KADIR HAS UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

GLOBAL MANY-TO-MANY ALIGNMENT OF MULTIPLE PROTEIN-PROTEIN INTERACTION NETWORKS

FERHAT ALKAN

August, 2013

Ferhat Alkan M.S. Thesis 2013

GLOBAL MANY-TO-MANY ALIGNMENT OF MULTIPLE PROTEIN-PROTEIN INTERACTION NETWORKS

by

Ferhat Alkan

Bachelor's degree, Telecommunication Engineering, Istanbul Technical University, 2010

Submitted to the Graduate School of

Science and Engineering in partial fulfillment of the requirements for the degree of

Computer Engineering

Master of Science

Kadir Has University

2013

GLOBAL MANY-TO-MANY ALIGNMENT OF MULTIPLE PROTEIN-PROTEIN INTERACTION NETWORKS

APPROVED BY:

Assoc. Prof. Dr. Cesim Erten ....
(Thesis Supervisor)

Asst. Prof. Dr. Öznur Yaşar Diner ....

Asst. Prof. Dr. Tınaz Ekim Aşıcı ....

DATE OF APPROVAL: . . . . . .

ABSTRACT

GLOBAL MANY-TO-MANY ALIGNMENT OF MULTIPLE PROTEIN-PROTEIN INTERACTION NETWORKS

Proteins are the essential parts of organisms and almost every biological process within a living cell is mediated by proteins and their interactions. Due to such importance, proteins are at the core of many researches in systems biology and evolutionary biology. In particular, defining the function of a protein and identifying functionally orthologous proteins are crucially important in many research areas and precise function of a protein can only be defined by biochemical and structural studies. However, many computational methods are also developed for such purposes and they use the sequence and interaction data of proteins since it provides a presumption about the chemical structure of a protein. For example, network alignment studies aims to find clusters of functionally related proteins across given protein interaction networks usually by implementing the given networks as graphs and employing some graph theoretical approaches. In this thesis, we focus on the problem of global many-to-many alignment of multiple protein-protein interaction networks. We define the problem as an optimization problem and this is the first combinatorial definition that is given for the problem in the literature. Then, we prove the computational intractability of this problem and we propose a new heuristic algorithm for the solution. We provide the test results of the proposed algorithm on both actual and synthetic PPI networks and it outperforms the existing algorithms, that serve at similar purpose, in terms of many evaluation aspects.

ÖZET

BİRDEN ÇOK PROTEİN ETKİLEŞİM AĞININ ÇOKA ÇOK OLARAK HİZALANMASI

Proteinler canlı organizmaların temel yapıtaşlarını oluşturur ve hücreler içerisindeki birçok biyolojik süreci düzenlerler. Bu büyük önemleri nedeniyle de sistem biyolojisi ve evrimsel biyoloji alanlarında birçok araştırmanın odağı halindedirler. Özellikle proteinlerin fonksiyonlarının tanımlanması ve fonksiyonel olarak benzer proteinlerin gruplanması birçok araştırma alanı için büyük önem taşımaktadır. Fakat bir proteinin kesin fonksiyonu ancak biyokimyasal ve yapısal analizlerle bulunabilmektedir. Bununla beraber proteinlerin dizilim ve etkileşim bilgilerini kullanarak bu amaçlara hizmet eden hesapsal yöntemler de geliştirilmektedir. Örneğin ağ hizalama çalışmaları bunlardan biridir ve verilen protein ağları içerisinden fonksiyonel olarak birbirine benzeyen proteinleri kümelemeyi amaçlar. Bu çalışmalar genellikle verilen ağların çizgeler olarak tanımlanmasını ve bu çizgeler üzerinde çeşitli çizge teorik yaklaşımlar uygulanmasını içerirler. Bu tez kapsamında ise birden çok protein ağının çoka çok olarak hizalanması problemi ele alınmaktadır. Bu tez ile bu hizalama problemi bir optimizasyon problemi olarak tanımlanmakta ve bu tanım bu problem için literatürde verilmiş olan ilk kombinatöryel tanımdır. Daha sonra bu problemin işlemsel karmaşıklığı analiz edilmekte ve problemin çözümü için bir buluşsal algoritma önerilmektedir. Sunulmuş olan BEAMS algoritmasının hem gerçek hemde sentetik ağlar üzerindeki test sonuçları sunulmakta ve bu sonuçlar literatürde aynı amaca hizmet eden diğer algoritmalar ile karşılaştırıldığında, BEAMS algoritmasının birçok açıdan diğer benzer algoritmalardan daha etkili çalıştığı görülmektedir.

ACKNOWLEDGEMENTS

I would like to thank many people who helped to make my graduate school days most valuable and productive but first, I sincerely thank Assoc. Prof. Cesim ERTEN, my major professor and thesis advisor. It was not only intellectually rewarding but also convivially pleasant to work with him in these two years. I also thank him for introducing me to the very interesting field of Bioinformatics.

Many thanks to faculty research assistants, who sincerely gave me a warm welcome when I first enrolled to this graduate programme. Particularly, special thanks to Aykut Çayır for his valuable friendship and Serkan Altuntaş for his informative comments about biological background of this thesis. I would also like to thank to all professors of computer engineering department for their highly instructive courses.

The last words of thanks go to my family and my beloved friends. I thank my parents Mete & Fatma Alkan and my brother Köksal Alkan for their endless support and encouragement. Lastly I thank my dearest friends Aysima Karçaaltıncaba, Doğuş Burak Akkoyunlu, Ekin Önsöz, Emre Sofyalı, Ezgi Toksoy, Taner Öz and all my other friends for their presence in my life.

This work is dedicated to the brave chapullers who are killed and injured during the Taksim Resistance in June 2013,

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This introductory chapter is divided into four sections to provide better understanding of the biological background of the thesis. We first start by giving information about genes, gene products, their functions and sequence alignment. Then, we give information about the functional orthology of genes and proteins and we continue by explaining protein interaction networks. In the last section, network alignment is introduced to the reader and existing alignment algorithms are summarized for a better understanding of global network alignment problem.

1.1. Genes, Proteins and Comparative Genomics

Every living organism is made of cell or cells and all living organisms carry out countless different biological activities during their lifetime. Most of these biological activities take place inside the cells and these cellular processes are always mediated by some specific molecules and their interactions with other molecules. In particular, proteins and their interactions are at the core of many cellular processes and these proteins are mostly synthesized within the cells of organisms.

The information to synthesize such proteins is mostly inherited from the ancestors of the cell and it is part of the genetic information that is handed down from generation to generation through their genomes. DNA molecules in genomes store this information in a chemical code with its chemical building blocks and the interpretation machinery of this code is essentially the same for every species [2]. According to the central dogma of biology, chemical code of the needed information is first transcribed into a chemically related set of molecules, messenger RNA (mRNA) and then, this coded information in mRNA is translated into a chemically related protein molecule in ribosome, a special organelle of the cell. The information-carrying transcribed parts of the DNA molecules are called as genes and thereby proteins are considered as the products of their coding genes.

Amino-acids are the fundamental building blocks of proteins and every protein has an amino-acid sequence which is determined by the chemical code sequence of its coding gene. The amino-acid sequence and the folding of protein determines its specific three dimensional structure and this structure lies at the core of determining its interactions and function within the cell.

Breakthrough advances in sequencing technology of genomes and proteins resulted in huge amount of fruitful sequence data of genomes and proteins for many species. Due to this progress, a new field of study, comparative genomics was born and it aims to provide insights about the evolutionary and functional mechanisms on genomes by comparing sequence data of different genes and gene products that are from different species. As mentioned by Fang *et al.*, all species originates from a common ancestor and these variations are because of the natural selection and being exposed to different complicated environmental changes. They continue by stating, with the field of comparative genomics, divergence of species from the common ancestor can be deciphered by comparing genomes of different organisms and, evolutionary processes such as gene deletion, gene speciation, gene duplication and horizontal gene transfer cause additional complexity for current comparisons [3]. Therefore, these complexities have forced researchers to develop different aspects for analyzing the data which is highly increasing in quantity and quality. So far, there have been many breakthrough progresses and many tools have been created for such comparisons. For example, ”Basic Local Alignment Search Tool (BLAST)” [4] is one of the most important tools that is used for measuring how similar two genes and two gene products are in their sequences.

1.2. Functional Orthology of Genes and Proteins

Evolutionary processes on genomes cause speciation, duplication and deletion of genes and that is the reason why species have different DNA sequences in their genomes despite the origination from a common ancestor. If two genes share a common ancestral gene, they are called as homologs and there are two types of homologous genes. If these genes belong to the different species they are called as orthologs and otherwise called as paralogs [5]. In other words, if two genes in different species have evolved through speciation processes they become orthologous and if two genes from the same specie have evolved through duplication processes they become paralogous [5]. Whether they are paralogous or orthologous, homologous genes are mostly similar to each other in their sequences.

In comparative genomics, accurate orthologous gene identification bears a crucial importance for many other research areas such as gene function prediction, phylogenetic analyses, and genomic context analyses [5] and there is a large variety of proposed orthologs prediction methods in literature [6, 7, 8, 9, 10, 11, 12, 13]. Orthologous genes and proteins are also analyzed to identify whether they share the same function in their cellular processes. Generally it is assumed that orthologous proteins are functionally related but they need further analyses to identify their functional orthology.

Defining a function for proteins is still an extensively studied area of proteomics science and sequence alignment tools and interaction data of proteins are intensively used in this area. Accurate function prediction for a protein can only be achieved by biochemical and structural studies, however, due to the high quantity of proteins, it is impossible to perform such studies for every protein in all species. For this reason, development of reliable computational methods for protein function prediction bears a crucial importance to progress in genomics science. So far developed computational methods for such predictions mostly rely on the sequence alignment and interaction data of proteins and they are proven to be reliable in many ways. Besides, representing the interaction data of proteins as networks comes in handy for such function prediction studies.

1.3. Protein-Protein Interaction (PPI) Networks

As mentioned before, cellular activities of all organisms are mostly mediated by proteins and their interactions. Technological advances, several high throughput measurement techniques and computational methods enabled to discover these protein-protein interactions (PPIs) for many species and these interactions are at the center of many researches in many areas. According to Singh *et al.* [14], ”the data from these techniques, which are still being perfected, are being supplemented by high-confidence computational predictions and analyzes of PPIs [15, 16, 17]”. For better analysis and representation, many complex systems in biology such as PPIs, metabolic processes, gene regulations and signal transductions are usually represented by networks and structural information of PPI networks for many species are becoming increasingly complete and accurate with those techniques. With the availability of these networks, new area for systematic studies of PPI networks was born and especially, cross-species network comparisons have taken a considerable interest from many researchers. In many computational comparison studies, these networks are implemented with graphs where nodes represent the proteins and the edges correspond to interactions between pairs of proteins.

Network comparison can provide valuable insights about the structural and organizational features of PPI networks [18] and by discovering network similarities of different species, valuable insights can be developed about the evolution, cellular biology and maybe diseases. For these reasons, as mentioned by Sharan and Ideker, three types of network comparison methods has been suggested in general and these methods are network integration, network querying and network alignment [1].

Network integration is the study of comparing PPI network of a specie with other networks of the same specie. This other compared network can be metabolic, signal transduction or gene regulatory network and by this method it is aimed to discover the interrelations within the specie which can also result in function prediction for the proteins in the PPI network [19]. Furthermore, network querying studies aim to find subnetworks in a PPI network that is similar to the desired subnetwork whether from the same specie or different species and by this querying, it is aimed to develop knowledge about the evolutionary processes as it is mentioned in such articles [20, 21, 22, 23, 24, 25, 26]. Network alignment, which is also the main topic of our interest in this thesis, is explained in more detail in the next section.

1.4. Network Alignment of PPI Networks

Network alignment is the study of comparing two or more networks to identify similar or dissimilar regions across given networks. Network alignment of PPI networks is a crucially important study area in comparative genomics since it provides a better understanding and gives valuable insights in many areas, such as functional module conversation across species, functional orthologous proteins identification, prediction of homologous proteins and creation of phylogenetic relationships between different organisms. For such different purposes, two different network alignment type exists which are *local network alignment* and *global network alignment*. Additionally, if alignment is performed only with two networks it is named as *pairwise alignment* but if performed with more networks, it is named as *multiple alignment*. Figure 1.1 illustrates the network alignment problem.

As mentioned by Singh *et al.*, whether it is local or global, network alignment algorithms generally aim to reveal one or more common subgraphs across the graphs of given input networks and the uniformity of these graphs make way for conserving edges of these subgraphs. They continue by stating that, this conservation leads to a mapping between the nodes (proteins) from different networks but the difficulty is to create such mappings

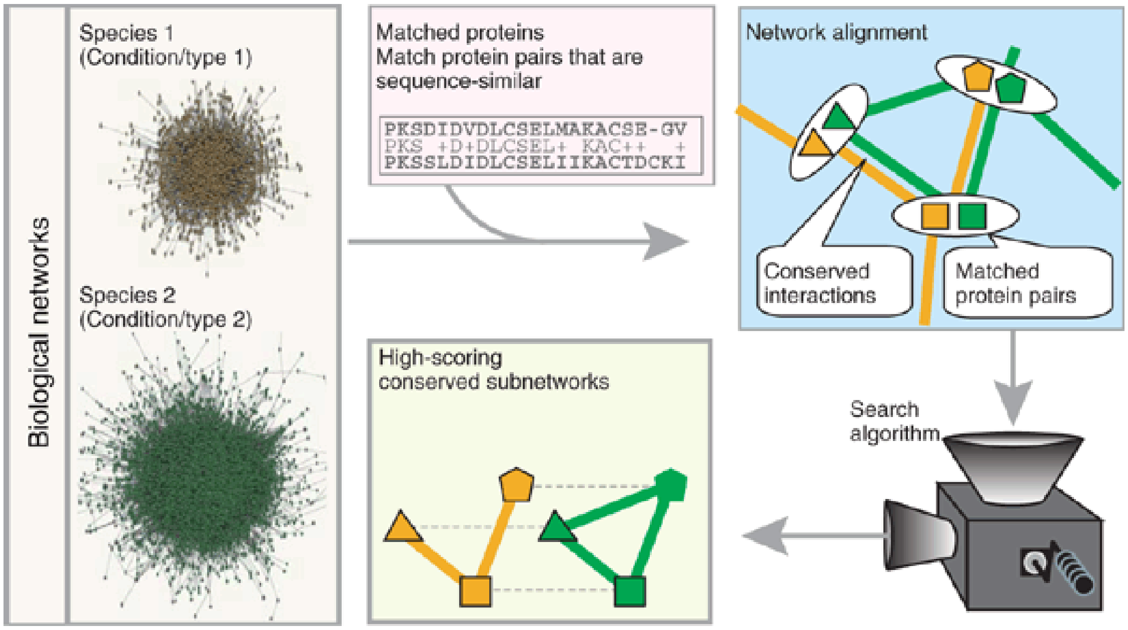


Figure 1.1. Visual description of network alignment problem (taken from [1]).

which are evolutionarily related [14]. Therefore, network alignment algorithms do not only deal with the network topologies of input networks to decide alignments, they also consider evolutionary relations of proteins such as their sequences since sequence similarities could represent evolutionary relations.

Additionally, alignment algorithms may also differ with respect to the types of mappings they provide. *One-to-one alignment* approaches aim to generate alignments where the output alignment either maps a protein in a network to exactly one protein from one of the networks or leaves the protein unmapped [14, 27, 28]. *One-to-many alignments* have been proposed for the global alignment of other biological networks including metabolic pathways, where each metabolic reaction in a pathway is mapped to a subset of reactions from another pathway [29, 30]. Finally, for *many-to-many alignments* the goal is to extract clusters of proteins where each cluster may include any number of proteins from the input networks [31, 32]. The proteins mapped to the same cluster as a result of the alignment are all expected to compose a functionally orthologous group. Note that among all three versions of the network alignments, the many-to-many version is the most general. Furthermore, as far as constraints from evolutionary molecular biology are concerned, it provides a more intuitive definition; the evolutionary distance between organisms under study may have large variations, leading to different numbers of proteins functioning similarly when considered in different networks.

In the following subsections, we continue by giving more detail about local and global network alignments.

1.4.1. Local Network Alignment

Local network alignment aims to discover highly similar structured subgraphs in given network graphs and it is performed for detecting similar functional modules in different species. At the early stages of network alignment algorithms, instead of global alignment algorithms, many local alignment algorithms have been developed and proposed. For example, NetworkBLAST [33] and PathBLAST [21] adapted the underlying ideas of BLAST sequence alignment algorithm; Graemlin [34] used protein modules for producing alignments; Berg and Lassig [35] has used Bayesian approach; MaWish [36] used evolution based scoring and Narayanan and Karp [37] performed a graph matching algorithm. However, Singh *et al.* states that, all these algorithms mostly rely on sequence similarities of proteins to reduce the complexity of problem so that they suffer from not considering network topologies in a significant level [14].

The outcomes of local network alignment algorithms provide clues about the functions of proteins by having many proteins of the same known function in the same detected common subgraph and in such situations, it is expected that remaining functionally unknown proteins of that subgraph have the same function as the rest.

1.4.2. Global Network Alignment

Informally, global alignment of multiple PPI networks is the problem of generating functional orthologous disjoint protein clusters through given networks. Since functional orthology is both about the interactions and sequences of proteins, global alignment seeks to create any kind of mapping between all proteins of given networks that will conserve the network topology and ensure the mapped proteins are highly sequence similar to each other. Zaslavskiy *et al.* states that, it is a more challenging problem than local alignment from a computational point of view since it searches for the best global mapping solution among all global possibilities [38]. They continue by stating, global alignment problem can also be considered as the problem of finding weighted graph matching between given PPI network graphs [38].

For all global network alignment algorithms, integration of network topology and sequence similarities has a crucial role. Aladağ and Erten states that, "Network alignment algorithms on the other hand incorporate the interaction data as well as the evolutionary relationships represented possibly in the form of sequence data. Based on the assumption that the interactions among functionally orthologous proteins should be conserved across species, such incorporation is usually achieved by aligning proteins so that both the sequence similarities of aligned proteins and the number of conserved interactions are large" [38].

Lately, global alignment problem has taken considerable interest and many algorithms are proposed for the problem solution. Some of them are GA [38], NATALIE [39], NetAlignBP, NetAlignMR [40], Graemlin [34], IsoRank [14], IsoRankN [32], MI-GRAAL [41] and variants [42], algorithm of Shih and Parthsarathy [37] and SPINAL [28]. Among all these existing algorithms, only IsoRankN algorithm is introduced in this thesis since it is the latest and so far best algorithm about global many-to-many alignment of multiple protein interaction networks.

IsoRankN algorithm generates the many to many clusters of global alignment results in two phases. In the first phase, it calculates a functional similarity score for each cross-species protein pairs, where it balances the the topological similarity and sequence similarity of the proteins with a user defined value α. Functional similarity score generation is performed by the original IsoRank algorithm and it uses a spectral graph theory for these calculations. Then, IsoRankN constructs a similarity graph with these scores and it performs a star aligned approach on this graph. After the creation of stars which is based on generated similarity scores, it performs spectral partitioning method on generated stars to decide final clusters.

1.5. The Scope and Contribution of the Thesis

The focus of this thesis is on global many-to-many alignment of multiple PPI networks. We first provide a formal combinatorial definition of the problem and it is the first formal definition in the literature. We proceed with proving its computational intractability even in a quite restricted case. We next provide a general framework for the problem, where we decompose the original problem into two subproblems; that of *backbone extraction* and *backbone merging*. Informally, each backbone in this framework corresponds to a closely related central group of proteins, at most one from each network. Once all the backbones are determined, the latter subproblem involves merging together the backbones with higher chances of coexistence in a cluster of orthologous proteins. We provide heuristic methods for both subproblems which together form our proposed algorithm based on backbone extraction and merge strategy, *BEAMS*. We experimentally evaluate the algorithm with regards to several biological significance metrics proposed in literature and compare it against a state-of-the-art and one of the most popular global many-to-many alignment methods, IsoRankN. The experimental results indicate that BEAMS alignments provide more consistent clusters than those of IsoRankN. Furthermore, considering the heavy computational load of the problem, the exceptional running time of BEAMS as compared to that of IsoRankN is a further improvement resulting from the provided framework and the algorithm.

2. METHODS AND ALGORITHMS

In this chapter, we first define the problem of global many-to-many alignment of multiple PPI networks as an optimization problem. Later on, we propose a new heuristic algorithm for the solution. Proposed algorithm is named as BEAMS after its method *Backbone Extraction And Merging Strategy* which will be explained in following sections.

2.1. Problem Definition

Although the one-to-one version of the problem has been formally defined in previous work [28], no formal combinatorial definition exists for the many-to-many version of the alignment problem apart from parameter learning based definitions [34]. We first provide a formally defined optimization goal for the problem that captures the essence of the informal definition provided in the Introduction. The definition is based on an intuitive generalization of the global one-to-one network alignment problem definition provided in [14, 28].

Let $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, $\ldots$, $G_k(V_k, E_k)$ be the input PPI networks where G_i corresponds to the i^{th} PPI network and V_i, E_i denote respectively the vertex set (proteins) and the edge set (interactions) of G_i. Let S indicate the edge-weighted complete $k-$partite *similarity graph* where the i^{th} partition of S is V_i and each edge (u, v) in S is assigned a positive real weight $w(u, v)$. This weight corresponds to the *sequence similarity score* $s(u, v)$ between u and v, usually assumed to be the Blast bit score of u and v, where $u \in G_i$, $v \in G_j$ and $i \neq j$. Let S_β be a subgraph of S with the same set of vertices. S_β represents a filtered version of the similarity graph S, so that only edges between pairs of proteins with relatively high sequence similarity are retained. For a fixed S_β, the *global many-to-many alignment* of all the input PPI networks is the problem of finding a *maximal* set of non-overlapping *clusters* $\mathcal{CL} = \{Cl_1, Cl_2, \ldots, Cl_m\}$ that maximizes the following alignment score:

$$AS(\mathcal{CL}) = \alpha \times CIQ(\mathcal{CL}) + (1-\alpha) \times \frac{\sum_{\forall Cl_i \in \mathcal{CL}} ICQ(Cl_i)}{|\mathcal{CL}|} \tag{2.1}$$

Here α is a real number between 0 and 1. It is a balancing parameter that determines the contribution weight of network topology as compared to homological similarity in the construction of output alignments. Each cluster Cl_i is defined to be a complete $c-$partite subgraph of S_β where $1 < c \leq k$. A set of clusters $\mathcal{CL}$ is maximal if no additional clusters can be added to $\mathcal{CL}$, that is no further complete $c-$partite subgraph remains in S_β. Note that maximizing the AS score does not automatically guarantee the maximality of the output set of clusters.

$CIQ(\mathcal{CL})$ denotes *cluster interaction quality* and is a measure of interaction conservation between all cluster pairs in $\mathcal{CL}$. Let E_{Cl_m,Cl_n} denote the set of all PPI edges with endpoints in distinct clusters Cl_m, Cl_n. We define a *conservation score* for each such edge (u, v), denoted with $cs(u, v)$. Let $s_{m,n}$ denote the number of PPI networks shared by the vertices in Cl_m, Cl_n and let $s'_{m,n}$ be the number of distinct PPI networks containing the edges in E_{Cl_m,Cl_n}. We assign $cs(u, v) = 0$ if $s'_{m,n} = 1$ and $cs(u, v) = s'_{m,n}/s_{m,n}$ otherwise. This is a generalization of edge conservation definition of pairwise network alignments. Note that for pairwise alignments edge conservation is assigned a binary value, that is a PPI edge in one network is either conserved in the other network or not. However for multiple alignments the employed definition may assign rational conservation values. We formally define $CIQ(\mathcal{CL})$ as follows:

$$CIQ(\mathcal{CL}) = \frac{\sum_{\forall Cl_m,Cl_n} \sum_{\forall (u,v) \in E_{Cl_m,Cl_n}} cs(u, v)}{\sum_{\forall Cl_m,Cl_n} |E_{Cl_m,Cl_n}|} \tag{2.2}$$

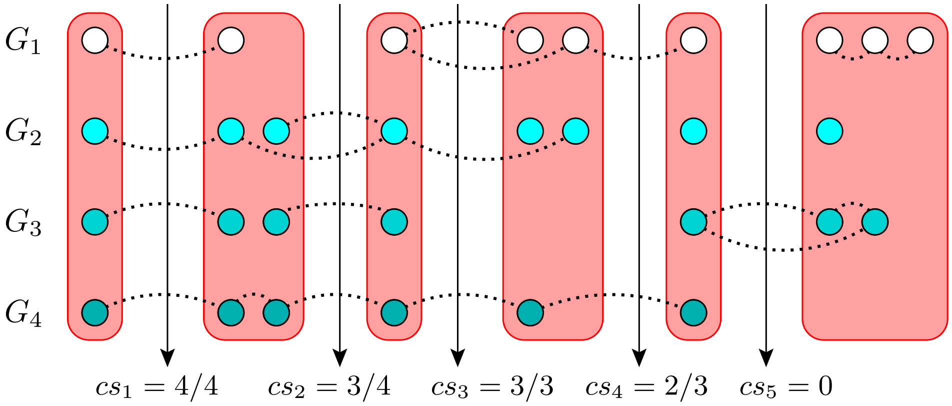


Figure 2.1. Conservation scores on a sample alignment covering all notable cases.

For a sample ciq calculation, see Figure 2.1. Note that, rectangular groups represent the clusters of the alignment. The dotted edges represent the protein-protein interactions. Proteins of each PPI network are drawn at separate horizontal layers. The CIQ score for this alignment is $(4 \times 4/4 + 4 \times 3/4 + 4 \times 3/3 + 2 \times 2/3 + 0)/16 = 0.771$.

In Equation 2.1, $ICQ(Cl_i)$ stands for the *internal cluster quality* of a given cluster Cl_i and is a measure of sequence similarities of involved proteins. Let $w_{max}(u)$ denote the maximum weight of any edge incident on u in S_β. Denote the edge set of Cl_i with $E(Cl_i)$. $ICQ(Cl_i)$ is defined as follows:

$$ICQ(Cl_i) = \frac{\sum_{\forall(u,v)\in E(Cl_i)} \frac{w(u,v)^2}{w_{max}(u) \times w_{max}(v)}}{|E(Cl_i)|} \tag{2.3}$$

2.2. BEAMS Algorithm

We first show that for a fixed S_β, the global many-to-many network alignment problem is computationally intractable. Due to clarity considerations we leave the proof to the Chapter 3.

Proposition 2.2.1. *For all $\alpha \neq 0$, the global many-to-many alignment problem is NP-hard* *even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*

Considering this NP-hardness result, it is necessary to devise efficient heuristics for the problem. Regarding the cluster definition of Equation 2.1 we make the following observation. Each cluster Cl_i which is a complete $c-$partite graph, can be subdivided into a set of n_i disjoint cliques, where n_i denotes the size of the maximum partition of Cl_i. In fact, n_i is the minimum possible size for such a set and each clique in the set has size c' where $1 \leq c' \leq c$. Therefore we view the original alignment problem of being composed of two subproblems: *backbone extraction* and *backbone merging*. A *backbone* is defined as a clique in S_β and a set of appropriate backbones together form a cluster. The first subproblem is that of extracting a *minimal* set of disjoint cliques from S_β which covers S_β completely and that maximizes the alignment score AS when each nontrivial clique of size greater than one is considered a cluster in the definition of Equation 2.1. The set is minimal in the sense that no output pair of cliques can be merged together to form a larger clique. Informally, each backbone corresponds to an orthologous set of proteins with at most one protein from each of the input networks. Thus the backbone extraction problem can actually be viewed as the global one-to-one alignment of multiple networks. A group of backbones is called *mergeable* if their union provides a valid cluster, that is a complete $c-$ partite graph. We define the second subproblem as finding a minimal set of mergeable backbone groups such that no further mergeable group remains and that maximizes the resulting AS score when each mergeable backbone group is considered a cluster in the definition of Equation 2.1. Note that a mergeable group represents a cluster of proteins that are highly homologous since every pair of proteins from different networks are connected by large weight edges in the filtered similarity graph S_β. Thus imposing the constraint that no further merging can be done on the set implies the intuition that no two pairwise homologous clusters should be part of the output alignment separately. We show that even these subproblems are computationally hard and we provide efficient heuristics for each one. In what follows, we first present the details of S_β construction, then proceed to provide descriptions of the two main steps of the BEAMS algorithm.

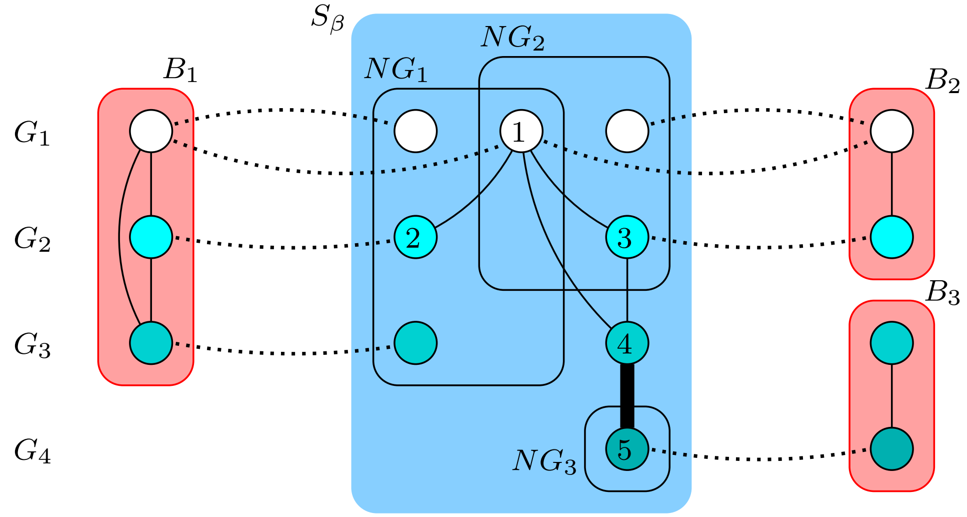
2.2.1. Construction of S_β

Considering the sizes of the networks under consideration and the fact that multiple networks constitute the study subject, a suitable filtration on the complete sequence similarity graph S is necessary for mainly two reasons. Firstly, even the suboptimal polynomial-time heuristic algorithms require large amounts of computational power as the size of S increases. Furthermore, taking into account the complete graph S may lead to incorrect alignments as far as biological significance measures are concerned. Most pairs of proteins from different networks do not bear any significance in terms of sequence similarity scores and employing an alignment with the unfiltered similarity graph S may align proteins with almost no homological similarity. As the evolutionary distance between pairs of input networks might be quite different, we employ a *relative filtration* that takes into account the relative differences in sequence similarities of pairs of networks. For some user-defined threshold β, we construct the filtered similarity graph S_β, so that each edge (u, v) is removed from S if $w(u, v) < \beta \times max(u, v)$, where $max(u, v)$ denotes the maximum of $w(u, v')$ or $w(u', v)$ for any u', v' from the networks of u and v respectively.

2.2.2. Backbone Extraction

Regarding the first subproblem defined within the BEAMS framework, we show that the backbone extraction problem is NP-hard even for quite a restricted case. The full proof can be found in the Chapter 3.

Proposition 2.2.2. *For all values of $\alpha \neq 0$, the backbone extraction problem is NP-hard even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*



$C_0 = \{4, 5\}$ $\quad$ $C_1 = M_1 = \{1, 2\}$

$M_2 = \{1, 3\}$ $\quad$ $C_2 = \{1, 3, 4\}$

$C_3 = M_3 = \emptyset$

Figure 2.2. Sample neighborhood graph construction and candidate generation for a small instance.

Since the backbone extraction problem is NP-hard, we devise an iterative greedy heuristic that runs in polynomial time assuming the number of networks under consideration is constant. Our algorithm employs concepts related to *maximum edge weighted cliques* (MEWC), candidate generation based on neighborhood graph constructions, and a greedy selection heuristic aiming to optimize the AS score. The pseudocode is shown in Algorithm 1.

We start with an empty backbone set and a candidate set that consists only of C_0 which is the MEWC of S_β. The j^{th} iteration of the main loop of the algorithm consists of four main steps: Selecting a new backbone B_j among already existing j candidates, removing the backbone from S_β, generating the new candidate C_j, and finally updating all existing candidates. The first step simply involves selecting the new backbone as the candidate providing the maximum AS score when considered together with all existing backbones. Each candidate C_j is defined with respect to an already existing backbone B_j other than the special candidate C_0 which is updated throughout iterations as S_β is updated. To generate a new candidate C_j via the function call $Generate\_Cand(S_\beta, B_j)$, we first construct the *neighborhood graph* of B_j, which is the induced subgraph in S_β of the set of PPI neighbors of all the nodes in B_j. If the neighborhood graph does not contain any S_β edges, then the candidate C_j is empty. Otherwise, we find the MEWC, M_j, of this neighborhood graph and we generate C_j by constructing the *G-MEWC* of M_j in S_β. Here G-MEWC corresponds to *generalized* MEWC which is defined as the maximum edge weighted clique in S_β that is required to include all the nodes of M_j; see Figure 2.2 for a sample neighborhood graph construction and candidate generation on a small instance. In Figure 2.2, the dotted edges represent protein interactions and each network is drawn at a separate horizontal layer. Edges between different layers represent S_β edges. Besides, the bold S_β edge between 4 and 5 represents high homological similarity between the corresponding proteins. Candidates are generated with respect to S_β and backbones B_1, B_2, and B_3.

Note that on top of the interaction conservation advantages brought by neighborhood graphs, constructing the MEWC of the neighborhood graph guarantees a highly similar backbone candidate as far as homological sequence similarities represented by S_β edges are concerned. The G-MEWC construction on the other hand, is a precautionary measure to enable possible extensions of a candidate towards networks other than those of its respective backbone. As the last step within an iteration, we generate each candidate anew, again with respect to its corresponding backbone and the updated S_β, if it shares any nodes with the new backbone B_j. The iterations continue until S_β contains only isolated nodes, that is those of degree zero.

2.2.2.1. Computing Generalized MEWC. We employ a branch-and-bound type algorithm to find the generalized maximum edge weighted clique of S_β that is required to contain a given set of nodes, M_j. Note that assigning $M_j = \emptyset$, the problem reduces to that of finding the maximum edge weighted clique.

As is the case with usual branch-and-bound type algorithms, we traverse the search tree $\mathcal{T}$ in a depth first manner. Each node at level-i of $\mathcal{T}$ represents a clique of size $i + |M_j|$ in S_β, that must include nodes in M_j. During the traversal, for each traversed node $\eta = \{u_1, \ldots, u_{i+|M_j|}\}$ of $\mathcal{T}$ representing a clique containing nodes u_1, $\ldots$, $u_{i+|M_j|}$, we store the *neighborhood set* of η, denoted with N_η which contains nodes that are in the common S_β neighborhood of nodes u_1, $\ldots$, $u_{i+|M_j|}$. The total edge weight of η is denoted with $EW(\eta)$. Let $Rep(N_\eta)$ denote the set of partition numbers of S_β (the set of PPI networks) that has a node in the set N_η. Throughout the traversal, we store the best node of the search, denoted with $best_\eta$ and its weight with $EW(best_\eta)$. To complete the description of the algorithm, we need only to specify the rules for *branching* and the *bound* formulation of the search. An upper bound for the potential weight of a node η in $\mathcal{T}$ is assigned to, $EW(\eta) + \sum_{\forall u_t \in \eta} \sum_{\forall r \in Rep(N_\eta)} w_{max}(u_t, r) + PW_{max}(N_\eta)$, where $w_{max}(u_t, r)$ denotes the weight of the maximum weighted edge between u_t and any node in the r^{th} partition of S_β, and $PW_{max}(N_\eta)$ represents the maximum potential weight of a possible clique in N_η. Formally, $PW_{max}(N_\eta)$ is defined as the sum of the edge weights of the $\frac{|Rep(N_\eta)| \times (|Rep(N_\eta)|-1)}{2}$ heaviest edges of S_β. If the defined potential weight of a node η is greater than $EW(best_\eta)$ we branch at node η, which implies creating a new node η' at the next level $i+1$, where $\eta' = \{u_1$, $\ldots$, $u_{i+|M_j|}$, $u_{i+|M_j|+1}\}$ such that $u_{i+|M_j|+1} \in N_\eta$.

2.2.3. Backbone Merging

We previously defined the backbone merging problem as finding a minimal set of mergeable backbone groups that maximizes the resulting AS score. With regards to the second main step of the BEAMS algorithm, we first state the following proposition about the computational complexity of the corresponding problem. The full proof can be found in the Chapter 3.

Proposition 2.2.3. *For $\alpha \neq 0$, the backbone merging problem is NP-hard even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*

We provide an iterative greedy heuristic for the backbone merging step. Let MB denote the set of mergeable backbone groups. Initially MB contains all backbones provided by the first backbone extraction step. It is updated at every iteration of the algorithm by a greedy selection strategy which, similar to the backbone extraction step, employs a candidate generation and selection idea. At each iteration we construct all pairs of mergeable groups in MB which all together provide the set of all candidates of that iteration. For each candidate we compute the AS score of MB considering the candidate pair as a single group. Note that some groups in MB may consist of a single node. Such groups are excluded from the AS score computations. We then select the candidate which provides the maximum score and update MB by merging the pair. The algorithm stops when no mergeable pair remains which provides a minimal set MB. We finally remove groups with a single node and provide the resulting set as the output set of clusters. A full discussion of several implementation details regarding this step and the algorithm as a whole are left to the Chapter 4.

Algorithm 1 *EXTRACT_BACKBONES*

```
Input: $S_\beta, G_1, G_2, \ldots, G_k, \alpha$
Output: Set of backbones $B = \{B_1, B_2, \ldots, B_n\}$
$B = \emptyset; C = \emptyset$
//Initial candidate
$C_0 = MEWC(S_\beta)$; $C = C \cup \{C_0\}$
repeat
    $B_{new} = Select\_Cand(C, B)$; $B = B \cup \{B_{new}\}$
    Remove $B_{new}$ from $S_\beta$
    //Generate new candidate
    $C_{new} = Generate\_Cand(S_\beta, B_{new})$; $C = C \cup \{C_{new}\}$
    //Update each candidate in C
    for all $C_i \in C$ do
        if $C_i \cap B_{new} \neq \emptyset$ then
            if $i == 0$ then
                $C_0 = MEWC(S_\beta)$
            else
                $C_i = Generate\_Cand(S_\beta, B_i)$
            end if
        end if
    end for
until $S_\beta$ contains only isolated nodes
//Each isolated node is a backbone itself
for all nodes $u \in S_\beta$ do
    $B_{new} = \{u\}$; $B = B \cup \{B_{new}\}$
end for
```

3. NP-HARDNESS PROOFS

In this chapter, we provide the NP-hardness proofs of the propositions in Section 2. The following propositions correspond in the same order to Propositions 2.1, 2.2, and 2.3. All the proofs are based on reductions from $Monotone\ 1in3SAT$ which is a restricted version of the $3SAT$ problem [43]. In Monotone 1in3SAT exactly one literal in each clause is required to be true and none of the clauses contains negated literals.

3.1. NP-Hardness Proof of Global Many-to-Many Alignment Problem

Proposition 3.1.1. *For all $\alpha \neq 0$, the global many-to-many alignment problem is NP-hard even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*

Proof. Given a Monotone 1in3SAT instance Φ, we show how to construct an instance of the global many-to-many alignment problem that consists of two interaction networks G_1, G_2 and S_β the filtered sequence similarity graph. The variable and the clause gadgets are as shown in Figure 3.1. Note that each lp node in the auxiliary group is connected to all 6 of the lq and lr nodes of the auxiliary group, each lq node is connected to all 6 of the lp and lr nodes, and finally each lr node is connected to all 6 of the lp and lq nodes. These PPI interactions are not shown in the figure for clarity. The variable gadget corresponding to a variable x_p consists of two nodes v_p^T and v_p^F in G_1, and a single node v_p in G_2. Corresponding to a clause $c_i = (x_p \vee x_q \vee x_r)$ of Φ there are three nodes a_p^i, a_q^i, a_r^i in G_1. In G_2 12 nodes are created for the same clause. The nodes l_p^i, l_q^i, l_r^i make up the *main group.* Additionally there are three *auxiliary groups*, one for each literal in c_i. The nodes lp_p^i, lp_q^i, lp_r^i make up the auxiliary group for p; lq_p^i, lq_q^i, lq_r^i make up the auxiliary group for q; lr_p^i, lr_q^i, lr_r^i make up the auxiliary group for r. In terms of the PPI edges, the variable gadget contains no edges

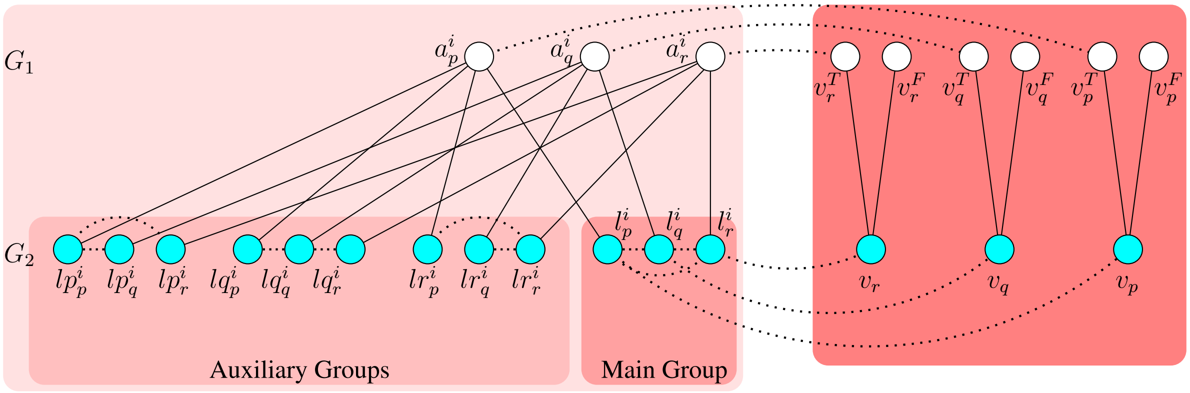


Figure 3.1. Construction of the clause gadget for a clause $c_i = (x_p \vee x_q \vee x_r)$ and the variable gadgets for x_p, x_q, x_r of Proposition 3.1.1.

between its own nodes. In the clause gadget the main group is a K_3 in G_2. The auxiliary groups altogether is almost a K_9 in G_2, except the auxiliary group of p has a missing edge between lp_q^i and lp_r^i, the auxiliary group of q has a missing edge between lq_p^i and lq_r^i, and finally the auxiliary group of r has a missing edge between lr_p^i and lr_q^i. With regards to the edges between variable gadget nodes and clause gadget nodes, each node v_p^T is connected in G_1 to every node a_p^j for every clause c_j such that $x_p \in c_j$. Similarly in G_2, the node v_p is connected to every node l_p^j such that $x_p \in c_j$ for some clause c_j. Regarding similarity edges, there are edges (v_p, v_p^T), (v_p, v_p^F) in the variable gadget. In the gadget for clause c_i, a_p^i is connected to $l_p^i, lp_p^i, lq_p^i, lr_p^i$; a_q^i is connected to $l_q^i, lp_q^i, lq_q^i, lr_q^i$; a_r^i is connected to $l_r^i, lp_r^i, lq_r^i, lr_r^i$ in the similarity graph. For simplicity we call S_β edges incident on a main group node a *main group edge* and those incident on an auxiliary group node an *auxiliary group edge*. All similarity graph edges have equal weight.

We show that Φ is satisfiable if and only if the constructed graph admits a global many-to-many alignment with an AS score of 1. Assume Φ is satisfiable. From the variable gadget of a variable x_p we choose (v_p^T, v_p) as a cluster if x_p is assigned True in Φ and (v_p^F, v_p) if it is assigned False. For a clause gadget corresponding to $c_i = (x_p \vee x_q \vee x_r)$, without loss of generality let x_p be the True literal. We choose three clusters $(a_p^i, l_p^i), (a_q^i, lp_q^i)$, and

(a^i_r, lp^i_r). Note that the nodes a^i_q, a^i_r are clustered with their corresponding nodes from the auxiliary group of p. The provided clustering is a valid alignment according to the problem definition provided in the section 2.1. We show that with such a clustering the AS score is 1. The ICQ score of each cluster is exactly one since each sequence similarity edge is assumed to have equal weight. We only need to prove that the CIQ score of the output clusters is exactly 1. Note that this corresponds to a cluster selection where the interactions between all cluster pairs are conserved. We only need to show this for a pair that consists of a cluster from a clause gadget and a cluster from a variable gadget, since the clause clusters are chosen so that no PPI edge exists between any pair of clause clusters, and the variable gadget itself contains a single cluster. Both G_1, G_2 PPI edges connecting to the cluster (a^i_p, l^i_p) are conserved since a^i_p is connected to v^T_p, l^i_p is connected to v_p, and (v^T_p, v_p) is one of the constructed clusters. The clusters (a^i_q, lp^i_q), and (a^i_r, lp^i_r) do not have PPI edges to variable gadget clusters; $(v^F_q, v_q), (v^F_r, v_r)$ are their variable gadget clusters and no edge exists between the pairs $\prec a^i_q, v^F_q \succ$ $\prec a^i_r, v^F_r \succ$, $\prec lp^i_q, v_q \succ$, and $\prec lp^i_r, v_r \succ$.

For the reverse direction we show that the existence of a legal alignment with AS score 1 implies the satisfiability of Φ. If such an alignment exists then it must be the case that its CIQ score is also 1, that is every edge between any pair of clusters in the alignment must be conserved. Every G_1 node in a clause gadget is neighbors in the similarity graph with nodes that have a single similarity edge which implies that every G_1 node must be involved in a cluster by the maximality property of a legal alignment. Since the G_1 nodes in the clause gadget do not have any common similarity graph neighbors this further implies that each one must be in a separate cluster and that for every clause gadget there must exactly be three disjoint clusters. We first show that one of these clusters is a main group edge and the other two are auxiliary group edges. Furthermore the auxiliary group edges are incident on nodes that belong to the auxiliary group of the node that the main group edge is incident on.

Note that there are no three auxiliary group nodes that are pairwise disjoint in G_2. This implies that one of the clusters must involve a main group node for otherwise there would be a G_2 edge that is not conserved in G_1. Without loss of generality let l_p^i be that node for the gadget corresponding to the clause $c_i = (x_p \vee x_q \vee x_r)$. The clusters of a_q^i and a_r^i can not include any main group node since that would introduce a nonconserved edge. Their clusters must then respectively be (a_q^i, lp_q^i), (a_r^i, lp_r^i), since among the similarity graph neighbors of a_q^i, a_r^i the only auxiliary group nodes that are disjoint in G_2 are lp_q^i and lp_r^i, and including any other node in the clusters would introduce a nonconserved edge. Note that lp_q^i, lp_r^i are PPI neighbors in G_2 with every other node among the auxiliary group. This implies that the cluster of l_p^i must be (a_p^i, l_p^i) since including any other auxiliary group node that are neighbors of a_p^i in the similarity graph would introduce a nonconserved edge.

For the truth assignment of ϕ we assign every literal that corresponds to a main edge cluster in a clause gadget to True and every literal that corresponds to an auxiliary edge cluster to False. Thus obviously only one literal per clause is assigned True. We finally need to show that this assignment is a valid assignment in the sense that a variable assigned to True in some clause gadget is not assigned to False anywhere else and vice versa. Let x_p be a variable assigned True due to the main edge cluster selection in a cluster c_i. It must be the case that in the variable gadget corresponding to x_p the node v_p^T must belong to a cluster, for otherwise there would be a nonconserved PPI edge between l_p^i and v_p. This implies x_p can not be assigned False anywhere else due to auxiliary edge clustering, since no auxiliary group nodes are connected to v_p in G_2 and there would be a nonconserved edge. □

3.2. NP-Hardness Proof of Backbone Extraction Problem

Proposition 3.2.1. *For all values of $\alpha \neq 0$, the backbone extraction problem is NP-hard even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*

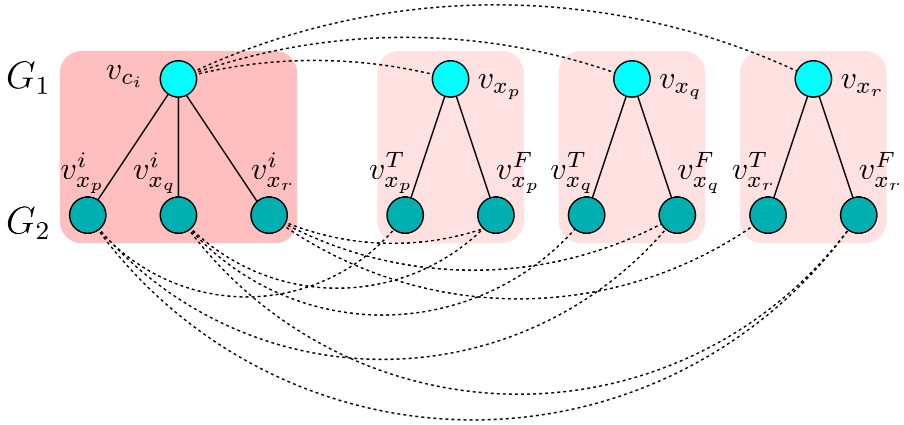


Figure 3.2. Construction of the clause gadget for a clause $c_i = (x_p \vee x_q \vee x_r)$ and the variable gadgets for x_p, x_q, x_r of Proposition 3.2.1.

Proof. Given a Monotone 1in3SAT instance Φ, we show how to construct an instance of the backbone extraction problem that consists of two interaction networks G_1, G_2 and S_β the filtered sequence similarity graph; see Figure 3.2 for the construction of clause and variable gadgets. For each clause c_i of Φ we create a *clause node* v_{c_i} in G_1. Additionally, for each variable x_p of Φ we create a *variable node* v_{x_p} in G_1. For a clause node v_{c_i} where $c_i = (x_p \vee x_q \vee x_r)$, we create three PPI edges $(v_{c_i}, v_{x_p}), (v_{c_i}, v_{x_q}), (v_{c_i}, v_{x_r})$ in G_1. Corresponding to each clause node v_{c_i} of G_1 we create three nodes $v^i_{x_p}, v^i_{x_q}, v^i_{x_r}$ in G_2. We call these nodes *clause nodes* of G_2. Also for each variable node v_{x_p} of G_1 we create two variable nodes $v^T_{x_p}, v^F_{x_p}$ in G_2, each of which is called a *literal node* of G_2. Each node $v^i_{x_p}$ of G_2 is connected with three PPI edges with $v^T_{x_p}, v^F_{x_q}, v^F_{x_r}$ in G_2. The filtered similarity graph S_β is constructed as follows. We add three edges between v_{c_i} of G_1 and each of its corresponding clause nodes in G_2, that is $v^i_{x_p}, v^i_{x_q}, v^i_{x_r}$. Additionally we add two similarity graph edges between each variable node v_{x_p} of G_1 and the literal nodes $v^T_{x_p}, v^F_{x_p}$ of G_2. Note that all the sequence similarity edges are assumed to have equal weight. We show that Φ is satisfiable if and only if the AS score of the optimum solution to the backbone extraction problem on the instance G_1, G_2, S_β is exactly 1. Assuming Φ is satisfiable the backbone involving a clause node v_{c_i} of G_1 is the edge $(v_{c_i}, v^i_{x_p})$ where x_p is the true literal in c_i and the backbone involving a variable node v_{x_t} of G_1 is the edge $(v_{x_t}, v^T_{x_t})$ if x_t is assigned True in Φ and it is the edge $(v_{x_t}, v^F_{x_t})$ if it is assigned False in Φ. We show that this assignment of backbones provides a legal output for the backbone extraction problem and that its AS score is 1. It is easy to see that the assignment is legal since the output set of backbones is a minimal disjoint set of cliques. The ICQ score of each backbone is exactly one since each sequence similarity edge is assumed to have equal weight. We only need to prove that the CIQ score of the output backbones is exactly 1. Note that this corresponds to a backbone selection where the interactions between all backbone pairs are conserved. For a backbone $(v_{c_i}, v^i_{x_p})$ assigned by the construction, we show that every edge incident on the backbone be it in G_1 or G_2 is conserved. The node v_{c_i} is connected to $v_{x_p}, v_{x_q}, v_{x_r}$ in G_1 and the node $v^i_{x_p}$ is connected to $v^T_{x_p}, v^F_{x_q}, v^F_{x_r}$ in G_2. Since each of $(v_{x_p}, v^T_{x_p}), (v_{x_q}, v^F_{x_q}), (v_{x_r}, v^F_{x_r})$ is also selected as a backbone every edge involving v_{c_i} and $v^i_{x_p}$ is conserved. Note that considering only the backbones involving clause nodes of G_1, G_2 is sufficient since there are no PPI edges between any variable node pair of G_1 and the same is true for any literal node pair of G_2.

For the other direction, we show that if there exists a legal backbone extraction that provides an AS score of 1, then we can find an assignment of variables that gives rise to a satisfiable assignment of Φ that is valid with respect to the definition of Monotone 1in3SAT. First we note that every node in G_1 must be involved in a backbone due to the full-coverage condition in the definition of a legal backbone set. Furthermore this backbone cannot be a trivial backbone containing only the node itself for otherwise the backbone set would not be minimal; a clause node in G_1 is connected to three nodes from G_2 in S_β which have no other similarity edges and similarly a variable node in G_1 is connected to two nodes from G_2 in S_β which also have no other similarity edges. Given the output backbone set, for each backbone $(v_{c_i}, v^i_{x_p})$ involving a clause node of G_1 we assign x_p True and x_q, x_r False. First we show that with this assignment every variable x_p is assigned either True or False.

We start by showing that a variable assigned True by a backbone assignment must not be assigned False by the rest of the backbone assignments. In addition to clause c_i, let c_j be another clause containing variable x_p. Assuming $(v_{c_i}, v^i_{x_p})$ is a backbone, we need to show that $(v_{c_j}, v^j_{x_p})$ is also a backbone and thus its assignment of x_p does not conflict with that of the former backbone. We show that the backbone $(v_{c_i}, v^i_{x_p})$ implies that $(v_{x_p}, v^T_{x_p}$ is also a backbone. The variable node v_{x_p} has two candidates for a nontrivial backbone, $(v_{x_p}, v^T_{x_p})$ and $(v_{x_p}, v^F_{x_p})$. Thus (v_{c_i}, v_{x_p}) is a PPI edge in G_1 that must be conserved and this conservation is only possible by selecting the former backbone candidate since $v^i_{x_p}$ is connected only to $v^T_{x_p}$ with a PPI edge in G_2. The existence of the backbone $(v_{x_p}, v^T_{x_p}$ further implies the existence of the backbone $(v_{c_j}, v^j_{x_p})$. This follows from an argument similar to the one above. The clause node v_{c_j} has three candidates for a nontrivial backbone among which $(v_{c_j}, v^j_{x_p})$ has to be selected as only $v^j_{x_p}$ has a PPI edge with $v^T_{x_p}$ in G_2 that conserves the edge (v_{c_j}, v_{x_p}). Next we show that a variable assigned False by a backbone assignment must not be assigned True by the rest of the backbone assignments. Assuming $(v_{c_i}, v^i_{x_q})$ is not a backbone, we need to show that there exists no other backbone $v_{c_j}, v^j_{x_q}$. If $(v_{c_i}, v^i_{x_q})$ is not a backbone $(v_{x_q}, v^F_{x_q}$ must be a backbone. This follows from the fact that $(v_{c_i}, v_{x_p}$ or (v_{c_i}, v_{x_r}) must be a backbone and both of v_{x_p}, v_{x_r} are connected to $v^F_{x_q}$ rather than $v^T_{x_q}$ in G_2. The existence of the backbone $(v_{x_q}, v^F_{x_q})$ implies the nonexistence of $(v_{c_j}, v^j_{x_q})$ since there exists no PPI edge $(v^j_{x_q}, v^F_{x_q})$ in G_2 to conserve the PPI edge (v_{c_j}, v_{x_q}) of G_1. Finally due to the truth value assignment rule, it is obvious that for each clause exactly one literal is assigned True which implies a valid satisfiable Monotone 1in3SAT instance. □

3.3. NP-Hardness Proof of Backbone Merging Problem

Proposition 3.3.1. *For all values of $\alpha \neq 0$, the backbone merging problem is NP-hard even for the restricted case where two PPI networks are aligned, all backbones are 2-cliques and all edge weights in S_β are equal.*

Proof. We similarly construct a reduction from the Monotone 1in3SAT problem. For a given Monotone 1in3SAT instance Φ we provide the construction of G_1, G_2, S_β and the backbone

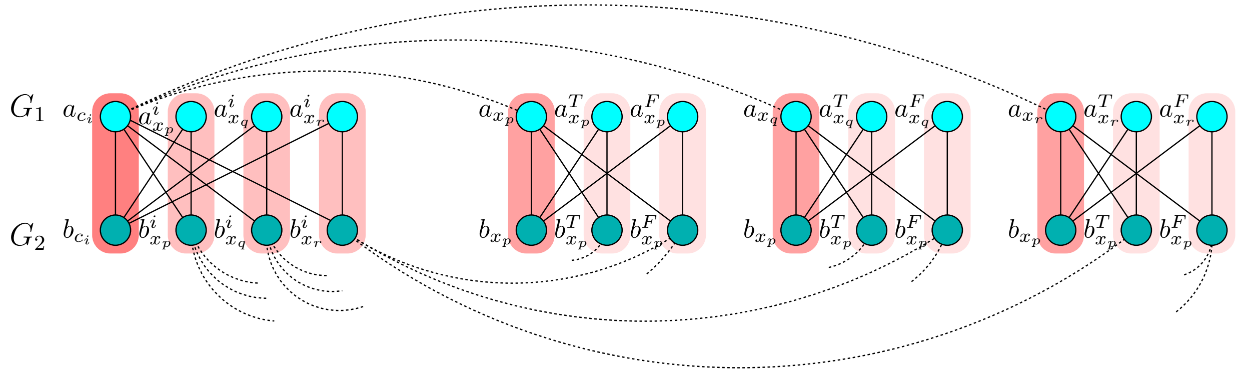


Figure 3.3. Construction of the clause gadget for a clause $c_i = (x_p \vee x_q \vee x_r)$ and the variable gadgets for x_p, x_q, x_r of Proposition 3.3.1.

set B. For each variable x_p of Φ three nodes $a_{x_p}, a^T_{x_p}, a^F_{x_p}$ are created in G_1. Corresponding to each clause $c_i = (x_p \vee x_q \vee x_r)$ we create four nodes $a_{c_i}, a^i_{x_p}, a^i_{x_q}, a^i_{x_r}$. The edge set of G_1 consists of edges (a_{c_i}, a_{x_t}) for each clause c_i, where x_t is a literal in c_i. The node set of G_2 is similar to that of G_1, that is for each variable x_p three nodes $b_{x_p}, b^T_{x_p}, b^F_{x_p}$ and for each clause c_i four nodes $b_{c_i}, b^i_{x_p}, b^i_{x_q}, b^i_{x_r}$ are created. Regarding the edges of G_2, for each clause c_i, we add the edge $(b^i_{x_t}, b^T_{x_t})$ for each literal $x_t \in c_i$ and the edge $(b^i_{x_t}, b^F_{x_w})$ for each literal pair $x_t, x_w \in c_i$ and $x_t \neq x_w$. For each clause c_i, we add the following similarity edges: (a_{c_i}, b_{c_i}) and for each $x_t \in c_i$, $(a_{c_i}, b^i_{x_t})$, $(a^i_{x_t}, b_{c_i})$, $(a^i_{x_t}, b^i_{x_t})$. For each variable x_p the following similarity edges are added: $(a_{x_p}, b_{x_p}), (a_{x_p}, b^T_{x_p}), (a_{x_p}, b^F_{x_p})$, $(a^T_{x_p}, b_{x_p}), (a^T_{x_p}, b^T_{x_p}), (a^F_{x_p}, b_{x_p}), (a^F_{x_p}, b^F_{x_p})$. We assign a similarity score of 0.5 for each similarity edge. Finally the backbone set B consists of single edges. For each clause c_i we have four backbones denoted with *clause backbones*: (a_{c_i}, b_{c_i}) and $(a^i_{x_t}, b^i_{x_t})$ for each $x_t \in c_i$. For each variable x_p we have three backbones denoted with *literal backbones*: $(a_{x_p}, b_{x_p}), (a^T_{x_p}, b^T_{x_p}), (a^F_{x_p}, b^F_{x_p})$. Note that this backbone set includes all nodes from the input networks and it is minimal, that is no pair of backbones can be merged together to form a larger clique. The construction is illustrated in Figure 3.3.

A key observation is that the maximum CIQ score attainable in any backbone merging of such an input instance is 0.5. This is due to the fact that the cluster backbone (a_{c_i}, b_{c_i})

can only be merged with only one of $(a^i_{x_t}, b^i_{x_t})$ for some $x_t \in c_i$ which further implies that two backbones in the clause gadget can not be merged with any other backbones. Six G_2 edges are incident on those two backbones and none of them can be conserved due to lack of G_1 edges incident on them and at most 6 PPI edges out of all 12 in the gadgets involving a clause and all its literals can be conserved. Thus the maximum AS score achievable in any alignment is 0.5.

We show that Φ is satisfiable if and only if the constructed instance has a backbone merging that provides a legal alignment with maximum score of 0.5. Assume Φ has a satisfying assignment. For each clause c_i, the clusters resulting from mergings is as $\{\{(a_{c_i}, b_{c_i}), (a^i_{x_p}, b^i_{x_p})\}, \{(a^i_{x_q}, b^i_{x_q})\}, \{(a^i_{x_r}, b^i_{x_r})\}\}$ where each set in this multiset represents a set of merged backbones into a cluster and x_p is the only variable assigned True in c_i. The clusters resulting from backbone merging in the corresponding variable gadgets is as $\{\{(a_{x_p}, b_{x_p}), (a^T_{x_p}, b^T_{x_p})\}, \{(a^F_{x_p}, b^F_{x_p})\}\}$ for the True literal x_p and $\{\{(a_{x_q}, b_{x_q}), (a^F_{x_q}, b^F_{x_q})\}, \{(a^T_{x_q}, b^T_{x_q})\}\}$, and $\{\{(a_{x_r}, b_{x_r}), (a^F_{x_r}, b^F_{x_r})\}, \{(a^T_{x_r}, b^T_{x_r})\}\}$ for the False literals x_q, x_r. Note that with the provided mergings, the resulting clusters conserve 6 out of all 12 PPI edges from G_1, G_2 between the clusters, when clusters related to a single clause and its variables are considered. Since every variable has the same truth value assignment in all the clauses, the AS score of the constructed alignment is exactly 0.5, the maximum possible score. Furthermore it is easy to verify that the provided alignment is legal with respect to the main problem definition; each cluster is a complete $c-$partite subgraph of S_β for $1 < c \leq 2$ and the set of clusters is maximal, that is no further complete $c-$partite subgraph remains in S_β.

For the reverse direction, assume we have a legal alignment with AS score 0.5. In any legal alignment, it should be that for a cluster $c_i = (x_p \vee x_q \vee x_r)$, any backbone merging must include three resulting clusters: $\{(a_{c_i}, b_{c_i}), (a^i_{x_t}, b^i_{x_t})\}$ for some $x_t \in c_i$ and $\{(a^i_{x_w}, b^i_{x_w})\}$ for $x_w \in c_i$ and $x_w \neq x_t$. We construct a truth value assignment for Φ by considering each cluster and assigning x_t, the variable involved in a merging, to True and the remaining two variables to False. We show that this is a legal Monotone1in3SAT assignment and it evaluates to True.

Since for each cluster exactly one variable is assigned True, it easy to verify that the provided assignment of truth values makes Φ True. It remains to show that this assignment is legal in the sense that a variable x_t assigned True due to a clause gadget must be assigned True in every clause gadget. As both the AS score and the ICQ score of the alignment is 0.5, it should be that the CIQ score is also 0.5. This implies that for every clause gadget and the gadgets involving its variables exactly 6 out of all 12 edges must be conserved, that is all three G_1 edges involved in the gadgets must be conserved. Given a clause $c_i = (x_p \vee x_q \vee x_r)$, without loss of generality let x_p be the variable involved in merging for the clause gadget of c_i, that is the clusters resulting from merging is as $\{\{(a_{c_i}, b_{c_i}), (a^i_{x_p}, b^i_{x_p})\}, \{(a^i_{x_q}, b^i_{x_q})\}, \{(a^i_{x_r}, b^i_{x_r})\}\}$. All three G_1 edges incident on the first cluster must be conserved. To conserve the edge (a_{c_i}, a_{x_p}) the clusters resulting from mergings in the variable gadget of x_p must be $\{\{(a_{x_p}, b_{x_p}), (a^T_{x_p}, b^T_{x_p})\}$, $\{(a^F_{x_p}, b^F_{x_p})\}\}$. To conserve the edge (a_{c_i}, a_{x_q}) the clusters resulting from mergings in the variable gadget of x_q must be $\{\{(a_{x_q}, b_{x_q}), (a^F_{x_q}, b^F_{x_q})\}, \{(a^T_{x_q}, b^T_{x_q})\}\}$. Finally, to conserve the edge (a_{c_i}, a_{x_r}) the clusters resulting from mergings in the variable gadget of x_r must be $\{\{(a_{x_r}, b_{x_r}), (a^F_{x_r}, b^F_{x_r})\}, \{(a^T_{x_r}, b^T_{x_r})\}\}$. We show that for any clause c_j such that $x_p \in c_j$, it must be that x_p is the variable involved in merging for the clause gadget of c_j, that is the resulting three clusters of c_j's gadget must be $\{(a_{c_i}, b_{c_i}), (a^i_{x_p}, b^i_{x_p})\}$ and $\{(a^i_{x_w}, b^i_{x_w})\}$ for $x_w \in c_i$ and $x_w \neq x_p$. Assume for the sake of contradiction, x_p is not involved in merging for the gadget of c_j, that is one of the resulting three clusters is $\{(a_{c_i}, b_{c_i}), (a^i_{x_w}, b^i_{x_w})\}$ for some $x_w \in c_j$ and $w \neq p$. Then it is impossible to conserved the edge (a_{c_j}, a_{x_p}) incident on the cluster $\{(a_{x_p}, b_{x_p}), (a^T_{x_p}, b^T_{x_p})\}$ since the cluster is incident on only one G_2 edge which is $(b^i_{x_p}, b^T_{x_p})$. This further implies a CIQ score strictly less than 0.5 which is a contradiction. □

4. IMPLEMENTATION DETAILS AND RUNNING TIME ANALYSIS

We provide a discussion of BEAMS in terms of its running time requirements and describe implementation details when necessary. The initial preprocessing step of S_β construction is trivial and requires $O(|E|)$ time where E represents the set of edges in the $k-$partite graph S.

With regards to the running time analysis of the backbone extraction step described in pseudocode in Algorithm 1 of the Chapter 2, we first provide a description of our implementation of finding the generalized maximum edge weighted clique, G-MEWC. Since the input to the G-MEWC algorithm changes throughout the execution of the algorithm we provide a description of the algorithm on a general $k'-$partite graph $G' = (V', E')$ and a given M which denotes the set of nodes required to be in the output maximum edge weighted clique. As a preprocessing step of the G-MEWC algorithm, for each node $u_t \in V'$, we first compute and store $w_{max}(u_t, r)$ for each $1 \leq r \leq k'$ edges. The preprocessing also includes the computation of the sum of the weights of the largest $\frac{r\times(r-1)}{2}$ edges. All this information is then employed to speed up the bound calculations; when computing an upper bound for the potential weight of a node η of the branch-and-bound tree this preprocessed data is used rather than computing it repeatedly for each tree node. The only remaining information during a bound phase of a node η is the common neighborhood of all the nodes stored at η which is computed employing the neighborhood information of η's parent in the tree. This requires $O(\Delta)$ time, where Δ denotes the maximum degree of any node in G'. The number of nodes of the branch-and-bound tree is bounded by $O(|V'|^{k'})$ if $M = \emptyset$ and $O(\Delta^{k'-|M|})$ otherwise, since the common neighborhood of M can be of size at most Δ. The total running time of G-MEWC is then $O(\Delta|V'|^{k'})$ if $M = \emptyset$ and $O(\Delta^{k'-|M|+1})$ otherwise. Note that the former version of G-MEWC is denoted with MEWC in Algorithm 1.

Let V denote the set $V_1 \cup \ldots \cup V_k$. The running time of Algorithm 1 is dominated by the time spent in the main *repeat* loop of lines 6 through 21. Note that the number of iterations of the loop is $O(|V|)$, since the maximum number of output backbones can be at most $|V|$, each iteration finds a new backbone, and the iterations continue until no new backbones remain. The function $Select\_Cand$ at line 7 finds the candidate that scores the best when considered with the already existing backbone set. Both the new ICQ and the CIQ scores are calculated by computing the contribution of the new backbone and combining this contribution with the existing values. To compute the contribution of the ICQ score of a given candidate with the existing backbones requires $O(k^2)$ time, whereas the contribution to the CIQ score is computed in time $O(k^2\Delta_{max})$, where Δ_{max} is the maximum degree of any node in V in its respective PPI network. Since the number of candidates at a specific iteration is bounded by $O(|V|)$, the running time required by line 7 is $O(|V|k^2\Delta_{max})$. For the $Generate\_Candidate$ function calls of lines 10 and 17, one call to MEWC is made on the neighborhood graph of the input backbone and one call to G-MEWC is made on S_β with the set M containing at least two elements. Note that the size of the neighborhood graph is at most $k\Delta_{max}$. The total running time of these two calls is $O(\Delta(k\Delta_{max})^k + \Delta^{k-1})$, where the first term indicates the time required for the first call and the second term stands for the running time of the second call. Hereinafter Δ denotes the maximum degree of any node in S_β, since Δ gets its maximum value when G-MEWC is called on S_β. For the calls at line 15 the set M is empty, thus each call requires the heavier version of G-MEWC, namely MEWC on S_β which requires running time $O(\Delta|V|^k)$. Therefore to speed up the algorithm, we do not actually compute MEWC at each execution of line 15, but rather employ some preprocessing and proceed with updates when necessary. As a preprocessing, the G-MEWC is initially computed for $M = \{u\}$ for each node u in V and all these G-MEWC sets are stored in a list which requires $O(|V|\Delta^k)$ time in total. At each iteration two main operations regarding line 15 are implemented: $find\_max$ and *update*. The former finds the maximum weighted G-MEWC stored in the current list, whereas the latter recomputes G-MEWC of the nodes in the list that contain nodes already assigned to some backbone. Since each iteration of the *repeat* loop assigns at most k nodes to a new backbone, these nodes can be part of at most $k\Delta$ G-MEWC sets. Thus all the updates at a specific iteration of the *repeat* loop requires $O(k\Delta)$ updates each of which requires $O(\Delta^k)$ time. In total the running time required by line 15 is then bounded by $O(|V| + k\Delta^{k+1}))$. Note that line 15 is executed only once for the update of C_0 within the *for* loop of lines 12 through 20. However line 17 is executed $O(|V|)$ times since the number of candidates at a specific iteration can be at most $|V|$. Thus the total running time of the main *repeat* loop and in turn that of the whole Algorithm 1 is $O(|V|^2\Delta(k\Delta_{max})^k + |V|^2\Delta^{k-1} + |V|k\Delta^{k+1})$. Assuming $\Delta_{max} = O(\Delta)$ and k a small constant, which usually is the case for the PPI networks under study, the running time is $O(|V|^2\Delta^{k+1})$.

For the second main phase of BEAMS which consists of backbone merging, assume a backbone list MB is given. We treat MB as a cluster list, iteratively update it, and finally the list remaining at the end of this phase becomes the set of output clusters. First a list of all mergeable pairs of backbones, C_{MB} is constructed. Note that this is done only once, at the beginning of this phase. Next we iteratively select the best pair from C_{MB}, one that provides the best AS score with the rest of the clusters in MB, remove the pair from MB, insert the merged pair back into MB, and update C_{MB} by removing the two candidates corresponding to the merged pair from C_{MB} and inserting their intersection back into C_{MB}. Throughout iterations the most time consuming task is that of computing the best pair in C_{MB}. Let C_{max} denote the size of the maximum cluster output by the algorithm. Computing the ICQ contribution of a single candidate requires time $O(|C_{max}|^2)$. The CIQ contribution can be computed in time $O(|C_{max}\Delta_{max}|)$, since this is an upper bound on the total number of PPI edges incident on the nodes of a candidate. Thus a single execution of this step requires $O(|V|^2|C_{max}|^2 + |V|^2|C_{max}|\Delta_{max})$ time since the number of candidates at each iteration is bounded by $O(|V|^2)$. There are $O(|V|)$ iterations in total. Thus the total running time is bounded by $O(|V|^3C_{max}|^2 + |V^3||C_{max}|\Delta_{max})$. Note that C_{max} is usually a small constant. For our experimental instances the average size of an output cluster is usually almost equal

Table 4.1. Required CPU times in minutes for both algorithms executing on the IsoBase data for five networks. The BEAMS algorithm is executed with the parameter setting of $\beta = 0.4$.

	BEAMS	IsoRankN
$\alpha = 0.3$	65	1407
$\alpha = 0.4$	64	1511
$\alpha = 0.5$	62	1784
$\alpha = 0.6$	62	3619
$\alpha = 0.7$	69	7117

to k. Again assuming $\Delta_{max} = O(\Delta)$ the running time of this phase becomes $O(|V|^3)\Delta$. Since Δ^k is usually much larger than $|V|$, the execution time required for the initial phase of backbone extraction dominates that required by the backbone merging. With the reasonable assumption that $|V| = O(\Delta^k)$, we have that the running time of the BEAMS algorithm is $O(|V|^2\Delta^{k+1})$. We note that gains in running time such as those achieved via the branch-and-bound computations are not reflected in this upper bound and the actual execution time of the algorithm is actually much less than that represented in the bound. It is not possible to compare this formal bound with that of the IsoRankN algorithm, since no running time analysis is provided for IsoRankN. A major advantage of the BEAMS algorithm as compared to IsoRankN [32] is the speed of execution. We evaluated both algorithms in terms of their required CPU times on IsoBaSe [44], the database employed in the experimental evaluations of the section 5.1. We present the required CPU times for all the tested networks in Table 4.1. The required times are shown for each α setting employed in the experimental evaluations of the section 5.1. The average time required by IsoRankN over all α settings is 3487 minutes, almost 58 hours, whereas the average time required by BEAMS is almost one hour. These results are obtained by running both algorithms on an Intel(R) Xeon(R) CPU 2.67GHz with 24GB of memory.

5. DISCUSSION OF RESULTS

We implemented the BEAMS algorithm in C++ employing the LEDA library [45]. We experimented on both real and synthetic PPI networks. Regarding the former, we present a discussion of the global many-to-many alignment results for the PPI networks of five extensively studied species: *Caenorhabditis elegans* (worm), *Drosophila melanogaster* (fly), *Homo sapiens* (human), *Mus musculus* (mouse) and *Saccharomyces cerevisiae* (yeast). As input data, the BEAMS algorithm requires the PPI networks and the pairwise sequence similarity scores of aligned proteins. All this data is retrieved from the IsoBase [44] database which is the same as that used by the IsoRank, IsoRankN, and the SPINAL algorithms. These PPI networks are formed by combining the network data from various databases including DIP [46], BIOGRID [47], HPRD [48], MINT [49] and IntAct [50]. The *C. Elegans* network has 19756 proteins and 8639 interactions, the *D. Melanogaster* network has 14098 proteins and 49467 interactions, the *H.Sapiens* network has 22369 proteins and 105232 interactions, the *M. Musculus* network has 24855 proteins and 776 interactions, the *S. Cerevisiae* network has 6659 proteins and 164718 interactions, and in total there are 87737 proteins and 328832 interactions. Pairwise sequence similarity scores correspond to the BLAST Bit-values of the protein sequences retrieved from Ensembl [51]. With regards to the experimental results on synthetic data, we employed synthetic PPI networks retrieved from the NAPAbench [18]. It is a recently proposed network alignment benchmark intended mainly for a comparative study of different global many-to-many network alignment algorithms.

IsoRankN is one of the most popular algorithms in the global many-to-many network alignment literature. It has been shown that compared to other popular alignment algorithms such as Graemlin, NetworkBLAST-M, and MI-GRAAL, it provides better performance under measures suitable for network alignment quality determination [32, 18]. Furthermore the informal optimization goals of both IsoRankN and the BEAMS algorithms are quite similar in the sense that they both aim at maximizing a suitable optimization scoring function that balances the contribution of homological similarities of clustered proteins and the edge conservation between pairs of clusters via a suitably assigned constant α. We therefore extensively compare the BEAMS algorithm with IsoRankN. Herein we present the experimental results for different values of α varying from 0.3 to 0.7 in the increments of 0.1. The BEAMS algorithm has an additional user-defined parameter β, the filtering ratio, which is set to 0.4. Below we provide a detailed evaluation of the alignment results produced by the two algorithms. We present our experimental evaluations regarding these synthetic and actual networks separately in two sections.

5.1. Alignment of Actual PPI networks

In the next two sections, we first analyze the output clusters in terms of properties formalized in Section 2.1. Following this discussion we next provide an evaluation based on biological significance of the resulting alignments for actual PPI networks of five species.

5.1.1. Analysis of Output Clusters

Table 5.1 provides a summary of a quantitative analysis of the alignments produced by the BEAMS and the IsoRankN algorithms. For the first five multirows of the table, the top row corresponds to the number of generated clusters and the bottom row provides the total number of proteins in the output clusters. For a more detailed analysis, in addition to the total coverage values provided by all the clusters, we also provide a separate analysis by subdividing the output set based on the number of networks represented in the clusters. The first four rows provide these results for $c = 2, 3, 4, 5$ respectively where c denotes the number of networks in the clusters under consideration. It is easy to verify that the clusters produced by the BEAMS algorithm alignments has far better total coverage than those of the IsoRankN alignments; for each α, the BEAMS algorithm aligns almost 50% more proteins

Table 5.1. Analysis of Output Clusters

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$c = 2$	7251	7238	7242	7249	7245	0	0	0	0	0
	20540	20359	20419	20399	20392	0	0	0	0	0
$c = 3$	3259	3261	3277	3280	3277	4717	4716	4708	4714	4699
	12089	12187	12259	12286	12204	15891	15860	15827	15859	15807
$c = 4$	3281	3287	3283	3286	3291	3058	3052	3036	3035	3040
	16254	16353	16311	16322	16450	14651	14611	14540	14533	14550
$c = 5$	2090	2092	2081	2081	2074	2099	2101	2104	2084	2083
	13117	13094	13012	12978	12940	12834	12844	12868	12718	12697
Total	15881	15878	15883	15896	15887	9874	9869	9848	9833	9822
Cov.	62000	61993	62001	61985	61986	43376	43315	43235	43110	43054
Inter.	7060	7286	7425	7317	7407	5978	5956	6024	5653	5766
	114889	114919	114323	114839	114306	109364	108778	108374	107310	106642
	%6.15	%6.34	%6.49	%6.37	%6.48	%5.47	%5.48	%5.56	%5.27	%5.41
AS	0.5978	0.5175	0.4372	0.3563	0.2762	0.4909	0.4254	0.3606	0.2941	0.2288

than IsoRankN. Considering the clusters as claimed orthologies, this implies that the BEAMS algorithm leaves out much less unexplained data by proposing orthology relations for most of the proteins. Out of all the 87737 proteins, around 62000 are assigned to clusters by our algorithm. The main reason behind this discrepancy is the lack of IsoRankN clusters containing only proteins from two networks. Such a deficiency may lead to unreasonable conclusions, as it is quite natural to expect orthologous groups with proteins from only two species given that the pairwise evolutionary distances of the species under consideration have large variations.

The top row in the multirow indicated with *Inter.* provides the number of *conserved interactions* resulting from the output alignments, the middle row indicates the total number of interactions between clusters, and the bottom row provides their ratios. A protein-protein interaction is assumed to be conserved if its *cs* score is greater than 0, that is the interaction is between a pair of proteins from different clusters which further contain at least one more pair of interacting proteins from another PPI network. The number of conserved interactions is a common performance indicator employed in the alignment studies since it is a measure of the topology conservation achieved by the alignment. For all instances of α the BEAMS algorithm provides more conserved interactions than IsoRankN. Furthermore this superiority is not simply due to the large number of clusters produced by the BEAMS alignments; considering the ratio of the number of conserved interactions to the total number of interactions between clusters, it can be observed that the BEAMS alignments conserve a larger ratio of existing edges between all clusters. Finally, the last row of the table provides the AS score of alignments as defined in Equation 2.1. Comparing the scores under corresponding α values, the AS scores of BEAMS is larger than those of IsoRankN in all cases.

5.1.2. Evaluations based on Biological Significance

Similar to previous PPI network alignment studies, our biological significance evaluations are based on the hierarchical GO categorization, where proteins are annotated with appropriate GO categories organized as a directed acyclic graph (DAG) [52]. In order to standardize the GO annotations of proteins, similar to the evaluation methods of [14, 32, 28], we restrict the protein annotations to level 5 of the GO DAG by ignoring the higher-level annotations and replacing the deeper-level category annotations with their ancestors at the restricted level. The protein annotations are used to measure the consistency of generated clusters. A cluster is *annotated* if at least two of its proteins are annotated by some GO categories. An annotated cluster is considered *consistent* if all of its proteins share at least one common standard GO annotation. The consistency evaluations of the BEAMS and the IsoRankN alignments are provided in the first five multirows of Table 5.2. The top row in each of these multirows indicates the number of annotated clusters, the middle row provides the number of consistent clusters, and finally the bottom row indicates the ratio of consistent clusters to annotated clusters. This ratio for all the clusters altogether is shown

Table 5.2. Biological Significance Evaluations.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$c = 2$	2150	2143	2147	2139	2132	0	0	0	0	0
	1997	1992	1997	1992	1985	0	0	0	0	0
	%92.9	%92.9	%93.0	%93.1	%93.1	-	-	-	-	-
$c = 3$	1791	1787	1792	1786	1784	2523	2516	2524	2528	2524
	1478	1469	1479	1468	1466	1926	1924	1938	1944	1943
	%82.5	%82.2	%82.5	%82.2	%82.2	%76.3	%76.5	%76.8	%76.9	%77.0
$c = 4$	2497	2503	2499	2503	2517	2275	2272	2253	2252	2255
	1843	1852	1840	1842	1853	1616	1613	1608	1606	1601
	%73.8	%74.0	%73.6	%73.6	%73.6	%71.0	%71.0	%71.4	%71.3	%71.0
$c = 5$	1971	1974	1961	1962	1954	1958	1960	1963	1941	1943
	1375	1382	1384	1382	1371	1309	1308	1305	1293	1298
	%69.8	%70.0	%70.6	%70.4	%70.2	%66.9	%66.7	%66.5	%66.6	%66.8
$Total$	8409	8407	8399	8390	8387	6756	6748	6740	6721	6722
	6693	6695	6700	6684	6675	4851	4845	4851	4843	4842
$Specificity$	79.59	79.64	79.77	79.67	79.59	71.80	71.80	71.97	72.06	72.03
$Sensitivity$	22231	22258	22304	22234	22218	16350	16333	16334	16315	16301
$Relative$ $Sensitivity$	7473	7468	7497	7507	7495	1592	1543	1527	1588	1578
MNE	1.2881	1.2908	1.2902	1.2909	1.2890	1.4685	1.4679	1.4672	1.4682	1.4672
$NGOC$	0.3093	0.3075	0.3086	0.3097	0.3096	0.2413	0.2410	0.2424	0.2427	0.2422

as a separate row indicated by *specificity* to be consistent with the terminology employed in previous alignment studies [18]. Considering the complete set of annotated clusters, it is clear that the BEAMS alignments outperform those of IsoRankN in terms of the number of consistent clusters. Furthermore the aligned clusters are more specific than those produced by IsoRankN. To measure how sensitive the provided alignment results are, we employ the *sensitivity* definition as in [18]. It represents the total number of annotated proteins in all the consistent clusters. Additionally, we provide an alternative sensitivity definition, *relative sensitivity*. A relative sensitivity value shown under a BEAMS column provides the number of annotated proteins in consistent clusters in a BEAMS alignment and in inconsistent clusters in an IsoRankN alignment under the same α settings. The relative sensitivity value under an IsoRankN column provides the exact opposite. The BEAMS alignments provide much better sensitivity and relative sensitivity than those of IsoRankN. This is especially evident with the relative sensitivity measure; taking the average over all α settings BEAMS has a relative sensitivity that is almost five times better than that of IsoRankN. In other words, the proteins aligned into consistent clusters by BEAMS but not by IsoRankN is far more than the exact opposite.

Mean normalized entropy (MNE) is another consistency evaluation metric employed in previous studies [32, 28]. The normalized entropy of an annotated cluster Cl_x is defined as $NE(Cl_x) = -\frac{1}{\log d} \times \sum_{i=1}^{d} p_i \times \log p_i$, where p_i is the fraction of proteins in Cl_x with the annotation GO_i, and d represents the number of different GO annotations in Cl_x. For MNE the sum of these values are averaged over the total number of annotated clusters. Note that lower MNE values indicate better consistency. Yet another consistency evaluation metric is *GO consistency (GOC)* defined in [28]. Since GOC is defined for the one-to-one alignment of a pair of networks, we extend the definition to many-to-many alignments of multiple networks by normalizing the score. For an annotated cluster Cl_x let $GO_{int}(Cl_x)$ and $GO_{uni}(Cl_x)$ indicate respectively the intersection set of GO annotations of proteins in Cl_x and the union set of GO annotations of all the proteins in Cl_x. The normalized GOC score denoted with $nGOC$ is defined as the weighted mean of $|GO_{int}|/|GO_{uni}|$ over all annotated clusters, where the weight of each cluster is the number of annotated proteins it contains. In terms of better consistency larger $nGOC$ values are more desirable. With respect to both metrics, MNE and $nGOC$, the BEAMS algorithm clearly outperforms IsoRankN.

In addition to these evaluation metrics, intended to measure biological significance of output alignments, we also provide a specific clustering instance resulting from the alignments of BEAMS and IsoRankN on the same dataset. Figure 5.1 illustrates the specific

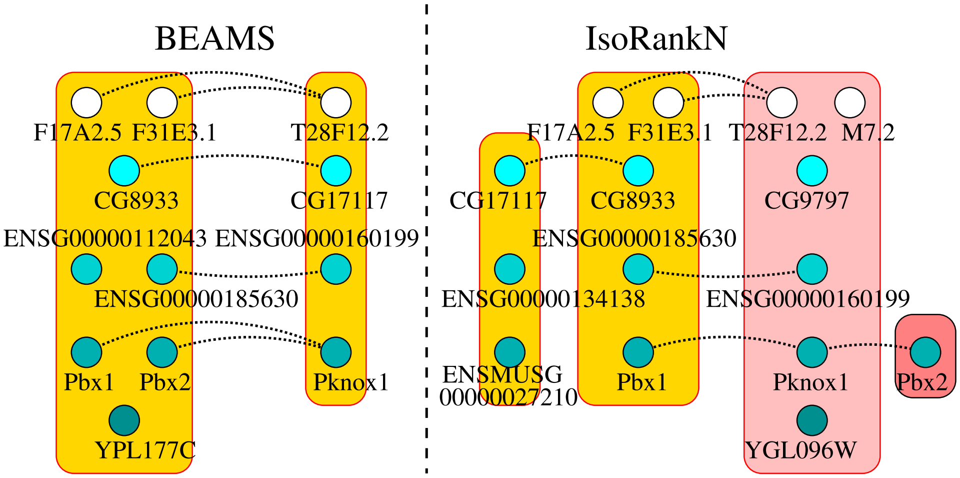


Figure 5.1. Comparative visualization of a sample clustering produced by the BEAMS and the IsoRankN algorithms running on the IsoBase data. Both clusters of the BEAMS alignment are consistent. Only the two leftmost clusters of the IsoRankN alignment are consistent.

clusters. The alignments are obtained under the setting of $\alpha = 0.5$ for both algorithms. Two clusters from the alignment of BEAMS, one with eight proteins from five networks and one with four proteins from four networks, are depicted. Former cluster includes the proteins F17A2.5, F31E3.1, CG8933, ENSG00000185630, ENSG00000112043, Pbx2, Pbx1, and YPL177C. This cluster is consistent since all its annotated proteins share the same GO annotation, GO:0006355, *regulation of transcription, DNA-dependent*. The second cluster includes the proteins T28F12.2, CG17117, ENSG00000160199, and Pknox1. This cluster is also consistent; all annotated proteins share the annotation GO:0043565, *sequence specific DNA binding*. We note that not only the pair of clusters are consistent with respect to a low-level GO annotation, indicating a high functional orthology among the members, but also provides high interaction conservation; all six interactions between these clusters are conserved with the maximum possible conservation score of 1. On the other hand, the clusters from the IsoRankN alignment including respective proteins are defective in several ways. Proteins T28F12.2, ENSG00000160199, and Pknox1 are clustered into an inconsistent cluster with three other proteins, and protein Pbx2 is not aligned with any other protein by the IsoRankN algorithm. The other proteins of this inconsistent cluster, M7.2, CG9797, and YGL096W, are aligned into a consistent cluster by BEAMS which is not depicted in the figure for compactness. The clustering produced by the IsoRankN alignment further suffers from poor interaction conservation. Out of the six conserved by the BEAMS alignment, four are conserved by IsoRankN with a conservation score of 0.75, one is not conserved, and one is not included in the alignment at all. All this facts indicates the superiority of BEAMS algorithm over IsoRankN when these proteins and their interactions are considered.

5.2. Alignment of Synthetic PPI Networks

The *Network Alignment Performance Assessment Benchmark* (NAPAbench) is a recently proposed network alignment benchmark intended mainly for a comparative study of different global network alignment algorithms [18]. Three different datasets are provided for the pairwise, 5-way, and 8-way alignments each standing for the alignment of two, five, and eight networks respectively. For each dataset, there are three different network families that are generated with different network growth heuristics: *Crystal growth* (CG) model, *duplication-mutation-complementation* (DMC) model, and *duplication with random mutation* DMR model. We present experimental evaluations of global many-to-many alignment results on 8-way alignment datasets for each network family. Experiments are performed under different settings of α varying from 0.3 to 0.7 in the increments of 0.1, whereas β is fixed to 0.2 for the BEAMS algorithm. Note that for the experiments applied on the IsoBase data presented in the previous section we employed the $\beta = 0.4$ setting. This discrepancy stems from the fact that the NAPAbench networks are completely synthetic and fewer sequence similarity data should be filtered out compared to alignments on actual networks of IsoBase. We present the quantitative information and the functional consistency evaluations of the resulting alignments in two tables per network family. Each pair of tables are similar to the pair presented in the previous section for the IsoBase evaluations, that is the rows and columns represent analogous data. Note that functional consistency of synthetic network alignments corresponds to the biological significance of actual PPI network alignments presented in the previous section. Functional group id assignments, again synthetically constructed within the NAPAbench data, are used for the functional consistency evaluations. Since these functional groups do not have any hierarchical organization, all functional group ids are treated as if they belong to the standard level.

All eight synthetic networks of the CG family have the same size; each has 1000 proteins and 3985 interactions. In the DMC and DMR network families, networks have 1000 proteins, whereas number of interactions vary. In the DMC family, the number of interactions for each network are 1919, 1853, 1923, 1840, 1867, 1848, 1818, and 1867. In the DMR family, the number of interactions are 2031, 2092, 1967, 1977, 1959, 1998, 2030, and 2056. Similar to the evaluations of actual PPI networks of IsoBase, we first provide the quantitative analysis of the alignment results in Tables 5.3- 5.5 for the CG, DMC, and DMR alignments respectively. It is easy to verify that the clusters produced by the BEAMS algorithm alignments has better total coverage than those of the IsoRankN alignments on all network families. This indicates the ability of the BEAMS algorithm to explain more data. For the alignment of the CG family, BEAMS conserves more interactions than IsoRankN. Note that this superiority in terms of interaction conservation which affects the resulting AS scores in turn, does not hold for the DMC and DMR network family alignments; IsoRankN provides alignments with larger interaction conservation. This discrepancy is mainly due to the sizes of output clusters and the synthetic nature of the employed data. Generated clusters of the BEAMS alignments have an average size of 6.1, whereas IsoRankN clusters have an average size of 9.2. Interaction conservation is trivially proportional to the sizes of output clusters; the larger the clusters, the better the interaction conservation. A second implication of large interaction conservation is the larger AS scores achieved by the IsoRankN alignments as compared to those of BEAMS. Note that under normal circumstances large clusters would be expected to decrease the ICQ scores representing the normalized homological similarity of the proteins within each cluster, which would in turn balance out the affects of larger interaction conservation and finally lead to similar overall AS scores. However due to the synthetic nature of the employed data, the ICQ scores of the alignment are underrated as far as their contribution to the AS score; as α approaches 0, the AS scores of the alignments become almost equal. Note large output clusters, although may lead to good interaction conservation and in turn to higher AS scores in some cases, have the potential deficiency of misleading results by including mostly inconsistent members. This actually is the case for the networks under study and is discussed as part of the functional consistency evaluations described next.

For the functional consistency comparison of the two algorithms' alignment results, we present the evaluations of performed alignments in Tables 5.6- 5.8 for the CG, DMC, and the DMR families respectively. All evaluation metrics defined in the previous section are computed and the functional consistency evaluation tables similar to the biological significance evaluation table are provided. By inspecting the tables, it can immediately be verified that, the BEAMS alignments are far more consistent than those of IsoRankN. Furthermore BEAMS alignments are more specific and sensitive. Especially the BEAMS algorithm outperforms IsoRankN with regards to the specificity of the alignments, which is at least 80 for the BEAMS alignments, whereas it is at most 67 for those of IsoRankN. Sensitivity, that is the number of annotated proteins assigned to a consistent cluster, of the BEAMS alignment is also %50 larger than that of IsoRankN alignments on average. Moreover, the relative sensitivity of the BEAMS alignments are 7 times better than those of IsoRankN on average. Finally, BEAMS clearly outperforms IsoRankN by the overall quality of generated clusters measured through the evaluation metrics, MNE and NGOC. The gap between the MNE and NGOC score of two algorithms' alignments is significantly large which indicates the superiority of the BEAMS algorithm over IsoRankN when generating functionally consistent clusters.

Table 5.3. Analysis of Output Clusters for CG Network Family.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	207	200	203	198	196	0	0	0	0	0
	569	548	543	528	525	0	0	0	0	0
$k = 3$	184	182	187	188	200	184	199	190	181	169
	680	661	695	691	740	645	709	676	651	609
$k = 4$	189	187	179	182	182	128	103	101	122	129
	923	895	857	865	860	706	565	552	647	702
$k = 5$	127	131	138	136	126	59	74	76	65	56
	728	763	797	788	730	440	516	514	432	369
$k = 6$	115	113	110	114	113	79	67	60	54	55
	786	778	760	796	785	749	633	566	519	479
$k = 7$	81	84	85	79	83	49	59	66	72	65
	625	657	669	600	647	608	743	840	897	765
$k = 8$	360	361	360	361	362	277	275	280	282	299
	3477	3484	3430	3508	3493	4246	4237	4251	4278	4488
Total	1263	1258	1262	1258	1262	776	777	773	776	773
Coverage	7788	7786	7788	7776	7780	7394	7403	7399	7424	7412
Interactions	24950	25119	25113	25096	25167	22416	22349	22512	22821	22972
	29687	29676	29709	29657	29736	25491	25552	25618	25865	25848
	%84.0	%84.6	%84.5	%84.6	%84.6	%87.9	%87.5	%87.9	%88.2	%88.9
AS	0.5704	0.5888	0.6059	0.6289	0.6480	0.5096	0.5339	0.5645	0.5952	0.6305

Table 5.4. Analysis of Output Clusters for DMC Network Family.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	210	218	218	213	224	0	0	0	0	0
	577	592	599	583	614	0	0	0	0	0
$k = 3$	169	165	164	168	154	185	197	183	173	168
	621	601	594	606	548	624	673	626	605	572
$k = 4$	184	177	173	187	175	141	142	150	154	150
	892	873	838	903	842	752	751	795	834	792
$k = 5$	134	134	135	141	144	66	71	78	93	104
	774	750	781	792	813	468	500	540	629	716
$k = 6$	150	158	162	154	150	77	89	95	85	85
	1037	1092	1110	1053	1007	741	872	952	839	859
$k = 7$	101	94	101	105	110	76	68	75	85	82
	764	724	775	791	828	893	836	906	987	951
$k = 8$	336	338	332	329	331	271	266	254	244	239
	3127	3155	3096	3071	3127	3935	3790	3624	3513	3478
Total	1284	1284	1285	1297	1288	816	833	835	834	838
Coverage	7792	7787	7793	7799	7779	7413	7422	7443	7407	7368
Interactions	9690	9979	9971	9959	9948	11174	11144	11145	11106	11019
	14375	14373	14419	14406	14415	13200	13364	13345	13294	13229
	%67.4	%67.4	%69.2	%69.1	%69.0	%84.7	%83.4	%83.5	%83.5	%83.3
AS	0.4932	0.5007	0.5007	0.5037	0.5023	0.5092	0.5269	0.5470	0.5740	0.5938

Table 5.5. Analysis of Output Clusters for DMR Network Family.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	250	243	245	247	262	0	0	0	0	0
	691	654	650	668	705	0	0	0	0	0
$k = 3$	173	175	178	176	177	214	205	200	214	188
	633	638	630	633	648	755	709	715	751	668
$k = 4$	188	182	192	188	191	137	126	122	112	130
	863	841	893	875	889	702	681	640	603	700
$k = 5$	122	115	123	125	104	72	80	84	88	82
	701	660	718	715	584	502	544	576	615	598
$k = 6$	116	130	123	115	135	70	78	80	79	94
	756	859	805	762	881	630	682	698	695	797
$k = 7$	98	97	98	93	97	69	75	75	85	79
	770	779	761	732	743	830	955	961	1003	909
$k = 8$	361	362	357	363	357	266	260	263	256	255
	3416	3396	3375	3445	3391	4036	3864	3853	3779	3725
Total	1308	1304	1316	1307	1323	828	824	824	834	828
Coverage	7830	7827	7832	7830	7841	7455	7445	7443	7446	7397
Interactions	10477	10395	10389	10341	10577	12037	12042	12205	12081	12184
	15740	15756	15774	15733	15766	14807	14835	14750	14802	14692
	%66.6	%66.0	%65.9	%65.7	%67.1	%81.3	%81.2	%82.7	%81.6	%82.9
AS	0.4899	0.4831	0.4819	0.4808	0.4790	0.4960	0.5123	0.5426	0.5616	0.5883

Table 5.6. Functional Consistency Evaluation for the Alignment of CG Network Family.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	123	112	113	112	108	0	0	0	0	0
	103	94	94	89	87	0	0	0	0	0
	%83.7	%83.9	%83.2	%79.5	%80.5	-	-	-	-	-
$k = 3$	181	176	181	183	198	174	188	178	168	153
	151	146	146	155	169	120	128	113	106	104
	%83.4	%82.9	%80.7	%84.7	%85.3	%69	%68.1	%63.5	%63.1	%68.0
$k = 4$	189	187	178	181	182	125	101	97	118	126
	150	156	148	150	148	86	73	72	87	95
	%79.4	%83.4	%83.1	%82.9	%81.3	%68.8	%72.3	%74.2	%73.7	%75.4
$k = 5$	127	131	138	136	126	59	73	75	64	56
	102	107	108	107	102	38	45	49	39	33
	%80.3	%81.7	%78.3	%78.7	%80.9	%64.4	%61.6	%65.3	%61.0	%59.0
$k = 6$	115	113	110	114	113	79	67	60	54	55
	88	91	91	93	89	50	47	43	41	37
	%76.5	%80.5	%82.7	%81.6	%78.8	%63.3	%70.1	%71.7	%76.0	%67.3
$k = 7$	81	84	85	79	83	49	59	66	72	65
	76	80	82	74	78	28	30	35	37	42
	%93.8	%95.2	%96.5	%93.7	%94.0	%57.1	%50.8	%53.0	%51.3	%64.6
$k = 8$	360	361	360	361	362	277	275	280	282	299
	354	356	355	360	359	162	160	157	158	164
	%98.3	%98.6	%98.6	%99.7	%99.2	%58.5	%58.2	%56.1	%56.0	%54.8
$Total$	1176	1164	1165	1166	1172	763	763	756	758	754
	1024	1030	1024	1028	1032	484	483	469	468	475
$Specifity$	87.07	88.49	87.90	88.16	88.05	63.43	63.30	62.04	61.74	63.00
$Sensitivity$	6508	6580	6569	6596	6592	4078	4054	3992	3965	4021
$Relative$ $Sensitivity$	2707	2784	2838	2881	2825	277	258	261	250	254
MNE	0.1176	0.1047	0.1116	0.1089	0.1097	0.2898	0.2855	0.2924	0.2893	0.2750
$NGOC$	0.9008	0.9125	0.9101	0.9129	0.9128	0.5949	0.5901	0.5823	0.5765	0.5857

Table 5.7. Functional Consistency Evaluation for the Alignment of DMC Network Family.

	BEAMS					IsoRankN				
α	0.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	118	129	127	119	134	0	0	0	0	0
	90	93	87	86	97	0	0	0	0	0
	%76.3	%72.1	%68.5	%72.3	%72.4	-	-	-	-	-
$k = 3$	165	160	161	163	152	173	186	176	163	158
	134	133	132	134	125	111	129	126	109	111
	%81.2	%83.1	%82.0	%82.2	%82.2	%64.2	%69.3	%71.6	%66.9	%70.2
$k = 4$	184	177	173	187	175	141	142	149	154	150
	147	145	140	146	146	121	116	119	122	123
	%79.9	%81.9	%80.9	%78.1	%83.4	%85.8	%81.7	%79.9	%79.2	%82.0
$k = 5$	134	134	135	141	144	66	71	78	91	104
	111	109	111	118	119	43	51	58	64	64
	%82.8	%81.3	%82.2	%83.7	%82.6	%65.1	%71.8	%74.4	%70.3	%61.5
$k = 6$	150	158	162	154	150	77	89	95	85	85
	108	112	115	111	108	50	54	62	55	58
	%72.0	%70.9	%80.0	%72.1	%72.0	%64.9	%60.7	%65.3	%64.7	%68.2
$k = 7$	101	94	101	105	110	76	68	75	85	82
	77	70	72	76	84	50	41	40	52	49
	%76.2	%74.5	%71.3	%72.4	%76.4	%65.8	%60.3	%53.3	%61.2	%59.8
$k = 8$	336	338	332	329	331	271	266	254	244	239
	299	300	303	298	298	158	156	148	144	137
	%89.0	%88.7	%91.3	%90.6	%90.0	%58.3	%58.6	%	58.3 %59.0	%57.3
$Total$	1188	1190	1191	1198	1196	804	822	827	822	818
	966	962	960	969	977	533	547	553	546	542
$Specifity$	81.31	80.84	80.60	80.88	81.69	66.29	66.54	66.87	66.42	66.26
$Sensitivity$	6027	5996	6005	6009	6048	4242	4205	4200	4244	4173
$Relative$ $Sensitivity$	2234	2278	2288	2238	2322	449	487	483	473	447
MNE	0.1736	0.1799	0.1817	0.1782	0.1708	0.2625	0.2567	0.2506	0.2561	0.2544
$NGOC$	0.8356	0.8311	0.8326	0.8331	0.8391	0.6194	0.6123	0.6098	0.6178	0.6103

Table 5.8. Functional Consistency Evaluation for the Alignment of DMR Network Family.

	BEAMS					IsoRankN				
α	.3	0.4	0.5	0.6	0.7	0.3	0.4	0.5	0.6	0.7
$k = 2$	127	118	115	117	136	0	0	0	0	0
	94	92	87	88	107	0	0	0	0	0
	%74.0	%78	%75.6	%75.2	%78.7	-	-	-	-	-
$k = 3$	162	164	167	164	166	189	183	179	195	170
	130	133	133	133	130	127	120	120	130	122
	%80.2	%81.1	%79.6	%81.1	%78.3	%67.2	%65.6	%67.0	%66.7	%71.8
$k = 4$	187	180	191	186	190	133	124	119	112	128
	140	139	143	144	147	104	90	88	83	94
	%74.9	%77.2	%74.9	%77.4	%77.4	%78.2	%72.6	%73.9	%74.1	%73.4
$k = 5$	122	115	123	125	104	72	79	84	88	81
	102	93	102	105	81	42	50	58	61	52
	%83.6	%80.9	%82.9	%84.0	%77.9	%58.3	%63.3	%69.0	%69.3	%64.2
$k = 6$	116	130	123	115	135	70	77	79	78	94
	84	95	93		85	93	43	47	48	63
	%72.4	%73.1	%75.6	%73.9	%68.9	%61.4	%61.0	%60.8	%61.5	%67.0
$k = 7$	98	97	98	93	97	69	75	75	85	79
	76	72	69	66	73	35	34	36	39	43
	%77.5	%74.2	%70.4	%71.0	%75.3	%50.7	%45.3	%48.0	%45.9	%54.4
$k = 8$	361	362	357	363	357	266	260	263	256	255
	328	331	330	332	323	147	150	157	158	140
	%90.9	%91.4	%92.4	%91.5	%90.5	%55.3	%57.7	%59.7	%61.7	%54.9
$Total$	1173	1166	1174	1163	1185	799	798	799	814	807
	954	955	957	953	954	498	491	507	519	514
$Specifity$	81.33	81.90	81.52	81.94	80.51	62.33	61.53	63.45	63.76	63.69
$Sensitivity$	5997	6006	6007	6025	5928	3786	3815	3976	4050	3927
$Relative$ $Sensitivity$	2653	2660	2544	2504	2545	442	469	513	529	544
MNE	0.1727	0.1678	0.1711	0.1691	0.1813	0.2860	0.2917	0.2792	0.2813	0.2721
$NGOC$	0.8389	0.8416	0.8414	0.8447	0.8294	0.5549	0.5597	0.5838	0.5934	0.5789

6. CONCLUSION AND FUTURE RESEARCH

6.1. Conclusion

Frst of all, with this thesis, we provide the first formal combinatorial definition in the literature for the problem of global many-to-many network alignment of multiple PPI networks and it is an important task to turn biological problem into a combinatorial problem for better analyses. We proceed with another important study, proving the computational intractability of this problem even in a quite restricted case. We next propose a new algorithm BEAMS for the solution of the problem and we experimentally evaluate our proposed algorithm with regards to several biological significance metrics proposed in literature. The results indicate that BEAMS algorithm generates highly reliable protein clusters and most of these generated clusters are biologically consistent. We also compare this new algorithm against one of the most popular global many-to-many alignment methods, IsoRankN. The experimental results indicate that BEAMS algorithm outperforms IsoRankN in generating more consistent clusters. Furthermore, considering the heavy computational load of the problem, the exceptional running time of BEAMS algorithm as compared to that of IsoRankN can be considered as another important improvement of BEAMS algorithm.

6.2. Future Research

BEAMS algorithm is proven to be the state-of-the-art algorithm for the global many-to-many alignment of multiple PPI networks but still it can be improved by some future researches. Instead of using only sequence similarities in the similarity graph S, different methods could be developed for this similarity score computation. If this score is computed through some measure of functional similarity, this could increase the performance of BEAMS algorithm. Additionally, since the BEAMS algorithm has been developed heuristically, some other heuristic strategies could be developed for the solution to the problem. Besides, backbone extraction and merging problems could also be handled by some other heuristic strategies, too. If these heuristic strategies perform well within the BEAMS algorithm, overall quality of the alignments that is generated with the algorithm would increase.

REFERENCES

1. R. Sharan and T. Ideker. Modeling Cellular Machinery Through Biological Network Comparison. *Nature Biotechnology*, 24(4):427–433, April 2006.

2. B. Alberts, A. Johnson, J. Lewis, M. Raff, D. Bray, K. Hopkin, K. Roberts, and P. Walter. *Essential Cell Biology.* Garland Science/Taylor & Francis Group, 2nd edition, September 2003.

3. G. Fang, N. Bhardwaj, R. Robilotto, and M. B. Gerstein. Getting Started in Gene Orthology and Functional Analysis. *PLoS Comput Biol*, 6(3):e1000703+, March 2010.

4. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, October 1990.

5. A. M. Altenhoff and C. Dessimoz. Phylogenetic and Functional Assessment of Orthologs Inference Projects and Methods. *PLoS Comput Biol*, 5(1):e1000262+, January 2009.

6. R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A Genomic Perspective on Protein Families. *Science*, 278(5338):631–637, October 1997.

7. M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons. *Journal of Molecular Biology*, 314(5):1041–1052, December 2001.

8. L. Li, C. J. Stoeckert, and D. S. Roos. OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. *Genome Research*, 13(9):2178–2189, September 2003.

9. C. Dessimoz, G. Cannarozzi, M. Gil, D. Margadant, A. Roth, A. Schneider, and G. H. Gonnet. Oma, a comprehensive, automated project for the identification of orthologs from complete genome data: Introduction and first achievements. *Lecture Notes in Computer Science*, 3678, 2005.

10. T. F. DeLuca, I-Hsien Wu, J. Pu, T. Monaghan, L. Peshkin, S. Singh, and D. P. Wall. Roundup: A Multi-Genome Repository of Orthologs and Evolutionary Distances. *Bioinformatics*, 22(16):2044–2046, August 2006.

11. D. L. Wheeler, D. M. Church, S. Federhen, A. E. Lash, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, E. Sequeira, T. A. Tatusova, and L. Wagner. Database Resources of the National Center for Biotechnology. *Nucleic Acids Research*, 31(1):28–33, January 2003.

12. T. J. Hubbard, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, J. Herrero, R. Holland, K. Howe, K. Howe, N. Johnson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, C. Melsopp, K. Megy, P. Meidl, B. Ouverdin, A. Parker, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, J. Severin, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. Vilella, J. Vogel, S. White, M. Wood, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, A. Kasprzyk, G. Proctor, S. Searle, J. Smith, A. Ureta-Vidal, and E. Birney. Ensembl 2007. *Nucleic Acids Research*, 35(Database issue), January 2007.

13. L. J. J. Jensen, P. Julien, M. Kuhn, C. von Mering, J. Muller, T. Doerks, and P. Bork. eggNOG: Automated Construction and Annotation of Orthologous Groups of Genes. *Nucleic Acids Research*, 36(Database issue):D250–254, January 2008.

14. 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*, 105(35):12763–12768, September 2008.

15. J. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L.V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal. Evidence for Dynami-

cally Organized Modularity in the Yeast Protein-Protein Interaction Network. *Nature*, 430(6995):88–93, July 2004.

16. E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-Proteome Prediction of Protein Function via Graph-Theoretic Analysis of Interaction Maps. *Bioinformatics*, 21(suppl 1):i302–i310, June 2005.

17. S. H. Yook, Z. N. Oltvai, and A. L. Barabási. Functional and Topological Characterization of Protein Interaction Networks. *Proteomics*, 4(4):928–942, April 2004.

18. S. M. Sahraeian and B. J. Yoon. A Network Synthesis Model for Generating Protein Interaction Network Families. *PLoS ONE*, 7(8):e41474+, August 2012.

19. D. R. Rhodes, S. A. Tomlins, S. Varambally, V. Mahavisno, T. Barrette, S. Kalyana-Sundaram, D. Ghosh, A. Pandey, and A. M. Chinnaiyan. Probabilistic Model of the Human Protein-Protein Interaction Network. *Nat Biotech*, 23(8):951–959, August 2005.

20. B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved Pathways within Bacteria and Yeast as Revealed by Global Protein Network Alignment. *Proceedings of the National Academy of Sciences*, 100(20):11394–11399, September 2003.

21. 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*, 32(Web Server issue), July 2004.

22. R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson. Alignment of Metabolic Pathways. *Bioinformatics*, 21(16):3401–3408, August 2005.

23. T. Shlomi, D. Segal, E. Ruppin, and R. Sharan. QPath: A Method for Querying Pathways in a Protein-Protein Interaction Network. *BMC Bioinformatics*, 7, 2006.

24. B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. QNet: A Tool for Querying Protein Interaction Networks. *Research in Computational Molecular Biology*, pages 1–15, 2007.

25. S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan. Topology-free Querying of Protein Interaction Networks. *Journal of Computational Biology*, 17(3):237–252, March 2010.

26. X. Qian, S. H. Sze, and B. J. Yoon. Querying Pathways in Protein Interaction Networks based on Hidden Markov Models. *Journal of Computational Biology*, 16(2):145–157, February 2009.

27. L. Chindelevitch, C. S. Liao, and B. Berger. Local optimization for global alignment of protein interaction networks. In *Pacific Symposium on Biocomputing*, pages 123–132, 2010.

28. A. E. Aladag and C. Erten. SPINAL: Scalable Protein Interaction Network Alignment. *Bioinformatics*, 29(7):917–924, April 2013.

29. Ferhat Ay, Manolis Kellis, and Tamer Kahveci. Submap: aligning metabolic pathways with subnetwork mappings. *Journal of Computational Biology*, 18(13):219–235, 2011.

30. Gamze Abaka, Turker Biyikoglu, and Cesim Erten. Campways: constrained alignment framework for the comparative analysis of a pair of metabolic pathways. *Bioinformatics*, 29(13):i145–i153, 2013.

31. Jason Flannick, Antal Novak, Chuong B. Do, Balaji S. Srinivasan, and Serafim Batzoglou. Automatic parameter learning for multiple network alignment. In *Proceedings of the 12th annual international conference on Research in computational molecular biology*, RECOMB'08, pages 214–231, Berlin, Heidelberg, 2008. Springer-Verlag.

32. C. S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. IsoRankN: Spectral Methods for

Global Alignment of Multiple Protein Networks. *Bioinformatics*, 25(12):i253–i258, June 2009.

33. R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved Patterns of Protein Interaction in Multiple Species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1974–1979, February 2005.

34. 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*, 16(9):1169–1181, September 2006.

35. J. Berg and M. Lässig. Cross-Species Analysis of Biological Networks by Bayesian Alignment. *Proceedings of the National Academy of Sciences*, 103(29):10967–10972, July 2006.

36. M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama. Pairwise Alignment of Protein Interaction Networks. *Journal of Computational Biology*, 13(2):182–199, March 2006.

37. M. Narayanan and R. M. Karp. Comparing Protein Interaction Networks via a Graph Match-and-Split Algorithm. *Journal of Computational Biology*, 14(7):892–907, September 2007.

38. M. Zaslavskiy, F. Bach, and J. P. Vert. Global Alignment of Protein-Protein Interaction Networks by Graph Matching Methods. *Bioinformatics*, 25(12):i259–1267, June 2009.

39. G. W. Klau. A New Graph-based Method for Pairwise Global Network Alignment. *BMC Bioinformatics*, 10 Suppl 1, 2009.

40. M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang. Algorithms for Large, Sparse Network Alignment Problems. *Data Mining, IEEE International Conference on*, 0:705–710, 2009.

41. O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj. Topological Network Alignment Uncovers Biological Function and Phylogeny. *Journal of The Royal Society Interface*, 7(50):1341–1354, March 2010.

42. T. Milenković, W. L. L. Ng, W. Hayes, and N. Przulj. Optimal Network Alignment with Graphlet Degree Vectors. *Cancer Informatics*, 9:121–137, 2010.

43. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1st edition, January 1979.

44. D. Park, R. Singh, M. Baym, C. S. S. Liao, and B. Berger. IsoBase: A Database of Functionally Related Proteins Across PPI Networks. *Nucleic Acids Research*, 39(Database issue):D295–D300, January 2011.

45. K. Mehlhorn and S. Näher. *LEDA: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, November 1999.

46. L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32:449–451, 2004.

47. B. J. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bahler, V. Wood, and et al. The biogrid interaction database: 2008 update. *Nucleic Acids Research*, 36(Database-Issue):637–640, 2008.

48. T. S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, and et al. Human protein reference database-2009 update. *Nucleic Acids Research*, 37(Database-Issue):767–772, 2009.

49. A. Ceol, A. Chatr Aryamontri, L. Licata, D. Peluso, L. Briganti, L. Perfetto, L. Castagnoli, and G. Cesareni. Mint, the molecular interaction database: 2009 update. *Nucleic Acids Research*, 38(Database-Issue):532–539, 2010.

50. B. Aranda, P. Achuthan, Y. Alam-Faruque, I. Armean, A. Bridge, C. Derow, and et al. The intact molecular interaction database in 2010. *Nucleic Acids Research*, 38(Database-Issue):525–531, 2010.

51. T. J. Hubbard, B.L. Aken, S. Ayling, B. Ballester, K. Beal, E. Bragin, S. Brent, Y. Chen, P. Clapham, L. Clarke, and et al. Ensembl 2009. *Nucleic Acids Research*, 37(Database-Issue):690–697, 2009.

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*, 25(1):25–29, May 2000.

Curriculum Vitae

Ferhat Alkan was born in 1988 at Dinar, Afyon. He graduated from Kuşadası Derici Mustafa Gürbüz Anatolian High School in 2005. Then, he enrolled into İstanbul Technical University in 2005 and graduated in 2010. He has a bachelor degree of Telecommunication Engineering. After a year in military service, he continued on his education in Kadir Has University at the graduate program of Computer Engineering. He also worked as a research and teaching assistant in Kadir Has University. His main research interests are combinatorial problems, bioinformatics and graph theory.