12-16-2014 12:00 AM

Protein-Protein Interaction Network Alignment

Yu Qian, *The University of Western Ontario*

Supervisor: Dr. Lucian Ilie, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Recommended Citation

Qian, Yu, "Protein-Protein Interaction Network Alignment" (2014). *Electronic Thesis and Dissertation Repository*. 2579.
https://ir.lib.uwo.ca/etd/2579

PROTEIN-PROTEIN INTERACTION NETWORK ALIGNMENT

(Thesis format: Monograph)

by

Yu Qian

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Masters of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Proteins are some of the building blocks of organisms. They usually perform their functions by interacting with each other and forming protein complexes. A protein-protein interaction network is a graph that consists of proteins as vertices and their interactions as edges. Protein-protein interaction network alignment is very important in identifying protein complexes and predicting protein functions. Many algorithms based on graph theory have been developed to improve the accuracy of alignment, but due to the sparsity of protein-protein interactions, the result is far from satisfactory.

We propose to improve the network alignment through adding protein interactions to existing PPI networks. In order to assess the improvement, we devise four groups of experiments and compare their results. The quality of PPI network alignment is assessed through the number of known protein complexes that are discovered. Significant improvement is obtained, up to 70% additional complexes being discovered after adding interactions. Other consequences are observed as well. Out of the two programs we compare, AlignMCL and MaWISH, the former performs significantly better whereas the latter is more stable. Further, adding predicted PPIs is not as efficient as adding PPIs from existing databases. Finally, we show that smaller but more reliable sets of interactions perform better than larger PPI sets.

Keywords: protein-protein interaction, network alignment, PPI prediction

Acknowlegements

My deepest gratitude goes first and foremost to my supervisor, Dr. Lucian Ilie, for his careful guidance and rigorous attitude in research. I will never reach here without his help.

Second, I'd like to show my heartfelt appreciation to all my labmates for offering me advice and assistance which helped me a lot in the research, especially Yiwei Li, who provided me huge support in my experiments.

I also want to express my gratitude to my friends Linfang Jin and Yanjun Xu, their long-lasting encouragement and trusts make me self-confident and brave.

Last but not least, I wish to thank my beloved parents, without their unyeilding support and endless encouragement, Thank you for the faith in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is well known that proteins in cells often interact with each other to perform various important functions. The protein-protein interaction (PPI) network of an organism is a graph representing these interactions as edges between proteins as vertices. The PPI network contains pathways, complexes and modules of crucial importance for the good functioning of the organism [18].

Our understanding of the PPI networks can be enhanced by network alignment. A significant amount of work has been done to reveal protein functions through analysis of PPI networks. Global and local network alignment are the two main categories of network alignment. As distant species do not have large regions of global network similarity, local network alignment is of more value than global network alignment. Excellent algorithms, such as MaWISH [21] and AlignMCL [24] were proposed to tackle local network alignment. MaWISH constructs the alignment graph according to a gene evolution model, while AlignMCL devised a novel approach to construct it through a new concept called *union graph*. When searching subgraphs in the alignment graph, MaWISH reduces the problem to a maximum-weight subgraph problem, while AlignMCL uses a cluster finding program based on Markov random walks.

There is no golden rule to measure the performance of network alignment. The most popular way is to compare the protein overlaps between subgraphs and some known protein complexes. According to this rule, AlignMCL is currently the best one [24]. But still, it can only detects less than 50% known protein complexes.

Another fact about PPIs worth noting is that the number of known PPIs is very small, the actual number might be significantly higher. Therefore, predicting new reliable PPIs may help us to reveal more unknown protein functions. Experimental methods such as Y2H (Yeast two-hybrid) [18] are used in early PPI prediction, but they are both labor and time consuming and their false positive rates are high. Due to these limitations, many computational approaches like Martin's [25] and PIPE [38] are proposed. Martin's program uses support vector machines (SVM) on signature products, while PIPE computes a prediction matrix based on sequence similarities between proteins.

According to the limitations of network alignment algorithms, we proposed a new idea for improving local network alignments by incorporating predicted PPIs. We investigate and select the best PPI prediction program to enrich existing PPI networks, then we perform local network alignment on these new PPI networks. The experimentals confirm our expectation: more protein complexes are detected after adding PPIs.

The thesis is organized as follows. Chapter 2 introduces biological concepts such as protein sequences, PPIs, followed by the definition of the network alignment problem in computer science. In Chapter 3, we described our new idea in detail. Our investigation of some state-of-the-art PPI prediction programs and network alignment algorithms is also given here. At the end of this chapter, we explain the design of our experiments. The evaluation and analysis is performed in Chapter 4 by comparing between different experiments. We conclude in Chapter 5 where several further research directions are proposed.

Chapter 2

Background

This chapter introduces some basic concepts about proteins, protein-protein interactions (PPIs) and PPI prediction. Then, the two main categories of PPI network alignment are discussed, along with some popular PPI databases.

2.1 Proteins and Interactions

Proteins are key components of cellular machinery. They consist of twenty kinds of amino acids and fold into various 2D and 3D structures. They play multiple roles, including transferring signals, controlling the function of enzymes, and regulating production and activities in the cell [52].

To achieve their key functions, proteins usually interact with each other. Some of the PPIs are permanent, while others are transient and happen only during certain cellular processes. Groups of proteins that interact together to perform a certain function are called protein complexes. Figure 2.1 shows an example of a protein complex in a cell. Protein pathways and modules are another two functional structures formed by proteins through PPIs. A PPI network consists of all the proteins in a species together with the interactions between proteins. Figure 2.2 shows a real PPI network constructed using 11,000 yeast interactions [18]. Protein complexes, pathways and modules are subgraphs of the PPI network. They are however not apparent from the network and need to be discovered.

PPIs are very important in molecular biology, as they help us to understand a protein's function and behavior. They also help us to predict the biological processes that a protein of unknown function is involved in, so it is meaningful to find as many protein complexes in the PPI networks as possible.

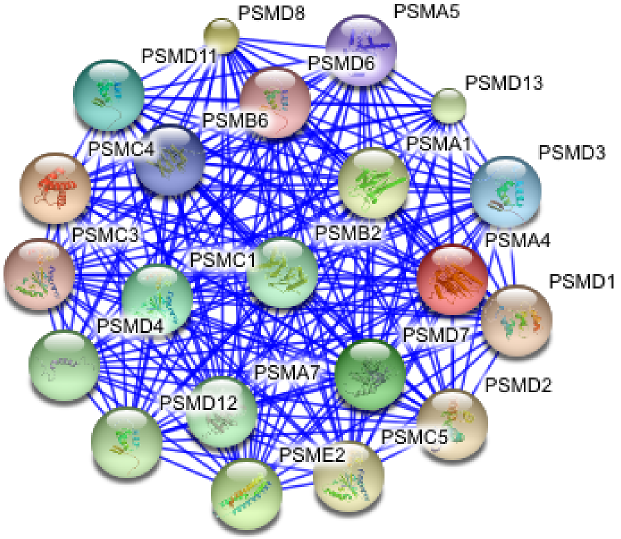


Figure 2.1: Protein Complex. (From: bioproximity.com)

2.2 Protein-Protein Interaction Prediction

PPIs are critical to the integrity of PPI networks [5, 34, 43]. In *Saccharomyces cerevisiae*, there are early estimates of between 10,000 and 40,000 PPIs, out of a potential 19 million possible protein pairs, and the actual number might be significantly higher than these predictions.

Over the recent years, there are some experimental techniques such as Y2H [13], TAP and Micro-arrays [40], that provided a vast quantity of information regarding PPIs. These methods are expensive and labor intensive. Additionally, these methods cannot detect some protein complexes, and current high-throughput screens for interactions are incomplete and contain many false-positives and false-negatives [40]. Also, there is a

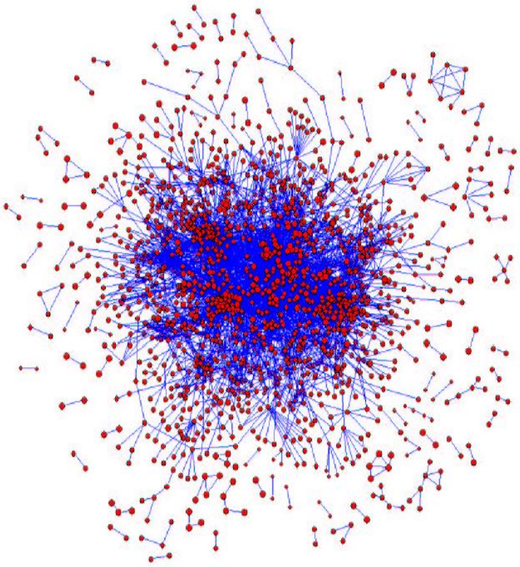


Figure 2.2: A Protein-Protein Interaction Network for yeast. (From: [39])

low level of overlap between different experimental methods [18]. Figure 2.3 shows the overlap of PPIs between traditional experiments.

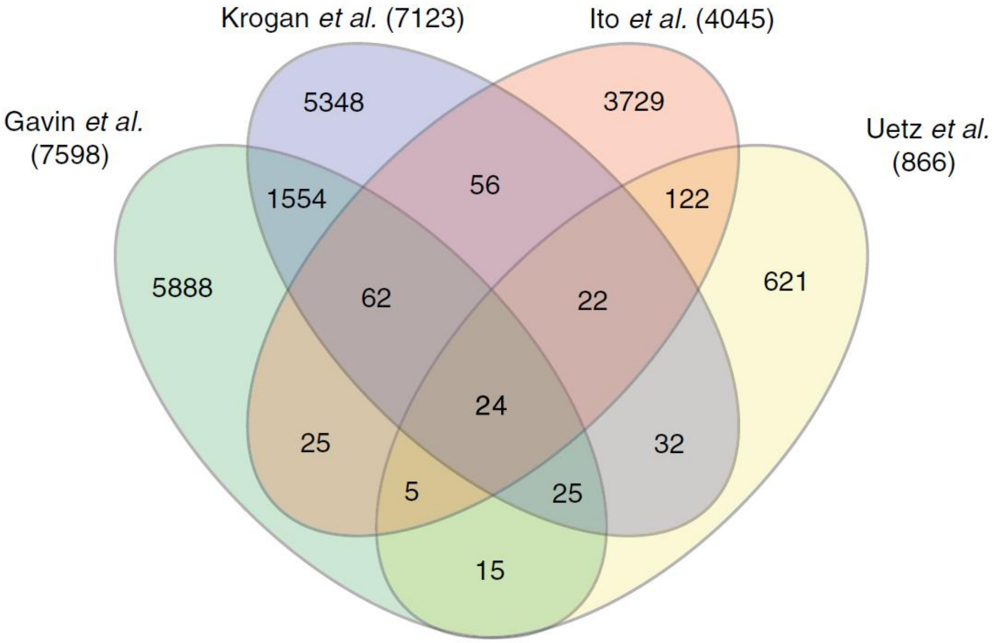


Figure 2.3: Overlap of PPIs between high-throughput traditional experiments. (From: [18])

More accurate methods to predict PPIs are computational techniques, which can be classified into six general categories: methods based on genomic information, evolutionary relationships, three-dimensional (3D) protein structure, protein domains, network analysis and primary protein structure [34]. Numerous methods have been developed to predict PPIs that take advantage of the sequence, structure or the genomic context of the query proteins [18]. Two leading programs are Martin's [25] and PIPE (Protein-Protein Interaction Prediction Engine) [38].

2.3 Protein Orthologs

With the rapidly growing amount of sequence data, many scientists are asking which genes one species has in common with other species. A particularly important question is which genes in one species are sharing the exact same biological function with genes in simpler organisms. To be able to infer which genes have the same function, we need to understand how the genes evolved.

In biology, two genes related by descent from a common ancestral DNA sequence are called homologs. More specifically, if two genes are separated by the event of speciation, we call them orthologs; otherwise, if they are separated by the event of genetic duplication, we call them paralogs. Figure 2.4 is an example of the formation of orthologs and paralogs. Orthologs retain the same function in the course of evolution, whereas paralogs evolve new functions, even if these are related to the original one.

As orthologs are related through evolutionary history, phylogenetic trees are the most natural way to detect orthologs. But unfortunately, construction of phylogenetic trees involves some poorly automatable steps and demands large resources of computing power. An alternative to phylogenetic methods is to use all-versus-all sequence comparison between two genomes to detect orthologs. The COG (Clusters of Orthologous Groups) database [48] is a collection of BLAST-based ortholog groups from multiple species. The members of a COG entry must belong to at least three species, it therefore represents sequences whose function is conserved across major phylogenetic lineages which is a problem if we need the orthologs of two certain species.

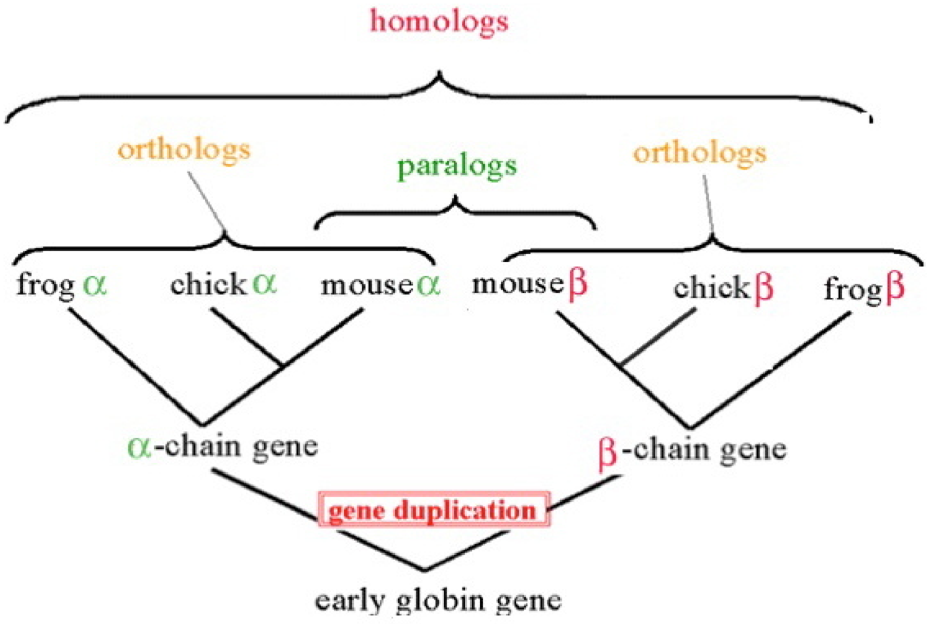


Figure 2.4: Gene homologs. (From: bioweb.uwlax.edu)

Current approaches of orthology assignment can be classified into: (1) graph-based methods, such as InParanoid [20], eggNOG [29], OrthoMCL [22] and (2) tree-based methods, such as Ensembl Compara [2].

2.4 Protein-Protein Interaction Network Alignment

PPI networks are commonly represented as graphs, with nodes corresponding to proteins and edges representing PPIs. Most of the PPI networks are undirected and represent only binary interactions. An alignment is a mapping between these graphs. Network alignment, in its general form, is a computationally hard problem, since it can be related to the subgraph-isomorphism problem, which is known to be NP-complete. Effective techniques for solving this problem rely on suitable formulations of the alignment problem, use of heuristics to solve these problems, or the use of alternate data to guide the alignment process [44].

At a high level, the network alignment problem can be classified as *global alignment* and *local alignment*; more details are given in the following section. Table 2.1 shows a summary of current popular network alignment programs.

	Local	Global
Pairwise	PathBLAST ([19]) MaWISH ([21]) AlignNemo ([10])	GRAAL family ([32]) HopeMap ([49]) GHOST ([36]) PINALOG ([37]) SPINAL ([1]) NETAL ([30])
Multiple	NetworkBLAST ([45]) Graemlin1.0 ([15])	IsoRank family ([9]) Graemlin2.0 ([14]) PIswap ([8]) SMETANA ([42])

Table 2.1: Network Alignment Algorithm Summary

2.4.1 Global Network Alignment

Given a set of networks $G = \{G_1, G_2, ..., G_k\}$, a global network alignment requires all nodes in all networks to be aligned. To achieve this goal, biological information from protein sequence similarity and topological information from a PPI network are used. Many algorithms make full use of this information to conduct global network alignment. The global network alignment is used to establish functional orthologs across species.

The two main steps in global network alignment are: 1) Compute node similarity between proteins from each input network and represent it as a matrix, 2) Extract node mapping from the matrix where seed and extension, bipartite matching or other heuristic strategy can be used. The IsoRank family [9] and the GRAAL family [32] are two traditional examples.

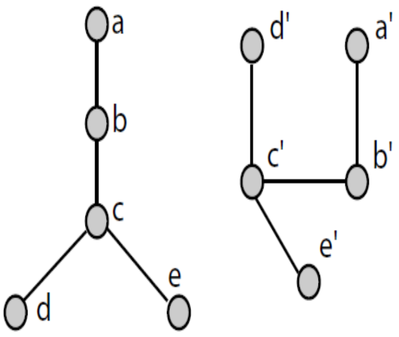
IsoRank

IsoRank is the first method to compute global alignment between the *S.cerevisiae* and *D.melanogaster* PPI networks. The idea is: a node i in G_1 is mapped to a node j in G_2 if the neighborhood topologies of i and j are similar [41]. It simultaneously uses both PPI network data and sequence similarity data to compute the alignment. This approach uses a similar idea as Google's PageRank technique [33]. Also it was formalized as an eigenvalue problem.

IsoRank uses the following formula to calculate the similarity score R_{ij} of protein i and protein j:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv} \quad i \in V_1, j \in V_2 \tag{2.1}$$

where V_1 and V_2 are the sets of nodes in network G_1 and G_2 and $N(a)$ is the set of neighbors of node a. The value of R_{ij} depends on the score of neighbors of i and j, which, in turn, depend on the neighbors of the neighbors and so on. An example is shown in Figure 2.5.



R

	a'	b'	c'	d'	e'
a	0.0312		0.0937		
b		0.1250		0.0625	0.0625
c	0.0937		0.2812		
d		0.0625		0.0312	0.0312
e		0.0625		0.0312	0.0312

$$R_{aa'} = \frac{1}{4} R_{bb'}$$

$$R_{bb'} = \frac{1}{3} R_{ac'} + \frac{1}{3} R_{a'c} + R_{aa'} + \frac{1}{9} R_{cc'}$$

$$R_{dd'} = \frac{1}{9} R_{cc'}$$

$$R_{cc'} = \frac{1}{4} R_{bb'} + \frac{1}{2} R_{be'} + \frac{1}{2} R_{bd'} + \frac{1}{2} R_{eb'} + \frac{1}{2} R_{db'} + R_{ee'} + R_{ed'} + R_{de'} + R_{dd'}$$

Figure 2.5: An example of score calculation in the IsoRank Algorithm (From: [41])

We can incorparate biological information to this model through assigning weights to the graph; the above equation becomes:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i,u)w(j,v)}{\sum_{r \in N(u)} \sum_{q \in N(v)} w(q,v)} R_{uv} \quad i \in V_1, j \in V_2 \tag{2.2}$$

Clearly, the first equation is a special case of the latter when all the edge weights are 1.

This equation can be further rewritten into a matrix form:

$$R = AR \tag{2.3}$$

where

$$A[i,j][u,v] = \begin{cases} \frac{1}{|N(u)||N(v)|} & \text{if } (i{,}u) \in E_1 \text{ and } (j{,}v) \in E_2 \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

A is a $|V_1||V_2| * |V_1||V_2|$ matrix and $A[i,j][u,v]$ refers to the entry at the row (i,j) and column (u,v).

It is clear that this is an eigenvalue problem, where R is the principal eigenvalue of A. Then, maximum-weight bipartite matching is used to find the best matching for this graph. A greedy algorithm can be used here: identify the highest score R_{pq} and output the pairing (p,q); then, remove all scores involving p or q; repeat this process until the list is empty.

IsoRankN [9] extends the same idea to aligning multiple networks.

GRAAL family

GRAAL [32] proposed an algorithm with the cost function based solely and explicitly on a strong, theoretically grounded and direct measure of network topological similarity. This method is flexible to incorporate biological data if we are trying to understand complex biological phenomena.

GRAAL first computes a matrix C of costs of aligning each node in G_1 with each node in G_2. The rows of C correspond to nodes in G_1 and the columns correspond to nodes in G_2. When computing the cost of aligning a node u from G_1 with a node v from G_2, GRAAL takes into account their signature similarity as well as their degrees. The cost of aligning nodes u and v is computed as:

$$C(u,v) = 2 - ((1-\alpha) * \frac{deg(u) + deg(v)}{max_{deg}(G_1) + max_{deg}(G_2)} + \alpha * S(u,v)) \tag{2.5}$$

A cost of 0 corresponds to a pair of topologically identical nodes v and u. The advantage of GRAAL lies in computing the similarity score $S(u, v)$ using graphlet degree signature and signature similarity [50]. Graphlets are small connected non-isomorphic induced subgraphs of a large network. Signature similarity counts the number of edges that a node touches in each graphlet; these numbers consist of the vector of graphlet degrees. We can understand the similarity simply as the number of shapes a node is part of.

In a general graph, two nodes can only form one shape; the 2-node graphlet is shown in Figure 2.6, while 3 nodes can form two non-isomorphic shapes. In each shape, nodes in the symmetric positions are called isomorphic. According to the same rule, 4 nodes have 6 graphlets, while 5 nodes can have at most 21 graphlets. Given these 29 graphlets in total, we then count how many non-isomorphic nodes are there in each graphlet. The two nodes in the 2-node graphlet are isomorphic, so we just count them once, and we call them an orbit, In the 2 3-node graphlets, we find 3 orbits. There are 73 orbits in total in all 2-node to 5-node graphlets.

Figure 2.7 is an example of graphlet degree vector, for the graph in the figure; node v appears in 5 different 2-node graphlets, so $GDV(0) = 5$. v also appears as end node in two 3-node graphlets, so $GDV(1) = 2$. The Graphlet Degree Vector is computed this way.

Spheres are then built after the similarity matrix C is computed. A sphere of radius r around a node u in a network G is the set of nodes $S_G(u, r) = \{v \in G : d(u, v) = r\}$ that are at distance r from node u. $d(u, v)$ is the length of the shortest path from u to v. The node pair (u, v) with lowest score in C is selected as initial seed. The sphere of u and the sphere of v are constructed in G_1 and G_2, respectively. Spheres with the same radius are then aligned greedily and the algorithm stops when all the nodes in one input network are aligned to nodes in the other network.

C-GRAAL [27] and MI-GRAAL [31] are two improved version of GRAAL. C-GRAAL developed an algorithm to compute common-neighbor information, which resulted in a more reliable similarity score matrix, while MI-GRAAL can integrate more node similarity.

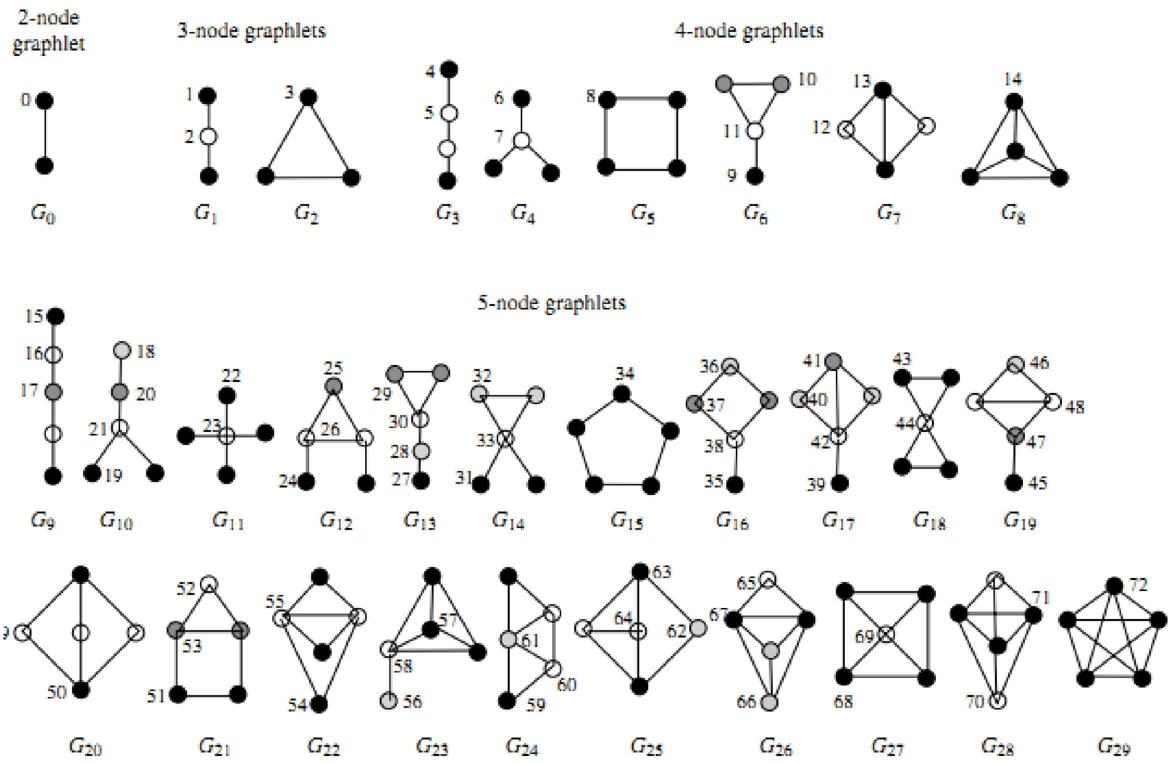
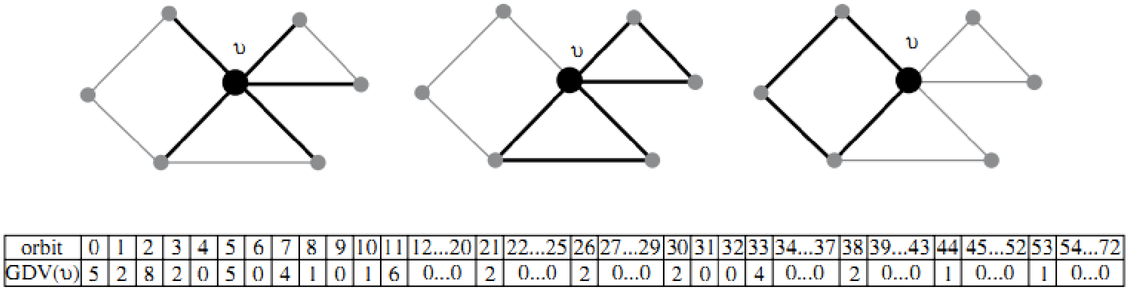


Figure 2.6: 2- to 5- Graphlets (From: [50])



orbit	0	1	2	3	4	5	6	7	8	9	10	11	12...20	21	22...25	26	27...29	30	31	32	33	34...37	38	39...43	44	45...52	53	54...72
GDV(υ)	5	2	8	2	0	5	0	4	1	0	1	6	0...0	2	0...0	2	0...0	2	0	0	4	0...0	2	0...0	1	0...0	1	0...0

Figure 2.7: Graphlet Degree Vector (From: [50])

Other Algorithms

NETAL [30] and SPINAL [1] are two new algorithms for global network alignment. They use the two general steps described above. While NETAL updates the interaction score dynamically in the alignment, SPINAL defines the score of aligning two nodes based on their neighborhood bipartite graph. GEDEVO [17] is a novel program using the graph edit distance and an evolutionary algorithm for alignment. According to Rashid's study, GEDEVO outperforms SPINAL and GRAAL in computing edge correctness [17].

Although many algorithms have been designed to tackle the global network alignment problem, the result is far from perfect due to the biological characteristics of the PPI networks. Another difficulty is that there is no golden rule to measure the performance of network alignment. For global network alignment, some often used parameters are edge correctness, node correctness, interaction correctness [30] and so on. Edge correctness is the percentage of edges (interactions) of the first network that are aligned to edges in the second network, while interaction correctness is the percentage of interactions of the first network that are aligned with correct interactions in the second alignment. Table 2.2 gives a comparison between some state-of-the-art global network alignment algorithms for a variety of networks; see [17] for more details. Note that none of them can reach a correctness rate greater than 50%.

Network 1	Network 2	Edge Correctness(%)			
		GEDEVO	MI-GRAAL	C-GRAAL	SPINAL
Campylobacter jejuni	*Escherichia coli*	33.70	24.60	22.56	22.09
Mesorhizobium loti	*Synechocystis sp.(PCC6803)*	43.60	39.88	33.19	25.86
Saccharomyces cerevisiae	*Homo Sapiens*	38.14	21.38	22.20	19.33
Homo Sapiens	*Saccharomyces cerevisiae*	30.40	26.15	24.15	25.59
Saccharomyces cerevisiae	*Drosophila Melanogaster*	20.79	17.73	20.59	21.07
Drosophila Melanogaster	*Homo Sapiens*	21.88	-	27.36	27.04
Homo Sapiens	*Homo Sapiens*	89.37	-	47.07	-

Table 2.2: Global Network Alignment Algorithm Comparison (From: [17])

2.4.2 Local Network Alignment

Global network alignment attempts to capture a global picture of the whole input networks. Distant species do not have large regions of global network similarity and it is more appropriate to search for local similarities. Local network alignment corresponds to a relationship defined over a subset of vertices in the input network, which is used to extract many conserved substructures (modules, pathways and complexes) from a set of species. A number of algorithms have been proposed for local network alignment.

The general way to perform local network alignment is: 1) Merge the input networks into an alignment graph, which is also represented as a graph of nodes and edges, where each node denotes a pair of protein orthologs a and $a^{'}$, and each edge denotes a conserved protein interaction, that is, an interaction observed for both (a, b) and $(a^{'}, b^{'})$. Sometimes, in order to tolerate a certain amount of missing interaction data, "indirect" edges are also defined if a pair of proteins interacts in one species (e.g. a and b interact) and the other pair of proteins (e.g. $a^{'}$ and $b^{'}$) is at distance at most two in their corresponding networks [47]. 2) Extract dense subgraphs from the alignment graph.

PathBLAST [19] and NetworkBLAST [45] are two early attempts for local network alignment. *Protein sequence similarity* is used to merge nodes from input networks. In PathBLAST and NetworkBLAST, a cut-off BLAST E-value (e.g., 10^{-7}) is used to define the protein sequence similarity of two proteins. PathBLAST then defines *match*, *mismatch* and *gap* as in sequence alignment to represent the alignment graph (see Figure 2.8). After the alignment graph is built, PathBLAST extracts high-scoring pathways. It generates a sufficient numbers of acyclic subgraphs and then uses dynamic programming to compute highest-scoring pathways.

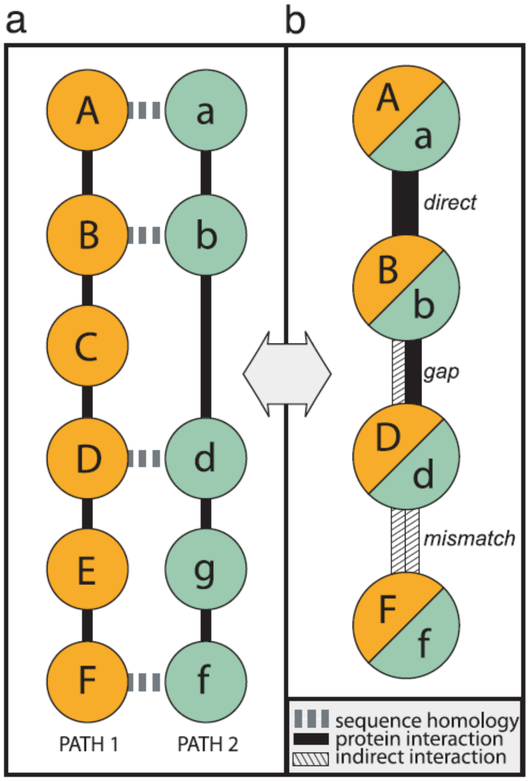


Figure 2.8: Alignment Graph in PathBLAST (From: [19])

PathBLAST was improved to a more efficient program called NetworkBLAST ([45]). The construction of the alignment graph in NetworkBLAST is more clear and flexible as proteins at distance 2 are assumed to interact, while in PathBLAST, gaps and mismatches are not allowed to occur consecutively. Subnetwork search over the alignment graph is defined to find high-scoring subnetworks based on a probabilistic model. The probabilistic model includes a *real model* in which every interaction should be present with high probability, and a *random model*, where the probability of an interaction depends on the total connections in the network. Then the log likelihood ratio defined on the two models are the score of the subnetwork over the subset of vertices U. Figure 2.9 shows the framework of NetworkBLAST used in local multiple network alignment.

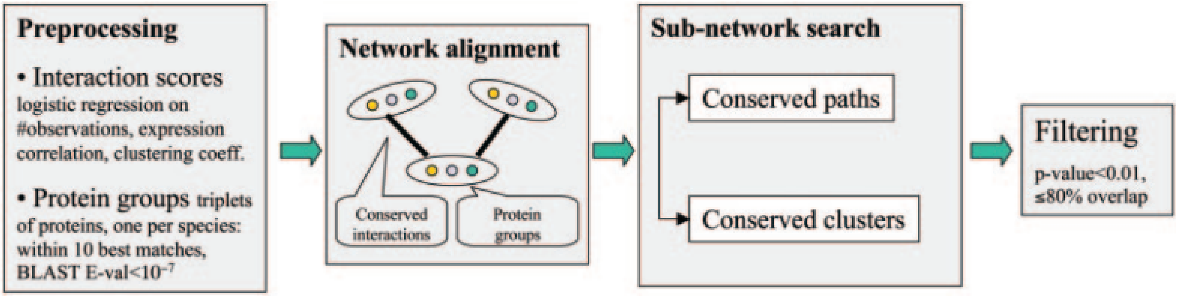

Preprocessing
• Interaction scores logistic regression on #observations, expression correlation, clustering coeff.
• Protein groups triplets of proteins, one per species: within 10 best matches, BLAST E-val<10⁻⁷
Network alignment
Conserved interactions
Protein groups
Sub-network search
Conserved paths
Conserved clusters
Filtering
p-value<0.01, ≤80% overlap

Figure 2.9: NetworkBLAST Framework (From: [45])

According to the model above, the local network alignment problem can be reduced to finding heavy subgraphs heuristicaly in the alignment graph. NetworkBLAST first computes seeds with highest score at size 3, and adds a node which will increase the score to the seed iteratively. The size of subgraphs is limited to 15 nodes. For each node, it outputs the 4 highest-scoring subgraphs as the final result.

A popular evaluation method to measure local network alignment is to compare the subgraphs with *known protein complexes*. A good algorithm is expected to find more known complexes. Although NetworkBLAST is reported to perform well in protein function prediction and protein-protein interaction prediction, few known complexes can be found by it when compared to two state of the art algorithms, MaWISH [21] and AlignNemo [10].

A novel algorithm, MaWISH, is based on duplication/divergence model with focus on understanding the evolution of PPI networks ([26]). AlignNemo improved local network alignment in two ways: 1). Adopted the newest protein orthologs information collected by Inparanoid algorithm [20], 2). The score assigned to each edge is based on an efficient construction of the alignment graph which incorporates topological information present in the original networks in terms of number, reliability and significance of paths of length less than or equal to 2 between two nodes [10].

AlignNemo and MaWISH are currently the leading programs for local network alignment, which also find many known complexes. Table 2.3 shows some comparison of Networkblast, MaWISH and AlignNemo. In this comparison, the test datasets include 22,969 interactions in fly (*Drosophila melanogaster*), 22,254 interactions in yeast (*Saccharomyces cerevisiae*) and 78,559 weighted interactions in human (*Homo sapien*), more details are described in [10]. But due to the sparsity of protein interactions, many complexes remain undiscovered. AlignNemo was improved by adopting a new clustering algorithm; the new software is named AlignMCL [28].

The challenge in network alignment is not merely in designing an excellent algorithm to find more complexes. Overcoming the limitation of lacking protein-protein interactions is of the same importance, and a lot of work remains to be done in this area.

Algorithms	fly-yeast		fly-human	
	Solution size	Complexes Founded	Solution size	Complexes Founded
NetworkBLAST	329	**30**	45	13
MaWISH	175	29	87	60
AlignNemo	242	**52**	115	**87**

Table 2.3: Local Network Alignment Algorithm Comparison. (One subnetwork (or subgraph) found by an alignment algorithm is called a solution.) (From: [10])

2.5 Existing PPI-related Databases

Large scale identification of PPIs generated hundreds of thousands of interactions, which were collected together in specialized biological databases that are continuously updated in order to provide complete interactomes. The first of these databases was the *Database of Interacting Proteins* (DIP). Since that time, the number of public databases has been increasing. Databases can be subdivided into primary databases, meta-databases, and prediction databases [11].

- **Primary databases** collect information about published PPIs proven to exist via small-scale or large-scale experimental methods. Examples: DIP, I2D, Hint, DroID, BIND, BioGRID, HPRD, IntAct Molecular Interaction Database, MINT, MIPS-MPact, and MIPS-MPPI [11].

  DIP (Database of Interaction Proteins) combines information from a variety of sources to create a single, consistent set of PPIs. The data stored in DIP has been curated, both manually, by expert curators, and automatically, using computational approaches that utilize knowledge about the PPI networks extracted from the most reliable, core subset of DIP data. In addition to the interaction information, DIP includes additional data regarding the proteins participating in PPI networks. This database is available at http://dip.doe-mbi.ucla.edu/. DIP is an early database from PPIs, and the number of PPIs in it is small when compared to other new databases.

  I2D (Interologous Interaction Database) is an on-line database of known and predicted mammalian and eukaryotic PPIs. It has been built by mapping high-throughput data between species. Thus, until experimentally verified, these interactions should be considered "predictions". I2D remains one of the most comprehensive sources of known and predicted eukaryotic PPIs. I2D is available at http://ophid.utoronto.ca/ophidv2.204/. Table 2.4 shows some statistics of PPIs.

- **Meta-databases** normally result from the integration of primary databases information, but also collect some original data. Example: Agile Protein Interaction Data Analyzer (APID), The Microbial Protein Interaction Database (MPID8), and Protein Interaction Network Analysis (PINA) platform.

- **Prediction databases** include many PPIs that are predicted using several techniques. Examples: Michigan Molecular Interactions (MiMI), Human Protein-

Protein Interaction Prediction Database (PIPs), On-line Predicted Human Interaction Database, Known and Predicted Protein-Protein Interactions (STRING), and Unified Human Interactome (UniHI).

SPECIES	DIP	I2D
Saccharomyces cerevisiae	24,516	338,493
Drosophila melanogaster	23,241	177,875
Homo sapiens	7,205	318,717
Escherichia coli	13,378	121,767

Table 2.4: PPIs data from DIP and I2D. (From: DIP and I2D web interface)

The most important information in the PPI databases that we need is the interacting proteins and their reliability scores.

Besides PPIs databases, some related databases about protein complexes, pathways and modules are also very important in PPI network alignment. These databases are not only important in protein functions research, they are also a vital standard to measure network alignment algorithms. *CORUM* (Comprehensive Resource of Mammalian protein complexes) provides a resource of manually annotated protein complexes from mammalian organisms. All information is obtained from individual experiments published in scientific articles. Data from high-throughput experiments is excluded [4]. *DPiM* has 556 protein complexes for Drosophila, it defines potential novel members for several important protein complexes and assigns functional links to 586 protein-coding genes lacking previous experimental annotation [16]. DPiM is a large protein complex database and provides a valuable resource for analysis of protein complex evolution. *CYC2008* is a comprehensive catalogue of 408 manually curated yeast protein complexes [46]. These gold standard datasets on protein complexes are key to inferring and validating PPIs.

Databases for protein sequence are more complete due to the development of high-throughput sequencing technology. It is easy to obtain protein sequences from NCBI, Entrez, Ensembl, SGD, FlyBase, wormBase and so on. Another notable database is Uniprot KB. It provides the scientific community with a comprehensive, high-quality and freely accessible resource of protein sequence and functional information. Many PPI databases use the Uniprot KB Accession/ID to represent proteins.

Chapter 3

Improving PPI Network Alignment

Designing a new alignment algorithm is not the only way to improve local network alignment. The result can be improved if many reliable PPIs are provided. We propose a new idea for improving network alignment through the combination of PPIs prediction and existing local network alignment algorithms.

3.1 Our Proposed New Idea

Our idea works as follows:

- Test existing PPI prediction algorithms and select the best one.
- Add high-scoring interactions to existing PPI networks.
- Perform network alignment on the new PPI networks.
- Verify the quality of the alignments obtained.

3.2 Software Used For Investigation

3.2.1 PPI prediction

According to the study of Park, Martin's program [25] and PIPE [38] are the best two candidates for PPIs prediction [35]. PIPE can successfully predict new interactions based on re-occurring short sub-sequences in an existing interaction database. Its major working steps are shown in Figure 3.1. Given a protein sequence A of length N, it checks every sliding window of size w starting from position 0 to position $N - w$; the neighbors of the protein in the network which have the same window are added to the neighbors' list of the query protein. When the neighbors' lists of all the query windows are constructed, it searches all sliding windows of the same length w in another protein sequence B, then compares the window with all the protein sequences in the neighbors' list and records every hit into the matrix M. If the maximum score in the final matrix is greater than a given threshold, then protein A and protein B are predicted to interact.

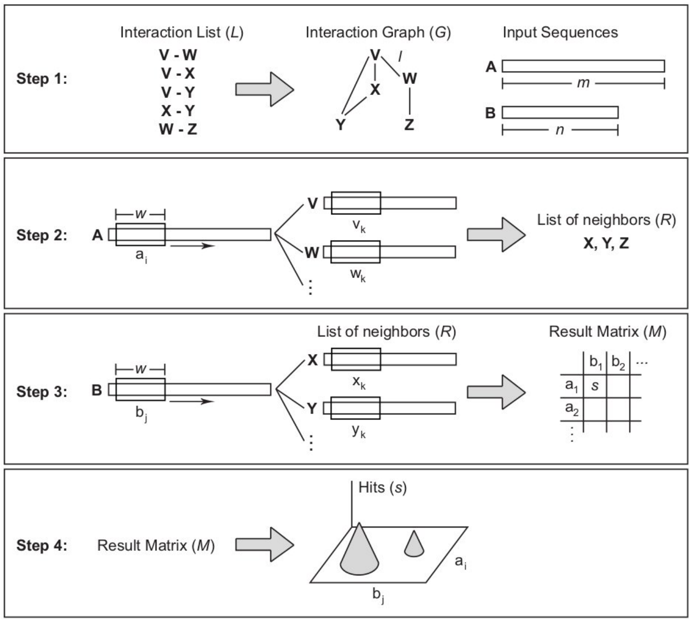


Figure 3.1: Steps for PIPE algorithm. (From: [38])

As PIPE source code is not public, we implemented the algorithm using C++ and OpenMP. PIPE is time-consuming even after optimization. Table 3.1 shows the estimated running time of PIPE using *yeast* (with 6632 proteins and 22,377 interactions) as test dataset. The newest version of PIPE, PIPE3, uses a 256 core PC cluster and a 1168 core Sun T2 "Victoria Falls" Cluster to support the prediction [38]. Given such a strong computing cluster, it still takes 3 weeks to predict *C.elegans* (23,684 sequences) and 3 months to predict *Homo sapiens* (22,513 sequences).

Table 3.1: PIPE run time analysis

PIPE version	Time (one pair)	Total(estimated)
original PIPE	115 minutes	9,623 years
window cached PIPE	4 minutes	334 years

Martin's program uses signature products [25]. In this method, an amino acid sequence is represented as a signature, which is a combination of three letters. For example, for the amino acid sequence $LVMTTM$, the trimers are: LVM, VMT, MTT and TTM. They are made of a root (middle letter) and two neighbors (side letters). Then the signatures of those trimers are V (LM), M (VT), T (MT), T (MT), respectively. By adding those vectors together, the signature of the sequence is $V(LM)+M(VT)+2T(MT)$ [25]. Note that $2T(MT)$ comes from $T(MT) + T(TM)$. To define the signature for protein-protein pairs, it uses the tensor product between vectors. In order to use support vector machines (SVM) in PPI prediction, the signature kernel is defined as the dot product of two signatures of two proteins. Then using machine learning methods, predictions will be made. In order to improve the performance, Martin adopted some adjustments, for example, it enforces symmetry in the protein-protein order to improve the signature product, then it normalizes the signature product to compensate for potential differences in the length of the amino acid sequences. More details are described in [25].

Martin's program is faster than PIPE and the code is freely available, so we select Martin's program for PPIs prediction.

3.2.2 PPI network alignment

According to the most popular evaluation method and the comparison we introduced in Chapter 2, MaWISH and AlignMCL are selected in our investigation.

MaWISH

MaWISH's advantage lies in the use of duplication/divergence model [26]. A number of studies show that the PPI network are power-law graphs, that is, the relative frequency of proteins that interact with k proteins is proportional to k^{γ}, where γ is a network-specific parameter [6]. A network-growth model based on preferential attachment is proposed to generate networks with degree distribution similar to PPI networks. Duplication/divergence model is such a model of evolution that explains preferential attachment and power-law nature of PPI networks.

Figure 3.2 shows an example of protein duplication. Protein $u_1{}'$ is duplicated from protein u_1, and u_1 loses its interaction with u_3 (dotted line), while, an interaction between u_1 and $u_1{}'$ is added to the network (dashed line). According to this model, MaWISH defines the local network alignment as a set of matches, mismatches (insertion or deletion) and duplications. For PPI networks $G(U, E)$ and $H(V, F)$, a protein subset pair $P = (\tilde{U}, \tilde{V})$ is defined as a pair of protein subsets $\tilde{U} \subseteq U$ and $\tilde{V} \subseteq V$. Any protein subset pair P induces a local alignment $A(G, H, S, P) = (M, N, D)$ of G and H with respect to S, a similarity function between each pair of proteins in $U \cup V$:

- M - set of matches: Each match $m \in M$ is associated with a score $\mu(m)$ to reflect the confidence of protein orthologs.
- N - set of mismatches: Each mismatch $n \in N$ is associated with a score $v(n)$ to penalize an insertion or deletion.
- D - set of duplications: Each duplication $d \in D$ is associated with a score $\delta(d)$ to reflect duplications in one species.

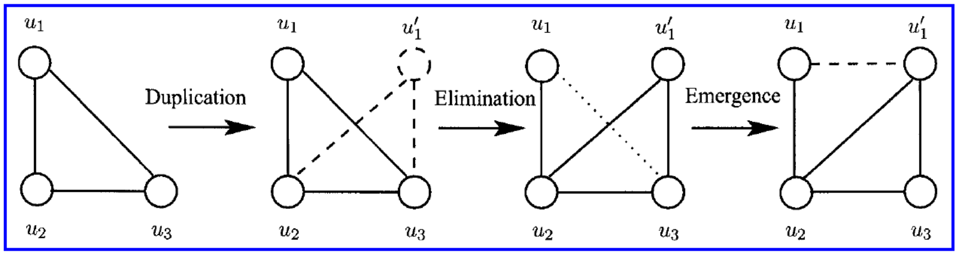


Figure 3.2: duplication/divergence Model. (From: ([26]))

The total score of a local alignment based on the above scoring system is:

$$\sigma(A) = \sum_{m \in M} \mu(m) + \sum_{n \in N} v(n) + \sum_{d \in D} \delta(d) \tag{3.1}$$

Every score in the formula is related to a similarity function S. The similarity score $S(u,v)$ quantifies the likelihood that protein u and v are orthologous. Equation 3.2 calculates the similarity score; in this equation, the COG database is used as a reference [48]. Let O be the set of all orthologous protein pairs derived from COG. Then $O(uv)$ represents the event that u and v are orthologous. $\tilde{E}$ is the threshold of BLAST E-value, MaWISH assumes the probability of a protein pair being orthologous is a monotonically decreasing function of the E-value. For example, for 100 orthologs in COG, if 30 orthologs among them have E-value smaller than $\tilde{E}$ and given a protein pair whose E-value is $\tilde{E}$, then the probability of this protein pair to be ortholog is 0.3. COG can be substituted by any other ortholog database.

$$S(u,v) = P(E(u,v) < \tilde{E}|O_{uv}) = \frac{\{u'v' \in O : E(u',v') < \tilde{E}\}}{|O|} \tag{3.2}$$

Given this function, we define the score of match, mismatch and duplication as follows:

$$\mu(uu', vv') = \overline{\mu} S(uu', vv') \tag{3.3}$$

$$v(uu', vv') = -\overline{v} S(uu', vv') \tag{3.4}$$

$$\delta(uu', vv') = \overline{\delta}(S(u,u') - \overline{d}) \tag{3.5}$$

$\overline{\delta}$, $\overline{v}$ and $\overline{\delta}$ are all coefficients to tune the relative weight of corresponding events. $S(u,u')$ is the similarity score of protein u and protein u'.

Figure 3.3 is an example of the pairwise local alignment problem. The alignment graph is constructed according to the model above (Figure 3.4). Weights are assigned to the edges, which differs from NetworkBLAST who assign the protein ortholog information to nodes only. Then the local network alignment can be reduced to a maximum-weight subgraph problem. This problem is NP-complete.

According to the power-law property, a protein in a particular module interacts with most proteins in the same module either directly or through a common module hub,

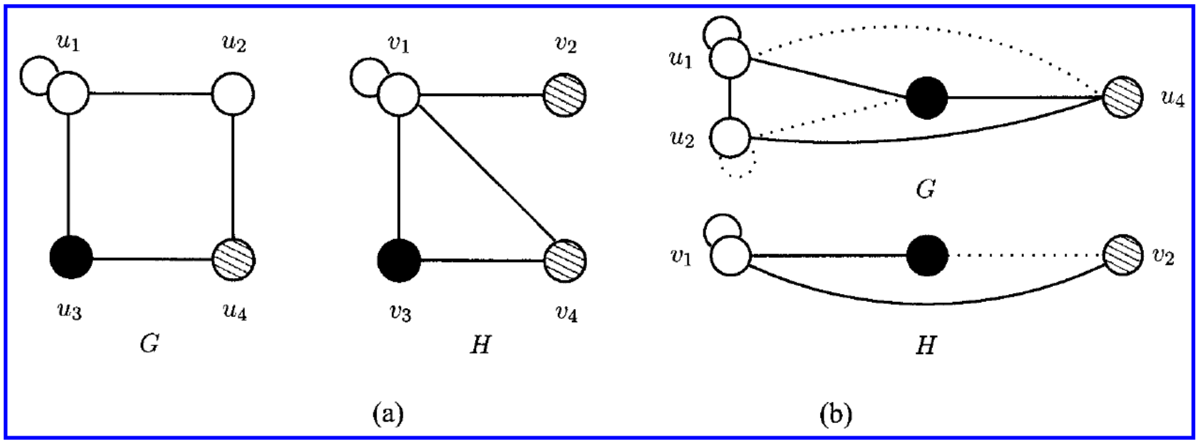


Figure 3.3: (a) An instance of local network alignment. The proteins that have nonzero similarity scores are shaded the same (b) A local alignment induced by the protein subset pair $\{u_1, u_2, u_3, u_4\}$ and $\{v_1, v_2, v_3\}$. (From: [26])

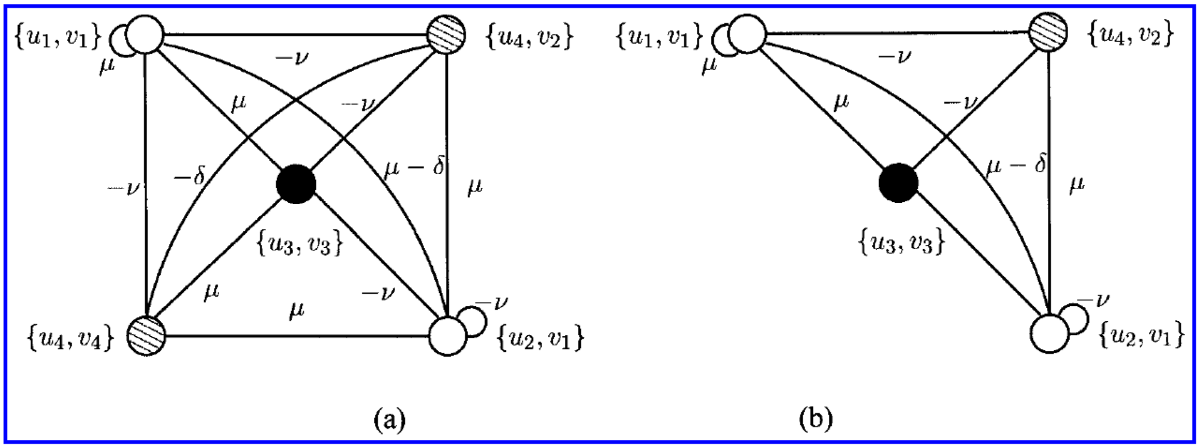


Figure 3.4: Alignment Graph of the example in Figure 3.3. (From: [26])

while it is only loosely connected to the rest of the network. MaWISH adopted a greedy heuristic strategy for this problem because proteins that belong to a conserved module will induce heavy subgraphs, while being loosely connected to other parts of the graph. It works for maximum-weight subgraphs. After finding one subgraph, all aligned nodes are marked, then the process is repeated on unmarked nodes until no more subgraphs with positive weight can be found. The time complexity of MaWISH is $O(|V||E|)$.

AlignMCL

The existing definitions of the alignment graph differ in the way of edge setting between two nodes. AlignMCL uses the same alignment graph construction method as AlignNemo, but it is more efficient in subgraphs mining [10].

Given G_1, G_2 and a set of protein orthologs $H = (u, v), u \in V_1, v \in V_2$ between the nodes of G_1, G_2, a *union graph* is defined on two kinds of nodes: (i) **composite nodes** representing pairs of protein orthologs and (ii) **simple nodes** that don't have homologs in the other network. Any edge contained in one of the input networks is represented in the *union graph.* An alignment graph in AlignNemo is a reduced version of the union graph in which only *composite nodes* are retained and an edge connects two nodes if there is at least one path of length less than or equal to 2 between the two nodes in the union graph; see Figure 3.5 for example.

An efficient edge scoring strategy based on union graph makes AlignNemo prevail over other methods. To score edges of two proteins (composite nodes), *direct path* and *indirect path* at most distance 2 are treated separately (See Equation 3.6, 3.7, 3.8).

$$S_1(a,b) = \frac{w(E_1(a) \cap E_1(b))}{w(E_1(a) \cup E_1(b))} \tag{3.6}$$

$$S_2(a,b) = \frac{w(E_2(a) \cap E_2(b))}{w(E_2(a) \cup E_2(b))} \tag{3.7}$$

$$ELI(a,b) = S_1(a,b) + S_2(a,b) \tag{3.8}$$

We use ELI to stands for the score of edges, $ELI(a, b)$ is the score of the edge (a, b). $S_k(a, b)$ is the score of all paths at distance k between a and b. $E_k(a)$ is the set of paths connecting a to its neighbors at distance k and $w(E_k(a))$ is the sum score of all paths connecting a to its neighbors at distance k. MaWISH also allows for gaps or mismatches to connect conserved proteins at distance 2 in the aligned graph, but it is not able to account for the reliability of direct and indirect paths. Figure 3.5 is an example of alignment graph in AlignNemo and AlignMCL.

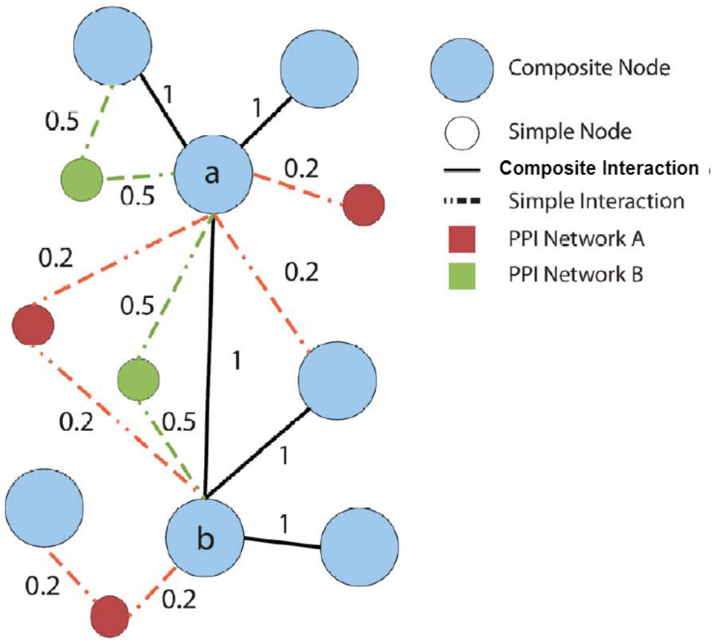


Figure 3.5: An Alignment Graph in AlignNemo. (From: [10])

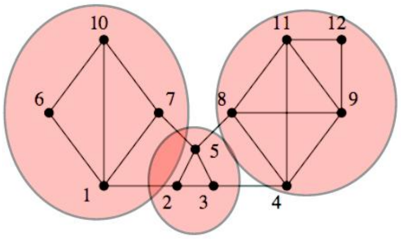
Finding enough reliable subgraphs is always at the heart of the local network alignment problem. It can be reduced to a clustering problem. The MCL (Markov CLustering) algorithm [12] is an excellent solution to extract clusters in a graph, but has never been used in PPI networks before. It is first adopted by AlignMCL in local network alignment to improve AlignNemo [24, 28].

MCL is also based on the power-law intuition that a collection of nodes of a potential cluster in the network are more connected to each other than to other nodes in the network. This is further extended to a conclusion that a random walk in a network that visits a dense cluster will likely not leave the cluster until most of the vertices in the cluster have been visited. MCL simulates a stochastic flow on the network that resembles a set of random walks on the graph. It means that a random walk is expected to cover a dense subgraph in an alignment graph.

There are two steps in MCL:

- **Expand** For each node in the graph, a stochastic flow spreading out from the node towards all the other nodes. This step is performed by repeatedly multiplying the normalized adjacency matrix of the graph by itself (See Figure 3.6). Nodes connected by multiple (and shorter) paths will be the endpoints of stronger flows.

- **Inflation** This step aims at enhancing flows within clusters and weakening inter-cluster flows. Inflation is simply squaring the matrix (see Figure 3.7) by r (the inflation level). It is important to tune an inflation level as it controls the extent of this strengthening/weakening. After inflation, the matrix is normalized on each column.



$$\begin{pmatrix}
5 & 2 & 1 & 0 & 2 & \mathbf{3} & \mathbf{3} & 0 & 0 & \mathbf{4} & 0 & 0 \\
2 & 4 & \mathbf{3} & 1 & \mathbf{3} & 1 & 2 & 1 & 0 & 1 & 0 & 0 \\
1 & \mathbf{3} & 4 & 2 & \mathbf{3} & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\
0 & 1 & 2 & 5 & 2 & 0 & 0 & \mathbf{4} & \mathbf{4} & 0 & \mathbf{4} & 2 \\
2 & \mathbf{3} & \mathbf{3} & 2 & 5 & 0 & 2 & 2 & 1 & 1 & 1 & 0 \\
\mathbf{3} & 1 & 0 & 0 & 0 & 3 & 2 & 0 & 0 & \mathbf{3} & 0 & 0 \\
\mathbf{3} & 2 & 1 & 0 & 2 & 2 & 4 & 1 & 0 & \mathbf{3} & 0 & 0 \\
0 & 1 & 2 & \mathbf{4} & 2 & 0 & 1 & 5 & \mathbf{4} & 0 & \mathbf{4} & 2 \\
0 & 0 & 1 & \mathbf{4} & 1 & 0 & 0 & \mathbf{4} & 5 & 0 & \mathbf{5} & \mathbf{3} \\
\mathbf{4} & 1 & 0 & 0 & 1 & \mathbf{3} & \mathbf{3} & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 1 & \mathbf{4} & 1 & 0 & 0 & \mathbf{4} & \mathbf{5} & 0 & 5 & \mathbf{3} \\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & \mathbf{3} & 0 & \mathbf{3} & 3
\end{pmatrix}$$

Figure 3.6: MCL of 2-path weighted 1, cells in bold in each row indicates a cluster. (multiply the original matrix by itself once) (From: [12])

These two steps are repeated until a steady state is reached. The resulting matrix is very sparse, so we can easily extract modules from it.

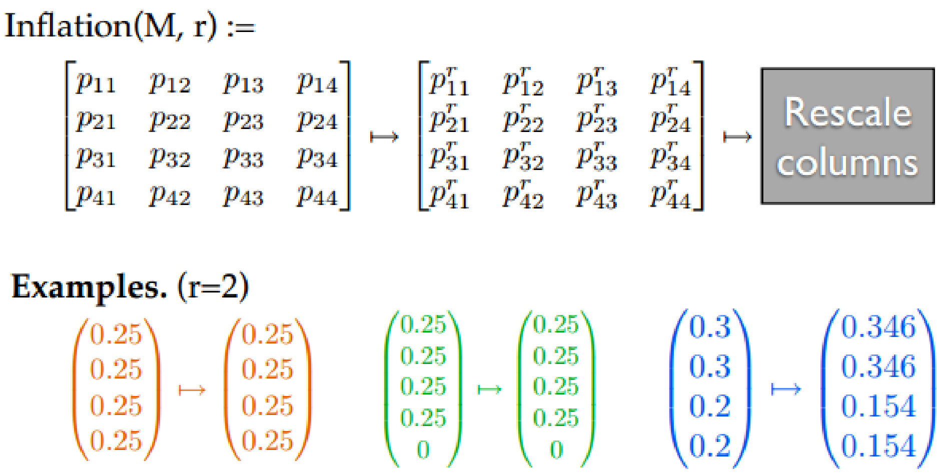


Figure 3.7: Inflation of the matrix in MCL, r is the inflation level, rescale means normalize all the columns (From: [12])

3.3 Data Preparation

Due to the variety of different protein sequences and PPI databases used in different steps, a significant amount of data preparation has to be performed before any experiments.

3.3.1 Datasets

We will use the dataset of *fly* (*Drosophila melanogaster*) and *yeast* (*Saccharomyces cerevisiae*) in all experiments. Protein sequences of *fly* and *yeast* (see Table 3.2) were downloaded from Uniprot KB on April 10, 2014.

Table 3.2: Protein Sequences of *fly* and *yeast*

Species	Number of Proteins
fly	20,980
yeast	6,621

The format of protein sequences used in Uniprot KB is FASTA. An example is shown in Figure 3.8.

Actually, we tested many PPI databases, but it is not necessary to introduce all of them. To verify our idea clearly, we will use the PPI dataset from DIP [23] and I2D [7].

```
>A0AQH0
MQPDFDFTDTPVSTGTTIMAVEFDGGVVIGADSRTSSGAYVANRVTDKLTRITD
KVYCCRSGSAADTQAIADIVAYSLNYHENQTNKDALVFEAASEFRNYCYSYRES
LLAGIIVAGWDEQRGGQVYSIPLGGMLTRESCTIGGSGSSFIYGFVREHYRPNM
ALEDCVTFVKKAVQHAIYHDGSSGGVVRIGIITKDGIERRIFYNTESGASAVSS
TPSFISSE
```

Figure 3.8: A protein sequence in FASTA format; the first line with ">" is description, then sequence follows.

I2D is used as a comparison of Martin's predicted interactions. We will add interactions from I2D and Martin's predictions separately. Table 3.3 shows the number of *fly* and *yeast* PPIs.

Table 3.3: PPIs of *fly* and *yeast*. (DIP&I2D stands for the common interactions shared by both DIP and I2D)

Species	DIP	I2D	DIP&I2D
fly	24,220	37,979	18,575
yeast	22,377	147,407	17,834

3.3.2 Protein Orthologs Collection

Protein ortholog information is essential in network alignment as it is the basis of alignment graph construction. Earlier alignment programs like PathBLAST and NetworkBLAST, use BLAST E-value [3] to represent the sequence similarity. Since evolution events are involved in the formation of protein orthologs, many methods and databases are developed to collect ortholog information using more evolution information. The Inparanoid eukaryotic ortholog database [20] is a collection of pairwise ortholog groups between 17 whole genomes; it is available from the DIOPT web portal. The DIOPT also incorporates other ortholog prediction algorithms like *Compara*, so it is a good choice for collecting protein orthologs.

Orthologs are easy to obtain from DIOPT web portal or accessible data files, but identifier mapping is an important procedure we need to perform. Although Uniprot KB Accession/ID is widely adopted by many databases, we still need to transfer it for datasets that do not use the same identifier. Many existing identifier mapping tools [51] differ in various aspects, such as coverage of species, coverage of identifier types, access speed and frequency of database updates. DAVID is a popular mapping tool, but only gene ID can be used as input. Other services such as PIR do not have up-to-date databases, that may cause information loss. Due to these limitations, we devised our own identifier mapping program. The work flow using *fly* as an example is as follows:

- Extract all the Uniprot KB Accession numbers from the FASTA file and save them to a Uniprot KB ID file, each line holding a unique Uniprot KB Accession.
- Search the web portal or database files for orthologs in *yeast* of all *fly* proteins.
- Generally, the identifier of another species is not UniprotKB Accession (for *yeast*, it uses the system name from the SGD database). For different species, we downloaded different databases which contain the two IDs we need. Sometimes, for example, there is no such file for *yeast* containing both Uniprot accession and SGD system name, so more than two files are needed to be combined together in order to find the mapping. Some databases are not complete, and it is better to combine as many files as we can to avoid information loss. In our experiments, sequences from Ensembl, NCBI, Uniprot and Entrez are all considered.
- Repeat mask; just keep one unique copy for each orthologs.

Figure 3.9 shows the work-flow of our identifier mapping program. The program is implemented using Python 2.7.4. The steps are described below: (a) Extract Uniprot Accession numbers from protein sequences files to a name list file. (b) search orthologs using DIPOT for every proteins in the name list file. The raw orthologs file should contain three columns: the first is Uniprot Accession number for *fly*, *ID* is the SGD name for *yeast* and *SCORE* is the reliability score of two proteins. (c) We take sequence files from Ensemble and NCBI as input; other files should be used if ID-Uniprot KB Accession mapping can not be found from Ensemble and NCBI. For every *ID*, we search the input files until its corresponding Uniprot Accession number is found, otherwise, we have to abandon that pair.

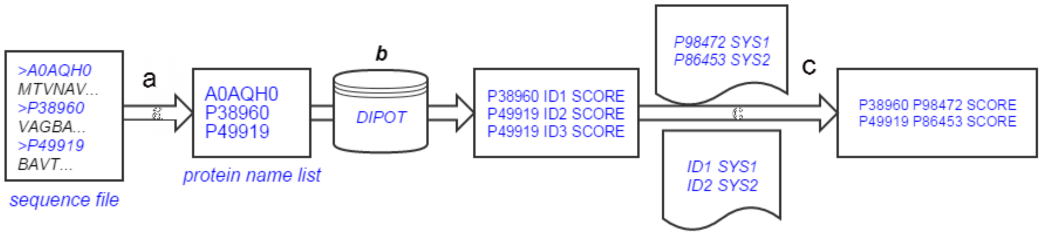


Figure 3.9: Identifier mapping Pipeline.

Finally, we collected 14,837 orthologs for *fly* and *yeast*; this is more than the 10,744, as claimed by AlignMCL [28].

3.4 Experimental Setup

In this section, we describe our novel procedure for predicting interactions and performing network alignment.

3.4.1 PPI Prediction

The procedure of predicting new PPIs is as follows:

- Transfer PPIs and protein sequences to numeric representation, each protein being assigned a unique number.

- Create the kernel matrix using Martin's program: *kernel_mat*.

- Train an SVM using Martin's program: *svm_learn*.

- Make the proteome wide predictions using Martin's program: *make_preds*.

- Process the resulting matrix to be used for alignment improvement, each cell in the matrix corresponds to a potential PPI and its reliability score.

It took 20 days to run Martin's prediction program for *fly* and nearly 40 hours for *yeast* on Linux. The resulting matrix is an all-versus-all proteins mapping with reliability score in corresponding cells. Only half of the matrix cells are needed to be filled. We extract every column and row and recover their UniprotKB Accession/ID using the identifier mapping file saved at the first step.

Of all the 21,915,510 interactions we obtained, there is no rule to judge which is true interaction. A higher score means a higher possibility of interacting, so we sorted all the interactions in descending order by their scores. We add interactions gradually starting with the most reliable ones.

3.4.2 Network Alignment

We designed four groups of experiments. We test on two datasets, and for each dataset, we add interactions from Martin's program and I2D separately.

- Test on the DIP dataset: perform alignment on the original DIP dataset first, then
  - Add PPIs from Martin's predicted interactions to both *fly* and *yeast* gradually and perform alignment after each addition.
  - Add PPIs from I2D gradually and perform alignment in the same way as above.
- Test on DIP&I2D dataset: again, we perform alignment on the original DIP&I2D dataset first, then
  - Add PPIs from Martin's predicted interactions to both *fly* and *yeast* gradually and perform alignment after each addition.
  - Add PPIs from I2D gradually and perform alignment in the same way as above.

For each alignment, we test using both MaWISH and AlignMCL. The numbers of PPIs we added at each step are: 2000, 3000, . . . , 8000, 10000 and 15000.

The input of an alignment program consists of two PPI networks and a protein ortholog file. In MaWISH, we set the parameter of a match to 1, while the parameters of mismatch and duplication are set at -0.1 and 0.1 especially. Only direct paths are considered in our experiments because indirect path in MaWISH can not produce better results, while slowing down the program significantly. In all the experiments using MaWISH, the parameter for match distance is 1.

In AlignMCL, edges at distance 2 are allowed. The inflation level is set to 2.8. We have not tuned the parameters instead used the best parameters as reported in the AlignMCL paper [28].

MaWISH is much faster than AlignMCL with less than 1 second in running every alignment, while AlignMCL took nearly 2 minutes on the average for every alignment. We obtained 76 alignment files in total, each file containing all the subnetworks found by the corresponding alignment. Each such subnetwork is called a *solution*.

Chapter 4

Results and Evaluation

In this chapter, we introduce the evaluation guidelines for our experiments, perform the tests and discuss the results. Besides significant improvement of the PPI network alignments, many interesting phenomena are observed.

4.1 Evaluation guidelines

A popular method for evaluating a local network alignment is to count the number of known protein complexes recovered by the alignment. For every alignment result, we check the solutions and evaluate them in the two species separately. A formal description of this method is given below. Given a solution S_i and a known protein complex module M_j, a simple approach to compare them is calculating the number of proteins in their overlap. More specifically, we use precision (π, also called Positive Predictive Value) and recall (ρ, also known as Sensitivity) to quantify the quality of overlap. *Precision* represents the percentage of proteins in the solution that are also present in the module. See Equation 4.1; while *recall* measures the percentage of proteins in the module that are in common with the solution; see Equation 4.2. Then we use the *F-index* function to measure the overall performance; see Equation 4.3.

$$\pi = \frac{|M_j \cap S_i|}{|S_i|} \tag{4.1}$$

$$\rho = \frac{|M_j \cap S_i|}{|M_j|} \tag{4.2}$$

$$\textit{F-index} = \frac{2\pi\rho}{\pi + \rho} \tag{4.3}$$

The *F-index* ranges in the interval [0, 1], with 1 corresponding to perfect agreement. For every known complex, we compute its overlap with every solution and record the solution with highest *F-index.* Given a protein complex M_i of one species and solutions $S = \{S_1, S_2, \ldots, S_n\}$ of an alignment, we define the best matching for complex M_i as follows:

$$B_{M_i} = argmax_j \quad \text{F-index}(M_i, S_j) \tag{4.4}$$

We compare the protein complex M_i with all the solutions and select the one which produces highest *F-index.* Given a set of protein complexes $M = \{M_1, M_2, \ldots\}$, the best matching complexes vector is defined as:

$$BestMatch\{M, S\} = \{B_{M_1}, B_{M_2}, \ldots\} \tag{4.5}$$

For every protein complex, we compute their best matching solution and form this *BestMatch* vector. To evaluate our alignments, we fix a lower bound on the F-index of a best match and count these matches in *BestMatch*$\{M, S\}$ that have the F-index above this threshold. As considered in the literature, we use as lower bounds for the F-index the values 0.3 and 0.5.

4.2 Test Datasets

The *known protein complexes* used in the evaluation are downloaded from *CYC2008* and *DPIM.* The summary of raw complexes we downloaded is shown in Table 4.1. Within each complexes dataset, there are some complexes with similar biological functions and highly overlapping with each other. This might lead to a biased evaluation, since a solution can overlap with more than one *known complex*, therefore be counted more than once. Moreover, these overlapping complexes are often quite small (2-4 proteins). AlignMCL merges these complexes together, see [28] for details. Therefore, we used merged complexes for evaluation. A summary of merged complexes is also shown in Table 4.1.

Table 4.1: Statistics of *known protein complexes*

Species	Dataset	Raw Complexes	Merged Complexes
fly	DPIM	556	153
yeast	CYC2008	408	345

4.3 Results and analysis

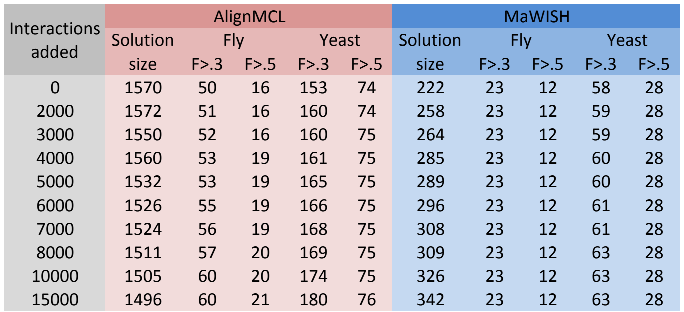
As described in the previous chapter, we use interactions from *fly* and *yeast* from the DIP dataset and the subset of DIP that appears also in I2D, denoted DIP&I2D. The interactions added are from Martin's program ([25]) and the I2D dataset.

As we compare the two top programs AlignMCL and MaWISH for two thresholds of the F-index, 0.3 and 0.5, we obtain 32 different tests. In each test, the number of interactions added has been varied from 0 to 15,000, where 0 correspond to the original datasets, that is, no interactions were added. In total, we have obtained 320 values that are shown in Tables 4.2-4.5. The first two tables, 4.2 and 4.3, contain the results for adding to the DIP dataset Martin's predicted interactions and I2D interactions, respectively. The last two tables, 4.4 and 4.5, contain the corresponding results for the DIP&I2D dataset instead of DIP. In all tables F stands for F*-index*.

4.3.1 Improvement of PPI network alignment

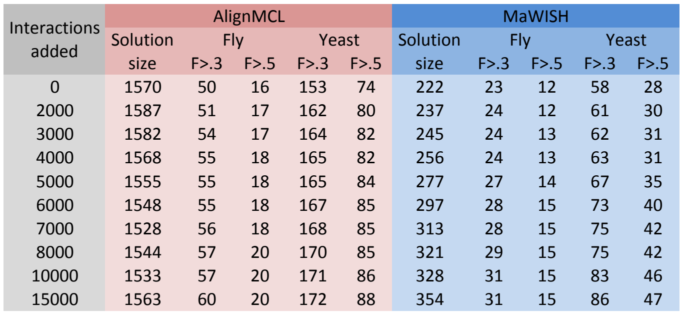
Our most important result and contribution is the improvement of the alignment of the PPI networks. This is clearly seen in Table 4.6 where the improvement in each case is

Table 4.2: Results for adding Martin's predicted interactions to DIP dataset.



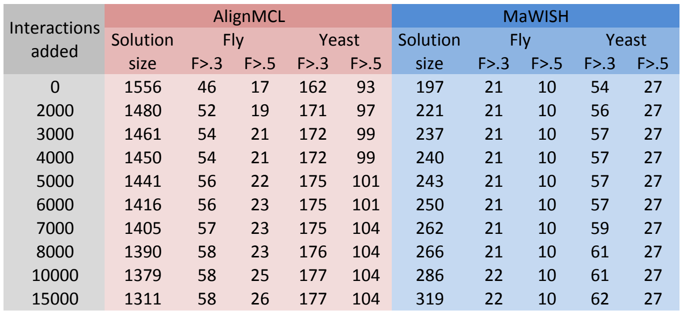
Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1570	50	16	153	74	222	23	12	58	28
2000	1572	51	16	160	74	258	23	12	59	28
3000	1550	52	16	160	75	264	23	12	59	28
4000	1560	53	19	161	75	285	23	12	60	28
5000	1532	53	19	165	75	289	23	12	60	28
6000	1526	55	19	166	75	296	23	12	61	28
7000	1524	56	19	168	75	308	23	12	61	28
8000	1511	57	20	169	75	309	23	12	63	28
10000	1505	60	20	174	75	326	23	12	63	28
15000	1496	60	21	180	76	342	23	12	63	28

Table 4.3: Results for adding I2D's interactions to DIP dataset.



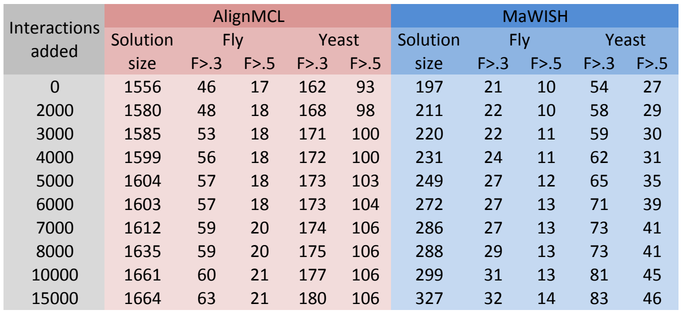
Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1570	50	16	153	74	222	23	12	58	28
2000	1587	51	17	162	80	237	24	12	61	30
3000	1582	54	17	164	82	245	24	13	62	31
4000	1568	55	18	165	82	256	24	13	63	31
5000	1555	55	18	165	84	277	27	14	67	35
6000	1548	55	18	167	85	297	28	15	73	40
7000	1528	56	18	168	85	313	28	15	75	42
8000	1544	57	20	170	85	321	29	15	75	42
10000	1533	57	20	171	86	328	31	15	83	46
15000	1563	60	20	172	88	354	31	15	86	47

Table 4.4: Results for adding Martin's predicted interactions to DIP&I2D dataset.



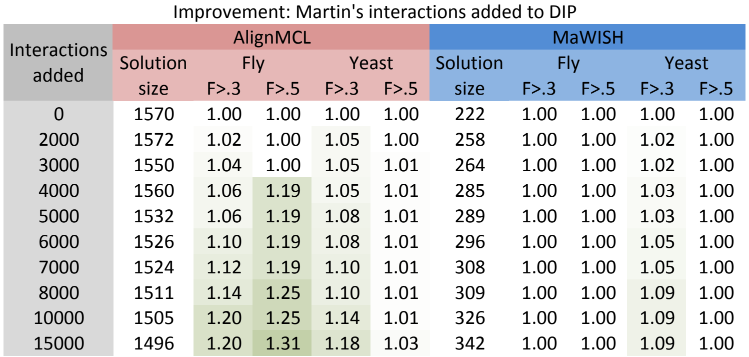
Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1556	46	17	162	93	197	21	10	54	27
2000	1480	52	19	171	97	221	21	10	56	27
3000	1461	54	21	172	99	237	21	10	57	27
4000	1450	54	21	172	99	240	21	10	57	27
5000	1441	56	22	175	101	243	21	10	57	27
6000	1416	56	23	175	101	250	21	10	57	27
7000	1405	57	23	175	104	262	21	10	59	27
8000	1390	58	23	176	104	266	21	10	61	27
10000	1379	58	25	177	104	286	22	10	61	27
15000	1311	58	26	177	104	319	22	10	62	27

Table 4.5: Results for adding I2D's interactions to DIP&I2D dataset.



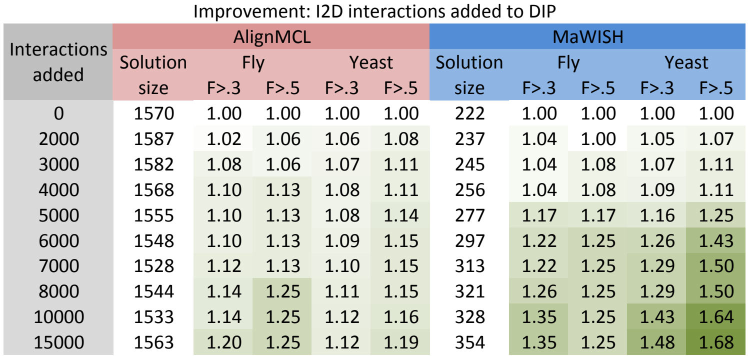
Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1556	46	17	162	93	197	21	10	54	27
2000	1580	48	18	168	98	211	22	10	58	29
3000	1585	53	18	171	100	220	22	11	59	30
4000	1599	56	18	172	100	231	24	11	62	31
5000	1604	57	18	173	103	249	27	12	65	35
6000	1603	57	18	173	104	272	27	13	71	39
7000	1612	59	20	174	106	286	27	13	73	41
8000	1635	59	20	175	106	288	29	13	73	41
10000	1661	60	21	177	106	299	31	13	81	45
15000	1664	63	21	180	106	327	32	14	83	46

Table 4.6: PPI network alignment improvement due to adding interactions. The tables correspond in order to the data in Tables 4.2-4.5.



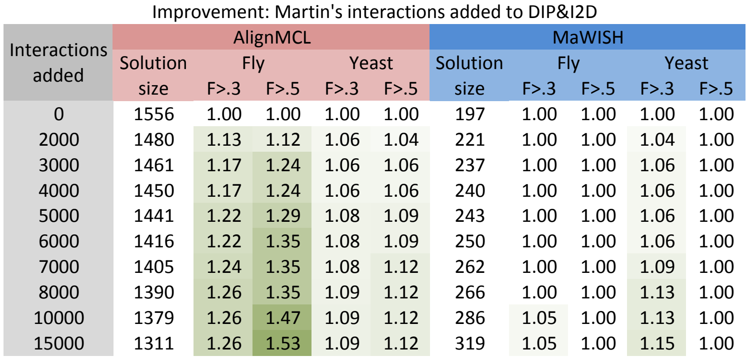
Improvement: Martin's interactions added to DIP

Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1570	1.00	1.00	1.00	1.00	222	1.00	1.00	1.00	1.00
2000	1572	1.02	1.00	1.05	1.00	258	1.00	1.00	1.02	1.00
3000	1550	1.04	1.00	1.05	1.01	264	1.00	1.00	1.02	1.00
4000	1560	1.06	1.19	1.05	1.01	285	1.00	1.00	1.03	1.00
5000	1532	1.06	1.19	1.08	1.01	289	1.00	1.00	1.03	1.00
6000	1526	1.10	1.19	1.08	1.01	296	1.00	1.00	1.05	1.00
7000	1524	1.12	1.19	1.10	1.01	308	1.00	1.00	1.05	1.00
8000	1511	1.14	1.25	1.10	1.01	309	1.00	1.00	1.09	1.00
10000	1505	1.20	1.25	1.14	1.01	326	1.00	1.00	1.09	1.00
15000	1496	1.20	1.31	1.18	1.03	342	1.00	1.00	1.09	1.00



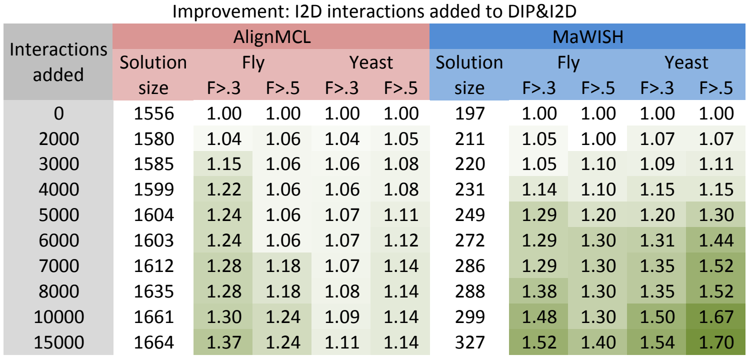
Improvement: I2D interactions added to DIP

Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1570	1.00	1.00	1.00	1.00	222	1.00	1.00	1.00	1.00
2000	1587	1.02	1.06	1.06	1.08	237	1.04	1.00	1.05	1.07
3000	1582	1.08	1.06	1.07	1.11	245	1.04	1.08	1.07	1.11
4000	1568	1.10	1.13	1.08	1.11	256	1.04	1.08	1.09	1.11
5000	1555	1.10	1.13	1.08	1.14	277	1.17	1.17	1.16	1.25
6000	1548	1.10	1.13	1.09	1.15	297	1.22	1.25	1.26	1.43
7000	1528	1.12	1.13	1.10	1.15	313	1.22	1.25	1.29	1.50
8000	1544	1.14	1.25	1.11	1.15	321	1.26	1.25	1.29	1.50
10000	1533	1.14	1.25	1.12	1.16	328	1.35	1.25	1.43	1.64
15000	1563	1.20	1.25	1.12	1.19	354	1.35	1.25	1.48	1.68



Improvement: Martin's interactions added to DIP&I2D

Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1556	1.00	1.00	1.00	1.00	197	1.00	1.00	1.00	1.00
2000	1480	1.13	1.12	1.06	1.04	221	1.00	1.00	1.04	1.00
3000	1461	1.17	1.24	1.06	1.06	237	1.00	1.00	1.06	1.00
4000	1450	1.17	1.24	1.06	1.06	240	1.00	1.00	1.06	1.00
5000	1441	1.22	1.29	1.08	1.09	243	1.00	1.00	1.06	1.00
6000	1416	1.22	1.35	1.08	1.09	250	1.00	1.00	1.06	1.00
7000	1405	1.24	1.35	1.08	1.12	262	1.00	1.00	1.09	1.00
8000	1390	1.26	1.35	1.09	1.12	266	1.00	1.00	1.13	1.00
10000	1379	1.26	1.47	1.09	1.12	286	1.05	1.00	1.13	1.00
15000	1311	1.26	1.53	1.09	1.12	319	1.05	1.00	1.15	1.00



Improvement: I2D interactions added to DIP&I2D

Interactions added	AlignMCL					MaWISH				
	Solution	Fly		Yeast		Solution	Fly		Yeast	
	size	F>.3	F>.5	F>.3	F>.5	size	F>.3	F>.5	F>.3	F>.5
0	1556	1.00	1.00	1.00	1.00	197	1.00	1.00	1.00	1.00
2000	1580	1.04	1.06	1.04	1.05	211	1.05	1.00	1.07	1.07
3000	1585	1.15	1.06	1.06	1.08	220	1.05	1.10	1.09	1.11
4000	1599	1.22	1.06	1.06	1.08	231	1.14	1.10	1.15	1.15
5000	1604	1.24	1.06	1.07	1.11	249	1.29	1.20	1.20	1.30
6000	1603	1.24	1.06	1.07	1.12	272	1.29	1.30	1.31	1.44
7000	1612	1.28	1.18	1.07	1.14	286	1.29	1.30	1.35	1.52
8000	1635	1.28	1.18	1.08	1.14	288	1.38	1.30	1.35	1.52
10000	1661	1.30	1.24	1.09	1.14	299	1.48	1.30	1.50	1.67
15000	1664	1.37	1.24	1.11	1.14	327	1.52	1.40	1.54	1.70

presented as a ratio of the original alignment. The table is presented as a heat map, with darker colours representing better results. No improvement corresponds to 1.00, coloured in white.

The improvement varies very much with the dataset being used, the interactions added, and the alignment program. As many as 70% new complexes can be detected (for MaWISH, when adding 15,000 I2D interactions to the DIP&I2D dataset), which is a very large improvement of the original alignment.

AlignMCL alignments are always improved, between to 3% and 50% with 15,000 interactions added. MaWISH has a very different behaviour, observing little or no improvement when Martin's interactions are used and a very large improvement, between 25% and 70% when 15,000 I2D interactions are used.

4.3.2 Performance comparison of AlignMCL and MaWISH

The comparison between the two top aligning programs AlignMCL and MaWISH is shown in the plots in Figure 4.1 for *fly* and Figure 4.2 for *yeast*. Each figure contains four plots, corresponding to adding Martin's interactions (top plots in each figure) or I2D interactions (bottom plots) to the DIP dataset (left plots) and DIP&I2D dataset (right plots). Each plot contains four curves, two for each program AlignMCL and MaWISH. The two curves corresponding to the same program are identically coloured and shaped, the higher one corresponding to F-index threshold 0.3 and the lower to F-index threshold 0.5.

While the improvement varies with different parameters, AlignMCL clearly outperforms MaWISH on all datasets. On the original datasets, AlignMCL discovers between 1.33 and 3.44 more complexes, and it can discover up to 3.85 times more complexes when interactions are added.

In the case of adding Martin's interactions to the DIP&I2D dataset for *fly*, the number of complexes discovered by AlignMCL with F-index at least 0.5 gradually exceeds the number of those discovered by MaWISH with F-index at least 0.3 when interactions are added.

We have discussed the advantage of AlignMCL in Chapter 3. The use of *union graph*

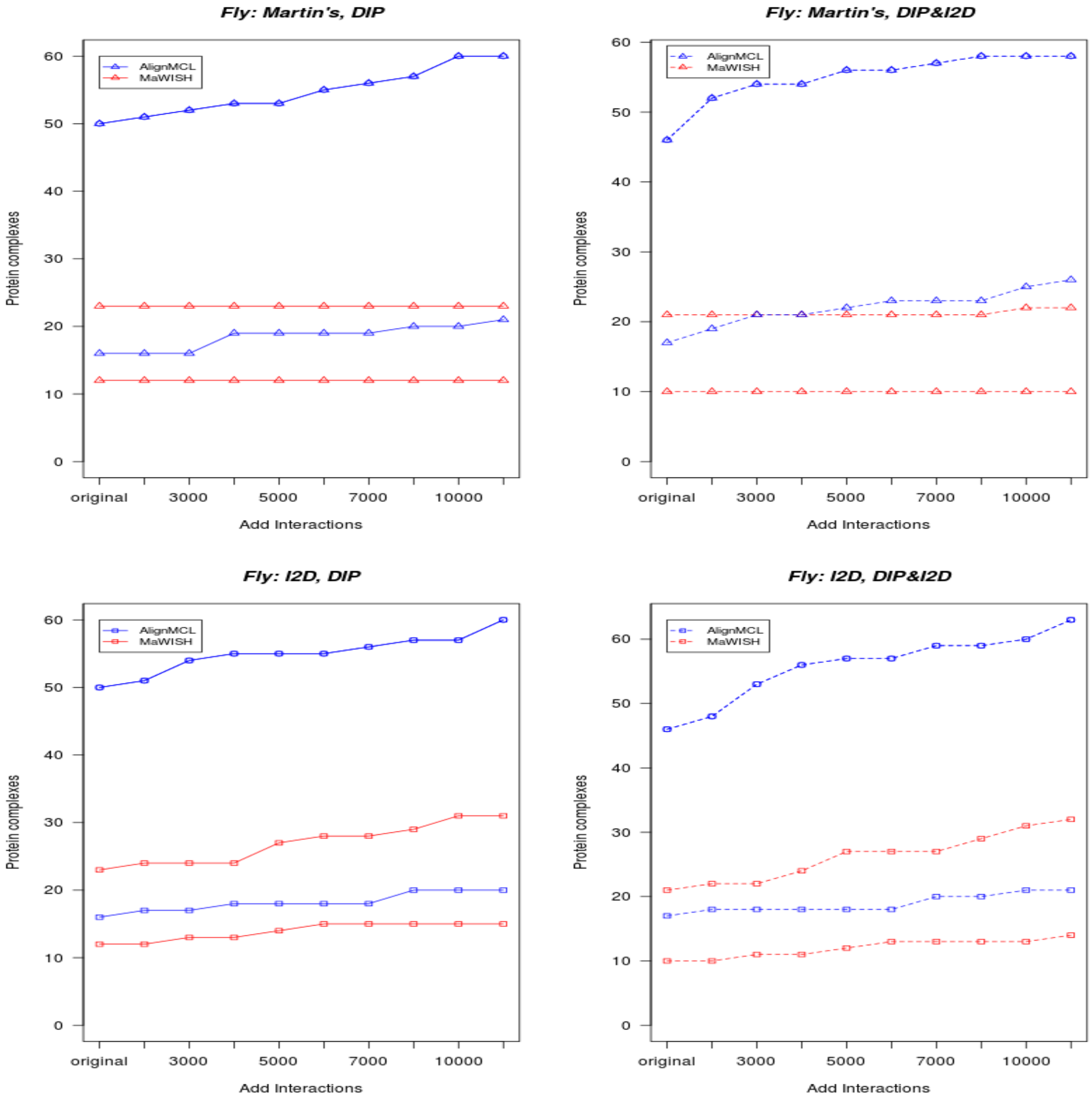


Figure 4.1: AlignMCL vs MaWISH for *fly*.

in AlignMCL is able to score direct and indirect edges separately, while in MaWISH, it is less efficient to use indirect paths. Besides this, the MCL used in AlignMCL seems more powerful than the greedy strategy of MaWISH for mining heavy subgraphs.

Still, MaWISH has some advantages as well. An interesting phenomenon we observed is that MaWISH produces more robust results even though its overall performance is worse than AlignMCL. AlignMCL will lose some complexes it found before adding interactions, so when calculating the high quality solutions, we need to investigate how many

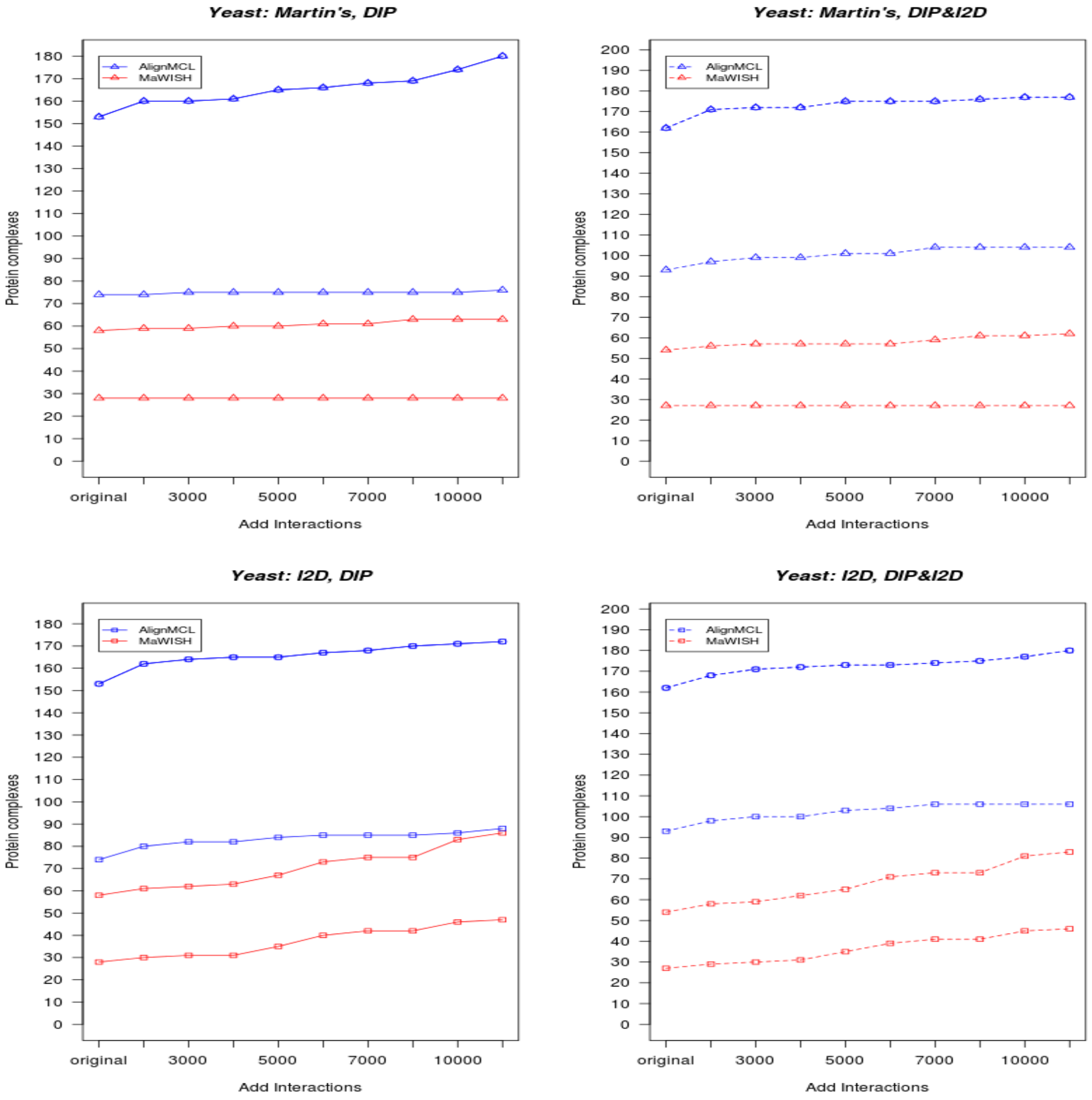


Figure 4.2: AlignMCL vs MaWISH for *yeast*.

complexes were lost and how many new ones discovered. MaWISH is stable in all the experiments; none of the complexes found before adding interactions is lost. This advantage is due to the strict model MaWISH uses. It enables reliable interactions to keep a relatively high weight, while AlignMCL depends more on the graph structure when using MCL.

Another observation is that MaWISH has a much higher percentage of discovered complexes out of the total number of solutions. That is, AlignMCL discovers more complexes but it also produces many more solutions, many of which do not discover any complexes. The ratio of discovered complexes can be as high as 27% compared with only up to 14% for AlignMCL. Also, the total number of solutions is expected to increase with the addition of more interactions. This happens indeed for MaWISH but not for ALignMCL.

4.3.3 Reliability of predicted PPIs

Predicting PPIs is a very difficult problem and we have discussed this aspect earlier in the thesis. No gold standard exists for testing the reliability of predicted PPIs. Our work provides an indirect way of assessing how reliable predicted PPIs are. We compare in this section the PPIs predicted by Martin's program with those in the I2D dataset.

The comparison is shown in Figure 4.3 for *fly* and Figure 4.4 for *yeast*. Each figure is organized as before, with four plots, the top two for the DIP dataset and bottom two for DIP&I2D dataset; the two left plots correspond to AlignMCL and the right ones to MaWISH.

The impact of adding interactions predicted by Martin's program or from I2D is similar to the alignments computed by AlignMCL while there is a very large difference on those of MaWISH. In fact, for *fly*, there is essentially no improvement when adding Martin's predicted interactions; a single new complex is discovered when adding 10,000 interactions to the DIP&I2D dataset. For *yeast* some improvement is detected, however much smaller than that obtained by using I2D interactions. This gives a good indication that the predictions of Martin's program may not be as reliable as believed.

Note also that the total number of solutions for AlignMCL decreases significantly when adding Martin's interactions. It increases or exhibits non-monotonic behaviour when adding I2D interactions.

4.3.4 Intersection dataset performance

Intuitively, an interaction is expected to be more reliable if it was verified by more experiments or appears in more databases. In order to further improve local network

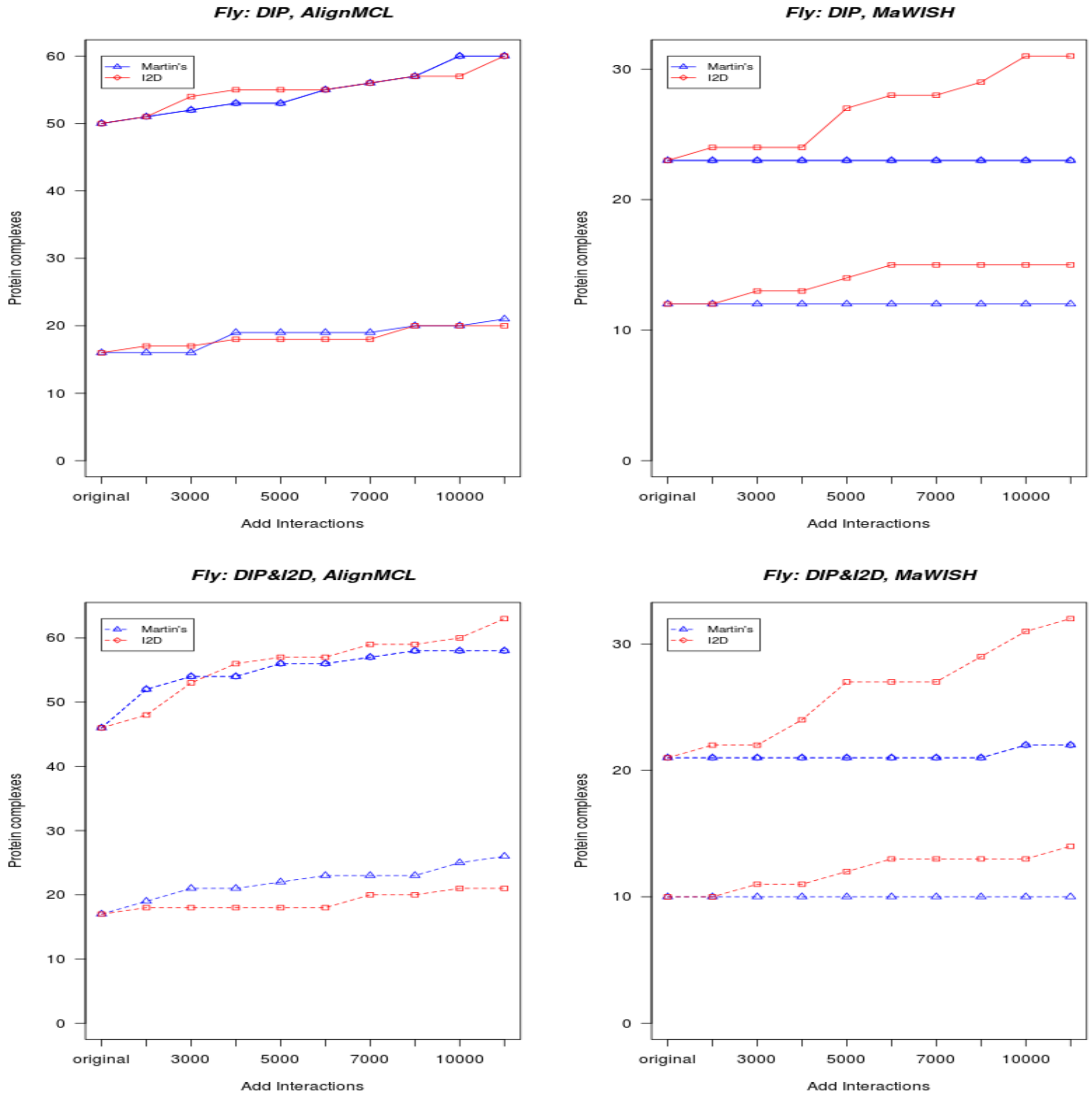


Figure 4.3: Martin's vs I2D for *fly*.

alignment, we performed our tests also on the DIP&I2D dataset, that is, we considered only those interactions that are included in both datasets DIP and I2D. The results are plotted in Figure 4.5 for *fly* and Figure 4.6 for *yeast*. As before, each figure includes four plots for the remaining combinations: the two on the top consider the cases when Martin's predicted interactions are added, while the two bottom ones when I2D interactions are added; AlignMCL alignments are shown in the left plots, and MaWISH in the right.

Considering the fact that DIP&I2D has considerably less interactions than DIP (see

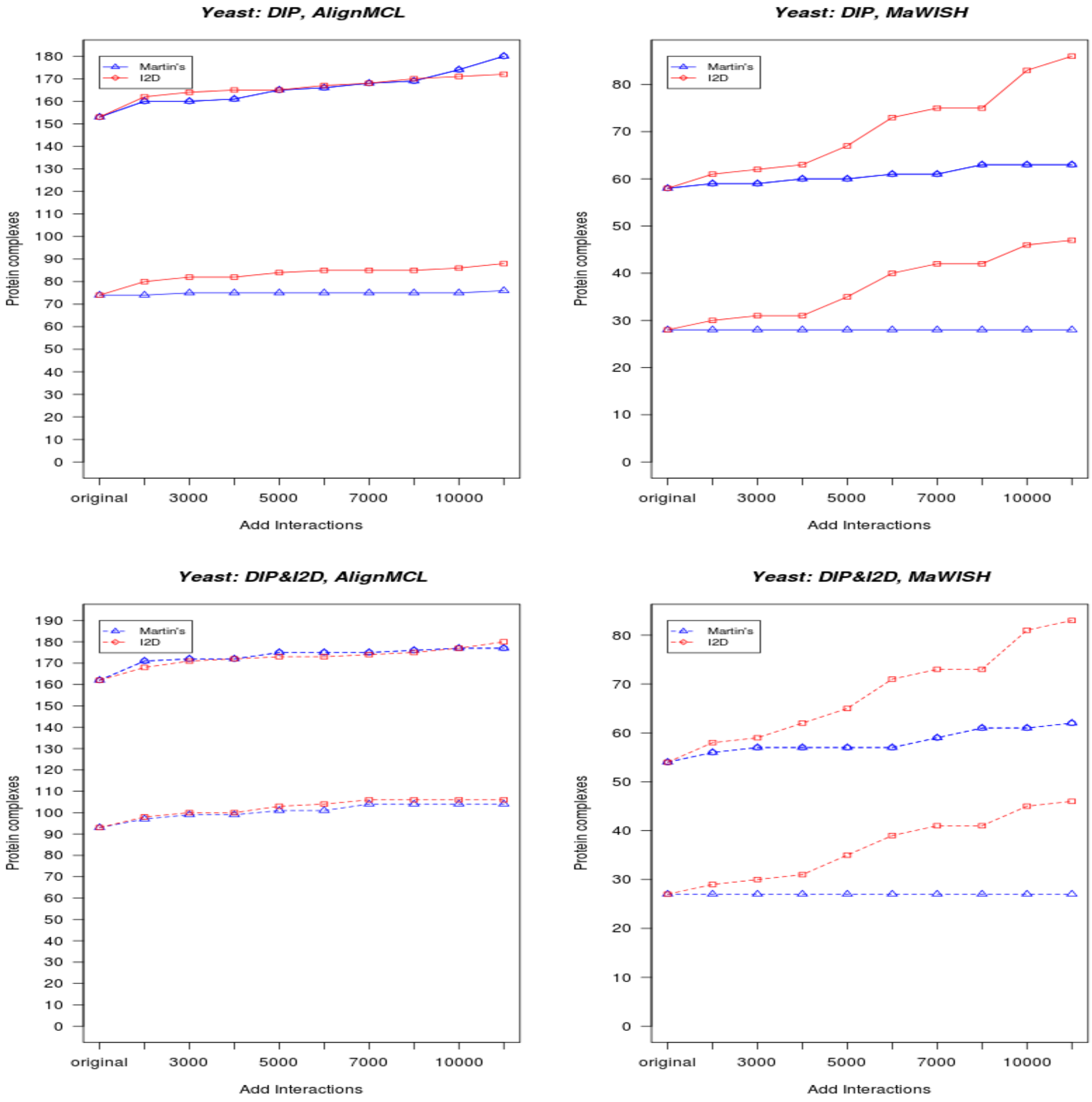


Figure 4.4: Martin's vs I2D for *yeast*.

Table 3.3), it is both surprising and very interesting that DIP&I2D performs better, and sometimes significantly better than DIP for the alignments computed by AlignMCL. For MaWISH, the results using DIP&I2D are only slightly behind those using DIP.

This shows clearly that it is not the number of the interactions but their quality that is the most important attribute. However, since DIP interactions are expected to be highly reliable, the results we obtained deserve more investigation.

PPI network alignment is complex due to the structure of the graph. A protein

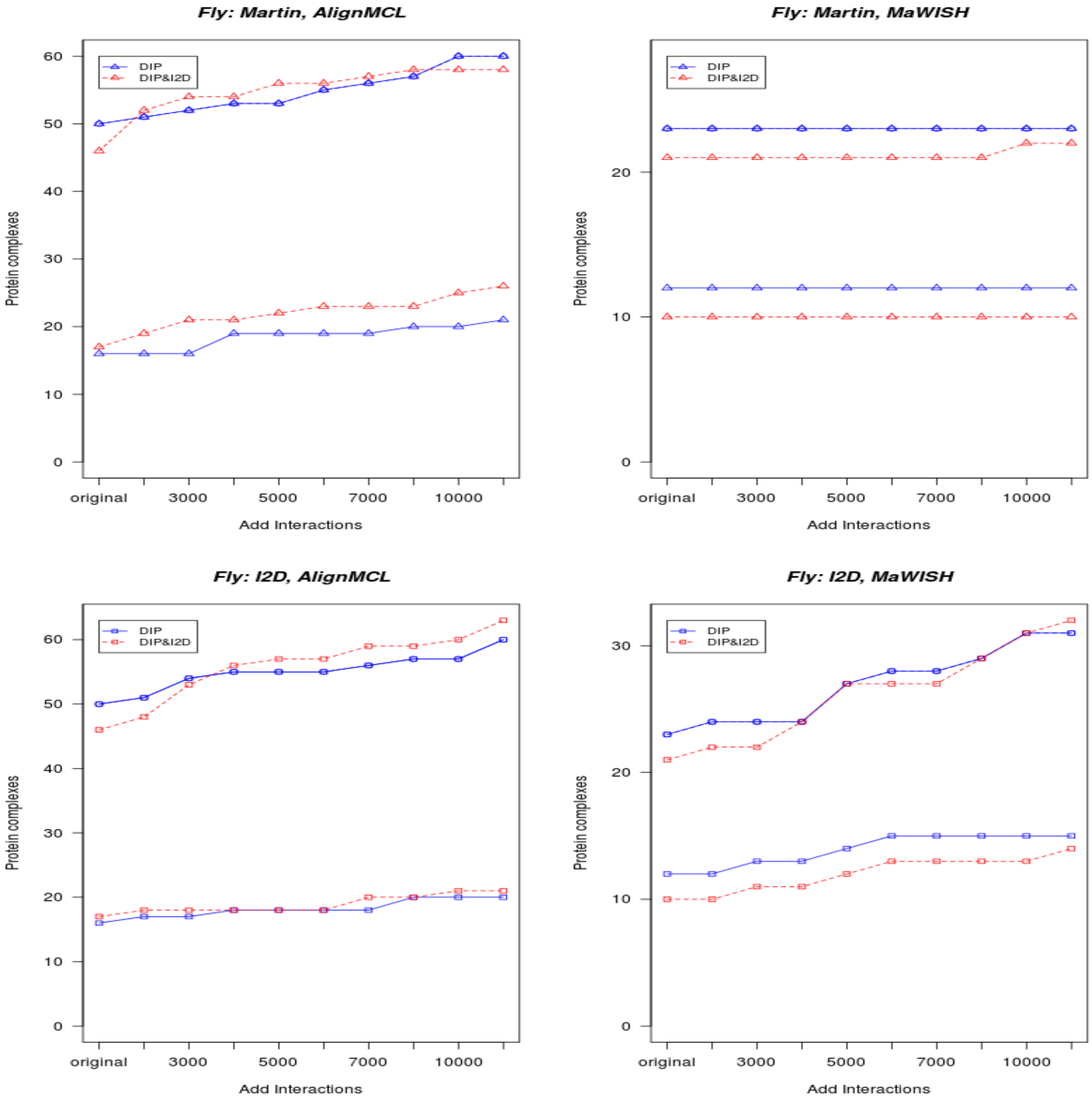


Figure 4.5: DIP vs DIP&I2D for *fly*.

complex usually contains only few proteins, but the total number of interactions in a PPI network is very large and different complexes also interact with each others, so it is difficult to find a perfect match between a solution and a protein complex. *F-index* is not a golden rule to assess the result of alignment. Many protein complexes exist in other solutions, but these solutions can not produce a high *F-index* as many unrelated proteins are included. For example, one solution may contain most of the proteins of a protein complex, but many other proteins also exist in this solution due to the large

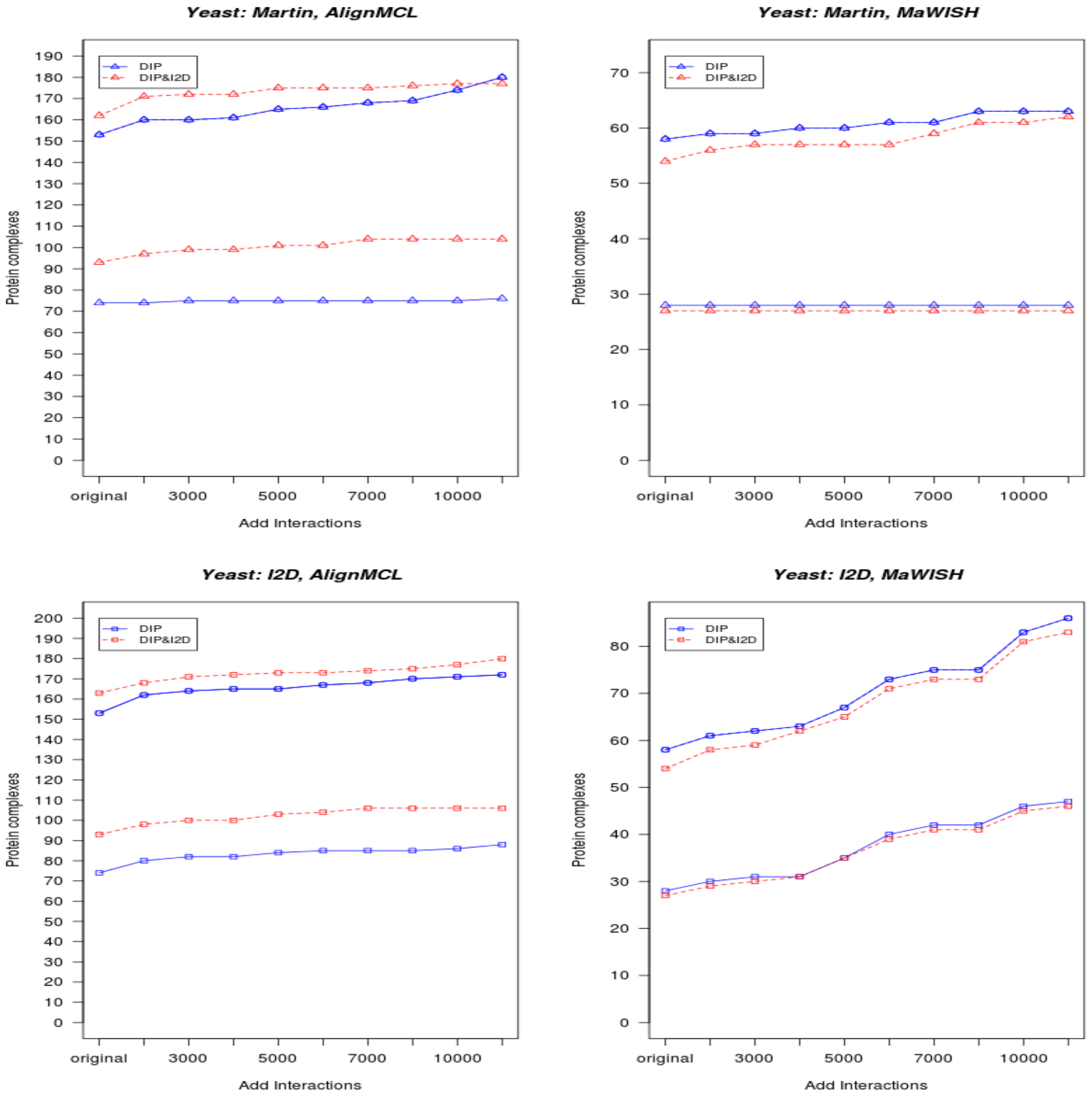


Figure 4.6: DIP vs DIP&I2D for *yeast*.

number of interactions. These abandoned solutions could be used to predict new PPIs or complexes.

Also, when constructing an alignment graph, only protein orthologs are preserved. According to our statistics, 6,094 out of 7,718 proteins in *fly* and 3,743 out of 5,033 proteins in *yeast* have orthologs. Many protein complexes composed of proteins that have no orthologs in other species remain undiscovered. For example, the OCA complex is a complex in *yeast* that contains P53965, P38738, P53949, P25366 and Q12454, but none of these five proteins have ortholog in *fly* according to current ortholog databases.

Chapter 5

Conclusion and Further Research

In this thesis, we studied the problem of protein-protein interaction network alignment. In spite of a large amount of work on this problem, significant room for improvement remains.

Our main contribution is the improvement of PPI network alignment by adding predicted interactions to the networks, prior to the computation of the alignment. The improvement we obtain can be quite large.

On our way to implementing the new idea, a number of additional preparations were necessary, such as reviewing, understanding, testing and selecting the best local network alignment and PPI prediction programs. In some cases, implementing the programs were necessary. Also, due to the complexity of multiple data sources, we also devised and implemented our own identifier mapping program to assist us in our research. Datasets had to be analyzed and their suitability assessed. Many experiments were carefully designed and performed.

After collecting the results of all experiments, we designed an evaluation method based on known protein complexes. While in general the performance of the aligning programs is increased by the addition of predicted interactions, a number of interesting phenomena were observed. They deserve further investigation as they may provide valuable insight into the inner workings of the algorithms as well as into the interplay between PPI networks and the aligning algorithms.

AlignMCL performs better than MaWISH but MaWISH seems more stable, as it never loses complexes that it has found. Understanding this phenomenon may lead to an improved way to align PPI networks.

Our result also shed light on the reliability of the interactions predicted by one of the top programs for PPI prediction. Unexpectedly, such interactions do not improve the alignments produced by MaWISH, whereas they do improve those of AlignMCL. Viewed in connection with the stability discussed above, this is an interesting phenomenon that needs further investigation.

Another unexpected behaviour has been detected when using only the interactions common to both DIP and I2D datasets. These interactions, much fewer than those in DIP, performed unexpectedly well and it is not very clear why, since the interactions in DIP are assumed highly reliable. It is of crucial importance that such behaviour is properly explained.

Finally, the current way of evaluating the quality of the alignment has the problem that, of thousands of solutions found by alignment algorithms, only hundreds of them are considered of high quality. The information contained in the remaining ones is unused even if it has potentially useful information.

Bibliography

[1] Ahmet Emre Aladag and Cesim Erten. Spinal: Scalable protein interaction network alignment. *Bioinformatics*, 15, February 2013. 8, 12

[2] Andrey Alexeyenko, Julia Lindberg, Åsa Pérez-Bercoff, and Erik LL Sonnhammer. Overview and comparison of ortholog databases. *Drug Discovery Today: Technologies*, 3(2):137–143, 2006. 7

[3] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W.Myers, and David J.Lipman. Basic local alignment search tool. *Jounral of Molecular Biology*, 215:403–410, 1990. 29

[4] Ruepp Andreas, Brauner Barbara, and Dunger-Kaltenbach et al Irmtraud. Corum: the comprehensive resource of mammalian protein complexes. *Nucleic acids research*, 36:D646–D650, 2008. 18

[5] Gary D Bader, Adrian Heilbut, Brenda Andrews, Mike Tyers, Timothy Hughes, and Charles Boone. Functional genomics and proteomics: charting a multidimensional map of the yeast cell. *Trends in cell biology*, 13(7):344–356, 2003. 4

[6] A Barabasi and R Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999. 22

[7] K. R. Brown and I. Jurisica. Online predicted human interaction database. *Bioinformatics*, 21(9):2076–2082, May 2005. 28

[8] Leonid Chindelevitch, Chung-Shou Liao, and Bonnie Berger. Local optimization for global alignment of protein interaction networks. In *Pacific Symposium on Biocomputing*, volume 15, pages 123–132. World Scientific, 2010. 8

[9] Liao Chung-Shou, Lu Kanghao, Baym Michael, Singh Rohit, and Berger Bonnie. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009. 8, 10

[10] Giovanni Ciriello, Marcco Mina, and Pietro H. Guzzi et al. Alignnemo: A local network alignment method to integrate homology and topology. *PLoS One*, 7(6), June 2012. vi, viii, 8, 15, 16, 25, 26

[11] J De Las Rivas and C Fontanillo. Protein-protein interactions essentials key concepts to building and analyzing interactome networks. *PLoS computational biology*, 6(6), 2010. 17

[12] S. Enright, AJ. Van Dongen, and C. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research*, 30(7):1575–1584, 2002. vi, 26, 27, 28

[13] Stanley Fields and Ok-kyu Song. A novel genetic system to detect protein protein interactions. 1989. 4

[14] Jason Flannick, Antal Novak, Chuong B Do, Balaji S Srinivasan, and Serafim Batzoglou. Automatic parameter learning for multiple network alignment. In *Research in Computational Molecular Biology*, pages 214–231. Springer, 2008. 8

[15] Jason Flannick, Antal Novak, and Balaji S. Srinivasan et al. Graemlin:general and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, September 2006. 8

[16] K.G. Guruharsha, Jean-Franc, ois Rual, and Bo Zhai et al. A protein complex network of drosophila melanogaster. *Cell*, 147:690–703, 2011. 18

[17] Rashid Ibragimov, Maximilian Malek, Jiong Guo, and Jan Baumbach. Gedevo: An evolutionary graph edit distance algorithm for biological network alignment. *German Conference on Bioinformatics 2013*, February 2013. viii, 12, 13

[18] Matthew Jessulat, Sylvain Pitre, Yuan Gui, Mohsen Hooshyar, Katayoun Omidi, Bahram Samanfar, Le Hoa Tan, Md Alamgir, James Green, Frank Dehne, et al. Recent advances in protein-protein interaction prediction: experimental and computational methods. *Expert Opinion on Drug Discovery*, 6(9):921–935, 2011. vi, 1, 2, 3, 5, 6

[19] Brian P. Kelley and Sharan Richard M. Karp et al Roded. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, 100(20):11394–11399, September 2003. vi, 8, 14

[20] P. OBrien Kevin, Maido Remm, and Erik L. L. Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Research*, 33(5), 2005. 7, 16, 29

[21] Mehmet Koyuturk, Yohan Kim, and Umut Topkara et al. Pairwise alignment of protein interaction networks. *Journal of Computational Biology*, 13(2):182–199, 2006. 1, 8, 15

[22] Li Li, Christian J Stoeckert, and David S Roos. Orthomcl: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189, 2003. 7

[23] Salwinski Lukasz, Christopher S. Miller, Adam J. Smith, and Frank K. Pettit. The database of interacting proteins: 2004 update. *Nucleic acids research*, 32:D449–D451, 2004. 28

[24] Mina Marco and Pietro Hiram Guzzi. Alignmcl: Comparative analysis of protein interaction networks through markov clustering. *IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pages 174–181, 2012. 1, 26

[25] Shawn Martin, Diana Roe, and Jean-Loup Faulon. Predicting protein–protein interactions using signature products. *Bioinformatics*, 21(2):218–226, 2005. 2, 6, 19, 21, 36

[26] Koyuturk Mehmet, Grama Ananth, and Szpankowski Wojciech. Pairwise local alignment of protein interaction networks guided by models of evolution. *RECOMB*, 13(2):48–65, 2005. vi, 16, 21, 22, 24

[27] Vesna Memisevic and Natasa Przulj. C-graal:common-neighbors-based global graph alignment of biological networks. *The Royal Society of Chemistry Integrative Biology*, 4(7):734–743, December 2012. 11

[28] Marco Mina and Pietro Hiram Guzzi. Improving the robustness of local network alignment: Design and extensive assessment of a markov clustering-based approach. *IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, 11(3):561–572, June 2014. 16, 26, 31, 33, 36

[29] Jean Muller, Damian Szklarczyk, Philippe Julien, Ivica Letunic, Alexander Roth, Michael Kuhn, Sean Powell, Christian von Mering, Tobias Doerks, Lars Juhl Jensen, et al. eggnog v2.0: extending the evolutionary genealogy of genes with enhanced non-supervised orthologous groups, species and functional annotations. *Nucleic acids research*, 38(suppl 1):D190–D195, 2010. 7

[30] Behnam Neyshabur, Ahmadreza Khadem, Somaye Hashemifar, and Seyed Shahriar Arab. Netal: a new graph-based method for global alignment of protein protein interaction networks. *Bioinformatics*, 29(13):1654–1662, May 2013. 8, 12, 13

[31] Kuchaiev Oleksii and Przulj Natasa. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 00(7), 2010. 11

[32] Kuchaiev Oleksii, Milenkovic Tijana, and Memisevic et al Vesna. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 2010(7):1341–1354, March 2010. 8, 10

[33] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999. 9

[34] Akhilesh Pandey and Matthias Mann. Proteomics to study genes and genomes. *Nature*, 405(6788):837–846, 2000. 4, 6

[35] Yungki Park. Critical assessment of sequence-based protein-protein interaction prediction methods that do not require homologous protein sequences. *BMC bioinformatics*, 10(1):419, 2009. 19

[36] Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012. 8

[37] Hang TT Phan and Michael JE Sternberg. Pinalog: a novel approach to align protein interaction networksimplications for complex detection and function prediction. *Bioinformatics*, 28(9):1239–1245, 2012. 8

[38] S Pitre, C North, M Alamgir, M Jessulat, A Chan, X Luo, JR Green, M Dumontier, F Dehne, and A Golshani. Global investigation of protein–protein interactions in yeast saccharomyces cerevisiae using re-occurring short polypeptide sequences. *Nucleic acids research*, 36(13):4286–4294, 2008. vi, 2, 6, 19, 20

[39] Natasa Przulj. *Knowledge Discovery in Proteomics: Graph Theory Analysis of Protein-Protein Interactions*. Number Ed.1. Springer Express, 2005. vi, 5

[40] Guillaume Rigaut, Anna Shevchenko, Berthold Rutz, Matthias Wilm, Matthias Mann, and Bertrand Séraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature biotechnology*, 17(10):1030–1032, 1999. 4

[41] Jinbo Xu Rohit Singh and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *PNAS*, 105(35):12763–12768, 2008. vi, 8, 9

[42] Sayed Mohammad Ebrahim Sahraeian and Byung-Jun Yoon. Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PloS one*, 8(7):e67995, 2013. 8

[43] Benno Schwikowski, Peter Uetz, and Stanley Fields. A network of protein–protein interactions in yeast. *Nature biotechnology*, 18(12):1257–1261, 2000. 4

[44] Mohammadi Shahin and Grama Ananth. *Functional Coherence of Molecular Networks*. Number Ed.1 in 36. Springer Express, 2006. 7

[45] Roded Sharan, Silpa Suthram, and Ryan M. Kelley et al. Conserved patterns of protein interaction in multiple species. *PNAS*, 102(6):1974–1979, February 2004. vi, 8, 14, 15

[46] Pu Shuye, Wong Jessica, Turner Brian, Cho Emerson, and J. Wodak Shoshana. Up-to-date catalogues of yeast protein complexes. *Nucleic acids research*, 37(3):825–831, December 2009. 18

[47] Roded Sharan Sourav Bandyopadhyay and Trey Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome Research*, 16(1):428–435, 2006. 14

[48] Roman L Tatusov, Natalie D Fedorova, John D Jackson, Aviva R Jacobs, Boris Kiryutin, Eugene V Koonin, Dmitri M Krylov, Raja Mazumder, Sergei L Mekhedov, Anastasia N Nikolskaya, et al. The cog database: an updated version includes eukaryotes. *BMC bioinformatics*, 4(1):41, 2003. 6, 23

[49] Wenhong Tian and Nagiza F Samatova. Pairwise alignment of interaction networks by fast identification of maximal conserved patterns. 14:99–110, 2009. 8

[50] Milenkovic Tijana and Przulj Natasa. Uncovering biological network function via graphlet degree signatures. *RECOMB Satellite Conference on Systems Biology*, 2010(7):1–9, 2007. vi, 11, 12

[51] Martijn P van Iersel1, Alexander R Pico, and Thomas Kelder et al. The bridgedb framework: standardized access to gene, protein and metabolite identifier mapping services. *BMC Bioinformatics*, 11(5):1–7, 2010. 30

[52] Abraham White, Philip Handler, Emil Smith, DeWitt Stetten Jr, et al. Principles of biochemistry. *Principles of Biochemistry.*, (Edn 2), 1959. 3

Curriculum Vitae

Name:	Yu Qian
Post-Secondary Education and Degrees:	University of Western Ontario London, Ontario, Canada 2013 - present M.Sc. candidate
	Harbin Institute of Technology Harbin, Heilongjiang, China 2009 - 2013 B.Eng.
Honours and Awards:	National Scholarship Harbin Institute of Technology, China, 2012-2013
	Third Prize in "Energy Conservation and Emission Reduction" National Academic Science and Technology Contest Ministry of Education, China, 2012
Related Work Experience:	Teaching Assistant The University of Western Ontario 2013 - present
	Research Assistant The University of Western Ontario 2013 - present
	Lab Member Provincial Digital Ship Manufacturing System Development Center Harbin Institute of Technology, China, 2010 - 2013
	Software Developer Intern NeuSoft Corporation China, 2011