UNIVERSITÀ DELLA CALABRIA
Dipartimento di MATEMATICA

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Matematica

Dottorato di Ricerca in Matematica ed Informatica

XXII ciclo

Tesi di Dottorato

Knowledge Discovery in Bioinformatics: from Simple to Complex Structures

Valeria Fionda

UNIVERSITÀ DELLA CALABRIA

Dipartimento di Matematica

Dottorato di Ricerca in Matematica e Informatica

XXII ciclo

Settore Disciplinare: INF/01 INFORMATICA

Tesi di Dottorato

Knowledge Discovery in Bioinformatics: from Simple to Complex Structures

Valeria Fionda

Valeria Fionda

Coordinatore
Prof. Nicola Leone

Supervisore
Prof. Luigi Palopoli

Acknowledgements

I wish to thank my PhD advisor prof. Luigi Palopoli for the precious advices that led me through my personal and professional growth.

I wish to thank my parents and my sisters for their faithful support.

I wish to thank my best friend Ilario for always being close, despite the distance.

Last but not least, a special thank to Sweetie for his help and moral support, for sharing with me dreams and expectations, notwithstanding the dark road along which we are walking together.

To the horse on three hooves,
died in Place du Carrousel and

to Jacques Prévert
who portrayed him so well.

To Sweetie
for enlightening my way.

Preface

Life is the point upon which the whole universe lies. However, the knowledge about life machinery is very poor in comparison to the complexity of biological processes regulating it. Many efforts have been done to better understand mechanisms underlying life, grasp the key concepts about the processes of birth, growth and death and trim the incompleteness of the knowledge about life basic elements.

This thesis is meant to be helpful in this direction, trying to enlighten some shady issues relevant in bioinformatics. In particular, the work that has been done tries to clarify some biological processes regulating cell life cycle in different organisms, by comparing their simple and complex building blocks.

On the one hand, simple biological structures (i.e., proteins) have been analyzed. This way, the unknown functions of uncharacterized proteins or the biological processes in which they are involved can be determined. To this aim two approaches have been devised:

- PQSC-FCNN: a tool for predicting protein quaternary structure, which is related to the biological function of the protein when involved in specific biological processes.
- Bi-Grappin: a tool for annotating proteins with functional information by comparing protein-protein interaction networks.

On the other hand, complex biological structures (i.e, biological networks) of different organisms have been explored. This way, functional modules conserved during the evolution can be identified. In this respect, two approaches have been proposed:

- Sub-Grappin: a tool for the pairwise alignment of protein-protein interaction networks.
- PInG-Q: a tool for querying protein-protein interaction networks.

The above mentioned approaches have been proved, by experimental evaluations, to be able to discover significant biological results. This is promising since it allows to help in complementing the knowledge about biological processes regulating the cell life cycle. This way, by looking within the simple elements of life (i.e., living cells) the knowledge beyond these simple elements can be grasped.

A look to the future research perspectives, in this promising research area, hints that the efforts payed in this direction can be greatly rewarded through the results that will be obtained in the long term.

Rende (CS), Italy, *Valeria Fionda*
November 2009

Contents

Part I Bioinformatics: Background and Uptake

Part II Protein Function Prediction

Part III Network Alignment

Part IV Network Querying

Part V Conclusions and Future Trends

List of Figures

List of Tables

Part I

Bioinformatics: Background and Uptake

1

Introduction and Overview

The development of biotechnology, that is, the application of the principles of engineering and technology to the life sciences, has led to the birth of a new field of research: Bioinformatics. Bioinformatics was born at the end of the 70s when the emerging ICTs found a wide use in the project of genome sequencing.

Several definitions of Bioinformatics have been proposed, all of which underline the role of this research area as a bridge linking life science and computer science. The National Center for Biotechnology Information (NCBI), for instance, defines bioinformatics as:

"Bioinformatics is the field of science in which biology, computer science, and information technology merge into a single discipline. There are three important subdisciplines within bioinformatics: the development of new algorithms and statistics with which to assess relationships among members of large data sets; the analysis and interpretation of various types of data including nucleotide and amino acid sequences, protein domains, and protein structures; and the development and implementation of tools that enable efficient access and management of different types of information."

Generally speaking, bioinformatics tasks can be subdivided in two main groups: the first group concerns the design and maintenance of biological data banks while the second one is related to the design of algorithms and systems for data manipulation and knowledge discovery. As matter of fact, these two strands of research cross-fertilize each other, as also exemplified in Figure 1.1. In particular, on the one hand, software and algorithms are developed by exploiting biological data banks, from which biological data useful for their evaluation are extracted. On the other hand, by exploiting bioinformatics systems, new information is inferred and, possibly, used to enrich available data banks.

In this general context, this thesis focuses in particular on the design and implementation of new algorithmic and software solutions to address relevant bioinformatics problems, such as *protein function prediction*, *network alignment* and *network querying*.

In the last few years, biological data banks were populated with a very large amount of data produced by research in *Systems Biology*. These data convey infor-

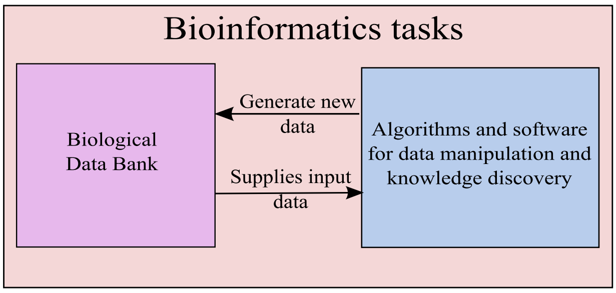


Fig. 1.1. Relations between bioinformatics tasks.

mation about single macromolecules such as proteins and genes which can be seen as the cell building blocks, as well as the interactions among such macromolecules. Starting from these interaction data it is possible to build more complex bioinformatics structures as shown in Figure 1.2. For instance, interactions among proteins are exploited to build *protein-protein interaction networks*, whereas biochemical reactions involving enzymes and metabolites are used to build *metabolic networks*.

To properly look up the large amount of biological data, available in the plethora of biological data, banks and mine useful information, the design and development of automatic tools has become crucial.

At the beginning, the interest of researchers was focused merely on tools to mine bio-sequences. In fact, several efforts have been paid for genome sequencing and designing procedures to compare biological sequences to search for similar regions. In this respect, notable examples are the Needleman and Wunsch algorithm [146] for global sequence alignment, and the Smith and Waterman algorithm [189] for local sequence alignment. These basic tools, then, evolved giving birth to very popular sequence alignment tools, such as FASTA [160] and BLAST [202]. At the same

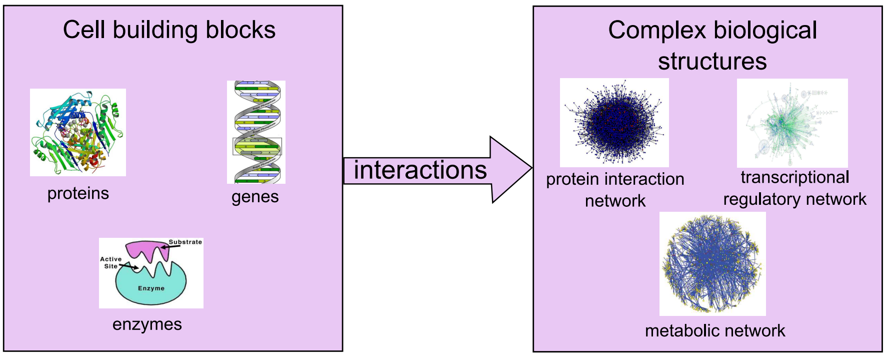


Fig. 1.2. Simple and complex structures exploited in bioinformatics.

time, algorithms for motif search[1] and identification of coding regions in genomic sequences were also developed.

More recently, the study of proteins, protein relations and macromolecules complex structures has gained momentum. In particular, by looking at proteins as independent macromolecules, a relevant task has become the prediction of protein functions, with the aim of properly understanding the role of uncharacterized proteins within living cells. However, the observation that proteins, and macromolecules in general, can be better characterized by analyzing their interaction patterns has given birth to the definition of a formal model, grounded on the graph theory, to represent the set of molecular interactions of an organism referred to as *Biological Networks*. Hence, biological networks can be fed as input to graph-based techniques that would try to infer new information about cellular activity and evolutive processes of the species. Indeed, by comparing the biological networks of two different species the transfer of knowledge, from one species to another, is also possible by identifying similar regions in the two input networks.

The aim of this thesis is also that of investigating the applications and opportunities in this latter group of bioinformatics tasks and provide useful tools to overcome some of the relevant problems thereof.

1.1 Main Contributions

The goal of this thesis is to provide innovative software tools for knowledge discovery in bioinformatics concerning the analysis of both simple (i.e., proteins) and complex (i.e., protein-protein interaction networks) structures. In particular, some efforts have been paid to predict the functions of uncharacterized proteins and discover functional modules in protein-protein interaction networks. A comprehensive experimental evaluation is also provided to substantiate the effectiveness of the proposed approaches from a biological point of view.

1.2 Problem Description

Proteins are essential parts of organisms which participate in virtually every process within cells. Many proteins work as biochemical catalysators, also known as enzymes, that catalyze the reactions occurring in living organisms. Proteins can interact with other molecules to perform storage and transport functions. Moreover, these fundamental components provide mechanical support and shape to tissues and mechanical work as, for example, the muscular contraction. Finally, several proteins have an essential role in decoding cellular information. Therefore, understanding the functions performed by proteins within the cell is a key bioinformatics task.

The function of a protein is determined by its three-dimensional structure. The tools developed to face this task, providing information about the three-dimensional

[1] A motif is a nucleotide or amino-acid sequence pattern that is widespread and has, or is conjectured to have, a biological significance.

folding of a given protein, are also useful to understand the function performed by the latter.

The first part of this thesis is oriented to protein function prediction, accomplished by using two different methods:

- PQSC-FCNN: a protein quaternary structure prediction tool that leverages the number of polypeptidic chains within a given protein;
- BI-GRAPPIN: based on the analysis of protein-protein interactions and, in particular, on the idea that similar proteins have similar interaction profiles.

However, biological processes regulating the cell life cycle stem from complex interactions among cell constituents. Therefore, the behavior of the cell can be deeply understood if the analysis is not limited to a mere individual study of cell building blocks (e.g., proteins, genes) but also encompasses more complex structures (e.g., protein complexes). In this respect, recently, some automatic tools have been developed, which aim at mining new knowledge about cellular processes by exploiting interaction data. These tools exploit *Biological Networks* as a formal model to encode molecular interactions among cell building blocks.

In this context, there are several ways to compare biological networks, but *network alignment*, *network integration* and *network querying*, have surely to be regarded as the most significant ones [181]. In Figure 1.3 these concepts are summarized.

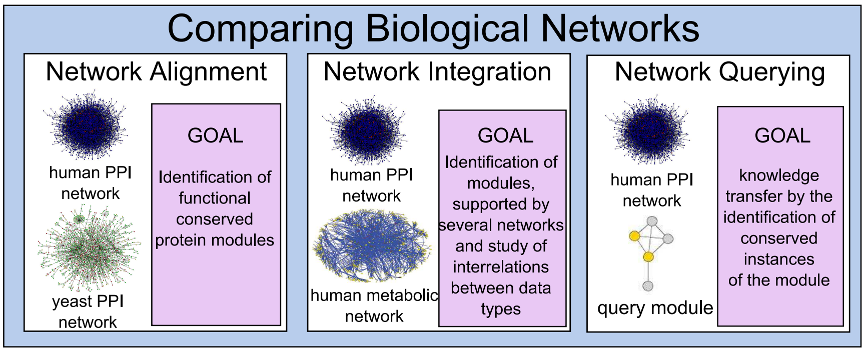

Comparing Biological Networks
Network Alignment
human PPI network
yeast PPI network
GOAL
Identification of functional conserved protein modules
Network Integration
human PPI network
human metabolic network
GOAL
Identification of modules, supported by several networks and study of interrelations between data types
Network Querying
human PPI network
query module
GOAL
knowledge transfer by the identification of conserved instances of the module

Fig. 1.3. The three main ways of comparing biological networks.

Network alignment is the process of globally comparing two or more networks of the same type belonging to different species, in order to identify similarity and dissimilarity regions. Network alignment is commonly applied to detect conserved sub-networks, which are likely to represent common functional modules [184].

Network integration is the process of combining several networks of the same species, representing different kinds of interactions (e.g., protein, metabolic), to study their interrelations. For instance, network integration techniques have been

used to predict protein interactions and identify protein modules [98, 236].
Finally, *network querying* techniques search a whole biological network to identify conserved occurrences of a given query module, which can be used for transferring biological knowledge from one species to another (or possibly within the same species). Indeed, since the query generally encodes a well-characterized functional module (e.g., the *MAPK cascade* in *yeast*), its occurrences in the queried network (e.g., the *MAPK cascade* in *human*) suggest that the latter (and then the corresponding organism) features the function encoded by the former.

The second part of this thesis concerns the comparative analysis of biological networks and, in particular, protein-protein interaction (or PPI) networks. More precisely, two tools have been developed, namely:

- Sub-Grappin: a tool to preform network alignment;
- PInG-Q: a tool to query PPI networks.

1.3 Outlook

This section provides the reader with an overview on the content of this thesis. Moreover, a chapter dependency schema is also sketched. This schema is intended to help the reader in following the path that motivated each individual chapter and understand how chapters are connected to one another.

1.3.1 Thesis' Structure

Part I comprises two chapters (i.e., chapters 1 and 2). Chapter 1 introduces and motivates the work presented in the other chapters. Moreover, a reader's guide is presented. Chapter 2 provides some background; in particular, some information is given both on simple (i.e., proteins) and complex (i.e., biological networks) biological strictures . Moreover, an overview on the motivations behind the study of both these structures (i.e., protein function prediction and biological network analysis) is provided. Overall, the aim of this chapter is to grip the reader's interest and create a well-founded motivation for the work done in later chapters.

Part II comprises three chapters (i.e., chapters 3, 4 and 5). Chapter 3 charts the state of the art in protein function prediction. This background is necessary to properly understand the motivations of the work presented in the two subsequent chapters (i.e., chapters 4 and 5). In particular, Chapter 4 is focused on the prediction of the quaternary structure of proteins, which characterizes the biological function of a protein when involved in specific biological processes. In Chapter 5, the tool Bi-Grappin, whose aim is to compute protein functional similarity across protein interaction networks of different organisms, is presented. This tool can be useful when comparing two networks, one of which is well-characterized while the other one is uncharacterized, to predict the unknown functions of proteins.

Part III comprises two chapters (i.e., chapters 6 and 7). In Chapter 6, the approaches presented in the literature for aligning two or more biological networks are described. This is useful to understand the advantages and disadvantages of the approach we developed for the same purpose, which is called Sub-Grappin and is discussed in Chapter 7.

Part IV consists of two chapters (i.e., chapters 8 and 9). In Chapter 8, a novel approach to querying protein interaction networks, called PInG-Q, is presented. The aim of Chapter 9 is that of analyzing and comparing tools devised to query biological networks, also considering the method presented in Chapter 8. This analysis is intended to help in understanding problems and research issues, state of the art and opportunities for researchers working in this area.

Finally, Part V sketches final conclusions and discusses future trends in the bioinformatics fields. Here, the contribution of the present thesis will be once more outlined w.r.t. the motivations and requirements identified at the beginning.

1.3.2 Reader's Guide

The present thesis has been written following a logical path interconnecting the various research contributions. However, it is possible to recognize two main threads. The first is related to bioinformatics simple structures (i.e., proteins) considered as single macromolecules and is discussed in Part II. The second part is related to Biological Network analysis and concerns Part III and IV.

As for the first thread, a reader interested in this specific problem can focus on Part II even if the content included in the introductory chapter and the second one have to be considered as compulsory premises to it. Part II has been logically divided in three sub-parts. The Chapter 3 gives some background necessary to understand the problem of protein function prediction and draws the state of the art in this area. This introductory chapter is a must to understand the subsequent two chapters (Chapter 4 and 5). In facts, Chapter 3 motivates the tools proposed in chapters 4 and 5.

As for the second thread (i.e., Biological Network analysis), a reader interested in this specific problem may only focus on parts III or IV; also in this case the introductory chapter and the second chapter become a must.

Part III has been logically divided in two sub-parts. Chapter 6 gives some background necessary to understand the problem of network alignment and draws an overview of the state of the art related to global and local alignment tools. This introductory chapter is a premise to the subsequent chapter (Chapter 7). In fact, Chapter 6 motivates the tool proposed in Chapter 7.

Part IV has been also logically divided in two sub-parts. In Chapter 8, a tool for querying protein interaction networks is presented and evaluated. In Chapter 9, this tool is compared with the state of the art. Moreover, Chapter 9 also provides a comparative overview of biological network querying systems, by exploiting an illustrative example.

Figure 1.4 summarizes chapters organization and provides links between the work presented in the different parts and chapters in order to allow the reader to choose the parts on which s/he is interested. In particular, two kinds of dependencies between parts and three between chapters are depicted. The relation *concludes* between Part I and Part V indicates that Chapter 10 analyzes the claims presented in Part I on the basis of the research discussed in the various parts. The relations *motivates* from a Part (or Chapter) to another Part (or Chapter) indicates that the content of the former provides the information necessary to motivate the contributions introduced in the latter. For example, the relations *motivates* from Part I to parts II, III and IV indicates that the content of Part I provides the information necessary to understand why the tools presented in Parts II, III and IV have been developed. The relation *exploits* indicates that the contribution introduced in Chapter 7 uses as a sub-procedure the method presented in Chapter 5. To correctly understand how the system discussed in Chapter 7 works, the reader is suggested to also read Chapter 5. The relation *analyzes and compares* indicates that the tool described in Chapter 8 is compared w.r.t. the state of the art in Chapter 9.

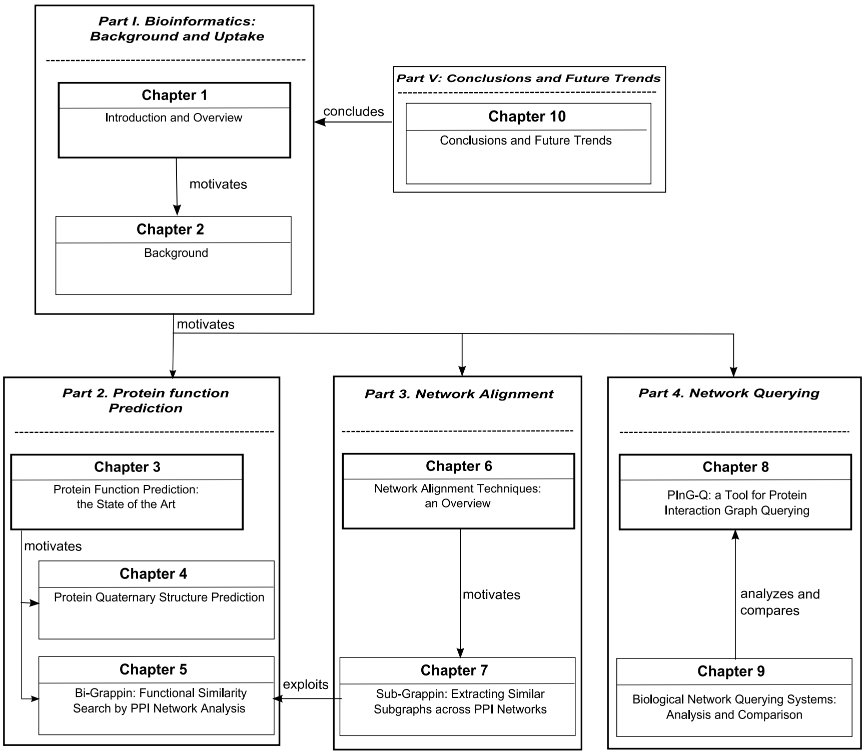

Part I. Bioinformatics: Background and Uptake
Chapter 1
Introduction and Overview
motivates
Chapter 2
Background
concludes
Part V: Conclusions and Future Trends
Chapter 10
Conclusions and Future Trends
motivates
Part 2. Protein function Prediction
Chapter 3
Protein Function Prediction: the State of the Art
motivates
Chapter 4
Protein Quaternary Structure Prediction
Chapter 5
Bi-Grappin: Functional Similarity Search by PPI Network Analysis
exploits
Part 3. Network Alignment
Chapter 6
Network Alignment Techniques: an Overview
motivates
Chapter 7
Sub-Grappin: Extracting Similar Subgraphs across PPI Networks
Part 4. Network Querying
Chapter 8
PInG-Q: a Tool for Protein Interaction Graph Querying
analyzes and compares
Chapter 9
Biological Network Querying Systems: Analysis and Comparison

Fig. 1.4. Structure of the thesis and chapter dependencies

1.3.3 Publications

Part of the material of the thesis has been published in some journals, conferences and books :

Journals

- V. Fionda, L. Palopoli, S. Panni, S. Rombo. “A technique to search for functional similarities in protein-protein interaction networks”. *International. Journal on Data Mining and Bioinformatics*. Vol. 3(4), pp. 431-453, 2009.

Book Chapters

- V. Fionda, L. Palopoli. “Network Querying Techniques for PPI network Comparison”. Chapter XVII. *In: Biological data mining in protein interaction networks (Xiao-Li Li, See-Liong Ng, Eds.),* IGI Publishing. ISBN:978-1605663982. pp. 312-334, 2009.

Conferences

- Valeria Fionda, Simona Panni, Luigi Palopoli and Simona E. Rombo. “sc Bi-GRAPPIN: Bipartite GRAph based Protein-Protein Interaction Networks similarity search”. *In Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM’07)*. Silicon Valley, USA, 2-4 November, pp. 355-361, 2007.
- Fabrizio Angiulli, Valeria Fionda and Simona E. Rombo. “Protein Data Condensation for Effective Quaternary Structure Classification”. *In Proceedings of International Conference on Intelligent Data Engineering and Automated Learning (IDEAL’07)*. Birmingham, UK, 16-19 December, pp. 810-820, 2007.
- Valeria Fionda, Simona Panni, Luigi Palopoli and Simona E. Rombo. “Singling out functional similarities in graph databases”. *In Proceedings of the Sixteenth Italian Symposium on Advanced Database Systems (SEBD 08)*. Mondello (PA), 22-25 June, pp. 271-278, 2008
- Valeria Fionda, Simona Panni, Luigi Palopoli and Simona E. Rombo. “Protein-protein interaction network querying by a ’focus and zoom’ approach”. *In Proceedings of the 2nd International Conference on Bioinformatics Research and Development (Bird’08)*. Vienna, 7-9 July, pp. 331-346, 2008.
- Valeria Fionda, Gialuigi Greco. “Charting the Tractability Frontier of Mixed Multi-Unit Combinatorial Auctions”. *In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI’09)*. Pasadena, CA, USA, 11-17 July. pp. 134-139, 2009.
- Valeria Fionda, Simona Panni, Luigi Palopoli and Simona E. Rombo. “Extracting similar sub-graphs across PPI Networks”. *In Proceedings of the 2nd International Conference on Bioinformatics Research and Development (ISCIS’09)*. Northen Cyprus, 1417 September. pp. 183-188, 2009.

2

Background

Summary. In this chapter, some background necessary to understand the topic and the motivation of this thesis is provided. In particular, across this chapter, some information will be given regarding both simple biological structures (i.e., proteins) (Section 2.1) and complex biological strictures (i.e., biological networks) (Section 2.2). Moreover, an overview of the motivation behind the study of both these types of structure (i.e., protein function prediction and biological network analysis) is provided in Section 2.3 and Section 2.4.

2.1 Background on Proteins

Genes are segments of DNA that code for proteins inside the cell. Transcription is the process by which the enzyme (that is a protein working as a biochemical catalysator) RNA polymerase, reads the sequence of bases on a gene and constructs an mRNA molecule from that sequence. Translation is the process by which a ribosome, a macromolecular assembly, reads the information contained in the mRNA molecule and synthesizes a protein molecule from the sequence on the mRNA molecule. Thus, each protein molecule is a product of the gene that codes for it. In turn, proteins are responsible for carrying out various functions inside the cell. For instance, many proteins work as enzymes that catalyze the reactions that occur in living organisms or they can interact with other molecules for performing storage and transport functions. Moreover, these fundamental components provide mechanical support and shape to tissues and mechanical work as, for example, muscular contraction. Finally, several proteins have an essential role in the decoding of cellular information and also regulate the transcription of a gene to an mRNA molecule.

Proteins are macromolecules composed by linear polymers, or chains, of amino acids. All organisms use the same set of 20 amino acids as building blocks in the protein synthesis. The variations of the order in which amino acids are connected and their total number let to obtain an almost unlimited number of proteins.

The primary structure of a protein is the sequence of its amino acids, forming the polypeptidic chain. The 20 amino acids are known as α-amino acids since they are composed by an amide group and a carboxylic group, bind to the C-2, also known as α carbon. The α carbon also binds hydrogen atoms and a side chain, called $-R$. The side chain is distinctive to each amino acid. The amino acids are bound to one another by the condensation of a α-carboxylic group of one amino acid to the amide group of another amino acid to form a chain. This bond is known as peptidic bond and the involved amino acids are called residues. The free amide and carboxylic groups at the opposite extremities of the peptidic chain are called N-terminal (amide terminal) and C-terminal (carboxylic terminal). Conventionally, all the residues of a peptidic chain are numbered starting from N-terminals.

On the basis of protein complexity, a protein can have at most four levels of structural organization (see Figure 2.1). The primary structure is the amino acid sequence and describes the one-dimensional structure of a protein. The other three levels encode the protein three-dimensional structure. In more detail, the polypeptidic chain patterns that regularly repeat into the protein denote the secondary structure. The tertiary structure is related to the three-dimensional structure of the whole polypeptide. The Quaternary Structure is related to the arrangement of two or more polypeptidic chains in one polymer.

Alterations of the conditions of the environment, or some chemical treatments, may lead to a destruction of the native conformation of proteins with the subsequent loosing of their biological activities. This process is called denaturation.

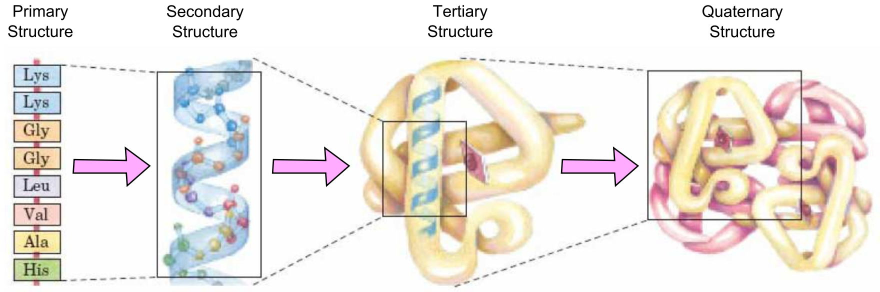


Fig. 2.1. Different levels of protein structures.

The central dogma of molecular biology was first enunciated by Francis Crick in 1958 [43] and re-stated in a paper appeared in the *Nature* journal published in 1970 [44]: “The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that information cannot be transferred back from protein to either protein or nucleic acid”. In other words, the central dogma of molecular biology is that genes may perpetuate themselves and work through their expression in form of proteins, but it is not possible to go the other way around and obtain the gene sequence from the protein. Note that the expression of a gene is its product, that is, the protein for which the gene encodes information.

The genetic information is encoded into the sequence of the bases of the DNA and perpetuate through the replication. More precisely, the genetic information is represented both by DNA and RNA. In fact, while cells use only DNA, some viruses, the *retroviruses*, have their genome encoded into the RNA, which is replicated into the infected cells.

The DNA uses four nucleotides: adenine (A), guanine (G), cytosine (C) e thymine (T). Since it is not possible to represent each of the 20 different amino acid by a nucleotide, each amino acid corresponds to a group of nucleotides. By choosing words composed by two nucleotides only $4^2 = 16$ combinations can be obtained. Instead, by choosing words composed by three nucleotides $4^3 = 64$ combinations can be obtained, that are sufficient to encode the 20 amino acids. Thus, a code of three or more nucleotides is necessary and the one made of three nucleotides seems to be valid for all organisms. Each triplet is called *codon*. All the 64 codons specify amino acids except three of them, that are stop triplets, and are stop signals in the transduction process. Since 61 codons are used to encode 20 amino acids, multiple triplets may encode for the same amino acid, and in general these have the same first two nucleotides and different third nucleotides. The starting triplet is the one encoding the methionine amino acid: all proteins start with this amino acid. The transduction process ends and the protein is released when one of the three stop triplets is recognized.

2.1.1 Protein Primary Structure

The primary structure of a protein is the linear sequence of its amino acids. The amino acid sequence of a protein is determined by the gene that encodes for it. The differences between two primary structures reflect the evolutive mutations. The amino acid sequences of related species are with high probability similar and the number of differences in their amino acid sequences are a measure of how far in the time the divergence between the two species is located: the more distant the species are the more different the protein amino acid sequences are.

The amino acid residues essential for a given protein to maintain its function are conserved during the evolution. On the contrary, the residues that are less important for a particular protein function can be substituted by other amino acids. It is important to note that some proteins have a higher number of substitutable amino acids than others, thus proteins can evolve at different speeds. Generally, the study of molecular evolution is focused on family of proteins. Proteins belonging to the same family are called homologous and the tracing of the evolution process starts from the identification of such families. Homologous are identified by using specialized amino acids sequence alignment algorithms that, by analyzing two or more sequences, search for their correspondences.

2.1.2 Protein Secondary Structure

The secondary structure is referred to the general three-dimensional form of local segments of proteins. It does not describe specific atomic positions in three-dimensional space, but is defined by patterns of hydrogen bonds between backbone amide and carboxylic groups. The secondary structure is related to the spacial arrangement of amino acid residues that are neighbors in the primary structure. The secondary structure is the repetition of four substructures that are: α helix, β sheet, β turn, Ω loop. The most common secondary structures are alpha helices and beta sheets (see Figure 2.2).

A common method for determining protein secondary structure is far-ultraviolet (far-UV, 170-250 nm) circular dichroism. A less common method is infrared spectroscopy, which detects differences in the bond oscillations of amide groups due to hydrogen-bonding. Finally, secondary-structure contents may be accurately estimated using the chemical shifts of an unassigned NMR spectrum.

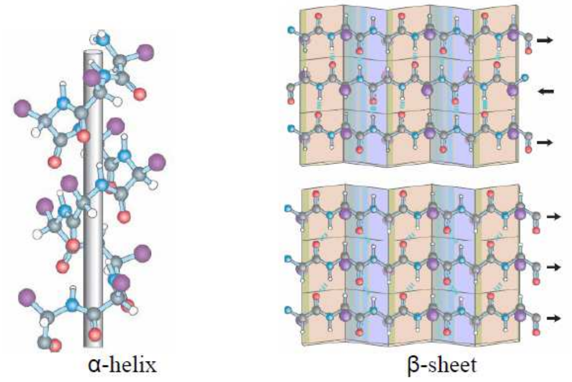


Fig. 2.2. Two examples of protein secondary structure: α helix and β sheet.

2.1.3 Protein Tertiary Structure

The tertiary structure of a protein is its three-dimensional structure, as defined by the atomic coordinates. The function of a protein is determined by its three-dimensional structure and the three-dimensional structure depends on the primary structure. Efforts to predict tertiary structure from the primary structure are generally known as protein structure prediction. However, the environment in which a protein is synthesized and allowed to fold are significant determinants of its final shape and are usually not directly taken into account by current prediction methods.

The biological activity of a protein is related to the conformation the protein assumes after the folding of the polypeptidic chain. The conformation of a molecule is a spacial arrangement that depends on the possibility for the bonds to spin. In physiologic conditions a protein has only one stable conformation, known as native conformation.

On the contrary of secondary structure, the tertiary structure also takes into account amino acids that are far in the polypeptidic sequence and belong to different secondary structures but interact with one another.

To date, the majority of known protein structures have been determined by the experimental technique of X-ray crystallography. A second common way of determining protein structures uses NMR, which provides somewhat lower-resolution data in general and is limited to relatively small proteins.

An example of tertiary structure as reported by the PDB database[1] is shown in Figure 2.3. This figure represents the tertiary structure of the *S-Adenosylmethionine Synthetase* with 8-BR-ADP.

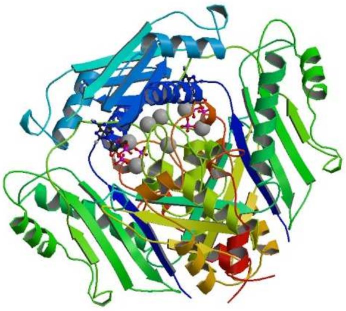


Fig. 2.3. An example of protein tertiary structure.

2.1.4 Protein Quaternary Structure

Many proteins are assemblies of more than one polypeptide chain, known as protein subunits. In addition to the tertiary structure of the subunits, multiple-subunit proteins possess a quaternary structure, which is the three-dimensional spacial arrangement of the several polypeptidic chains, corresponding to protein subunits.

According to this structure, the protein can be subdivided in two groups: homo-oligomers and hetero-oligomers. The first group is made of proteins composed by only one type of subunit, while the second one is made of proteins that are composed by different types of subunits. The proteins belonging to the first group are those having structural and supporting roles, while the proteins belonging to the second one have dynamic functions.

Protein quaternary structures can be determined using a variety of experimental techniques that require a sample of proteins in a variety of experimental conditions. The experiments often provide an estimate of the mass of the native protein and, together with knowledge of the masses and/or stoichiometry of the subunits, allow the quaternary structure to be predicted with a fixed accuracy. However, it is not always possible to obtain a precise determination of the subunit composition. The number of subunits in a protein complex can often be determined by measuring the hydrodynamic molecular volume or mass of the intact complex, which requires native solution conditions.

[1] http://www.rcsb.org/pdb/home/home.do

Table 2.1 reports the nomenclature used to identify protein quaternary structures. The number of subunits in an oligomeric complex are described using names that end in *-mer* (Greek for “part, subunit”).

Number of subunits	Name
1	monomer
2	dimer
3	trimer
4	tetramer
5	pentamer
6	hexamer
7	heptamer
8	octamer
9	nonamer
10	decamer
11	undecamer
12	dodecamer
13	tridecamer
14	tetradecamer
15	pentadecamer
16	hexadecamer
17	heptadecamer
18	octadecamer
19	nonadecamer
20	eicosamer

Table 2.1. The nomenclature used to identify protein quaternary structures

Figure 2.4 shows an example of the quaternary structure of a protein. The quaternary structure reported in the figure is a *tetramer* and is related to a potassium ion channel protein from *Streptomyces lividans*.

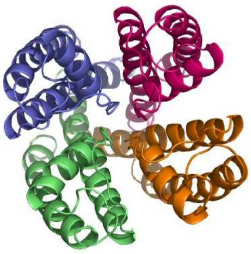
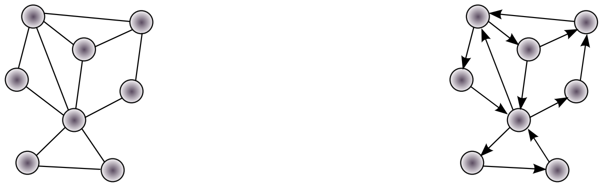


Fig. 2.4. An example of protein quaternary structure.

The quaternary structure is important, since it characterizes the biological function of proteins when involved in specific biological processes. Unfortunately, quaternary structures are not immediately deducible from protein amino acid sequences.

2.2 Biological Networks

Biological networks, which store information about molecular relations and interactions, can be conveniently represented as graphs. A graph is built from a set of nodes or vertices, representing cellular building blocks (e.g, proteins or genes), and a set of edges (directed or undirected), representing interactions (see Figure 2.5). A graph is a pair $G = (V, E)$, where V is the set of nodes and E is the set of edges, so that the elements from E are pairs of elements of V. In an undirected graph, an edge linking nodes A and B represents a mutual interaction. Conversely, in a directed graph, each edge represents the flow of material or information from a source node to a target node.



(a) An example of undirected graph (b) An example of directed graph

Fig. 2.5. Examples of graph structure.

Different types of graphs are used to represent different types of biological networks, each of which stores information about interactions related to specific entities or molecules [1]. Relevant kinds of networks include: *transcriptional regulatory networks*, *signal transduction networks*, *metabolic networks*, *protein-protein interaction networks* (or *PPI network*), *domain interaction networks*, *Gene Co-Expression Networks* and *genetic interaction networks*.

2.2.1 Transcriptional Regulatory Networks

As already pointed out in Section 2.1, the transcription of a gene to an mRNA molecule is regulated by proteins referred to as transcription factors. A transcription factor may activate or inhibit the expression of a gene inside the cell by binding to regions upstream or downstream of the gene on the DNA molecule. This process may, in turn, facilitate or prevent RNA polymerase from binding and initiating the transcription of the gene. Thus, the genes inside cells interact with each other via intermediate transcription factors to influence each others expression.

The set of genes interactions inside the cell is referred to as the *transcriptional regulatory network*. This kind of network can be modeled as a graph having two types of nodes, representing the transcriptional factors and the mRNAs of the target genes, respectively. Moreover, it has two types of directed edges, representing transcriptional regulation and translation, respectively. A simpler graph model combines genes with transcriptional factors they encode, to obtain a graph all the nodes of which represent genes. In this latter representation, an edge connects a source gene to a target gene if the former produces RNA or a protein acting as a transcriptional activator or inhibitor of the latter. An activator gene is the source of a positive regulatory connection, while an inhibitor gene is the source of a negative regulatory connection.

2.2.2 Signal Transduction Networks

Cells use signaling pathways and regulatory mechanisms to coordinate multiple functions. For instance, inside the cell, the proteins interact with each other to influence each other's activity. Moreover, extracellular signals are mediated to the inside of a cell by protein-protein interactions of signaling molecules. The *signal transduction networks* store information about the processes through which a cell converts one kind of signal or stimulus into another by protein-protein interactions. In particular, the signal transduction corresponds to the propagation of molecular or physical signals (for example, sensory stimuli) from a cell's exterior to its intracellular response mechanisms.

In the graphs modeling signal transduction networks, vertices represent proteins and directed edges represent the protein-protein interactions that work as signal converters.

2.2.3 Metabolic Networks

Metabolic networks represent the set of biochemical reactions that are responsible for the uptake of nutrients from the external environment and their conversion into other molecules required for the growth and maintenance of the cell. Each reaction takes in input some metabolites and produces as output other metabolites. Moreover, metabolic reactions are catalyzed by enzymes.

Metabolic networks can be represented as weighted tripartite graphs with three types of nodes (i.e., metabolites, reactions and enzymes) and two types of edges representing mass flow and catalytic regulation, respectively. The first type of edge connects reactants to reactions and reactions to products. The second type connects enzymes to the reactions they catalyze.

Simpler graph models have also been proposed. In particular, metabolic networks can be represented as bipartite graphs consisting of two types of nodes, which are metabolites and reactions. Each reaction node has an incoming edge from each reactant metabolite and one outgoing edge to each product metabolite. In the bipartite metabolic graph, there are no direct links between either two metabolites or two reactions. Another bipartite graph representation considers as the two partitions of nodes the chemical compounds and the enzymes, respectively. For each enzyme node, an incoming edge occurs with each of its substrate nodes and an outgoing edge occurs with each of its product nodes.

The metabolic networks sometimes are also represented as unipartite graphs (which could be directed or undirected) in which there is only one type of node. For instance, a simple model is a directed graph in which nodes represent enzymes and directed edges connect pairs of enzymes for which the product of the source enzyme is a substrate of the sink enzyme. In another simple model, nodes represent metabolites and directed edges represent enzymes that catalyze a reaction having the source metabolite as the reactant and the sink metabolite as the product.

2.2.4 Protein-Protein Interaction Networks

A protein-protein interaction network stores the information about the interactome of a given organisms, that is the whole set of its protein-protein interactions. In graphs modeling *protein-protein interaction (PPI) networks* , the nodes represent proteins and the edges are undirected and possibly weighted, with two proteins connected if they bind. Edge weight may be used to incorporate reliability information concerning the interaction.

Since protein-protein interactions are very important in regulating cell life cycle, there are a multitude of methods to detect them. Each of these method has its own strengths and weaknesses, especially with regard to the sensitivity and specificity. A high sensitivity means that many real interactions are detected. A high specificity indicates that most of the interactions detected are also occurring in reality. Thus, the reliability weights are important to take into account reliability, in terms of sensitivity and specificity, of the method used to detect interactions.

It is important to note that, since protein interactions are often obtained from protein complex detection and not really as binary interactions, a more complex model may be more informative. In fact, the use of hyper-graphs, instead of simple graphs, might be usefully adopted to model protein complexes.

2.2.5 Domain Interaction Networks

Domains are independently folded modules of a protein. A *domain-domain interaction (DDI) network* is constructed when each protein in a PPI network is replaced by one or more nodes representing its constituent domains. In this type of network, edges connecting two proteins are transformed to connect the corresponding domain nodes. Since most of the known proteins are composed by more than one domain, a domain-domain interaction network usually gets much larger than the original protein-protein interaction network. However, different proteins (often functionally unrelated) frequently share identical domains and, therefore, one domain node in a DDI network usually appears multiple times in the context of different proteins.

A similar type of network is the domain co-occurrence network, in which each domain is represented by a single node. In this type of network two nodes are connected by an edge when the corresponding domains occur in the same protein at least once.

2.2.6 Gene Co-Expression Networks

The *gene co-expression networks* store information about transcription that takes place at the same time or under the same conditions. In these networks, each gene corresponds to a node and edges connect genes that are co-expressed. These networks are constructed by large-scale DNA microarray experiments, and the unordered composition of a pair of co-expressed genes leads to the undirected nature of the networks. Starting from microarray gene expression data, the concordance of gene expression is measured with a Pearson correlation producing a Pearson correlation matrix. According to a first type of model, this matrix is dichotomized to arrive at an adjacency matrix. Binary values in the adjacency matrix correspond to an unweighted graph. Using this representation some genes are connected and all connections are equivalent.

A more complex model takes into account edge weights to store information about the absolute value of the Pearson correlation. In this type of representation all genes are connected and edge weights denote connection strengths between gene pairs.

2.2.7 Genetic Interaction Networks

Inactivation of most genes, in any organism, has little discernible effects on cell functioning under laboratory conditions. However, inactivating specific rare combinations of such non-essential genes can have profound effects on the organism under exactly the same conditions. In general, two genes are said to genetically interact if a mutation in one gene either suppresses or enhances the phenotype of a mutation in its partner gene. In the graphs modeling genetic interaction networks, nodes are genes and edges represent genetic interactions.

2.2.8 The Cell: a Network of Networks

It is important to underline that all the kinds of biological networks discussed above (e.g., metabolic, transcriptional regulatory or protein-protein interaction networks) are not independent of each other inside the cell. For instance, the state of the genes in the transcriptional regulatory network determines the activity of the metabolic network. On the other hand, the concentration of metabolites in the metabolic network determines the activity of transcription factors or proteins which regulate the expression of genes in the regulatory network. Thus, the biological networks together form a network of networks inside the cell that determines the overall behaviour of the corresponding organism.

2.2.9 Biological Network Modeling

On the more formal side, considering only unipartite graphs, a biological network N is commonly represented by a (possibly directed) graph $G^N = \langle V^N, E^N \rangle$ (see Figure 2.6). In this graph, the set of nodes (or vertices) V^N denotes a set of cell building blocks (e.g., proteins, enzymes, metabolites, genes) and the set of edges E^N encodes the interactions between pairs of nodes.
In the most general definition, each edge $e_{ij} \in E^N$ takes the form of a triplet $e^N_{ij} = \langle v_i, v_j, l_{i,j} \rangle$ where $v_i, v_j \in V^N$ are the interacting cell components and $l_{i,j}$ is the label associated to that edge (in PINs, for example, the edge label may encode the reliability of that interaction to actually occur).

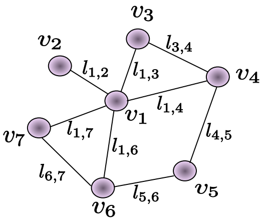


Fig. 2.6. An example of biological network graph G^N.

2.3 Protein Function Prediction

Proteins are essential parts of organisms and participate in virtually every process within cells. Many proteins work as biochemical catalysators, known also as enzymes, that catalyze the reactions occurring in living organisms. Proteins can also interact with other molecules to perform storage and transport functions. Moreover, these fundamental components provide mechanical support and shape to tissues and mechanical work as, for example, the muscular contraction. Finally, several proteins have an essential role in decoding cellular information. Therefore, understanding the functions performed by proteins within the cell is a key issue in bioinformatics.

Recently, a large amount of protein sequences has been made available as a result of whole genome sequencing project of many organisms. However, it is almost impossible to reveal their potential functions by experimental methods only. Moreover, there is a fast increasing in the number of proteins whose structures are known but whose functions are not. Seeing that experimental methods alone are not sufficient, a great attention is given to the computational approaches in which plenty of protein functions can be predicted simultaneously with reasonable accuracy. Therefore, computational protein function prediction methods prove themselves a powerful tools for biological research.

Protein function prediction methods can be basically classified according to information sources:

- Sequence-based approaches, that are the most basic methods. They exploit sequence alignment, sequence motif and domain information;
- Structure-based approaches, that make use of structural information. They compare whole three-dimensional shapes;
- Protein-protein interaction-based approaches. There are several different methods such as global mapping of unknown proteins or evidence integration in PPI networks.

The fundamental idea of sequence-based protein function prediction is the detection of similar protein sequences by database searching, assuming that similar sequences might have similar functions. For this purpose, several alignment algorithms, such as BLAST [202], can be used. But it is important to note that: *(a)* on the one hand, similar sequences not always have similar function and *(b)* on the other hand, dissimilar sequences have similar function at times. Thus, sequence space do not correspond with function space.

Structure-based protein function prediction uses structure information and is similar to sequence-based prediction. The basic assumption is that proteins with similar structure might have similar function. Protein function is strongly related with its structure since a protein works by interacting with other proteins or chemicals and its structure limits the possibility of its interaction modes. Moreover, structure similarity could fill the gap that is overlooked with sequence-based methods. In fact, low sequence similar proteins may have a significant structural similarity.

Protein-protein interaction (PPI) information have determined protein physical interaction maps for several organisms. These physical interactions are complemented by the other types of information discussed in Section 2.2 and shared evolutionary history. The protein-protein interaction data can be used to predict protein function by the observation that if protein p and protein p' interact, they are functionally close to each another. Moreover, similar proteins have similar interacting patterns. Thus, if p and p' interact with p_1 and p'_1, and p_1 and p'_1 are similar, it is possible to infer that also p and p' are functional related.

2.4 Biological Network Analysis

Cell behavior and function cannot be deeply understood through a mere analysis of its individual *building blocks* (e.g., proteins, genes). In fact, biological processes regulating cell life cycle stem from complex interactions among cell constituents. In the last few years, several techniques have been developed to discover such interactions and the amount of data made available in several databases (e.g., DIP [175], MINT [33], KEGG [94]) has grown steadily. These datasets promise new and exciting insights into the molecular machinery underlying biological systems. However, their analysis is fraught with a range of mathematical and statistical problems. This is particularly true for protein-protein interaction datasets, which suffer from being incomplete and subject to high error rates (both false positive and false negative). However, to properly look up the large amount of available data and mine useful information, the design and development of automatic tools has become crucial. These tools leverage Biological Networks as a formal model to encode molecular interactions among cell building blocks. As already pointed out in Section 2.2, at their most basic abstraction level, biological networks can be represented as graphs, where groups of connected biomolecules (corresponding to nodes of the graph) "collaborate" to form relatively isolated biological functional unit (corresponding to subgraphs). Biological graphs can be fed as input to suitable graph-based techniques able to perform topological and functional comparisons. Such techniques exploit specialized algorithms to infer new information about cellular activity and evolutive processes of the species, which allows to gain better understanding about the mechanisms underlying life processes [237].

A wide range of statistical and computational methods for the structural, functional and comparative analysis of biological networks have been developed. In particular, there are several ways to compare biological networks, but *network alignment*, *network integration* and *network querying*, have surely to be regarded as the most significant ones [181]. Figure 2.7 summarizes the goal of each of these tasks.
Network alignment is the process of globally comparing two or more networks of the same type belonging to different species in order to identify similarity and dissimilarity regions. Network alignment is commonly applied to detect conserved subnetworks, which are likely to represent common functional modules. As can be seen in Figure 2.7, the input of a network alignment algorithm are two (or, possibly more) biological networks of different organisms and the output are pairs (or, possible sets) of subgraphs (or, possibly simpler structures, such as paths), one for each input network, that have been recognized to be similar. For instance, the identification of conserved linear paths may lead to the discovery of signaling pathways, as well as conserved clusters of interactions (subgraphs) may correspond to protein complexes.
Network integration is the process of combining several networks of the same species, representing different kinds of interactions (e.g., protein, metabolic), to study their interrelations. Since each type of network lends insight into a different slice of biological information, integrating different network types may paint a more comprehensive picture of the overall biological system under study. Commonly, networks to be integrated are defined over the same set of elements (e.g., the set of proteins of a certain species), and the integration is achieved by merging them into a single network with multiple types of interactions, each drawn from one of the original networks. As shown in Figure 2.7, the input of a network integration algorithm are two (or, possibly more) biological networks defined over the same set of elements (corresponding to graph nodes) that store different types of information (painted in green for the first input network and in red for the second one). The output is a new network, defined over the same set of elements, that integrates all types of input interactions. In particular, in the figure, the interactions belonging to only one of the input networks are reported with the same color used in the corresponding network (green or red), while the interactions stored in both networks are painted in black. A fundamental problem is to identify, in the merged network, functional modules that are supported by interactions of multiple types (for instance, the cluster of nodes $\{n_1, n_2, n_4, n_5, n_6\}$ in Figure 2.7).

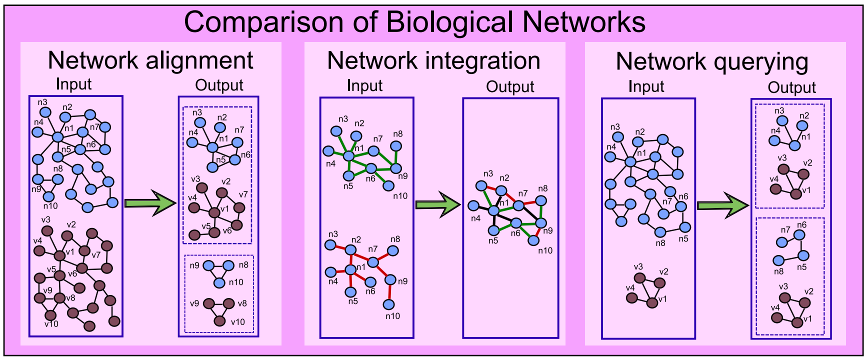


Fig. 2.7. Comparing biological networks: the three main ways.

Finally, *network querying* techniques search a whole biological network to identify conserved occurrences of a given query module, which can be used for transferring biological knowledge from one species to another (or possibly within the same species). Indeed, since the query generally encodes a well-characterized functional module (e.g., the MAPK cascade in yeast), its occurrences in the queried network (e.g., the MAPK cascade in human) suggest that the latter (and then the corresponding organism) features the function encoded by the former. As shown in Figure 2.7, the input of a network querying algorithm are a whole biological network (painted in blue) and a query module (colored in violet) of the same type (for instance, both reporting protein-protein interaction information). The output are all the (possibly approximated) occurrences of the query module into the target network.

2.5 Concluding Remarks

In this chapter some biological and bioinformatics background knowledge, useful to understand the subsequent chapters, has been given. The subsequent parts of this thesis will illustrate the state of the art, and several innovative contribution in *protein function prediction* (Part II), *network alignment* (Part III) and *network querying* (Part IV).

Part II

Protein Function Prediction

3

Protein Function Prediction: the State of the Art

Summary. In this chapter, the state of the art about protein function prediction will be outlined. Firstly, in Section 3.1, the notion of "protein function" is discussed. Then, in Section 3.2 an overview of the different methods proposed in the literature to predict protein function is provided. Moreover, in the subsequent sections, two strands of research will be deepened: quaternary structure prediction (Section 3.3) and protein function prediction by PPI networks analysis (Section 3.4).

3.1 Protein Function

The concept of protein function is not very well-defined. In fact, this concept typically includes all the types of activities that a protein is involved in, from molecular to physiological ones. Some categorizations of the types of functions a protein can perform have been proposed in the literature [23, 7]. The first categorization [23] distinguishes among:

- Molecular function: the biochemical function performed by a protein, such as ligand binding, catalysis of biochemical reactions and conformational changes;
- Cellular function: the function performed when many proteins come together to perform complex physiological functions, such as operation of metabolic pathways and signal transduction, to keep the various components of the organism working well;
- Phenotypic function: the integration of the physiological subsystems, consisting of various proteins performing their cellular functions, and the interaction of this integrated system with environmental stimuli.

Clearly, these three categories are not independent. In fact, the molecular function category is a sub-category of the cellular function category, which is, in its turn, a sub-category of phenotypic function.
A widely used categorization is the Gene Ontology classification scheme [7], which categorizes protein functions into:

- Cellular component: referred to the parts of a cell or its extracellular environment where the protein is localized;
- Molecular function: the elemental activities of a protein at the molecular level, such as binding or catalysis;
- Biological process: operations or sets of molecular events with a defined beginning and end, pertinent to the functioning of integrated living units: cells, tissues, organs, and organisms.

Often, the protein function prediction is referred to one or more of these categories. In the sense that functional annotation of such categories are transferred from characterized to uncharacterized proteins.

3.2 Protein Function Prediction

There are plenty of proteins which have a totally unknown function. For some of these proteins only the amino acid sequences are known, while for others also protein structures have been provided by the structural genomics centers. Since proteins participate in virtually every process within cells, understanding the functions they perform therein is a key bioinformatics task. For this reason, several tools have been developed to infer protein function.
Among the existing software tools, some main strategies can be distinguished:

- homology search and transfer of annotations:
  - sequence alignment
  - structure alignment
- function inference by genomic context
  - genomic sequences
  - gene expression data
- phylogenomic approaches
- protein interaction networks

In the following paragraphs such strategies will be discussed in more detail.

3.2.1 Homology Search and Transfer of Annotations

The most basic strand of approaches proposed for predicting protein function is based on homology search. These methods try to infer the unknown function of a protein by finding a protein, with a known function, having either a similar sequence or a similar structure.

Sequence homology is the classical methodology used to infer the function of a novel protein. Indeed, sequence homology has been proved to be effective and reliable for inferring protein function, although its applicability is limited to protein for which substantial sequence similarity to annotated proteins can be found. In fact, in a study involving over a million sequence alignments [173], it was shown that alignments with at least 30% sequence identity correspond in the 90% of the cases to homologous proteins, while alignments whit the 25% of sequence identity or less identify homologous proteins only in the 10% of the cases. Hence, the coverage of methods that utilize sequence alignments may be limited to relevant sequence identity percentages to maintain reasonably low false positive rate.

Another methodology for protein function prediction is based on the observation that, in many biological processes, the interacting entities have to come into physical contact in order to accomplish the desired task. Starting from this observation a connection between structure and function can be detected, since the structure of a protein determines several of its functional features (as already pointed out in Chapter 2). Thus, proteins having similar structures have, with high probability, also similar function. The prediction methods that are based on this observation exploit the structural alignment of protein. Such alignments attempt to establish equivalences between two or more polymer structures based on their shape and three-dimensional conformation.

Sequence Alignment

As already pointed out in the previous section, sequence homology is the classical methodology used to infer the function of a novel protein.

The simplest way to discover sequence homology is to use an alignment software such as the Basic Local Alignment Search Tool (BLAST) [5], PSI-BLAST [4] or FASTA [161] to find possible homologs of a given protein in sequence databases. However, as already underlined in the previous section, simple transfer of function annotations from proteins having similar sequences may not produce very accurate results, due to the weak correlation between the sequence and the function of proteins.

This section discusses several approaches that have been proposed to improve sequence homology based techniques by exploiting several additional information. For instance, numerous approaches use standardized annotation schemes, such as the Gene Ontology. The use of GO annotations make the process of transferring functional annotations organism-independent, since it is based on a hierarchically-structured functional ontology. Several methods, such as Onto-Blast [235], GOblet [83] or GOtcha [134], that firstly align protein sequences and then filter the alignment result exploiting statistical and machine learning techniques have been proposed.

Another direction in which homology-based function transfer can be improved is by making the process probabilistic. This goal can be achieved, for example, by assuming that a protein can only belong to a functional class if its BLAST score distribution with the members of the class is the same as that of its members themselves [122].

Another family of approaches tackles subsequence analysis. The observation is that often only specific parts of the whole sequence are crucial for the protein to perform its function. Starting from this observation, some approaches try to identify useful portions of the protein sequence that may determine its function. However, the meaning of “useful portions” is ambiguous even if two main definitions are the most common:

- Motifs: that are subsequences conserved across a set of protein sequences belonging to the same family. These subsequences are candidates for functional sites in proteins, such as sites for ligand binding, DNA binding and interactions with other proteins. Thus, motifs can be usefully exploited for predicting the function of a protein.
- Domains: that are parts of protein sequences that can evolve, function, and exist independently of the rest of the protein chains. Each domain forms a compact three-dimensional structure and often can be independently stable and folded. Many proteins consist of several structural domains. The function of a protein is a combination of the functions of each of its domains.

The above definitions indicate that identifying domains and motifs can be useful for predicting protein function. As mentioned earlier, these subsequences provide a new way of encoding the protein sequence in terms of features, which encode whether a certain motif or domain is present in a sequence. Such feature representation can be modeled by a feature vector that must be calculated for each protein in the target set. Then, various statistics and data mining techniques, such as classification, could be used in the prediction process. Many approaches based on this idea have been proposed in the literature, which exploit both motif [80, 215, 127, 218, 22] or domains [177, 30, 163].

Unfortunately, the approaches in this category do not obtain notable results. One reason is the lack of an unambiguous definition of subsequences. Indeed, each of the above mentioned approaches models the subsequence patterns in a different manner. In addition, the programs used to extract these sequence patterns are approximated, and hence, add a source of error to the prediction process.

The third family of approaches for protein function prediction, which exploit sequence information are that based on features. The basic idea is to transform protein sequences into more biologically meaningful features, which make the distinction between proteins from different functional classes easier. Some examples of types of features that can be extracted from sequences are:

- Sequence based attributes: such as the number of residues of the different types, the length of the sequence, the molecular weight, normalized Van der Waals volume, polarity or n-grams.
- Phylogeny based attributes: computed for instance through the results of a PSI-BLAST search.
- Structure based attributes: such as secondary structure attributes.

Feature-based approaches use standard classification algorithms to learn models of functional classes from the set of features, and then utilize this model to make predictions for uncharacterized proteins [224, 223, 103, 104, 157, 90, 91, 29, 57, 35]. The most commonly used classifiers in this class of approaches are support vector machines (SVM) [29], neural networks (NN) [224, 223, 157] and the naive Bayesian classifier [35].

Overall, it is clear from the above discussion that feature-based approaches are better able to handle the function prediction task than homology or subsequence-based approaches: this is because of the inclusion of more biologically meaningful features. This enables the construction of a more robust model for the sequence-function mapping.

Concluding, techniques that predict protein function from sequence can be categorized into three classes, namely:

- Homology-based approaches: are those approaches based on the alignment of protein sequences and the discovery of significant sequence homology. These approaches are not always accurate and several efforts have been done to make the search more accurate by exploiting probabilistic approaches or leveraging other information (e.g., GO annotations).
- Subsequence-based approaches: often not the whole sequence, but only some segments of it (corresponding to motifs or domains) are important for determining the function of a given protein. Hence, the approaches in this category treat these segments or subsequences as features of a protein and map these features to protein function.
- Feature-based approaches: extract from the amino acid sequence some features related to several physical and functional protein characteristics. These features are used to construct a predictive model, which can map the feature-value vector of a query protein into its function.

Analyzing the above categorization, it is quite clear that the subsequence and feature-based approaches are very similar at the basic level, since they involve the construction of a model for the feature-to-function mapping. However, there are also significant differences between them. The most fundamental difference is that while subsequence-based approaches extract the features (i.e., meaningful subsequences) from a set of functionally related sequences, feature-based approaches derive and evaluate their features on the basis of individual protein sequences.

Structure Alignment

Sequence is only one aspect that has an influence on the function of a protein. In fact, to be able to perform their biological function, proteins fold into one, or more, specific spatial conformations, driven by a number of non covalent interactions such as hydrogen bonding, ionic interactions, Van Der Waals forces and hydrophobic packing. The functional behavior of a protein may hence be better understood by also looking at its structure.

Some approaches that analyze the secondary [217, 66] and tertiary structures [158, 117, 118] of proteins have been proposed in the literature. Tertiary structures reflect the physical characters of translated proteins, and offer clues to the actual mechanism of protein function. However, tertiary structures are derived using relatively costly and time-consuming experimental techniques such as X-ray crystallography and Protein nuclear magnetic resonance spectroscopy (NMR). The number of known tertiary structures is small as compared to the number of protein sequences known. Moreover, tertiary structures cannot be always reliably predicted from protein sequences, especially when appropriate template structures for homology modeling are not available. Secondary structures, on the other hand, can be effectively predicted from sequences and used to complement sequence homology for function prediction.

In several works it has been proved that the structure of a protein is not tightly correlated directly with its biological function, but it is correlated with lower-level functional features. Thus, functional features might be used for predicting the function of a protein from its structure [133, 82, 150, 203].

Leveraging these studies, some ideas for inferring functional features from the structure of a protein have been proposed [141, 188, 68, 143, 220], therefor, protein structure can be used to predict protein function. Indeed, several researchers have proposed various structural features and approaches for function prediction, which can be classified into the following four categories:

- Similarity-based approaches: are those approaches [158, 85, 180] that, given the structure of a protein, identify the protein with the most similar structure by using structural alignment techniques [110, 113], and transfer its functional annotations to the query protein.
- Motif-based approaches: attempt to identify three-dimensional motifs, that are substructures conserved in a set of functionally related proteins (e.g., the helix-turn-helix (HTH) motif [129]), and estimate a mapping between the function of a protein and the structural motifs it contains. This mapping is then used to predict the functions of unannotated proteins. However, note that structural motif finding programs, (e.g., TESS [213], FFF [69] and SPASM [107]) rely on their own definitions of a structural motif, since there does not exist a universally accepted definition of this concept.
- Surface-based approaches: these approaches do not consider the structure of a protein with respect to the distances between consecutive amino acids, but represent it by a continuous surface. This representation helps in identifying features such as voids or holes in the surface. The idea here is that interactions between proteins occur due to the complementarity of their molecular surfaces. The approaches in this category utilize these features to infer the function of proteins [105, 21, 65, 63].
- Learning-based approaches: this category employs effective classification methods, such as SVM and k-nearest neighbor, to identify the most appropriate functional class for a protein from its most relevant structural features [102, 49, 214, 13].

3.2.2 Function Inference by Genomic Context

In the context of exploiting genomic information for protein function prediction two strands of research can be recognized. The first strand concerns the analysis and alignment of genomic sequences while the second one makes use of gene expression data. In the two subsequent paragraphs these two strands of research will be discussed.

Genomic Sequences

This section discusses some approaches exploiting ideas which stem from the genome resource for function prediction. In this domain, most of the studies fall in the field of comparative genomics [131] and, thus, the applications are oriented to functional associations between genes or proteins rather than annotations for individual proteins. Also, it must be remarked that the approaches in this category are often justified by evolutionary mechanisms. The approaches proposed to derive functional associations from genomic data, and possible function prediction, can be divided in three categories [131]:

- Genome-wide homology-based annotation transfer: the most immediate impact of large-scale genome sequencing projects has been the wider application of existing sequence-homology based approaches [4] for functional annotation transfer. The availability of complete genomes of many organisms led to the creation of databases of gene sequences [19] and the database of Clusters of Orthologous Genes (COGs) [200]. The approaches in this category use existing databases for searching for homologous of the query protein, with the aim of transferring functional annotations from the closest results.
- Gene neighborhood-based or gene order-based approaches: these approaches are based on the hypothesis that proteins, whose corresponding genes are close to each other in multiple genomes, are expected to functionally interact [45, 152, 153, 111, 109, 123].
- Gene fusion-based approaches: these approaches attempt to discover pairs or sets of genes in one genome that are merged to form a single gene in another genome [132, 229, 60, 130]. Here, the underlying hypothesis is that these sets of genes are functionally related.

As can be seen, approaches in the latter two categories exploit genomic context, i.e. the location of a gene on the genome [Huynen et al. 2000].

Gene Expression Data

Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product (i.e., a protein or a functional RNA). Gene expression experiments try to quantitatively measure the transcription phase of protein synthesis and are obtained by microarray[1]. The primary advantage of gene expression experiments is that they offer an effective method for observing the simultaneous activity of thousands of genes under a given experimental condition. Thus, gene expression data holds great promise for determining the function and functional associations of proteins. Several repositories have been set up in order to make gene expression data publicly accessible [16, 156, 12].

[1] A 2D array on a solid substrate (usually a glass slide or silicon thin-film cell) that assays large amounts of biological material using high-throughput screening methods.

Usually, the format of gene expression data is very simple. They are represented by a rectangular matrix, in which the rows correspond to genes, the columns to conditions, and the entries denote the expression measurement of a gene under a particular condition. Another important factor in microarray data analysis is that the data used in research is generally of two kinds: static and temporal. The first category consists of datasets containing snapshots of the expression of certain genes in different samples under the same conditions. The second one, also known as time-series gene expression data, consists of datasets capturing the expression of certain genes of the same organism at different instants of time.

Early approaches identified functional associations between genes by measuring the similarity between their expression profiles using statistical methods [212]. However, these studies usually required significant human intervention, thus more generic techniques from data mining were proposed. These techniques can be grouped into the following three categories:

- Clustering-based approaches: an underlying hypothesis of gene expression analysis is that functionally similar genes have similar expression profiles, since they are expected to be activated and repressed under the same conditions. Approaches in this category use unsupervised learning techniques, particularly clustering, to group together genes on the basis of their gene expression profiles, and assign functions to the unannotated proteins using the most dominant function for the respective clusters [56, 18, 148, 242, 226, 199, 241, 36, 230, 27, 128, 155].
- Classification-based approaches: the functions of some genes may be known and may act as class labels. Thus, a more direct solution to the problem of predicting protein function from gene expression profiles is the classification. The approaches in this category firstly build various types of models for the expression function mapping by using classifiers and, then, exploit these models to annotate new proteins [24, 135, 115, 147, 239].
- Temporal analysis-based approaches: temporal gene expression experiments measure the activity of genes at different instances of time (e.g., during a disease) and this information can be used to predict protein function. The approaches in this category derive features from this temporal data and use classification techniques to predict the functions of unannotated proteins [15, 61, 81, 28, 88, 116, 8, 139, 208].

3.2.3 Phylogenomic Approaches

The biological species existing today have evolved from primitive forms of life over millions of years, and this process of evolution continues today. The changes in the physiologies of different organisms have been driven by the changes at the cellular level, which include the adoption and surrender of functions by proteins due to changes in the genes encoding them. Thus, it is essential to include the evolutionary perspective in any complete understanding of protein function.

As a result, several approaches for predicting protein function using evolution-based data have recently been proposed. The two most common forms of this data are known as phylogenetic profiles and phylogenetic trees, and the field of biology that deals with the evolutionary relationships among living organisms is also known as *phylogenetics*.

The phylogenetic profile of a protein is (generally) a binary vector whose length is the number of available genomes. The vector contains the value 1 in the i^{th} position if the i^{th} genome contains a homologue of the corresponding gene, and 0 otherwise. Some variations of these vectors use real numbers that reflect the extent of similarity between the original gene and the best match in the genome being searched. Thus, these profiles provide a way of capturing the evolution of genes across various organisms. This information becomes useful for functional genomics when it is seen in the light of the phenomenon of *speciation*, which is the evolutionary mechanism by which new species are created from currently existing ones.

It may be hypothesized that proteins which interact functionally correspond to genes that are inherited across several genomes during speciation events. Phylogenetic profiles are a powerful mathematical way of modeling this phenomenon, and thus offer a very innovative method for inferring functional associations between proteins, since the latter are expected to have very similar phylogenetic profile. This is the basic assumption made by all the approaches for function prediction on the basis of phylogenetic profiles [162, 125, 225, 58, 46, 240].

In several other studies, a more extensive representation of evolutionary knowledge is used. This representation is known as a phylogenetic tree. The leaves of this tree correspond to organisms and the internal nodes denote the hypothetical last common ancestor (LCA) of all its descendents. The branches represent evolution relationships. Surely, phylogenetic trees embody a much richer source of knowledge than phylogenetic profiles since the latter are constructed only on the basis of the leaf nodes of the former, thus ignoring the hierarchical structure of the evolutionary knowledge. The additional knowledge provided by the internal tree nodes can be used to extract further information about the pattern of evolution of a set of proteins. Thus, phylogenetic trees, if accurately constructed, can provide strictly richer information than simple profiles. Still, both of these forms of phylogenetic data together constitute a very rich pool of knowledge about evolution that can be utilized effectively for the prediction of protein function.

The studies that try to uncover gene/protein functions and functional linkages using phylogenetic data such as profiles and trees can be classified into three categories:

- Approaches using Phylogenetic Profiles: these approaches are based on the hypothesis that proteins with similar phylogenetic profiles are functionally related [162, 125, 225, 58, 46, 240].
- Approaches using Phylogenetic Trees: this category embodies those approaches that exploit phylogenetic trees to predict function. Most of these approaches use various data mining and machine learning techniques and produce better results than those based only on profiles [55, 50, 159, 169, 187, 59].

- Hybrid Approaches: these approaches use SVM-based techniques to combine the two forms of evolutionary knowledge stored in phylogenetic profiles and trees [210, 145].

3.2.4 Protein Interaction Networks

Proteins do not work alone, but interacts with other biological entities such as DNA, RNA, as well as other proteins to perform their function. Hence, the function of a protein may be inferred by looking at its interaction neighborhood.
The approaches that attempt to predict function from a protein interaction networks can be broadly categorized into the following five categories:

- Neighborhood-based approaches: utilize the neighborhood of the query protein in the interaction network to predict its function [178, 84, 106, 176, 26, 126, 137]. For instance, a basic technique belonging to this category assigns to the query protein the most prevalent function among its interacting proteins.
- Global optimization-based approaches: consider the structure of the entire network and try to optimize an objective function based on the annotations of all the proteins in the network [119, 121, 209, 197, 95, 142].
- Clustering-based approaches: are based on the hypothesis that dense regions in the interaction network represent functional modules in which proteins perform the same function. Thus, the approaches in this category apply graph clustering algorithms to PPI networks and then transfer the functions of characterized proteins to unannotated proteins belonging to the same module [190, 52, 172].
- Association-based approaches: use several algorithms for finding frequently occurring sets of interactions (subgraphs). The identified subgraphs are supposed to denote functional modules in which the majority of proteins perform the same function [86, 227, 34]. The basic idea of these approaches is similar to that of the previous category. The difference is that in this case, patterns of interactions, instead of clusters of nodes, are searched for.
- Comparison of protein-protein interaction networks: predict protein function by comparing the protein-protein interaction networks of two or more organisms. This way, an uncharacterized protein of one network is annotated with the known function of a protein in another network, so that the two proteins have the most similar interaction patterns [14, 186].

3.3 Protein Quaternary Structure Prediction

Many proteins are composed of two or more subunits, each associated with different polypeptide chains. The number and the arrangement of subunits forming a protein are referred to as *quaternary structure*, as already pointed out in Chapter 2. The quaternary structure of a protein is important, since it characterizes the biological function of the protein when it is involved in specific biological processes. Unfortunately, quaternary structures are not trivially deducible from protein amino acid sequences and, thus, recently, some techniques have been proposed to provide protein quaternary structure classification [38, 76, 191, 233, 238]. Most of them aim at classifying homo-oligomeric proteins.

The first software that has been proposed to predict protein quaternary structure, called Quaternary Structure Explorer (QSE) [76], classify proteins in two distinct classes: homodimers and non-homodimers. This software is based on the analysis of protein amino acid sequences and use the *C4.5* classification algorithm. The evaluation has been performed exploiting a dataset made of 1639 homo-oligomeric proteins, extracted from SWISS-PROT [11], composed by 914 homodimers and 725 non-homodimers. According to this approach, each protein is represented by 401 amino acid indices obtained by the AAindex database [100]. An amino acid index is a list of 20 numerical values corresponding to physical, chemical, and biochemical properties of the 20 common amino acids. The overall precision obtained during the evaluation was 70%.

Another method proposed by Song and Tang [191] that considers only the two classes of homodimers and non-homodimers introduced a new measure called function of degree of disagreement (FDOD). The FDOD is a measure of information discrepancy computed to measure discrepancies among sequences and the set of subsequence distributions. The subsequence distribution is useful to take into account the effect of residue order on protein structure. The approach by Song and Tang exploited the FDOD in the classification process and the evaluation has been performed on the same dataset exploited to evaluate QSE [76]. During the evaluation both the resubstitution test and the 10-fold cross-validation test were performed with different subsequence lengths ranging from 1 to 4. This technique obtained an overall precision of 82.5%.

The last approach [238], which has been proposed to classify protein quaternary structures into homodimers versus non-homodimers classes exploits protein primary sequences and uses both Support Vector Machines (SVM) and the covariant discriminant algorithm. Each protein is represented by the amino acid composition and four autocorrelation functions. According to the classical definition, amino acid composition consists of 20 components, representing the occurrence frequency of each of the 20 native amino acids in a given protein. Since the amino acid composition alone doesn't take into account any sequence information, the authors exploited also four autocorrelation functions computed by exploiting the amino acid index profile of the primary sequence. The autocorrelation functions exploited are: *(i)* FASG^{a}: the auto-correlation functions of amino acid residue index of Fasman; *(ii)* NISK^{b}: the auto-correlation functions of amino acid residue index of NishikawaOoi; *(iii)* WOLS^{c}: the auto-correlation functions of amino acid residue index of Wold et al; *(iv)* KYTJ^{d}: the auto-correlation functions of amino acid residue index of KyteDolittle. This approach obtained a precision of 87.5% on the same dataset used by the two previously discussed approaches [76, 191].

The techniques illustrated above [76, 191, 238] are able to distinguish just between two classes, that are homodimers and non-homodimers. However, some approaches able to discriminate among a large variety of classes have been proposed in the literature.

In this respect, the first proposed approach [38] exploits the pseudo amino acid composition of proteins for representing each protein as a set of discrete numbers. The pseudo amino acid composition consists of 20 + λ discrete numbers, in which the first 20 numbers are the same as the 20 components in the classical amino acid composition, and the others represent λ sequence-order correlation factors. This representation is more powerful than the standard amino acid composition, since it is able to take into account a considerable amount of sequence-order and sequence-length effects. The dataset used in the evaluation was extracted from SWISS-PROT. In particular, the training set was made of 3174 homo-oligomeric protein sequences, among which 382 were annotated with monomer, 817 with dimer, 593 with trimer, 884 with tetramer, 54 with pentamer, 287 with hexamer, and 157 with octamer. The independent dataset consisted in 332 protein sequences, of which 50 were annotated with monomer, 102 with dimer, 56 with trimer, 80 with tetramer, 6 with pentamer, 28 with hexamer, and 10 with octamer. This approach reached an overall success rate of 80.1% on the independent set by performing resubstitution, jack-knife, and independent data set tests.

The four approaches described above [76, 191, 238, 38] exploit only protein sequence information. Another notable approach [233], instead, exploits the functional domain composition of proteins. According to this representation, each protein is represented as a binary vector in which the *i*-th position is equals to 1 if the protein contains the *i*-th domain. This representation, as shown in some studies [222, 101, 39, 30, 232], is able to deliver important information about protein structures and functions. The approach is based on the nearest neighbor algorithm and was evaluated performing a two stages evaluation. The first stage was the jackknife cross-validation test on a non-redundant dataset of 717 proteins represented by 540 PFam domains. The second stage exploit the non-redundant dataset to classify an independent dataset of 9,951 proteins defined on the same set of 540 domains. This approach obtained an overall success rate of 75.17% in the first evaluation stage and 84.11% in the second one.

3.4 Protein Function Prediction by PPI networks analysis

In this section an overview of the methods proposed to predict protein function by comparing protein-protein interaction networks is provided. Three approaches have recently been proposed to address this issue [14, 184, 185]. In the following each of these approaches will be discussed in detail.

The first approach that relies on the comparison of PPI networks [14] is based on a strategy to identify functionally related proteins in two protein-protein interaction networks. This approach exploits both sequence-based protein comparisons and conserved protein-protein interactions across the two input networks. This approach works in two stages. In the first stage the two PPI networks are aligned using only protein sequence similarities and, in particular, by assigning proteins to sequence homology clusters using the *Inparanoid* algorithm [171]. In the second stage, pairs of proteins, one from each species, that are likely to retain the same function, are identified by performing probabilistic inference. In particular, the orthology relation between each pair of possibly corresponding proteins was modeled as a probabilistic function of the orthology relations among their immediate network neighbors. Note that, orthology relationships were inferred by using Gibbs sampling. In the evaluation phase, this approach has been used to resolve ambiguous functional orthology relationships between the *S. cerevisiae* and *D. melanogaster* PPI networks. In particular, 121 cases, for which functional orthology assignment was ambiguous when sequence similarity is used alone, were analyzed.

The second approach, called IsoRank [184], is an algorithm for pairwise global alignment of PPI networks aiming at finding a correspondence between nodes and edges of the input networks that maximizes the overall match. This approach uses both PPI network data and sequence similarity data to compute the alignment. Moreover, the relative weights of the two data sources are free parameters. The basic idea is that a node i in the first PPI network can be mapped to a node j in the second PPI network if the neighborhood topologies of i and j are similar, i.e., the neighbors of i can be mapped to the neighbors of j. In particular, the algorithm works in two stages. In the first stage, it associates a score with each possible match between the nodes of the two networks. The score R_{ij} is the score associated to the pair of proteins i, from the first network, and j, from the second network. The vector R, representing the set of $R_{ij}s$, is computed by constructing and solving an eigenvalue problem. This problem encompasses both network and sequence data. In the second stage, the algorithm builds the mapping by extracting from R high-scoring, pairwise, mutually-consistent matches. This stage is resolved by interpreting R as encoding a bipartite graph and finding the maximum-weight bipartite matching for this graph. In particular, each partition of the bipartite graph contains all the nodes from one network and the edge weights are set to the value from R. At the end of the alignment, any unmatched node represents a gap node. The system was used to align the *S. cerevisiae* and the *D. melanogaster* PPI networks and the common identified subgraph had 1420 edges. After the alignment was performed, the results have been also used to detect functional orthologs using the same dataset exploited by Bandyopadhyay et al. [14].

IsoRank [184] has been extended, in a subsequent work [185], to align multiple PPI networks. In particular the five PPI networks of *Saccharomyces cerevisiae*, *Drosophila melanogaster*, *Caenorhabditis elegans*, *Mus musculus* and *Homo sapiens* were aligned. In this respect, the computation of the vector R is made for each pair of input networks. Since the authors consider more than two networks in input, the node mapping can be computed according two different scenarios: *(i)* one-to-one mappings, that require that any node can be mapped to at most one other node per species; *(ii)* many-to-many mappings, in which a node can be mapped to more than one node in another networks. To compute the mapping, the authors exploit an approximate multipartite graph weighted matching algorithm.

3.5 Concluding Remarks

In this chapter, the state of the art about protein function prediction has been discussed. The subsequent two chapters will present two novel approaches proposed to solve the function prediction problem according to two different methodologies. In particular, in Chapter 4 an approach based on the prediction of protein quaternary structure is presented. In Chapter 5 a novel approach to functional annotation of proteins based on protein-protein interaction networks comparison is discussed.

4

Protein Quaternary Structure Prediction

Summary. This chapter describe a novel method for protein quaternary structure prediction. In particular, in Section 4.1 some background information on protein quaternary structure is recalled and in Section 4.2 a brief comparison with existing methods is outlined. Section 4.3 discusses the classification method exploited in the prediction process and in Section 4.4 the results of the experimental validation on both homo-oligomers and hetero-oligomers datasets are reported.

4.1 Introduction

As pointed out in Chapter 3, protein quaternary structure is related to the biological function of the protein when it is involved in specific biological processes.

While an increasing number of amino acid sequences is produced and stored in public databases, the geometric conformation of a protein can be determined by slow and expensive methods (such as crystallography and NMR spectrometry). Thus, a plenty of computational methods have been developed in the last few years to predict and classify protein secondary, tertiary and quaternary structures [30, 76, 138, 167, 233]. The focus of this chapter is protein quaternary structure prediction. In particular, we deal with the problem of efficiently exploiting available databases of amino acid sequences in order to predict the number of subunits of a given protein.

In the rest of this section, we first briefly recall some basic concepts concerning protein quaternary structure and then point out our contributions.

Background on protein quaternary structure

Several proteins (e.g., Hemoglobin) are a combination of two or more individual polypeptide chains or subunits. The arrangement into which such subunits assemble is called the *protein quaternary structure*. Quaternary structure refers to the number of subunits involved in forming a protein, to their interconnections and to their organization [108, 198]. Biological processes are often influenced by the quaternary structure of proteins involved therein; e.g., the subunit construction of many enzymes provides the structural basis for the regulation of their activities. Proteins having a quaternary structure are called *oligomers*, and may be further classified as *homo-oligomers*, consisting of identical subunits, and *hetero-oligomers*, made of subunits that are different from one another. Furthermore, based on the number of subunits linked together, an oligomer may be a *dimer* (composed by two subunits), a *trimer* (composed by three subunits), a *tetramer* (composed by four subunits), and so on. Proteins consisting of only one subunit are called *monomers*.

Contributions

In this chapter, a classification method to individuate the number of subunits of each protein of a given dataset is dicussed.
To this aim, protein functional domain information is exploited, as already successfully done in previous literature [233]. In particular, each protein is encoded by a vector whose elements are associated to PFam domains [164]. The number of subunits included in a given protein is then obtained by assigning that protein to a class (e.g., monomers, homodimers, etc.), on the basis of a previously classified dataset and of a suitable classification method.

As already discussed in Chapter 3, a few approaches have been recently introduced to support protein quaternary structure classification [38, 76, 191, 233, 238]. The most successful of them [233, 238] reach at most the 87.5% of overall accuracy, and the maximum dataset size they considered is of about 10,000 proteins. Furthermore, most of the quaternary structure classification methods proposed in the literature store the overall dataset, comparing each protein to be classified to each stored protein. This may result hard when large datasets are to be considered.

Our approach gives a contribution in the direction of reducing both the portion of dataset that is necessary to store and, consequently, the number of comparisons to carry out at classification time, allowing sensible space and time savings, while achieving very good accuracy figures.

In particular, we exploit a nearest neighbor condensation techniques (in particular, a recently introduced one [6]) to replace the whole protein dataset with a notable subset that can be then used for the sake of fast protein quaternary structure prediction. To this aim, we use a training set consistent subset for the nearest neighbor decision rule as reference dataset during classification. Let T be a dataset. Having fixed a meaningful distance metrics, a subset S of T is a training set consistent subset of T for the nearest neighbor rule, if S correctly classifies all the objects of T by means of the nearest neighbor rule.

To evaluate our method, we conducted two series of experiments. The first series involved homo-oligomeric proteins while the second one classified hetero-oligomeric proteins. As for homo-oligomeric proteins, we considered two different kind of tests. First, we performed the 10-fold cross-validation on a very large protein dataset including 20,068 proteins taken from the SWISSPROT [10] database. The results confirmed the effectiveness of our approach. In fact, we scored an overall accuracy of 97.74%, by using only the 6.51% of the total dataset. This result is important, since pinpoints that our method can be adopted to correctly classify proteins whose quaternary structures are unknown, significantly reducing the portion of dataset to analyze. Such a reduction is particularly attractive in the case of protein quaternary structures classification, where large datasets are often to be considered. The second kind of tests concerns the exploitation of the jackknife cross-validation on a non-redundant dataset already used to test another successful technique proposed in the literature [233]. Also in this case, the results we obtained show that our method is more powerful than the previous ones, being able to obtain comparable accuracy in the classification of quaternary structures, even if using only the 45.39% of the whole dataset.

As for hetero-oligomeric proteins we performed the 10-fold cross-validation on a very large protein dataset including 33,273 proteins again extracted from the SWISSPROT database. In this respect, we conducted two types of experiments considering only PFamA domains and both PFamA and PFamB domains. Also in this case we obtained high accuracy values. Indeed, we obtained a precision score in the range 98,03%-99,03% by using a condensed dataset having a size in the range 2,76%-4,13% of the original dataset.

The rest of this chapter is organized as follows. Section 4.2 briefly addresses differences among our approach and the approaches that have been proposed in the literature and discussed in Chapter 3. Section 4.3 describes our protein quaternary structure classification method and Section 4.4 presents some experimental results. Finally, Section 4.5 reports some conclusions.

4.2 Related Work

Recently, some techniques have been proposed for protein quaternary structure classification [38, 76, 191, 233, 238] and a detailed description of such methods can be found in Chapter 3.

Most of them aim at classifying homo-oligomeric proteins. Differently from all of them, our approach has been used for classifying both homo-oligomers and hetero-oligomers. Moreover, all the approaches presented in the literature use a dataset of protein with known quaternary structure as training set and, during the prediction stage, compare the query protein to each protein in the training set. Differently from them our approach extract a consistent subset of the training set to reduce both time and space requirement at classification time. In the following a more detailed comparison is carried of.

The techniques presented in [76, 191, 238] are able to distinguish just between two classes, that are homodimers and non-homodimers, whereas our approach is able to discriminate among any number of classes. In this respect, our method is more similar to other two recently proposed approaches [38, 233].

Moreover, four of the five approaches presented in the literature [76, 191, 238, 38] exploit only protein sequence information, without any regard for protein domain composition. Our method is different, as we consider the protein domain composition that, according also to other studies [222, 101, 39, 30, 232], is able to deliver important information about protein structures and functions, which may be related to protein quaternary structure. In this respect, the most similar approach is the one by Yu et al [233] which also exploits the functional domain composition of proteins and the nearest neighbor algorithm (NNA).

Therefore, in our experiments, we used the same non-redundant dataset exploited by Yu et al. [233], enriched in the number of considered domains, obtaining some accuracy improvements (see Section 4.4). But, differently from their method, which exploits a *generalized distance* (which is not a metric) in the classification method, we used the Jaccard distance as the distance metric. Furthermore, our technique is more efficient than the one proposed by Yu et al. and, in general, than the other related techniques, due to its ability of classifying proteins without the necessity of making comparisons with all the elements of the dataset. Indeed, we are able to extract a relatively small subset of the training set to carry out such a classification without any significant loose in precision.

To summarize, our approach is more general than some of the previous methods [76, 191, 238], that are specific for the classification of only two classes of protein quaternary structures. Furthermore, we exploited the protein representation which is shown to be the most complete in terms of protein functional information (i.e., functional domain composition), and we achieve high accuracy values even if exploiting small dataset portions. All these features grant to our method highest overall success rate than the other ones presented in the literature (97.74%), making it attractive especially when large protein datasets have to be handled.

4.3 Classification through PQSC-FCNN

In this section, the classification method exploited to individuate the number of subunits of each unclassified protein of a given dataset is described. In the following we will refer to as PQSC-FCNN, for Protein Quaternary Structure Classification through FCNN rule, to the classification method here presented. In order to design an effective and efficient classification method, different issues are to be addressed: *(i)* the feature space and distance metrics to adopt, *(ii)* the classification algorithm, and *(iii)* the suitability of the overall method.

As already pointed out, most of the quaternary structure classification methods proposed in the literature store and use the whole available dataset as training set, comparing each protein to be classified to each stored protein. This may result hard when large datasets are considered. Hence, we would like to drastically reduce the portion of the dataset that is necessary to store and, consequently, the number of comparisons to carry out, allowing sensible space and time savings.

To this end, we exploit protein functional domain information, and encode each protein by a binary vector whose elements are associated to PFam domains [17]. We adopt the Jaccard metric as our distance measure and exploit the *k nearest neighbor rule* [42, 196, 48], one of the most extensively used nonparametric classification algorithms, which is simple to implement and yet powerful. The rationale underlying this choice is that, for this classification rule, there exist efficient techniques to reduce both space and time requirements.

In the following, the adopted protein representation, distance metrics, classification rule, and data reduction method are detailed.

Protein representation

To characterize proteins, we adopted the functional domain composition, since this kind of representation has been proved to be successful both for the specific problem we analyzed [233], and for the solution of other related problems, such as the prediction of protein-protein interactions [222, 101], of protein structures [39] and of protein functions [30, 232]. Protein functional domains are elements of the protein structure that are self-stabilizing and often fold independently of the rest of the protein chain. According to the functional domain composition, a protein is represented by a binary vector with size equals to the number of exploited domains. In particular, let D be an ordered set of protein domains, which have been considered to characterize the proteins in a dataset P. Then, each protein $p \in P$ is represented by a vector v_p of $|D|$ elements. The element $v_p[i]$ is set to be one if p contains the *i-th* domain in D, zero otherwise.

Distance metrics

We used the Jaccard metrics as our distance metrics, which is very suitable for binary data. In particular, the Jaccard distance between two protein vectors v_{p1} and v_{p2} is defined as:

$$d(v_{p1}, v_{p2}) = \frac{n_2 + n_3}{n_1 + n_2 + n_3}$$

where:

- n_1 is the number of domains belonging to both p_1 ans p_2;
- n_2 is the number of domains belonging to p_1 and not to p_2;
- n_3 is the number of domains belonging to p_2 and not to p_1.

Classification rule

The *nearest neighbor rule* [42] is widely used as a classification algorithm. It is simple to implement and yet powerful, due to its theoretical properties guaranteeing that for all distributions its probability of error is bounded above by twice the Bayes probability of error.

The nearest neighbor decision rule can be generalized to the case in which the k nearest neighbors are taken into account. In such a case, a new object is assigned to the class with the most members present among the k nearest neighbors of the object in the training set. This rule has the additional property that it provides a good estimate of the Bayes error and that its probability of error asymptotically approaches the Bayes error [73].

The naive implementation of the NN rule has no learning phase, since it requires to store all the previously classified data, and then to compare each sample point to be classified to each stored point. In order to reduce both space and time requirements, several techniques to reduce the size of the stored data for the NN rule have been proposed (see [221] for a survey). In particular, among those techniques, the *training set consistent* ones, aim at selecting a subset of the training set that correctly classifies the remaining data through the NN rule.

Data reduction

In order to reduce the reference protein quaternary structure dataset used during classification, we exploited the Fast Condensed Nearest Neighbor rule [6], FCNN for short,that is an algorithm computing a training set consistent subset for the NN rule.

Informally, having fixed a meaningful distance metrics and a dataset T, a subset S of T is a training set consistent subset of T for the nearest neighbor rule, if S correctly classifies all the objects of T by means of the nearest neighbor rule. Thus, loosely speaking, the objects of the subset S can be regarded as representing the objects of T which are not in S, and training set consistent subset methods for the nearest neighbor rule can be regarded as methods to filter out dataset instances which can be considered unessential to correctly classify new incoming objects.

The method is recalled next. We provide some definitions first. We define T as a labeled training set from a metric space with distance metrics d. Let x be an element of T. Then we denote by $nn_k(x, T)$ the kth nearest neighbor of x in T, and by $nns_k(x, T)$ the set $\{nn_i(x, T) \mid 1 \leq i \leq k\}$. $l(x)$ will be the label associated to x. Given a point y, the k-NN rule $\mathrm{NN}_k(y, T)$ assigns to y the label of the class with the most members present in $nns_k(y, T)$. A subset S of T is said to be a *k-training set consistent subset of* T if, for each $y \in (T - S)$, $l(y) = \mathrm{NN}_k(y, S)$. Let S be a subset of T, and let y be an element of S. By $Vor(y, S, T)$ we denote the set $\{x \in T \mid \forall y' \in S, \mathrm{d}(y, x) \leq \mathrm{d}(y', x)\}$, that is the set of the elements of T that are closer to y than to any other element y' of S, called the *Voronoi cell* of y in T w.r.t. S. Furthermore, by $Voren(y, S, T)$ we denote the set $\{x \in (Vor(y, S, T) - \{y\}) \mid l(x) \neq \mathrm{NN}_k(x, S)\}$, whose elements are called *Voronoi enemies* of y in T w.r.t. S. $Centroids(T)$ is the set containing the centroids of each class label in T. The FCNN rule relies on the following property: a set S is a training set consistent subset of T for the nearest neighbor rule if for each element y of S, $Voren(y, S, T)$ is empty.

The FCNN algorithm initializes the consistent subset S with a seed element from each class label of the training set T. In particular, the seeds employed are the centroids of the classes in T. The algorithm is incremental. During each iteration the set S is augmented until the stop condition, given by the property above, is reached. For each element of S, a *representative* element of $Voren(y, S, T)$ w.r.t. y is selected and inserted into S. Such a representative element it is the nearest neighbor of y in $Voren(y, S, T)$, that is, the element $nn(y, Voren(y, S, T))$ of T.

As for the time complexity of the method, let N denote the size of the training set T and let n denote the size of the computed consistent subset S. Then the FCNN rule requires Nn distance computations to compare the elements of T with the elements of S. However, if the distance employed is a metric, a technique exploiting the triangle inequality further reduces this worst case computational cost [6].

Large dataset						
	% Accuracy					
Classes	**PQSC-FCNN**, k = 2		**PQSC-FCNN**, k = 3		**PQSC-FCNN**, k = 4	
	Corr/Tot	**% Accuracy**	**Corr/Tot**	**% Accuracy**	**Corr/Tot**	**% Accuracy**
	% Accuracy					
Classes	**2-FCNN**	**3-FCNN**	**4-FCNN**			
Monomer	6,114/6,184	99.45%	6,130/6,184	99.13%	6,135/6,184	99.21%
Homodimer	8,408/8,690	96.75%	8,427/8,690	96.97%	8,402/8,690	96.68%
Homotrimer	1,154/1,190	96.97%	1,150/1,190	96.64%	1,136/1,190	95.46%
Homotetramer	2,422/2,513	96.38%	2,452/2,513	97.57%	2,380/2,513	94.71%
Homopentamer	232/237	97.89%	232/237	97.89%	232/237	97.89%
Homohexamer	759/784	96.81%	761/784	97.07%	742/784	94.64%
Homoheptamer	4/5	80.00%	4/5	80.00%	4/5	80.00%
Homooctamer	457/465	98.28%	458/465	98.49%	458/465	98.49%
Overall	97.60%		97.74%		97.11%	
% Dataset Exploitation	6.43%		6.51%		6.70%	

Table 4.1. Precision of 2-FCNN, 3-FCNN, 4-FCNN on a 20,068 protein dataset.

4.4 Experiments

In this section, we illustrate the experimental evaluation of the method proposed in this chapter.

To build our datasets, we downloaded proteins from the SWISSPROT database[1] [10] and domains from the PFam database[2] [17]. We conducted two series of experiments. The first series involved homo-oligomers while the second one conserned hetero-oligomers.

Homo-oligomeric proteins

As for homo-oligomeric proteins, we considered two different experiments. The first experiment consisted in running the 10-fold cross-validation on a very large protein dataset consisting of 20,068 proteins. The number of considered domains is 1,816. The results of this experiment are shown in Table 4.1. The first column of the table contains the homo-oligomeric class names, the second, third and fourth ones report both the number of correctly predicted proteins w.r.t. their total number and the percentage of accuracy scored by *PQSC-FCNN* for $k = 2$, $k = 3$ and $k = 4$, respectively, for each class. In the last two rows of the table, the overall accuracy and the percentage of exploited dataset are reported. The obtained results confirmed the effectiveness of our approach. In fact, the maximum overall success rate obtained on the entire dataset is of the 97.74%, and the minimum dataset exploitation was drastically reduced to the 6.43% of the original dataset. In general, as for the classification accuracy the three values of k were comparable, being equivalent on the homopentamers and on the homoheptamers, while only for $k = 3$ and for $k = 4$ the method returned the same results for homooctamers.

Table 4.2 shows detailed information about the condensed set generated by the method on the overall dataset of 20,068 proteins. In particular, for each class, both

[1] http:/www.ebi.ac.uk/swissprot/

[2] http://www.sanger.ac.uk/Software/Pfam/

Condensed set						
Classes	**PQSC-FCNN**, k = 2		**PQSC-FCNN**, k = 3		**PQSC-FCNN**, k = 4	
	Number of elements	**Percentage**	**Number of elements**	**Percentage**	**Number of elements**	**Percentage**
Monomer	98/6,184	1.58%	153/6,184	2.47%	157/6,184	2.54%
Homodimer	643/8,690	7.40%	649/8,690	7.47%	718/8,690	8.26%
Homotrimer	145/1,190	12.18%	101/1,190	8.49%	108/1,190	9.08%
Homotetramer	197/2,513	7.84%	199/2,513	7.92%	157/2,513	6.25%
Homopentamer	17/237	7.17%	17/237	7.17%	17/237	7.17%
Homohexamer	74/784	9.44%	74/784	9.44%	79/784	10.08%
Homoheptamer	3/5	60.00%	3/5	60.00%	3/5	60.00%
Homooctamer	29/465	6.24%	29/465	6.24%	29/465	6.24%
Overall	1, 206	6, 01%	1, 225	6.10%	1, 268	6.32%

Table 4.2. Condensed sets related to the dataset of 20,068 proteins.

the number of elements of the condensed set belonging to that class, and the reduction percentage w.r.t. the total number of elements in that class, are reported. The number of elements and the reduction percentage of the overall condensed set are shown on the last row of the table. By using all the three values of k (that are 2, 3 and 4) the method extracted condensed sets with the same size per class for homopentamers, homoheptamers and homooctamers. For the homoheptamer class, the reduction percentage was notably higher than for the other classes, due to the few elements belonging to that class (only 5). The reduction percentage on the overall dataset was 6.01% for $k = 2$, 6.10% for $k = 3$ and 6.32% for $k = 4$. This shows the power of the method, as it is sufficient to explore only a bit more than the 6% of the overall dataset to (most probably) classify a new protein.

In order to compare our method with a related one, in the second kind of experiments we considered the non-redundant protein dataset discussed by Yu et al. in [233]. The main goal of this comparison is to show that our method may have accuracy comparable to those of related methods, while sensibly reducing the amount of labeled data to exploit during the classification. In particular, we point out that the method presented in [233] utilized a non redundant version of the overall protein dataset in order to cope with problems associated with management of large data sets. As we will show in the following, our method is able to halve even this non redundant dataset, while maintaining the same accuracy as the competitor method.

Yu et al. adopted an approach based on the functional domain composition and employed the nearest neighbor algorithm (NNA) to classify protein quaternary structures. They represented the 717 considered proteins by 540 domains. Here, we enlarged the number of considered domains to 1,253 in order to obtain a more accurate data representation. Thus, we compared *PQSC-FCNN* with *NNA* by running the jackknife cross-validation on the non-redundant dataset, by considering the same 1,253 domains representation for all methods.

We run *PQSC-FCNN* exploiting the Jaccard metric, whereas *NNA* has been run with the generalized distance exploited in [233]. The results are illustrated in Table 4.3. The first column of the table contains the homo-oligomeric classes, the second, third, fourth and fifth ones illustrate both the number of correctly predicted objects w.r.t. the total number of them and the percentage of accuracy scored by *PQSC-*

Non-redundant dataset								
Classes	**PQSC-FCNN**, $k = 2$		**PQSC-FCNN**, $k = 3$		**PQSC-FCNN**, $k = 4$		**NNA**	
	Corr/Tot	**% Accuracy**	**Corr/Tot**	**% Accuracy**	**Corr/Tot**	**% Accuracy**	**Corr/Tot**	**% Accuracy**
Monomer	177/208	85.10%	174/208	83.65%	178/208	85.58%	168/208	80.77%
Homodimer	250/335	74.63%	250/335	74.63%	263/335	78.51%	268/335	80.00%
Homotrimer	28/40	70.00%	27/40	67.50%	27/40	67.50%	28/40	70.00%
Homotetramer	53/95	55.79%	52/95	54.74%	42/95	44.21%	53/95	55.79%
Homopentamer	11/11	100.00%	11/11	100.00%	11/11	100.00%	11/11	100.00%
Homohexamer	9/23	39.13%	9/23	39.13%	10/23	43.48%	7/23	30.43%
Homooctamer	2/5	40.00%	2/5	40.00%	2/5	40.00%	2/5	40.00%
Overall accuracy	73.92%		73.22%		74.34%		74.90%	
% Dataset Exploitation	46.79%		47.35%		45.39%		100%	

Table 4.3. Comparison of precision scores obtained by PQSC-FCNN and NNA.

FCNN for $k = 2$, for $k = 3$ and for $k = 4$, and *NNA*, respectively, for each considered class. In the last two rows of the table, the overall accuracy and the percentage of exploited dataset for each method are reported. We can observe that all the considered techniques returned the same results for the two classes of homopentamers and homooctamers. The only class for which *PQSC-FCNN* does not obtain higher accuracy than the *NNA* is the homodimer class. However, for $k = 4$ it obtains a success rate of 78.51% for that class, w.r.t. the 80.00% scored by the *NNA*, and thus, also in this case, the two methods achieved comparable results.

For the homotrimer and the homotetramer classes, the *PQSC-FCNN* for $k = 2$ and the *NNA* had the same accuracy scores, whereas in the remaining cases (i.e., monomers and homoexamers), *PQSC-FCNN* always scored better accuracy than *NNA*, also with some sensible improvements. In particular, for $k = 4$, *PQSC-FCNN* scored the accuracy value of 85.58% for monomers, which is about 4.81% higher than the success rate obtained by the *NNA*, whereas it scored the accuracy value of 43.48% for homohexamers, which is about 13.05% higher than the success rate obtained by the *NNA* for the same class and represents the best accuracy improvement w.r.t. *NNA* we obtained.

These results are significant since, for monomers and homoexamers, our method has been able to obtain more accurate results than *NNA* while exploiting only the 45.39% of the overall dataset, whereas the methods by Yu et al. [233] does not feature any dataset reduction. Summarizing, the *PQSC-FCNN* method is able to reach an overall success rate that is greater than the *NNA*, even if it exploits only the 45.39 – 47.35% of the original dataset. This means that *PQSC-FCNN* is more efficient than the method [233], allowing both time and space savings without any significant loss in accuracy but, rather, often allowing success rate improvements.

Hetero-oligomeric proteins

As for hetero-oligomeric proteins, we performed two type of experiments. In the first type, we considered both PFamA and PFamB domains for protein representation, while in the second one we considered only the PFamA domains. We executed the 10-fold cross-validation on a very large protein dataset including 33,273 proteins,

Hetero-oligomers dataset PFamA+PFamB domains						
Classes	**PQSC-FCNN**, $k = 2$		**PQSC-FCNN**, $k = 3$		**PQSC-FCNN**, $k = 4$	
	Corr/Tot	**% Accuracy**	**Corr/Tot**	**% Accuracy**	**Corr/Tot**	**% Accuracy**
Monomer	14,681/14,801	99.19%	14,640/14,801	98.91%	14,653/14,801	99.00%
Heterodimer	9,331/9,632	96.88%	9,323/9,632	96.79%	9,357/9,632	97.14%
Heterotrimer	1,379/1,438	95.90%	1,376/1,438	95.69%	1,352/1,438	94.02%
Heterotetramer	6,017/6,157	97.73%	6,061/6,157	98.44%	6,077/6,157	98.70%
Heteropentamer	25/29	86.20%	25/29	86.21%	23/29	79.31%
Heterohexamer	778/798	97.49%	783/798	98.12%	782/798	98.00%
Heterooctamer	409/418	97.84%	409/418	97.84%	408/418	97.60%
Overall	98.04%		98.03%		98.13%	
% Dataset Exploitation	3.66%		3.84%		4.13%	

Table 4.4. Precision of 2-FCNN, 3-FCNN, 4-FCNN on a 33,273 protein dataset.

Hetero-oligomers Condensed sets PFamA+PFamB domains						
Classes	**PQSC-FCNN**, $k = 2$		**PQSC-FCNN**, $k = 3$		**PQSC-FCNN**, $k = 4$	
	Number of elements	**Percentage**	**Number of elements**	**Percentage**	**Number of elements**	**Percentage**
Monomer	135/14,801	0.91%	220/14,801	1.49%	203/14,801	1.37%
Homodimer	612/9,632	6.35%	600/9,632	6.23%	673/9,632	6.99%
Homotrimer	116/1,438	8.07%	96/1,438	6.68%	102/1,438	7.09%
Homotetramer	199/6,157	3.23%	203/6,157	3.30%	240/6,157	3.90%
Homopentamer	10/29	34.48%	10/29	34.48%	11/29	37.93%
Homohexamer	49/798	6.14%	50/798	6.27%	55/798	6.89%
Homooctamer	23/418	5.50%	23/418	5.50%	25/418	5.98%
Overall	1, 114	3.44%	1, 202	3.61%	1, 309	3.93%

Table 4.5. Condensed sets related to the dataset of 33,273 proteins.

extracted from the SWISSPROT database. As for the first experiment, involving both PFamA and PFamB domains, we exploited 3,389 domains. The obtained results are shown in Table 4.4. The maximum overall success rate obtained on the entire dataset was of the 98.13%, and the minimum dataset exploitation was drastically reduced to the 3.66% of the original dataset. In general, as for the classification accuracy the three values of k were comparable.

Table 4.5 shows detailed information about the condensed set generated by the method on the overall dataset of 33,273 proteins. The training set consistent subset extracted from the whole dataset has a size equals to the 3.44% for $k = 2$, 3.61% for $k = 3$ and 3.93% for $k = 4$ of the size of the original dataset. This shows the power of the method, as it is sufficient to explore less than the 4% of the overall dataset to (most probably) classify a new protein.

Table 4.6 and Table 4.7 show the results obtained on the same dataset of hetero-oligomers using only PFamA domains in the protein representation. As it can be noted, the size of the exploited dataset is of 31,807 proteins, thus it is even smaller than the previous one. The reason is that some proteins were composed only of PFamB domains and, in this experiment, they were deleted. The maximum overall success rate obtained on the entire dataset was of the 99.03%, and the minimum dataset exploitation was drastically reduced to the 2.76% of the size of original dataset. Also in this case, as for the classification accuracy the three values of k

Hetero-oligomers dataset Only PFamA domains						
Classes	**PQSC-FCNN**, $k = 2$		**PQSC-FCNN**, $k = 3$		**PQSC-FCNN**, $k = 4$	
	Corr/Tot	**% Accuracy**	**Corr/Tot**	**% Accuracy**	**Corr/Tot**	**% Accuracy**
Monomer	14,500/14,533	99.77%	14,479/14,533	99.63%	14,478/14,533	99.62%
Heterodimer	8,994/9,200	97.76%	8,979/9,200	97.60%	8,996/9,200	97.78%
Heterotrimer	1,355/1,378	98.33%	1,334/1,378	96.81%	1,334/1,378	96.81%
Heterotetramer	5,515/5,543	99.49%	5,511/5,543	99.42%	5,503/5,543	99.28%
Heteropentamer	22/23	95.65%	22/23	95.65%	22/23	95.65%
Heterohexamer	765/778	98.33%	765/778	98.33%	759/778	97.55%
Heterooctamer	347/352	98.58%	348/352	98.86%	345/352	98.01%
Overall	99.03%		98.84%		98.84%	
% Dataset Exploitation	2.76%		2.98%		3.08%	

Table 4.6. Precision of 2-FCNN, 3-FCNN, 4-FCNN on a 31,807 protein dataset.

Hetero-oligomers Condensed sets Only PFamA domains						
Classes	**PQSC-FCNN**, $k = 2$		**PQSC-FCNN**, $k = 3$		**PQSC-FCNN**, $k = 4$	
	Number of elements	**Percentage**	**Number of elements**	**Percentage**	**Number of elements**	**Percentage**
Monomer	97/14,533	0.67%	166/14,533	1.14%	161/14,533	1.11%
Homodimer	434/9,200	4.72%	6451/9,200	4.90%	496/9,200	5.39%
Homotrimer	84/1,378	6.10%	65/1,378	4.72%	69/1,378	5.01%
Homotetramer	133/5,543	2.40%	133/5,543	2.40%	114/5,543	2.06%
Homopentamer	5/23	21.74%	5/23	21.74%	6/23	26.09%
Homohexamer	44/778	5.66%	45/778	5.78%	50/778	6.43%
Homooctamer	17/352	4.83%	17/352	4.83%	18/352	5.11%
Overall	814	2.56%	882	2.77%	914	2.87%

Table 4.7. Condensed sets related to the dataset of 31,807 proteins.

were comparable and the 3-FCNN and 4-FCNN obtained the same results. As for the condensed sets generated by the method, the training set consistent subset extracted from the whole dataset has a size equals to the 2.56% for $k = 2$, 2.77% for $k = 3$ and 2.87% for $k = 4$ of the size of the original dataset.

Summarizing, the *PQSC-FCNN* method is able to reach a good classification precision even if it exploits only a very small portion of the original dataset. This means that *PQSC-FCNN* is a powerful tool, allowing both time and space savings.

4.5 Concluding Remarks

In this chapter a classification method for protein quaternary structures has been proposed. This method exploits protein functional domain information and the FCNN rule. Experimental evaluations showed that this approach is able to reduce the portion of protein dataset that is necessary to store, by extractiong a training set consistent subset, and, this, the number of comparisons to carry out during the classification of a new protein, allowing sensible space and time savings even guaranteeing high values of accuracy. Some tests carried out on homo-oligomeric and hetero-holigomeric proteins have been illustrated, confirming the validity of the approach.

In the next chapter a novel approach for the prediction of protein functions by comparing PPI networks will be described.

5

BI-GRAPPIN:Functional Similarity Search by PPI Network Analysis

Summary. This chapter describes a method for predicting protein function by comparing the protein-protein interaction networks of two species. Section 5.1 provides some background information about the comparison of PPI networks. Section 5.2 presents the BI-GRAPPIN algorithm along with some application cases. In Section 5.4, the results of the experimental evaluation of BI-GRAPPIN and the comparison with other algorithms is outlined. Finally, in Section 5.5 some conclusions are drawn.

5.1 Introduction

The problem of identifying conserved functional components across species is a central problem in biology. After the huge efforts that have been made toward completing the genome coding of several organisms [41], a large deal of attention is now turning toward the analysis of the ever increasing amount of annotated proteins. The observation that biological variations caused by evolution influence the ways proteins interact with one another, recently persuaded biologists that a protein cannot be analyzed independently of the other proteins participating into common biological processes [211]. The set of all the protein-protein interactions of a given organism is its *interactome*. The *interactome* is usually modeled by an indirect graph, i.e., the protein-protein interaction (PPI) network, where, as already discussed in Chapter 2, nodes represent proteins and edges encode their interactions. Protein interactions are usually discovered by high-throughput experimental techniques [89, 114] and computational methods [140, 211]. In both cases, the resulting interactions to hold are not completely reliable [193], as also testified by several specific studies [9, 47]. Clear enough, the limited reliability of such data may potentially affect any attempt to extract useful information from them.

In this chapter, we deal with the problem of searching for functional conservation across interaction networks of different organisms. This problem has been already considered in the literature [14, 71, 97, 182, 184]. Moreover, several approaches related to the work presented in this chapter have already been discussed in detail in Chapter 3. Our technique, called BI-GRAPPIN (Bipartite GRAph based Protein-Protein

Interaction Networks analysis), is inspired by an approach for matching database schemes [154]. Bi-Grappin is based on the computation of the maximum weighted matching [74] on bipartite graphs and aims at "measuring" the similarity between pairs of nodes of two networks. The intuition here is that a protein in one network should be actually considered similar to a protein in the other network as long as they are not only characterized by a good sequence similarity, but also by similar interaction profiles (here referred to as *neighborhoods*) [14, 62, 184]. In particular, we consider the sequence similarities between proteins of different networks and *refine* them by analyzing the similarities of their neighbor proteins. In more detail, we adopt a concept of "neighborhood" that is different from that adopted in some related work (e.g., [14, 184]). This new definition of neighborhood is not simply related to the number of edges connecting two nodes but, mostly, to the weights of edges. Surely, when information about weights is not available, our definition can be reduced to the one that assign to the *i*-neighborhood of a given protein all the proteins connected to it by a path of length *i*.

Bi-Grappin is independent of the topology of the analyzed networks and it provides the possibility to incorporate both quantitative and reliability information during the analysis. In particular, information about the strength of the interaction of two proteins, related to physical-chemical features [120, 194], is exploited independently of plausibility information about that interaction to reliably hold. On its turn, reliability depends on the way by which the interaction was discovered – laboratory, high-throughput or computational methods. Therefore, the two kinds of information are meant to play a different role in the similarity search. At the best of our knowledge, this is the first attempt in this direction.

The proposed approach can be summarized as follows. Given two PPI networks, Bi-Grappin considers each pair of proteins (p', p'') from the first and the second network, respectively. If the two proteins feature at least a weak sequence similarity (e.g., the BLAST E-value $\leq 10^{-2}$, as also done in [97]), the algorithm starts by exploring the first neighborhood of p' and p''. Such neighborhoods are used to build a bipartite graph on which a maximum weight matching w.r.t. sequence similarities is computed. The value thus obtained is combined with the sequence similarity of p' and p'' to compute a new refined similarity value between the proteins. This value will be further refined by iteratively analyzing the farthest neighborhood of p' and p''. The graph exploration stops when a given number of neighborhoods of p' and p'' has been analyzed.

To validate the effectiveness of Bi-Grappin, we ran it on the three PPI networks of *Saccharomyces cerevisiae* (the yeast), *Drosophila melanogaster* (the fly) and *Caenorhabditis elegans* (the worm) and performed two different kinds of experiments. The first one concerned the discovery of *functional orthologs* [14], that is, proteins performing the same biological function in different species. In this respect, we compared our results with those of two other approaches [14, 185]. Experimental evaluations confirmed that our method is successful in individuating functional orthologs. In the second kind of experiments, Bi-Grappin has been exploited to align the *S. cerevisiae* network with the *D. melanogaster* and the *C. elegans* networks. This analysis helped to verify that Bi-Grappin can be profitably exploited to individuate common processes in which proteins are involved. These latter experiments showed that Bi-Grappin is able to correctly single out proteins that are known to be involved in similar biological processes. That confirmed the correctness and reliability of this approach. Furthermore, those experiments highlighted the merits of the proposed technique in understanding the role of not yet well characterized proteins. In this respect, it is worth noting that we chose to align the *yeast* network with that of the *fly* and the *worm* since the *yeast* is a much more characterized organism than the other two.

The rest of the chapter is organized as follows. In the next section the Bi-Grappin algorithm is illustrated in detail. In Section 5.3, a comparison with some related work is provided. The experimental evaluations are reported in Section 5.4. Finally, in Section 5.5, some conclusions are drawn.

5.2 A Technique for Protein Similarity Refinement

In this section some useful definitions are introduced. Then, in Section 5.2.1, the algorithm Bi-Grappin is presented, and in Section 5.2.2 three examples showing the behavior of the algorithm on some artificial, yet significant, application cases are discussed.

The most common representation for protein-protein interaction networks is that of undirected graphs, where nodes represent proteins and edges denote their interactions.

Definition 5.1. *(Graph Protein-Protein Interaction Network)* Let $P = \{p_1, p_2, \ldots, p_n\}$ be the set of nodes denoting the proteins of a given organism (and identified by protein *ids*), and let I be the set of (undirected) labeled edges $\langle\{p_i, p_j\}, l\rangle$, associated to the interactions between pairs of proteins. Each edge label l is a pair of the form $\langle w, c\rangle$, where w and c are real numbers in the interval $[0, 1]$, called weakness and confidence, resp. A graph protein-protein interaction network (or *graph PPI network*) is then $\mathcal{G}_N = \langle P, I\rangle$.

Edge labels are used to encode both quantitative and reliability information about interactions, whenever available. For example, quantitative information, encoded in the term w of the label pair, might concern protein-protein interaction strength [120, 194], so that larger values of w denote weaker interactions. Beside quantitative information, we are also interested in representing the reliability associated with interactions [193]. Thus, the term c of the label pair represents a reliability coefficient that weighs to what extent a stored interaction should be reliably taken into account in the analysis.

Definition 5.2. *(Interaction Path$_i$, Cumulative Confidence C)* Given a graph PPI network $\mathcal{G}_N$, we call *Interaction Path* of rank i (shortly, *I-Path$_i$*) a path such that:

$$\mathcal{F}(i-1) \leq \sum_u w_u \leq \mathcal{F}(i)(i \geq 1)$$

where each w_u is the weakness value associated with edge u in the path and $\mathcal{F}$ is a user specified function, taking a nonnegative integer in input and returning a nonnegative integer as output, such that $\mathcal{F}(0) = 0$. The series $\{\mathcal{F}(i)\}_{i\geq 0}$ serves the purpose of encoding neighborhood border weight values and, as such, to suitably "shape" the graph neighborhood level structure.

The *Cumulative confidence* C of the I-Path$_i$ is defined as $C = \prod_u c_u$, where the c_u denote the confidences of edges $u \in$I-Path$_i$.

Definition 5.3. *(I-Shortest Path)* The *I-Shortest Path* between two nodes p and q in $\mathcal{G}_N$, denoted by $sp(p, q)$, is the path among those linking p to q such that $\sum_u w_u$ is minimum, where each w_u is the weakness value associated with edges occurring in the path. If more than one such a path exists, the one with maximum cumulative confidence is chosen (anyone of them, in case of a further tie).

Definition 5.4. *(i-th Neighborhood)* Given a node p in a graph PPI network $\mathcal{G}_N = \langle P, I\rangle$, the i-th neighborhood of p is the set:

$$\mathcal{N}(p,i) = \{q | q \in P, q \neq p, sp(p,q) \text{ is a I-Path}_i \text{ in } \mathcal{G}_N, i \geq 0\}.$$

$\mathcal{N}(p,i)$ is the set of nodes that can be reached from p through an I-Path$_i$ that is also an I-Shortest path.

Note that while the sum of weaknesses across an I-shortest path determines the i-neighborhood which a node p belongs to, the cumulative confidence is representative of the probability that p actually belongs to that i-neighborhood.

In the following, we shall assume that the graph representing the PPI network of a given organism is connected. This is reasonable in general and, whenever this condition is not satisfied, our technique can be thought as applied to each of the connected components of the graph PPI network by its own.

5.2.1 The Bi-Grappin Algorithm

Let $\mathcal{G}_{N_1}$ and $\mathcal{G}_{N_2}$ be two graph PPI networks, and assume that each pair of proteins (p', p''), with $p' \in \mathcal{G}_{N_1}$ and $p'' \in \mathcal{G}_{N_2}$, have been aligned using one of the available sequence alignment algorithms. Therefore, let SSD be a sequence similarity dictionary storing all the triplets $\langle p', p'', f_0\rangle$, where f_0 is a coefficient in the real interval $[0, 1]$ obtained from the alignment parameters[1] The larger f_0 the more similar the sequences of p' and p''. The output of our technique is a new set of triplets, called FSD (i.e., *Functional Similarity Dictionary*). In particular, FSD stores triplets of the form $\langle p', p'', f_p\rangle$, where $p' \in \mathcal{G}_{N_1}$, $p'' \in \mathcal{G}_{N_2}$ and f_p is a *protein-protein similarity coefficient* in the real interval $[0, 1]$ measuring the *refined* similarity between p' and p'', as computed by the Bi-Grappin algorithm. As before, the larger f_p the more similar p' and p''.

[1] In our experiments, in order to compute f_0, we have used the Blast 2 sequences algorithm [202] and the associated E-value parameter.

The algorithm starts by setting the FSD equals to the SSD. Then, each triplet $\langle p', p'', f_p \rangle$ in FSD with f_p larger than a fixed cut-off value ($f_{\text{cut-off}}$) is considered in order to refine its f_p value. To this end, the i-neighborhoods of p' and p'' ($i \geq 1$) are iteratively generated and compared by computing the objective function of a maximum weight matching. At the generic iteration i, the output of such an objective function is exploited to refine the value f_p. The neighborhood analysis stops at a fixed iteration i_{MAX} whose value is fixed as explained later in this section. The final refined value of f_p is that corresponding to the i_{MAX}-th iteration. Figure 5.1 shows the pseudocode of the algorithm.

The core of the algorithm is the evaluation of the similarity between two i-neighborhoods, which is based on a maximum weight matching computation. Given the two i-neighborhoods $\mathcal{N}(p', i) = \{p'_1, p'_2, \ldots, p'_{m_1}\}$ and $\mathcal{N}(p'', i) = \{p''_1, p''_2, \ldots, p''_{m_2}\}$, consider the sets:

- $\mathcal{S}'(p', p'', i) = \{p'_h \in \mathcal{N}(p', i)$ s.t. $\exists\, p''_k \in \mathcal{N}(p'', i)$ and a triplet $\langle p'_h, p''_k, f_0 \rangle \in SSD$, with $f_0 \geq f_{\text{match}}\}$;
- $\mathcal{S}''(p', p'', i) = \{p''_k \in \mathcal{N}(p'', i)$ s.t. $\exists\, p'_h \in \mathcal{N}(p', i)$ and a triplet $\langle p'_h, p''_k, f_0 \rangle \in SSD$, with $f_0 \geq f_{\text{match}}\}$;

Let X be set of edges $\{\langle p'_h, p''_k, g_{hk} \rangle | g_{hk} = C_{hk} \cdot f_{hk}\}$, where:

- $p'_h \in \mathcal{S}'(p', p'', i)$;
- $p''_k \in \mathcal{S}''(p', p'', i)$;
- f_{hk} is the sequence similarity between p'_h and p''_k as stored in the input SSD;
- $C_{hk} = \min\{C_h, C_k\}$, where C_h and C_k are the cumulative confidences of the I-shortest paths connecting p'_h to the target protein p' and p''_k to the target protein p'', respectively.

Moreover, consider the bipartite weighted graph $BG = (\mathcal{S}'(p', p'', i) \cup \mathcal{S}''(p', p'', i), X)$. The fixed threshold value f_{match} considered in the building of $\mathcal{S}'(p', p'', i)$ and $\mathcal{S}''(p', p'', i)$ is used to prune the set of nodes to be considered for the sake of the matching. Note that such a pruning is safe since it a-priori excludes only insignificant pairings, corresponding to pairs of proteins with a too low sequence similarity. The maximum weight matching for BG is a set $X' \subseteq X$ of edges such that for each node $x \in \mathcal{S}'(p', p'', i) \cup \mathcal{S}''(p', p'', i)$ there is at most one edge of X' incident onto x and $\phi(X') = \sum_{\langle p'_h, p''_k, g_{hk} \rangle \in X'} g_{hk}$ is maximum.
Note that $\phi(X')$ returns a measure of how much the two involved neighborhoods match, considering not only neighbor sequence similarities but also the associated cumulative confidences.

Let $\Pi_j(X')$, $1 \leq j \leq 2$ denote the projections of X' on the $j - th$ component of its triplets. Consider the set of nodes Γ of $\mathcal{N}(p', i) \cup \mathcal{N}(p'', i)$ that remained unmatched[2] in X', that is:

$$\Gamma = (\mathcal{N}(p', i) \setminus \Pi_1(X')) \cup (\mathcal{N}(p'', i) \setminus \Pi_2(X')).$$

Clear enough, in evaluating the similarity of the two neighborhoods analyzed at the generic step of the algorithm, unmatched nodes have to be taken into account by

[2] The unmatched nodes are called *gap nodes* in [184].

```
Algorithm Bi-Grappin
Input:
- a sequence similarity dictionary $SSD$
- two graph PPI networks $\mathcal{G}_{N_1}$ and $\mathcal{G}_{N_2}$
- the stop iteration $i_{\text{MAX}}$
- two real values $f_{\text{cut-off}}$ and $f_{\text{match}}$
- a real value $\alpha$
Ouput: a functional similarity dictionary $FSD$
set $FSD = SSD$
for each triplet $\langle p', p'', f_p \rangle$ in $FSD$
    if ($f_p \geq f_{\text{cut-off}}$)
        set $i = 1$
        while $i \neq i_{\text{MAX}}$
            generate the $i$-th neighborhoods $\mathcal{N}(p', i)$ and $\mathcal{N}(p'', i)$ of $p'$ and $p''$, resp.
            generate the sets $\mathcal{S}'(p', p'', i)$ and $\mathcal{S}''(p', p'', i)$
            compute the maximum weighted match $X'$ and the set of unmatched nodes $\Gamma$
            compute $\mu(\mathcal{N}(p', i), \mathcal{N}(p'', i), X', \Gamma, \alpha)$
            refine the value of $f_p$ as:
                $f_p(i) = \delta(i) \times \mu(\mathcal{N}(p', i), \mathcal{N}(p'', i), X', \Gamma, \alpha) + [1 - \delta(i)] \times f_p(i-1)$
            $i = i + 1$
return the functional similarity dictionary $FSD$
```

Fig. 5.1. The Bi-Grappin algorithm.

suitably decreasing the matching value, as their presence witnesses for differences in the two neighborhoods. Therefore, the following value is computed:

$$\mu(\mathcal{N}(p', i), \mathcal{N}(p'', i), X', \Gamma, \alpha) = (1 - \alpha \cdot \Lambda(\mathcal{N}(p', i), \mathcal{N}(p'', i), \Gamma)) \frac{\phi(X')}{\Theta(X')}$$

where:

- $\Lambda(\mathcal{N}(p', i), \mathcal{N}(p'', i), \Gamma) = \frac{\sum_{p_\gamma \in \Gamma} C_\gamma}{\sum_{p_\beta \in \mathcal{N}(p',i) \cup \mathcal{N}(p'',i)} C_\beta}$ denotes the proportion of the unmatched nodes weighted by the cumulative confidences C_γ associated with the I-shortest paths connecting p_γ to the target protein p' within the first network (resp. to p'' within the second one) over the sum of all the coefficients C_β, similarly associated with nodes in $\mathcal{N}(p', i) \cup \mathcal{N}(p'', i)$.
- The factor $\Theta(X') = \sum_{(p'_h, p''_k, g_{hk}) \in X'} C_{hk}$ is used to normalize $\phi(X')$ in the range $[0, 1]$, as $\sum_{(p'_h, p''_k, g_{hk}) \in X'} C_{hk}$ denotes the sizes of X' weighted by C_{hk}.
- α is a coefficient used to tune the weight of unmatched nodes w.r.t. that of matched ones.

At each step i, the value f_p of the considered triplet $\langle p', p'', f_p \rangle$ in FSD is modified according to the following formula:

$$f_p(i) = \delta(i) \times \mu(\mathcal{N}(p', i), \mathcal{N}(p'', i), X', \Gamma, \alpha) + [1 - \delta(i)] \times f_p(i-1)$$

where $\delta(i)$ represents the generic term of a succession $\{\delta(i)\}_{i \geq 1}$ of factors used to weaken the contribution of nodes belonging to farthest neighborhoods. Thus, $\{\delta(i)\}$ is monotone decreasing (in our experiments we set $\delta(i) = \frac{1}{1+i}$).

We recall that the neighborhood analysis stops at a fixed iteration i_{MAX}. Such a value i_{MAX} has to be chosen such that:

- the size of the analyzed neighborhoods does not get anyway comparable with the one of the entire graphs,
- the analyzed neighborhoods are not "too far" from the corresponding proteins,

since, otherwise, the results computed via the maximum weight matching would not be actually significant. Therefore we have the following result.

Theorem 5.5. *Let $\mathcal{G}_{N_1}$ and $\mathcal{G}_{N_2}$ be two graph PPI networks of n_1 and n_2 nodes, respectively, and let $n = max\{n_1, n_2\}$. Let i_{MAX} be the chosen iteration upper bound and let $n_{i_{MAX}}$ be the number of nodes in the $\mathcal{S}'(p', p'', i)$ ($\mathcal{S}''(p', p'', i)$, resp.) $1 \leq i \leq i_{MAX}$, of maximum size. Then, in the worst case, the algorithm* Bi-Grappin *runs in $O(max((n^3_{i_{MAX}} \cdot n^2), n^3))$ time.*

Proof. The I-Shortest path between each pair of nodes in each graph PPI network can be pre-computed by the Floyd-Warshall algorithm in $O(n^3)$. Thus, for each of the two networks, a matrix M of size n^2 can be built where each element $M[h, k]$ contains both the sum of the weaknesses and the cumulative confidences for the I-Shortest path connecting the node h and the node k. Hence, building the i-th neighborhood of a node costs $O(n)$.

The time required to compute the maximum weight matching of a bipartite graph including $\overline{n}$ nodes is $O(\overline{n}^3)$ [74]. Because of the definition of i_{MAX}, the number of nodes in each of the analyzed bipartite graphs is $O(n_{i_{\mathrm{MAX}}})$, thus the maximum weight matching costs $O(n^3_{i_{\mathrm{MAX}}})$. Both i-th neighborhood extraction and maximum weight matching have to be computed for each of the n^2 triplets in SSD. Thus, the overall cost of the FSD construction is $O(\max((n^3_{i_{\mathrm{MAX}}} \cdot n^2), n^3)$.

In particular, let n be the number of nodes in the largest of the two analyzed networks; even if other choices are possible, we recommend to fix i_{MAX} (which is what we done in our experiments) so that:

1. for each pair of nodes p' in $\mathcal{G}_{N_1}$ and p'' in $\mathcal{G}_{N_2}$, $|\mathcal{S}'(p', p'', i_{\mathrm{MAX}})| \leq \log^2(n)$ and $|\mathcal{S}''(p', p'', i_{\mathrm{MAX}})| \leq \log^2(n)$;
2. there is at least one pair of nodes $\overline{p'}$ in $\mathcal{G}_{N_1}$ and $\overline{p''}$ in $\mathcal{G}_{N_2}$ such that $|\mathcal{S}'(\overline{p'}, \overline{p''}, i_{\mathrm{MAX+1}})| > \log^2(n)$ or $|\mathcal{S}''(\overline{p'}, \overline{p''}, i_{\mathrm{MAX+1}})| > \log^2(n)$.

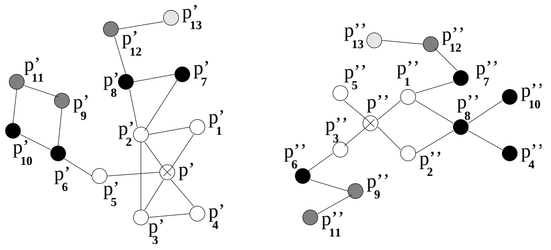
5.2.2 Application Cases

In this section, we illustrate some specific cases that we used to validate the algorithm. We built some ad hoc examples discussed below for this purpose, where the

Proteins		w	c
p'	p'_1	0.980	0.950
p'	p'_2	0.880	0.810
p'	p'_3	0.930	0.380
p'	p'_4	0.820	0.750
p'	p'_5	0.780	0.920
p'_1	p'_2	0.530	0.980
p'_2	p'_3	0.750	0.810
p'_2	p'_7	0.720	0.830
p'_2	p'_8	0.530	0.980
p'_5	p'_6	0.380	0.910
p'_6	p'_9	0.930	0.890
p'_6	p'_{10}	0.750	0.510
p'_7	p'_8	0.680	0.350
p'_8	p'_{12}	0.930	0.510
p'_9	p'_{11}	0.850	0.790
p'_{10}	p'_{11}	0.910	0.750
p'_{12}	p'_{13}	0.690	0.830

Proteins		w	c
p''	p''_1	0.950	0.920
p''	p''_2	0.910	0.860
p''	p''_3	0.850	0.880
p''	p''_5	0.870	0.580
p''_1	p''_7	0.780	0.690
p''_1	p''_8	0.730	0.510
p''_2	p''_8	0.490	0.990
p''_3	p''_6	0.630	0.850
p''_4	p''_8	0.530	0.910
p''_6	p''_9	0.540	0.770
p''_7	p''_{12}	0.910	0.970
p''_8	p''_{10}	0.580	0.780
p''_9	p''_{11}	0.930	0.880
p''_{12}	p''_{13}	0.580	0.850

(a)



(b)

p′-p″ (f₀ = 0.600)

i	$\mathbf{p'_h}$	$\mathbf{p''_k}$	$\mathbf{f_{hk}}$	$\frac{\phi(X')}{\Theta(X')}$	$\sum_{p_\gamma \in \Gamma} C_\gamma$	$\mathbf{f_p}$
1	p'_2	p''_2	0.850	0.810	0.92	0.692
	p'_3	p''_3	0.400			
	p'_1	p''_1	0.900			
	p'_5	p''_5	0.880			
2	p'_7	p''_7	0.880	0.808	0.775	0.721
	p'_8	p''_8	0.920			
	p'_6	p''_6	0.860			
	p'_{10}	p''_{10}	0.400			
3	p'_9	p''_9	0.880	0.881	0	0.761
	p'_{12}	p''_{12}	0.860			
	p'_{11}	p''_{11}	0.910			
4	p'_{13}	p''_{13}	0.850	0.850	0	0.779

(c)

Fig. 5.2. Example 1: increasing of the initial similarity value.

involved networks have small size only for ease of exposition. The behavior of Bi-Grappin does not change in analogous situations when larger networks are considered, but illustrating examples with thousand of nodes would have been, we argue, less explanatory.

The first situation we analyzed is that illustrated in Figure 5.2, where the starting similarity between the two target proteins p' and p'' is $f_0 = 0.600$. In particular, in Figure 5.2 (b) the subnetworks including p', p'' and their neighborhoods, up to the fourth level ones, are shown, highlighting with different gray tones proteins in different neighborhoods and showing edge labels using two tables, one for each network, in Figure 5.2 (a). Note that, for these synthetic examples, edge labels are often different from 1. Pairings between proteins in corresponding neighborhoods of the two target proteins, as returned by running Bi-Grappin, are shown in the table reported in Figure 5.2 (c). There, for each i-neighborhood of p' and p'', the second column reports the SSD values corresponding to the triplets $\langle p'_h, p''_k, f_{hk} \rangle$, where p'_h and p''_k are the best matched i-neighbors. The third and fourth columns contain the average neighborhood similarity and the sum of the cumulative confidences of the unmatched nodes of the i-neighborhood of p' and p'', respectively. The fourth column gives a measure of how much the unmatched nodes influence the final value of similarity. Finally, the fifth column shows p'-p'' similarity, as refined at each stage i. Analyzing in detail the intermediate outputs of Bi-Grappin, we can observe that the initial $f_0 = 0.600$ between p' and p'' is increased after analyzing the 1- neighborhood, due to the high similarity of proteins in corresponding neighborhoods paired during the matching process. Then, f_p further increases after the analysis of the 2-, 3-, and 4-neighborhoods, obtaining a final $f_p = 0.799$, for $i_{\text{MAX}} = 4$, that is, as expected, larger than f_0.

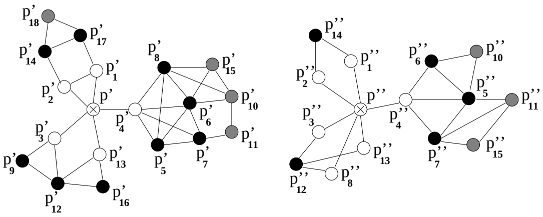
The second situation we consider is that illustrated in Figure 5.3, where the i-neighborhood of p' and p'' is explored up to $i_{\text{MAX}} = 3$ and their starting similarity is $f_0 = 0.850$. From the analysis of Figure 5.3, which is analogous to Figure 5.2 in terms of table structure, we can observe that the similarity between p' and p'' decreases after the analysis of the first and the second neighborhoods, where the average neighborhood similarities are relatively small and there are some unmatched nodes. Then, f_p weakly increases after the analysis of the 3-neighborhood, for which the average neighborhood similarity increases, while remaining lower than f_0. This is supposedly correct, since even if the sequence similarity between p' and p'' is high, their neighborhoods share a low average similarity and present some unmatched nodes, indicating low functional similarity.

The third example is illustrated in Figure 5.4, where $i_{\text{MAX}} = 2$ and $f_0 = 0.710$. In this case, the similarity between p' and p'' decreases after the analysis of the first neighborhoods, where the average neighborhood similarity is lower than f_0. Then, f_p increases after analyzing the 2-neighborhood, where the average neighborhood similarity is higher than the previous f_p. This example highlights that, as also suggested elsewhere [40], limiting the neighborhood analysis only to the first level would not be sufficient for the sake of obtaining coherent results w.r.t. functional conservation. This is because p' and p'' are supposedly involved in common biological processes but there is no evidence of that in their first level neighborhoods.

Proteins		w	c
p'	p'_1	0.980	0.970
p'	p'_2	0.940	0.930
p'	p'_4	1.000	0.940
p'	p'_3	0.930	0.920
p'	p'_{13}	0.920	0.970
p'_1	p'_{17}	0.890	0.200
p'_1	p'_2	0.880	0.950
p'_2	p'_{14}	1.000	0.880
p'_3	p'_{12}	0.870	0.930
p'_3	p'_9	0.940	0.300
p'_4	p'_5	0.860	0.800
p'_4	p'_6	1.000	0.800
p'_4	p'_7	0.920	0.800
p'_4	p'_8	0.870	0.800
p'_5	p'_6	0.990	0.970
p'_5	p'_7	0.850	0.580
p'_5	p'_8	0.970	0.670
p'_6	p'_7	0.980	0.690
p'_6	p'_8	0.880	0.460
p'_6	p'_{10}	1.000	0.770
p'_6	p'_{15}	0.930	0.770
p'_7	p'_{11}	0.960	0.890
p'_8	p'_{15}	0.950	0.970
p'_8	p'_{10}	0.850	0.970
p'_9	p'_{12}	0.890	0.970
p'_{10}	p'_{11}	0.870	0.440
p'_{10}	p'_{15}	0.910	0.380
p'_{12}	p'_{13}	0.790	0.920
p'_{12}	p'_{16}	0.760	0.870
p'_{13}	p'_{16}	1.000	0.850
p'_{14}	p'_{17}	0.970	0.780
p'_{14}	p'_{18}	0.890	0.910
p'_{17}	p'_{18}	0.930	0.710

Proteins		w	c
p''	p''_1	0.980	0.940
p''	p''_2	0.960	0.920
p''	p''_3	0.840	0.780
p''	p''_4	0.890	0.960
p''	p''_{13}	1.000	0.310
p''	p''_8	0.990	0.310
p''_1	p''_{14}	0.970	0.710
p''_2	p''_{14}	0.950	0.430
p''_3	p''_{12}	0.870	0.550
p''_4	p''_5	1.000	0.270
p''_4	p''_6	0.840	0.790
p''_4	p''_7	1.000	0.580
p''_5	p''_6	0.930	0.710
p''_5	p''_7	0.970	0.830
p''_5	p''_{10}	0.880	0.530
p''_5	p''_{11}	0.820	0.770
p''_6	p''_{10}	0.760	0.430
p''_7	p''_{15}	0.950	0.250
p''_7	p''_{11}	0.930	0.250
p''_8	p''_{12}	1.000	0.730
p''_{11}	p''_{15}	0.950	0.730
p''_{12}	p''_{13}	0.780	0.730

(a)



(b)

p′-p″ (f₀ = 0.850)

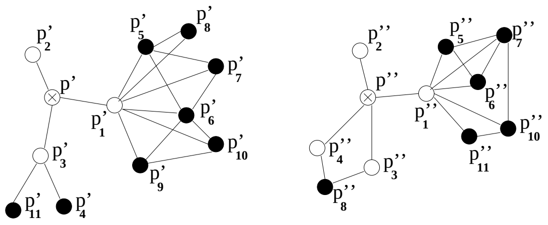
i	$\mathbf{p'_h}$	$\mathbf{p''_k}$	$\mathbf{f_{hk}}$	$\frac{\phi(X')}{\Theta(X')}$	$\sum_{p\gamma\in\Gamma} C_\gamma$	$\mathbf{f_p}$
1	p'_2	p''_2	0.050	0.104	0.310	0.476
	p'_3	p''_3	0.010			
	p'_1	p''_1	0.100			
	p'_{13}	p''_{13}	0.070			
	p'_4	p''_4	0.250			
2	p'_{14}	p''_{14}	0.120	0.376	2.047	0.434
	p'_{12}	p''_{12}	0.150			
	p'_6	p''_6	0.600			
	p'_7	p''_7	0.350			
	p'_5	p''_5	0.550			
3	p'_{10}	p''_{10}	0.600	0.533	0.138	0.457
	p'_{11}	p''_{11}	0.550			
	p'_{15}	p''_{15}	0.350			

(c)

Fig. 5.3. Example 2: decreasing of the initial similarity value.

Proteins		**w**	**c**
p'	p'_1	1.000	1.000
p'	p'_2	0.870	0.990
p'	p'_3	0.920	0.850
p'_1	p'_5	0.990	0.450
p'_1	p'_6	0.930	0.390
p'_1	p'_7	0.780	0.790
p'_1	p'_8	0.880	0.650
p'_1	p'_9	0.890	0.780
p'_1	p'_{10}	0.670	0.780
p'_3	p'_4	1.000	0.430
p'_3	p'_{11}	0.730	0.430
p'_5	p'_6	0.930	0.710
p'_5	p'_7	0.970	0.580
p'_5	p'_8	0.690	0.180
p'_6	p'_7	0.960	0.570
p'_6	p'_9	1.000	0.830
p'_6	p'_{10}	0.860	0.250
p'_9	p'_{10}	0.920	0.330

		w	c
p''	p''_1	0.990	0.890
p''	p''_2	0.870	0.630
p''	p''_3	1.000	0.980
p''	p''_4	0.950	0.580
p''_1	p''_5	0.920	0.780
p''_1	p''_6	0.880	0.630
p''_1	p''_7	0.970	0.570
p''_1	p''_{10}	0.890	0.630
p''_1	p''_{11}	1.000	0.570
p''_3	p''_8	0.780	0.720
p''_4	p''_8	0.910	0.280
p''_5	p''_6	0.900	0.350
p''_5	p''_7	0.940	0.810
p''_6	p''_7	0.980	0.280
p''_7	p''_{10}	0.860	0.210
p''_{10}	p''_{11}	1.000	0.390



(a)

p′-p″ ($\mathbf{f_0}$ = 0.710)

i	$\mathbf{p'_h}$	$\mathbf{p''_k}$	$\mathbf{f_{hk}}$	$\frac{\phi(X')}{\Theta(X')}$	$\sum_{p_\gamma \in \Gamma} C_\gamma$	$\mathbf{f_p}$
1	p'_2	p''_2	0.520	0.698	0.580	0.694
	p'_3	p''_3	0.535			
	p'_1	p''_1	0.980			
2	p'_5	p''_5	0.990	0.806	1.146	0.720
	p'_7	p''_7	0.920			
	p'_{10}	p''_{10}	0.880			
	p'_{11}	p''_{11}	0.940			
	p'_6	p''_6	0.970			
	p'_8	p''_8	0.350			

(b)

Fig. 5.4. Example 3: a final comprehensive example.

To summarize, Example 1 shows that if two proteins have a relatively low f_0 but very similar neighborhoods, then the final computed f_p is significantly larger than f_0. This confirms that Bi-Grappin is able to detect proteins with high similar interactors, thus possibly involved in common biological processes. Example 2 highlights that Bi-Grappin is also able to discern proteins that, even if characterized by high sequence similarity, have dissimilar interactors, and then they probably play different functional roles in the two organisms. This may be due, for example, to large changes caused by evolutive processes. Finally, Example 3 points out that the analysis of neighborhoods farther than the first one is necessary, in order to obtain a correct measure of functional similarity.

5.3 Related Work

The approaches which are most related to Bi-Grappin are reported in [14, 184, 185] and have been discussed in detail in Chapter 3.

If we consider the approach by Bandyopadhyay et al. [14], we will show in Section 5.4 that, differently from this technique, Bi-Grappin can be exploited not only to decide about functional orthology when sequence similarity may fail, but also to study proteins that are not yet well characterized in some species.

Moreover, as for the other two approaches proposed in [184, 185], unlike Bi-Grappin, quantitative information is not taken into account. Furthermore, the purpose of Bi-Grappin is that of refining protein similarities through neighborhoods exploration and differs from these two approaches that deal with the problem of global network alignment. Moreover, the exploitation of bipartite graph weighted matching as reported in [184] is quite different from that of Bi-Grappin. In fact, in [184] the bipartite graph weighted matching is only used for the final alignment of the two networks, whereas Bi-Grappin uses it step-wise on pairs of neighborhoods as the main computation task.

Other approaches [97, 182, 71] are more loosely related with Bi-Grappin. Similarly to these approaches, Bi-Grappin looks at conservation across PPI networks but, differently from them, it aims at singling out functional similarities between pairs of proteins, rather than focusing on the extraction of similar protein subnetworks.

Finally, Bi-Grappin is able to incorporate both quantitative and reliability information in its analysis, that is not simultaneously exploited in [14, 184, 185, 97, 182, 71].

5.4 Experimental Validation

To validate our approach, we tested it on the *S. cerevisiae* (yeast), *D. melanogaster* (fly) and *C. elegans* (worm) PPI networks. This evaluation was meant to study the ability of Bi-Grappin in individuating functional orthologs and to compare our results with those presented in [14, 185]. As will be illustrated in Section 5.4.1, the experimental results proved the effectiveness of our approach. In a second phase, we aligned the yeast network with those of the fly and the worm, respectively, and analyzed the most interesting results obtained by the alignments, as discussed in Section 5.4.2. We downloaded the interaction data for the three considered organisms from the DIP database [175][3]. To date, no explicit information about strength or reliability of interactions is available in DIP. Thus, in our experiments, we set $w = 1$ and $c = 1$ for all edge labels. The function $\mathcal{F}$ (introduced in Definition 5.2) was simply chosen to be the identity function. Following the recommendation in Section 5.2.1, i_{MAX} was set to 2 and, finally, α was set to 0.6.

In order to evaluate protein-protein sequence similarities needed to construct the SSD, we exploited the *Blast 2 sequences* algorithm [202][4] to align protein se-

[3] http://dip.doe-mbi. ucla.edu

[4] ftp://ftp.ncbi.nlm. nih.gov/blast/executables

quences, and referred to BLAST E-value parameter to measure the sequence similarity of pairs of proteins. In particular, after aligning two proteins p' and p'' of two different organisms, we computed the sequence similarity function f_0 according to the following transformation:

$$f_0 = \begin{cases} 0, & \text{if } E \geq 10^{-2} \\ 2^{\frac{20}{\log E}}, & \text{if } E < 10^{-2} \end{cases}$$

where E is the BLAST E-value on input p' and p''.

Note that the E-value may assume, in general, values greater than 1, and the lower it is, the more similar the protein sequences are. The formula reported above serves the purpose of both normalizing the sequence similarity function, to obtain a similarity value in the range $[0, 1]$, and obtaining a significant variations when the E-value reaches very small values (corresponding to very similar sequences).

The algorithm was implemented on a Pentium 4, 3.4 GHz with 4 GB of memory. The resulting running times were about 23 minutes for yeast and fly networks comparison, about 4 minutes for yeast and worm ones.

5.4.1 Functional Orthologs Detection

In this section, we discuss a set of experiments showing Bi-Grappin to be effective in detecting functional orthologs, that are, proteins codified by orthologs (i.e. genes in different species that originate from a single ancestor gene) performing the same function in two or more species [171, 14]. As pointed out in [14], the analysis of protein interactions can help in eliminating ambiguity where sequence similarity is not sufficient. In particular, the approach presented in [14] proves that it is possible to resolve ambiguous functional orthology relationships in the yeast and fly PPI networks. In [185], functional orthology detection was investigated for PPI networks of five different organisms.

We tested our method on two pairs of networks, that are, the yeast and the fly, and the yeast and the worm ones, respectively. We compared our results with those reported in [14] for the yeast and the fly correspondences, and also with those in [185] for both alignments. Table 5.1 shows the results obtained for the yeast and fly networks. We considered the yeast and fly pairs of proteins for which sequence similarity is not decisive to detect functional orthology as reported in the supplemental material of [14] [5]. Within the networks, we chose those protein clusters where the sequence similarity is sufficiently high, because in such cases the discrimination may be considered more significant. Furthermore, we discarded those clusters where some of the component proteins have no interactions, since Bi-Grappin works on connected networks. Therefore, we focused on the Inparanoid clusters [149], containing ambiguous functional orthologs, which are reported in the first column of Table 5.1. The second column contains the similarity values returned by our algorithm for each pair of proteins (reported in values between 0 and 100, to better appreciate

[5] http://www.cellcircuits.org/Bandyopadhyay 2006/

Yeast/Fly proteins	**Bi-Grappin** ($f_p \cdot 100$)	**Bandyopadhyay et al.**	**Singh et al.**
ssa2 – Hsc70-4	50.00 **(bs)**	53.22% **(bs)**	*out of cluster*
ssa1 – Hsc70-4	50.00 **(bs)**	48.10%	
Cmd1 – Cam	47.73	35.90%	*out of cluster*
Cmd1 – And	48.18 **(bs)**	44.39% **(bs)**	
Act1 – Act5c	58.32	39.56%	*out of cluster*
Act1 – Act42a	58.93 **(bs)**	39.24%	
Act1 – Act87e	51.68	43.53% **(bs)**	
Act1 – CG10067	51.89	38.20%	
Act1 – Act88f	55.07	40.17%	
kap104 – Trn	55.45 **(bs)**	40.64%	*out of cluster*
kap104 – CG8219	42.43	46.78% **(bs)**	
Hsp82 – Hsp83	57.23 **(bs)**	52.43%**(bs)**	*out of cluster*
Hsc82 – Hsp83	57.10	46.52%	
Myo4 – Didum	54.13 **(bs)**	37.06%**(bs)**	*out of cluster*
Myo2 – Didum	54.12	36.81%	
Gsy1 – CG6904	50.00 **(bs)**	48.97%**(bs)**	*out of cluster*
Gsy2 – CG6904	50.00 **(bs)**	38.13%	
Vph1 – CG18617	50.00 **(bs)**	41.87%**(bs)**	*out of cluster*
Stv1 – CG18617	50.00 **(bs)**	38.44%	
Rpt4 – Rpt4	58.97 **(bs)**	38.02%	*out of cluster*
Rpt4 – CG7257	56.45	44.43%**(bs)**	
Glc7 – Pp1-87b	50.00 **(bs)**	38.61% **(bs)**	*out of cluster*
Glc7 – Pp1α-96a	50.00 **(bs)**	37.31%	
Glc7 – Flw	50.00 **(bs)**	37.30%	
Rts1 – Pp2a-b'	50.00	56.83% **(bs)**	*out of cluster*
Rts1 – Wdb	55.54 **(bs)**	41.00%	
Pph22 – Mts	51.54 **(bs)**	49.68% **(bs)**	*out of cluster*
Pph21 – Mts	51.41	46.53%	
Tdh2 – Gapdh2	50.00	46.09% **(bs)**	
Tdh3 – Gapdh2	57.34 **(bs)**	38.08%	**(bs)**
Aac1 – Sesb	40.26	41.39%	
Aac3 – Sesb	41.19 **(bs)**	46.52% **(bs)**	**(bs)**
Aac1 – Ant2	39.64	41.43%	
Aac3 – Ant2	40.93	46.52% **(bs)**	**(bs)**
Utr1 – CG6145	47.80	63.07%**(bs)**	**(bs)**
YEL041W – CG6145	42.50	57.20%	
Utr1 – CG33156	49.30 **(bs)**	50.11%	**(bs)**
YEL041W – CG33156	49.21	48.60%	
YBR241C – Glut1	34.27 **(bs)**	55.18%	*out of cluster*
YBR241C – Sut1	29.92	60.18% **(bs)**	
Pre5 – Prosα6t	39.56	39.74%	**(bs)**
Pre5 – Pros35	49.58 **(bs)**	49.68% **(bs)**	
Cam1 – Ef1γ	36.41	44.02% **(bs)**	*out of cluster*
Tef4 – Ef1γ	42.35 **(bs)**	39.53%	
Clb3 – Cycb	35.21	36.90%	*out of cluster*
Clb5 – Cycb	33.46	36.53%	
Clb1 – Cycb	34.02	37.06%	
Clb2 – Cycb	35.70 **(bs)**	40.23% **(bs)**	
Clb4 – Cycb	35.04	37.00%	
Skp1 – Skpa	45.93 **(bs)**	38.68% **(bs)**	
Skp1 – CG12227	31.61	38.40%	**(bs)**
Skp1 – Skpc	29.34	36.35%	**(bs)**
Rps26a – Rps26	35.57 **(bs)**	40.32%	*out of cluster*
Rps26b – Rps26	35.57 **(bs)**	40.48% **(bs)**	
Cdc33 – Eif-4e	35.14 **(bs)**	39.12%	**(bs)**
Cdc33 – CG8023	33.01	39.65% **(bs)**	**(bs)**
Sso1 – CG31136	30.15	54.94%	**(bs)**
Sso2 – CG31136	31.00 **(bs)**	56.83% **(bs)**	**(bs)**
Egd2 – CG4415	18.57	43.99%	*out of cluster*
Egd2 – Nacα	26.85 **(bs)**	50.41% **(bs)**	
Rpp1a – Rplp1	37.23 **(bs)**	50.50% **(bs)**	*out of cluster*
Rpp1b – Rplp1	32.31	46.71%	
Cof1 – Tsr	37.23 **(bs)**	43.74% **(bs)**	*out of cluster*
Cof1 – CG6873	32.31	42.78%	

Table 5.1. Functional orthologs detection in *yeast* and *fly* networks.

the differences with [14]), whereas the third column reports the plausibility values returned by Bandyopadhyay et al. [14]. The symbol (*bs*) is used to indicate the best scoring pair. The last column contains the best scoring pairs according to [185]. In this respect, note that the purpose of the approach discussed in [185] is the global alignment of two or more input networks. Thus, it is not always the case that proteins recognized as functional orthologs by [185] corresponds to proteins in the same Inparanoid cluster. In these cases, a direct comparison between our method and that of [185] is not possible, and we referred as "out of cluster" the results corresponding to such cases.

Note that, our analysis agrees in most cases with either of [14] or [185] (20 out of 26 analyzed cases), and Bi-Grappin is able to discriminate functional orthologs in 21 out of 26 analyzed cases.

Table 5.2 shows the comparison between Bi-Grappin and [185] for the functional orthologs detection in the yeast and worm networks. Again, the first column shows the Inparanoid clusters, and the second and third columns illustrate the best scoring pairs according to our approach and that discussed in [185], respectively. Note that, in this case, Bi-Grappin always agrees with [185], whenever the two approaches are comparable but, notably and differently from [185], Bi-Grappin is always successful in discriminating among different protein pairs.

Yeast/c. elegans proteins	**BI-GRAPPIN** ($f_p \cdot 100$)	**Singh et al.**
RPL11A – T22F3.4	41.57 **(bs)**	**(bs)**
RPL11A – F07D10.1	41.53	
GSY1 – Y46D5A.31	55.23 **(bs)**	*out of cluster*
GSY2 – Y46D5A.31	50.00	
GSP1 – K01G5.4	51.10 **(bs)**	*out of cluster*
GSP2 – K01G5.4	50.68	
NPL4 – F59e12.5	41.94	**(bs)**
NPL4 – F59e12.4	50.02 **(bs)**	**(bs)**
BMH1 – M117.2	49.26 **(bs)**	**(bs)**
BMH1 – F52D10.3	42.02	
BMH2 – M117.2	47.73	**(bs)**
BMH2 – F52D10.3	42.07	
Aac1 – T27E9.1	40.42	*out of cluster*
Pet9 – T27E9.1	41.49 **(bs)**	
Aac3 – T27E9.1	41.21	
Cdc33 – B0348.6	33.42 **(bs)**	**(bs)**
Cdc33 – F53A2.6	32.29	**(bs)**
Cdc33 – R04A9.4	31.50	**(bs)**
GSY1 – Y46G5A.31	55.23 **(bs)**	*out of cluster*
GSY2 – Y46G5A.31	50.00	
YPL048W – F17C11.9	36.03 **(bs)**	*out of cluster*
YKL081W – F17C11.9	35.86	
Clb3 – T06E6.2	33.91 **(bs)**	*out of cluster*
Clb4 – T06E6.2	33.14	
Clb1 – T06E6.2	28.78	
Sso1 – F56A8.7	30.09	**(bs)**
Sso2 – F56A8.7	30.69 **(bs)**	**(bs)**
Rpp1a – Y37E3.7	20.68 **(bs)**	*out of cluster*
Rpp1b – Y37E3.7	14.54	

Table 5.2. Functional orthologs detection in *yeast* and *worm* networks.

5.4.2 Common Processes Detection

As a further set of experiments, we aligned the *S. cerevisiae* network with the *D. melanogaster* and the *C. elegans* ones, in order to individuate proteins involved in common biological processes. It is worth pointing out that the latter condition is different from functional orthology discussed in Section 5.4.1. Indeed, two proteins are recognized to be functional orthologs if, as already explained, they derive from orthologs and perform the same function in different organisms. On the other hand, proteins which are not necessarily functional orthologs might be anyway involved in common biological processes and it is known that commonalities between involved sets of interactors witness for this to hold.

We first discuss the pairs of proteins illustrated in Table 5.3, that are those scoring the highest refined similarity as computed by Bi-Grappin. We identify proteins by name, providing also the *SWISSPROT id* when the name may be ambiguous. The first two columns of Table 5.3 show the pairs of proteins corresponding to the first ten best scores for the *S. cerevisiae* and *D. melanogaster* networks, pointing out that Bi-Grappin is able to correctly pair proteins with similar functions. In fact:

- proteins PP2A are phosphatases involved in signal transduction;
- Actin, Actin42A and Actin5c are cytoskeleton constituents;
- alpha- and beta- PDHE1 are components of Pyruvate dehydrogenase complex;
- RPT4 are components of the proteasome;
- alpha Importin and CSE1 are involved in nuclear export
- Hsc70 are homologs to heat shock proteins.

In the third and four columns of Table 5.3, proteins corresponding to the top ten best scores for the *S. cerevisiae* and *C. elegans* are reported. Likewise, proteins with homologous functions are properly paired:

- tubulins, which constitute microtubules in both species;
- PMS1 and PMS2, required for DNA mismatch repair;
- phosphotases (PP1) and kinases (PKC, P53739, Q18846), involved in signal transduction;
- proteins RFC, which are subunits of the replication factor required for the duplication of the DNA strands.

There are also proteins of unknown function (Q08726, O01426, Q9XW68), which are all able to bind ATP, and proteins involved in glycogen synthesis, named Gsy1.

A further interesting issue that merits discussion concerns the possibility for our technique to infer connections of not always well characterized proteins to specific biological processes, even when involved sequence similarities are not particularly significant. Figure 5.5 (a) illustrates some examples of protein pairs where the refined similarity is higher than the sequence similarity, since a significant neighborhood similarity has been retrieved. It is understood that such an increasing in similarity is supposedly correct, since the proteins under consideration are actually biologically related. Indeed:

S. cerevisiae	*D. melanogaster*	*S. cerevisiae*	*C. elegans*
Hsc70	Hsc70	ATP BP Q08726	ATP BP O01426
alfa-PDHE1	PDHE1 (Q9W4H6)	ATP BP Q08726	ATP BP Q9XW68
PP2A (P31383)	PP2A	beta-Tubulin	beta-Tubulin
RPT4	RPT4	PMS1 P14242	PMS2 Q9TVL8
Actin	Actin42A	alfa-Tubulin	beta-Tubulin
CSE1	CSE1	PP1 (P20604)	PP1 (Q9XW79)
beta-PDHE1	PDH (Q7K5K3)	RFC4	RFC2
Actin	Actin5c	Kinase P53739	Kinase Q18846
PP2A (P20604)	PP2A (P23696)	PKC	PKC
alfa-Importin	alfa-Importin	Gsy1	Gsy1

Table 5.3. Best score pairs of proteins in: *yeast* and *fly*; *yeast* and *worm*.

- Cyclin B1, Cyclin B4, MSA2 and Cyclin D are key switches of cell cycle progression in yeast and fly, respectively;
- Cnb1 is the calcineurin B, a regulatory calcium binding protein such as the protein P48593;
- Cofilin,twinfilin and Abp1 are all involved in the regulation of the actin cytoskeleton;
- PTP2 and PTP-ER are both tyrosine phosphatases;
- YPT11 and Rab-RP4 are both Rab like proteins regulated by GTP hydrolysis.

Comparing yeast to worm (Figure 5.5 (b)):

- Prr1 and Mak-2 are kinases downstream of the MAPK activation;
- Tap42 is involved in Tor signaling pathway and Q9N4E9 protein is similar to it;
- Cdc37 and its worm homolog are kinase regulators;
- Fcy1 and Cdd2 are both pyrimidine deaminases.

It is important to note that the worm proteosome is less characterized than the yeast one, and that for many of its gene products, functions (and sometimes names) have been assigned automatically on the basis of sequence homology.

We believe that our method can be much helpful for either confirming or not this predictions by neighborhood analysis. This is, for instance, the case for the protein Q9N4E9 which is similar to Tap42 but no other information are available; for Q21021, similar to the yeast Ran BP2; and for Q21746, which contains TPR repeats like the co-chaperone yeast CNS1. The O44175 protein is also probably involved in cell duplication as the yeast CTF18. Finally, note that the pairs reported in the last three rows of the table in Figure 5.5 (a) and the last row in Figure 5.5 (b) score a sequence similarity close to zero.

Overall, this confirms that Bi-Grappin is able to correctly reconstruct useful information from neighborhoods analysis (whenever available), that is not predictable from the sole sequence similarity.

However, it is worth pointing out that Bi-Grappin results strictly depend on the correctness and completeness of interaction data stored in databases, where false positive/negative may occur. Unfortunately, as already pointed out, available data are sometimes characterized by low reliability [9, 47, 193]. The example illustrated below shows how Bi-Grappin results improve when interaction information become more accurate. We considered proteins Rnp11 of yeast and of fly, having a sequence

S. cerevisiae	*D. melanogaster*	Sequence similarity	Refined similarity
Cyclin B4	Cyclin D	0.424	0.444
YPT11	Rab-Rp4	0.409	0.442
Cyclin B1	Cyclin D	0.350	0.431
Cofilin	Cofilin	0.389	0.420
Cnb1	P48593	0.327	0.419
PTP2	PTP-ER (Q9W2F3)	0.293	0.405
Twinfilin	Cofilin	0.006	0.314
MSA2	Cyclin D	0.004	0.326
Abp1	Cofilin	0.002	0.251

(a)

S. cerevisiae	*D. melanogaster*	Sequence similarity	Refined similarity
Tap42	Q9N4E9	0.371	0.439
Prr1	Mak-2 (Q9TZ16)	0.322	0.394
RanBP2	Q21021	0.268	0.359
Cdc37	Cdc37	0.306	0.356
Ctf18	O44175	0.336	0.346
CNS1	Q21746	0.291	0.343
Fcy1	Cdd2 (Q20628)	0.015	0.352

(b)

Fig. 5.5. Some interesting pairings in (a) *yeast* and *fly*; (b) *yeast* and *worm*.

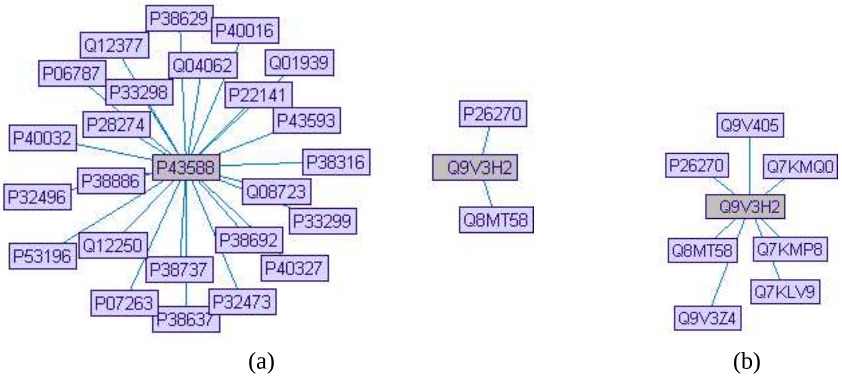


Fig. 5.6. *Yeast* and *fly Rnp11* 1-*neighborhood*: (a) available and (b) enriched.

similarity equal to 1, that is, the maximum possible value, which is decreased by neighborhood analysis as low as 0.685. Such a decrease is due to the fact that the yeast Rnp11 has 24 neighbors, whereas the fly one has only 2 neighbors (according to the data stored in the DIP database). Figure 5.6(a) illustrates such a situation. The figure has been drawn by using PIVOT [151], and *SWISSPROT ids* have been adopted as node labels to distinguish proteins. In particular, the yeast Rnp11 has *SWISSPROT id* equal to P43588, whereas that of the fly is Q9V3H2. We tried to complete the neighborhood of the fly Rnp11 with some missing data, referring to [77]. The neighborhood shown in Figure 5.6(b) was obtained, where the proteins that are not present in the DIP database as Rnp11 interactors have been added. By running Bi-Grappin on the so obtained new data network, we obtained a refined similarity of 0.777, that may be considered more correct than the rather smaller value 0.685

previously obtained, since both the yeast and fly Rnp11 proteins are indeed part of the well known proteasome complex performing ubiquitinated proteins degradation in both organisms.

5.5 Concluding Remarks

In this chapter, we dealt with the problem of searching similarities in PPI networks. In particular the aim of the proposed approach, called Bi-Grappin, is that of identifying functional similarities and detecting proteins involved in common biological processes.

The basic idea of Bi-Grappin is that proteins with similar neighborhoods are probably involved in similar biological processes, inducing a concept of similarity which is based on both sequence and network information. Indeed, the key step of Bi-Grappin is the refinement of protein sequence similarity by exploiting neighborhood similarities (i.e., similarity between interaction profiles). One of the peculiarities of Bi-Grappin is its capability of taking into account both quantitative (e.g., interaction strengths) and reliability information. The first is used to distinguish nodes belonging to different neighborhoods and the second one to weight the contributions of different interacting proteins in the refinement phase.

Experimental evaluations showed that our technique may be profitably exploited to detect functional orthologs when ambiguities may derive from the sole sequence similarity analysis, and also to correctly associate proteins involved in the same biological processes. Thus, we argue that Bi-Grappin can be regarded as a powerful tool to analyze PPI networks, whose already satisfactory accuracy will be further improved by the future availability of more complete and precise data about protein interactions.

In the next part of the thesis, involving Chapter 6 and Chapter 7, the problem of protein-protein interaction networks alignment will be faced. In particular, in Chapter 6 the state of the art about protein-protein interaction networks alignment will be outlined.

Part III

Network Alignment

6

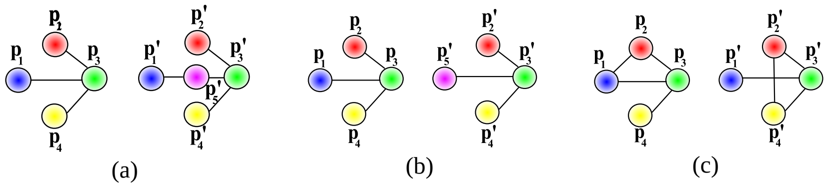
Network Alignment Techniques: an Overview

Summary. In this chapter, the state of the art about protein-protein interaction network alignment will be outlined. Firstly, in Section 6.1 the definition of the problem is provided. Then, in Section 6.2, an overview of the techniques proposed to align PPI Networks is reported. In particular, Section 6.2.1 discusses about local network alignment techniques while 6.2.2 introduces global network alignment techniques. Section 6.3 reports an overall comparison of the discussed methods and, finally, in Section 6.4 some conclusions are drawn.

6.1 PPI Network Alignment

Network alignment is the process of globally comparing two or more networks of the same type belonging to different species in order to identify similarity and dissimilarity regions. Network alignment is commonly applied to detect conserved subnetworks, which are likely to represent common functional modules. As already discussed in Chapter 2, the input of a network alignment algorithm are two (or, possibly more) biological networks of different organisms and the output are pairs (or, possible sets) of subgraphs (or, possibly simpler structures, such as paths), one for each input network, that have been recognized to be similar. For instance, the identification of conserved linear paths may lead to the discovery of signaling pathways, while conserved clusters of interactions (subgraphs) may correspond to protein complexes.

The word “conserved” means that the two (or more) identified subgraphs contain proteins performing similar functions and having similar interaction profiles. It is important to underline that the key word here is “similar” and not “identical”. In fact, the identified subgraphs often correspond to approximated rather than exact alignments. Approximation handling is needed for dealing with possible occurrences of evolution events modifying a network structure and also allows to suitably take into account the significant number of both false negative and false positive interactions found when looking up existing databases. Hence, different types of approximations should be taken into account: *(i) node insertions*, corresponding to the addition of nodes in one of the input networks (see, Figure 6.1(a)); *(ii) node mismatches*, corresponding to pairs of nodes characterized by a low similarity, but sharing similar



(a) (b) (c)

Fig. 6.1. (a) Node insertion; (b) node mismatch; (c) edge insertion.

biological characteristics (e.g., proteins performing the same function) (see, Figure 6.1(b)); and *(iii) edge insertions*, corresponding to the addition of interactions in one of the input networks (see, Figure 6.1(c)). Examples of evolution events that may affect protein interaction networks are gene duplication, that causes the addition of new nodes (proteins), and link dynamics, corresponding to gain or loss of interactions through mutations in proteins [20].

6.2 An Overview on PPI Network Alignment Techniques

As already pointed out in the previous section, the goal of network alignment approaches is to identify one or multiple possible mappings between the nodes of the input networks. Moreover, for each mapping, the set of conserved edges, corresponding to conserved interactions, have to be revealed. Mappings may be partial or complete and this distinction led to the definition of two classes of alignment algorithm:

- Local Network Alignment (LNA): comprises those algorithms that do not require that the identified mapping covers all the nodes in the input networks.
- Global Network Alignment (GNA): involves those algorithms that require that all the nodes of the input networks have to be involved.

LNA algorithms are intended for discovering similar motifs between two (or, possibly, more) networks, which may also lead, sometimes, to some inconsistencies to characterize discovered motifs. Indeed, a protein of one input network may correspond to different proteins of another input network if considering different matched subgraphs.

In GNA, instead, the goal is to find a single consistent mapping covering all nodes of the input networks. Thus, by solving the GNA problem some partial suboptimal mapping can be discarded in the light of a global alignment and all nodes have to be paired or explicitly marked as unpaired nodes.

In the two subsequent sections, the techniques belonging to LNA and GNA will be described. In particular, Section 6.2.1 focuses on LNA methods while Section 6.2.2 on GNA approaches.

6.2.1 Local Network Alignment Methods

The goal of LNA techniques is to find multiple, corresponding similar regions among the input networks. In this type of alignment, each partial mapping is independent from the others. Several local network alignment approaches have been proposed in the literature [97, 182, 112, 71, 70, 14]. The aim of this section is to survey on them.

PathBlast and NetworkBlast

PathBlast [97] is a procedure to align two PPI networks by combining interaction topology and protein sequence similarity, in order to identify conserved interaction pathways. The method searches for high scoring pathway alignments involving two paths, one for each network, in which the proteins of the first path are paired with putative homologs occurring in the same order in the second path. To this aim, this approach builds a network alignment graph where each node represents a pair of homologous protein (one for each input network) and each link between a pair of nodes represents a conserved protein interaction. To take into account possible errors in the available data and the role played by the evolution in network differences, *PathBlast* also allows for gaps and mismatches. A gap occurs when two corresponding pairs of proteins interact directly in one networks, and via a common protein in the other network (i.e., a node insertion). A mismatch occurs when two corresponding pairs of proteins interact via a protein in both networks and these proteins do not share relevant sequence similarity.

PathBlast has been extended into *NetworkBlast* in a subsequent work [182]. *NetworkBlast* is a tool for discovering conserved pathways and complexes across more than two PPI networks. Such an extension is based on the idea that each node of the alignment graph identifies a group of homologous proteins, instead of a mere pair of them. Moreover, this approach is able to search for both linear paths, corresponding to signal transduction pathways, and clusters of interactions, corresponding to protein complexes.

NetworkBlast, in its turn, has been subsequent extended to *NetworkBLAST-M* [92], which allows to identify protein complexes in protein-protein interaction networks based on a particular representation of the input networks that is linear in their size. *NetworkBLAST-M* is based on progressive alignments and avoids the explicit representation of every set of potentially orthologous proteins, thus gaining in efficiency.

Graemlin

Graemlin [71] is an algorithm for multiple network alignment and is meant to individuate conserved functional modules across species. This approach introduces a probabilistic formulation of the topology-matching problem. The method represents the input networks as weighted graphs in which the weights represent the interaction probabilities. The alignment produced by *Graemlin* is made of a set of subgraphs and a mapping between corresponding proteins.

It is important to note that, according to the algorithm formulation, the groups of aligned proteins are disjoint and must represent homologous groups that generally are proteins belonging to the same protein family. This observation leads to the definition of the alignment as a collection of protein families having conserved interactions. This way, it is possible to use evolutionary information to score the potential alignments.

To search for alignment between two input networks, *Graemlin* generates a set of seeds from each input network, where each seed is a set of close proteins, in order to cut the search space. Then, by enumerating the seeds between the two networks, it tries to transform each of them, in turn, into an high-scoring alignment. When applied to multiple networks, *Graemlin* uses a phylogenetic tree and successively aligns the closest pairs of networks. After each alignment, it obtains several new networks, each of which is placed as a parent of the two aligned networks. The method iterates this process until all the networks are at the root of the tree.

Graemlin has been extended with a novel scoring function, an algorithm that automatically learns the scoring function's parameters and an algorithm that uses the scoring function to globally align multiple networks giving birth to a GNA algorithm called *Graemlin 2.0* [70].

Bandyopadhyay

Bandyopadhyay et al. [14] proposed a strategy to identify functionally related proteins supplementing sequence-based comparisons with information on conserved protein-protein interactions. The idea is that the probability of functional orthology of a pair of proteins is influenced by the probability of functional orthology of their neighbor proteins.

This method first aligns two PPI networks using only sequence similarities, and in particular by Inparanoid clusters [149], for pairing the proteins of the two input networks. The result of the alignment is a graph where each node represents a pair of proteins and each edge is a conserved interaction. A state, indicating if the pairs of proteins are likely to identify a true functional orthology, is associated to each node. In particular, the protein pairs in each Inparanoid cluster having the lowest BLAST E-Value are marked as a true orthology and are said *strongly conserved*. Moreover, for each node of the alignment graph, a conservation index is defined. This index is a measure of the portion of strongly conserved interactions w.r.t the total number of interactions involving it.

Starting from the alignment graph, the approach performs probabilistic inference (based on Gibbs sampling) to identify pairs of proteins, one from each species, that are likely to feature the same function with the aim of resolving ambiguous functional orthologs in the Inparanoid clusters.

The approach has been specifically applied to resolve ambiguous functional orthology relationships in the *S. cerevisiae* and *D. melanogaster* PPI networks.

MaWISh

MaWISh [112] is a tool that implements a duplication divergence model to carry out pair-wise network alignment. In particular, this system merges pairwise interaction networks into a single alignment graph, formulates network alignment as a maximum weight induced subgraph problem and proposes several heuristics to solve it.

The duplication/divergence model is used to accurately identify and interpret conservation of interactions, complexes, and modules across species. Indeed, this model enables the introduction of the concept of match (conservation), mismatch (emergence or elimination) and duplication, which allow to discover alignments that take into account conjectures about the structure of the network in the common ancestor. These observations led to the possibility of also discovering indirect interactions.

A similarity score between two protein pairs is defined to take into account matches, mismatches and duplications. This allows to translate the problem of distinguishing orthologs[1] and in-paralogs[2] from out-paralogs[3] into an optimization problem that accounts for the trade-off between conservation of sequences and interactions.

QSim

QSim [64] is a tool proposed to align two protein-protein interaction networks obtained by an adaptation of an existing algorithm for network simulation. The peculiarity of this tool is that of performing an asymmetric search in the sense that it searches for local matches of one network into another. The approach is based on the same idea exploited by Bi-Grappin (see Chapter 5) according to which two proteins are similar if both they share a significant sequence similarity and their neighborhoods are similar.

QSim starts by computing an initial similarity value for each pair of proteins (the first protein belonging to the first network and the second one to the second network) based on the Inparanoid clusters and the BLAST E-values. In particular, a similarity value of 1 is assigned to protein pairs belonging to the same cluster, a value computed exploiting the BLAST E-value otherwise. As a second step, *QSim* refines the initial similarities by estimating the similarity of protein neighborhoods.

In more detail, *QSim* proceeds iteratively, computing a series of refinements, until the estimates converge to a unique global optimum. As compared to existing approaches, the peculiarity of *QSim* is that the alignment is asymmetrical in the sense that it internally exploits an asymmetric graph matching procedure.

Ali & Deane

Ali and Deane proposed a method [2] to align protein-protein interaction networks, which also exploits a protein functional similarity measure with the aim of detecting functional modules. The authors observed that the limitations of existing approaches

[1] homologous proteins of different species

[2] proteins that derive from an ancestral duplication and do not form orthologous relationships

[3] proteins that derive from a lineage-specific duplication, giving rise to co-orthologous relationships

for PPI networks alignment may derive from the mere use of sequence information to identify protein orhtologous. Thus, their tool is based on a different measure of protein similarity that exploits also functional information and, in particular, protein GO annotations related to the biological process sub-ontology (for more detail about the GO see Chapter 3).

Four scores are assigned to each edge (representing an interaction) to take into account different contributions. The first two contributions are obtained by two alignments of the input networks according to sequence and functional similarity, respectively. The two alignments provide, for each edge, two different alignment scores, one from the sequence based alignment and the other from the function based alignment. The third score is a graph based score computed by mixing a cluster coefficient, that is a local network measure of how close a node and its neighbors are to being a clique, and a normalized edge betweenness value, which takes into account, for each edge, the number of shortest paths between its ends. Finally, the fourth score encompasses the information obtained from co-expression data. These four scores are combined to obtain a single edge weight.

Starting from the so built graphs, the algorithm extracts a set of modules that potentially correspond to functional modules.

Dutkowski & Tiuryn

Dutkowski and Tiuryn proposed an approach [54] for protein-protein network alignment, based on the reconstruction of an ancestral PPI network. The alignment algorithm is based on the phylogenetic history of proteins and a stochastic evolutionary model of interaction emergence, loss and conservation.

The first stage of the approach is the reconstruction of the conserved ancestral PPI (CAPPI) network. In more detail, it concerns the reconstruction of the hypothetical sequence of evolutionary events (duplications, deletions and speciations), by which the proteins of the input PPI networks evolved from their counterparts in the common ancestral network.

In the second step, the posterior probabilities of interaction between proteins at each stage of evolution is determined. The probability of protein interaction is calculated under a proposed stochastic model of network evolution. The topology of the ancestral network (and each network at every stage of evolution) is determined by the most probable interactions. Finally, conserved ancestral interactions in the CAPPI network are identified and they are projected back onto the input networks to determine the alignment.

Domain

Domain [78] is a tool for domain-oriented alignment of protein-protein interaction networks. It follows an alternative direct-edge-alignment paradigm. According to this paradigm, the peculiarity of *Domain* is that it does not explicitly identify homologous proteins, but directly aligns protein-protein interactions (PPIs) across species by decomposing them in terms of their constituent domain-domain interactions (DDIs) and by looking for conservation of these DDIs.

In more detail, *Domain* consists of three stages. The first stage is the construction of a complete set of alignable pairs of edges (APEs). A pair of edges is said to be alignable if there exists a DDI that can plausibly mediate the two associated PPIs. A DDI is said to plausibly mediate a PPI if the corresponding interaction probability between the two domains is above some fixed threshold.

The second stage is the building of an APE graph. The APE graph is an undirected weighted graph, where nodes correspond to the identified APEs, and edges correspond to one of the following four evolutionary relationships: alignment extension, node duplication, edge indel and edge jump.

Finally, the last step is the exploitation of a heuristic search to identify high-scoring non-redundant subgraphs from the resultant APE graph.

HopeMap

HopeMap [204] is an iterative connected-components-based algorithm with linear cost for pairwise network alignment. This tool is focused on the fast identification of maximal conserved patterns across species.

HopeMap is based on the observation that the number of true homologous across species is relatively small compared to the total number of proteins in all species. Thus, *HopeMap* starts by picking up highly homologous groups and, then, it searches for maximal conserved interaction patterns according to a generic scoring schema. Finally, it validates the results across multiple known functional annotations. In particular, the results are evaluated in terms of statistical enrichment of Gene Ontology (GO) terms and KEGG ortholog groups (KO) within conserved interaction patters.

In more detail, *HopeMap* consists in five steps. The first step is a initial stage in which the data obtained from PPI network databases are preprocessed. In the second step, *HopeMap* uses homologous clustering to identify homologous groups and, thus, to find highly similar protein sequences across the species under consideration. In the third step, the tool uses the clustering results to build a network alignment graph, where nodes represent sets of proteins and edges represent conserved protein-protein interactions. In the fourth step, the network alignment graph is searched for the strongly connected-components (clusters) which are ranked by combining genomic similarity scores, interaction conservation, and functional coherence. At the end, in the fifth step, the functional coherence of the discovered homologous groups is evaluated in each species using the Gene Ontology (GO). After the fourth step, the local alignment procedure can be iteratively applied to improve the cluster scores, if necessary.

6.2.2 Global Alignment Methods

The aim of global network alignment is to find the best overall alignment between the input networks. This implies that all the nodes of the input networks must be covered by the mapping. Therefore, each node has either to be matched to some node or explicitly marked as a node insertion.

The GNA problem has received less attention than the LNA one in the last years. However, some global network alignment approaches have been proposed in the literature [185, 92, 124]. The aim of this section is to survey on them.

IsoRank

IsoRank [184] is an algorithm for pairwise global alignment of PPI networks aiming at finding a correspondence between nodes and edges of the input networks that maximizes the overall match between the two networks.

IsoRank works in two stages. In the first stage it associates a score with each possible match between nodes of the two networks. In the second one, it constructs the mapping for the global network alignment by extracting mutually-consistent matches according to a bipartite graph weighted matching performed on the two entire networks.

IsoRank has been extended to an approach for multiple network alignment, called *IsoRank-M* [185]. *IsoRank-M* is based on the exploitation of an approximate multipartite graph weighted matching. *IsoRank-M* has been subsequently extended to *IsoRankN* (IsoRank-Nibble) [124]. *IsoRankN* is a global multiple-network alignment tool, which relies on spectral clustering on the induced graph of pairwise alignment scores. Being based on spectral methods, IsoRankN is both error tolerant and computationally efficient.

Zaslavskiy et al.

Zaslavskiy et al. proposed an approach [234] to globally align protein-protein interaction networks by reformulating the PPI alignment problem as a graph matching problem.

Two types of problems have been considered in this work. The first problem considers strict constraints on the sequence similarity of matching proteins while the second one aims at finding an optimal compromise between sequence similarity and interaction conservation in the alignment. In particular, the authors investigate the use of modern state-of-the-art exact and approximate methods to solve the graph matching problem representing the GNA problem.

In more detail, the authors consider two possible formulations: the *Constrained GNA* where some constraints (e.g., edge weights) are provided, and the *Balanced GNA* where the aim is to automatically balance the matching of similar vertices with the conservation of interactions. Several algorithms to solve the above mentioned problems are considered and, in particular two algorithms for the first problem and three algorithms for the second one are discussed.

6.3 Discussion

In this section, an overall comparison of the techniques that have been presented in this chapter is provided. In general, network alignment algorithms may be classified along the two directions:

1. local versus global alignment;
2. pairwise versus multiple networks alignment.

In Table 6.3, the methods discussed in this chapter are compared with respect to the above mentioned directions, suggesting some observations.

The first observation is that the LNA problem has received more attention in the literature than the GNA one, indeed 11 of the 16 discussed methods concern LNA and only 5 have been proposed to solve the GNA. Moreover, the GNA techniques are more recent than the LNA ones, suggesting that the GNA problem has became relevant only in the last few years.

As for parwise vs. multiple network alignment techniques, both problem have received great attention in the literature. However, the pairwise alignment, that is the most simple one, was the first to be investigated. Then, in the last few years, with techniques becoming more efficient, the multiple network alignment problem has been receiving an increasing attention.

Tool	local	global	pairwise	multiple
PathBlast [97]	x		x	
NetworkBlast [182]	x			x
NetworkBlast-M [92]	x			x
Graemlin [71]	x			x
Graemlin 2.0 [70]		x		x
Bandyopadhyay [14]	x			x
MaWISh [112]	x		x	
QSim [64]	x		x	
Ali & Deane [2]	x		x	
Dutkowski & Tiuryn [54]	x			x
Domain [78]	x		x	
HopeMap [204]	x		x	
IsoRank [184]		x	x	
IsoRank-M [185]		x		x
IsoRankN [124]		x		x
Zaslavskiy et al. [234]		x	x	

Table 6.1. Overall comparison of the PPI network alignment methods.

Summarizing, alignment of protein-protein interaction networks went through three major generations. In the first generation, the pairwise alignment, conserved pathways/complexes between two species were indentified (e.g., *PathBlast* [97]). The second generation concerns the multiple alignment, in which tools such as *NetworkBlast* [182], aiming at aligning multiple networks, have been proposed. The tools belonging to the two first generations concern LNA, since their algorithms, searching for conserved regions, start from small local regions and then greedily expand. The third and last generation of alignment tools regards the GNA problem and has produced several methods, such as IsoRank [184].

6.4 Concluding Remarks

In this chapter an overview on the techniques proposed to align protein-protein interaction networks has been provided. This investigation has been useful to identify missing requirements in current PPI network alignment solutions and open paths of research in this context. This analysis has been also helpful to understand the collocation of the technique proposed in the next chapter in the PPI network alignment techniques landscape.

7

Sub-Grappin: Extracting Similar Subgraphs across PPI Networks

Summary. This chapter describes a novel method for discovering similar subgraphs, possibly representing similar functional modules, across the PPI networks of two different species. In particular, in Section 7.1, some background on protein-protein interaction network alignment is provided; in Section 7.2, some basic concepts useful to understand the proposed approach are defined. In Section 7.3, Bi-Grappin is briefly recalled. Moreover, here Sub-Grappin is described in detail along with an example showing how the method works. Section 7.4 provides a comparison with the main techniques proposed to align biological networks and discussed in Chapter 6. In Section 7.5, the experimental evaluations carried out to test Sub-Grappin are described and discussed in detail. Finally, in Section 7.6, some conclusions are drawn.

7.1 Introduction

One of the big challenges in computational biology is to understand how evolution influences the variation of functional components across species. In this context, studying how proteins interact inside the cell is necessary to understand several biological processes [211], and the analysis and comparison of protein-protein interaction networks associated to different organisms is becoming a key issue thereof. Discovering similar sub-networks in PPI networks of different organisms is useful both to uncover complex mechanisms at the basis of evolutionary conservations and to infer the biological meaning of groups of interacting proteins belonging to not yet well characterized organisms. As a result, a number of approaches have been recently presented in the literature for local [144, 112, 71, 182, 92] and global [184, 124] alignment of PPI networks. Since PPI networks are large, computationally demanding methods, such as those based on exact subgraph isomorphism checking [75], cannot be applied on real interaction networks. Moreover, due to the nature of high-throughput experimental techniques [89, 114] and computational methods [140, 211] often exploited to discover new protein interactions, stored information about interactions is not always completely reliable [193], as also testified by several studies [9, 47]. This may potentially affect any attempt to extract useful information from them.

In this chapter a technique, called Sub-Grappin, designed to extract conserved subgraphs across PPI networks, is presented. Bi-Grappin, presented in Chapter 5, based on computing the maximum weighted matching of certain bipartite graphs, is exploited to assess protein similarities according to both protein sequence and network structure similarities. Conserved subgraph extraction is then carried out by performing a node collapsing based technique (referred to as Collapse) which is the main topic of this chapter.

The basic intuition underlying the development accounted for in this chapter is as follows. Since Bi-Grappin is effective in discovering significant functional similarities among single proteins, then it is sensible to devise a technique based on collapsing subgraphs into nodes, by which the basic Bi-Grappin can be exploited to discover highly matching subgraphs in PPI networks as well. To summarize, while Bi-Grappin is used to characterize the functional orthology between pairs of proteins according to sequence and neighborhoods analysis (see also [184]), Sub-Grappin is a local search and collapse based technique which, by exploiting Bi-Grappin and Collapse, extracts similar protein modules in two input PPI networks.

Sub-Grappin works as follows: it takes in input two PPI-networks and additional information about similarities between their proteins, then it interleaves *(i)* a call to Bi-Grappin, by which node similarities are refined, and *(ii)* a call to Collapse, for collapsing subsets of nodes according to maximum similarities. This process is iterated until a fixed threshold on the similarity between pairs of collapsed nodes is reached, whereby highly matching subgraphs are recognized.

We point out that while exploiting Bi-Grappin as a submodule, Sub-Grappin has a rather different purpose. In fact, while the former algorithm only highlights functional similarities between pairs of proteins belonging to different networks, the technique presented here serves the purpose of extracting similar subgraphs across PPI-networks.

Differently from other known techniques (e.g., [97]) our method is able to recognize similar sub-networks of arbitrary structure, and to take into account both protein sequence similarity and network topology similarity by agreeing, in this respect, with most of the recent approaches presented in the literature (e.g., [184, 70, 92]). Differently from previous techniques, however, our method also uses both quantitative and reliability information about interactions. This is significant since quantitative information can be used, for instance, to characterize groups of proteins interacting with high strength [120, 194]. On the other hand, reliability information is useful to avoid mistaking mismatches caused by false positive. In fact, interactions data obtained by different methods (e.g., experimental or high-throughput methods) may be weighted differently, thus automatically handling problems related to the possible dirtiness of PPI networks.

We tested Sub-Grappin on the PPI networks of *Homo sapiens* (human) and *Saccharomyces cerevisiae* (yeast). Experimental results showed its ability in discovering biologically relevant associations. Interaction data has been collected from the MINT database [33], that also supplies reliability information about stored data. In order to assess the quality of computed results, we introduced a new accuracy parameter based on both Gene Ontology (GO) [7] annotations and protein sequence similarities. Eventually, we compared Sub-Grappin with NetworkBlast-M [92], a recently proposed technique for extracting similar subgraphs from PPI networks. This comparison showed that Sub-Grappin is actually able to find more biologically meaningful conservations.

The remainder of this chapter is organized as follows. In Section 7.2, we define some basic concepts useful for understanding Sub-Grappin. In Section 7.3, Bi-Grappin is briefly recalled. Moreover, Collapse and Sub-Grappin are described in detail and an example is provided to show how the method works. Section 7.4 provides a comparison with the main techniques proposed to align biological networks, which have been discussed in detail in Chapter 6. In Section 7.5, the experimental evaluations we carried out to test Sub-Grappin are described and discussed in detail. Finally, in Section 7.6, some conclusions are drawn.

7.2 Preliminaries

As already introduced in Chapter 5, the most common representation for the protein-protein interaction network of an organism is that of an undirected graph, where nodes represent proteins and edges denote interactions between proteins. We slightly generalize this definition, letting a node to represent also a set of proteins instead of a single protein.

Definition 7.1. *(Graph PPI Network) A graph (PPI) network is a labeled (undirected) graph $\mathcal{G}_N = \langle P, I \rangle$ where:*

- $P = \{p_1, p_2, \ldots, p_n\}$ *are the nodes (called also* objects *in the following), where each node denotes a (initially singleton) set of proteins, as better explained below.*
- $I = \{\langle\{p_i, p_j\}, \langle w, c\rangle\rangle\}$ *is the set of edges, each denoting that an interaction occurs between (a protein in)* p_i *and (a protein in)* p_j *($i \neq j$, $i, j = 1, \ldots, n$); the label $\langle w, c\rangle$ is a pair of real numbers in the interval* $[0, 1]$, *called weakness and confidence, resp.*

For completeness, some basic definitions, already introduced in Chapter 5, are recalled. Edge labels are used to encode both “quantitative” and “reliability” information about protein-protein interactions under analysis. Quantitative information, encoded in the coefficient w, may concern, e.g., protein-protein interaction strength [120, 194], where larger values of w denote weaker interactions. The term c of the label pair weighs to what extent a stored interaction can be reliably taken into account in the overall analysis [193]: interactions between proteins can be discovered using several not equally reliable techniques, which is mirrored in the c value. In the following, for an edge $e = \langle\{p_i, p_j\}, \langle w, c\rangle\rangle$, w_e denotes w and c_e denotes c.

Now, let π be a path connecting a node p_i to a node p_j in a graph network $\mathcal{G}_N$. Define the *length* of π as $len(\pi) = \sum_{e \in \pi} w_e$. Given two nodes p_i and p_j, define $short(i, j)$ as $short(i, j) = \mathop{\textbf{argmin}}\limits_{\{\pi \text{ path connecting } p_i \text{ and } p_j\}} len(\pi)$, that is, $short(i, j)$ denotes a shortest path connecting p_i and p_j. Given a path π, we define its overall confidence as follows.

Definition 7.2. *(Cumulative Confidence C)* Given a graph network $\mathcal{G}_N$, the *Cumulative confidence* $C(\pi)$ of a path π in $\mathcal{G}_N$ is defined as $C(\pi) = \prod_{\{e|e\in\pi\}} c_e$, for each edge e in π.

In order to gain the capability to finely distinguish nodes on the basis of their distance from a given node p (e.g., the case where all w_e's in the graph are very close to zero), we introduce a normalization function parameter $\mathcal{F}$ that, given an integer $i \geq 0$, allows to single out nodes "at distance i" from p and to define the i-neighborhood of a node.

Definition 7.3. *Given a graph network $\mathcal{G}_N$, a function $\mathcal{F} : \mathbb{N} \to \mathbb{N}$ and a node p; we say that a node $\overline{p}$* is at distance *i from p if $\mathcal{F}(i-1) < len(short(p, \overline{p})) \leq \mathcal{F}(i)$.*

Definition 7.4. *(i-th Neighborhood) Given a node p in $\mathcal{G}_N = \langle P, I\rangle$, the i-th neighborhood of p ($i > 0$) is the set $\mathcal{N}(p,i) = \{q|q \in P, q$ is at distance i from $p, i > 0\}$.*

Note that, while the length of a path determines the i-neighborhood which a node q belongs to, the associated cumulative confidence may be considered representative of the probability that q actually belongs to that i-neighborhood.

Example 7.5. Consider the two networks $\mathcal{G}'_N$ and $\mathcal{G}''_N$ represented in Figure 7.3(b). Assume that $\mathcal{F}$ (see Definition 7.3) is chosen to be the identity function. The 1-neighborhood of the node p'_1 in $\mathcal{G}'_N$ is the set $\{p'_2, p'_5, p'_6, p'_8, p'_9, p'_{14}\}$, while its 2-neighborhood is $\{p'_3, p'_7, p'_{10}, p'_{11}, p'_{12}\}$. For instance, the node p'_{14} belongs to the 1-neighborhood of the node p'_1 because $len(short(p'_1, p'_{14})) = 0.99 < 1$.

Given two labels $\langle w, c\rangle$ and $\langle w', c'\rangle$, we say that $\langle w, c\rangle \prec \langle w', c'\rangle$ if $w < w'$ or, otherwise, $w = w'$ and $c > c'$. The notion of $\prec$–minimality in a set of labels is then obviously defined. Next, we define an operator that *collapses* two or more nodes of a graph network.

Definition 7.6. *(Collapsing Operator) Let $\mathcal{G}_N = \langle P, I\rangle$ be a graph network. The collapsing operator $col(\mathcal{G}_N, \widehat{P}, \widehat{p})$, where $\widehat{P} \subseteq P$, returns a graph network $\widehat{\mathcal{G}_N}$ obtained from $\mathcal{G}_N$ by:*

- *substituting the subgraph induced by $\widehat{P}$ in $\mathcal{G}_N$ with the node $\widehat{p}$;*
- *deleting all edges of the form $\langle\{p_i, p_j\}, \langle w, c\rangle\rangle$ with $p_i \in P \setminus \widehat{P}$ and $p_j \in \widehat{P}$;*
- *adding one edge $\langle\{p_i, \widehat{p}\}, \langle\widehat{w}, \widehat{c}\rangle\rangle$ for each node $p_i \in P \setminus \widehat{P}$ such that the set $P_i = \{\langle\{p_i, p_j\}, \langle w, c\rangle\rangle, p_j \in \widehat{P}\}$ of deleted edges is not empty, where $\langle\widehat{w}, \widehat{c}\rangle$ is the $\prec$–minimum label occurring in P_i.*

Given $\widehat{p}$, $dec(\widehat{p})$ returns the set of nodes $\{p_1, \cdots, p_n\}$ which were (possibly iteratively) collapsed into $\widehat{p}$ (with $dec(\widehat{p}) = \{\widehat{p}\}$ for "singleton" nodes $\widehat{p}$).

Example 7.7. Consider the network $\widehat{\mathcal{G}'_N}$ represented in Figure 7.4(b); this is obtained as $\widehat{\mathcal{G}'_N} = col(\mathcal{G}'_N, \{p'_1, p'_5\}, \widehat{p'_1})$, where $\mathcal{G}'_N$ is illustrated in Figure 7.3(b). In particular, the subgraph induced by $\{p'_1, p'_5\}$ has been substituted by $\widehat{p'_1}$. The following edges have been deleted:

$\langle\{p'_1, p'_2\},\ \langle 0.78, 0.45\rangle\rangle$, $\langle\{p'_1, p'_6\},\ \langle 0.45, 0.97\rangle\rangle$, $\langle\{p'_1, p'_9\},\ \langle 0.78, 1\rangle\rangle$, $\langle\{p'_1, p'_8\},\ \langle 0.90, 0.80\rangle\rangle$, $\langle\{p'_5, p'_3\}, \langle 0.86, 0.96\rangle\rangle$, $\langle\{p'_5, p'_7\}, \langle 0.78, 0.99\rangle\rangle$.

The following edges have been added in $\widehat{\mathcal{G}'_N}$:

$\langle\{\widehat{p'_1}, p'_2\},\ \langle 0.78, 0.45\rangle\rangle$, $\langle\{\widehat{p'_1}, p'_6\},\ \langle 0.45, 0.97\rangle\rangle$, $\langle\{\widehat{p'_1}, p'_9\},\ \langle 0.78, 1\rangle\rangle$, $\langle\{\widehat{p'_1}, p'_8\},\ \langle 0.90, 0.80\rangle\rangle$, $\langle\{\widehat{p'_1}, p'_3\}, \langle 0.86, 0.96\rangle\rangle$, $\langle\{\widehat{p'_1}, p'_7\}, \langle 0.78, 0.99\rangle\rangle$.

Clear enough, the collapsing operator can be applied iteratively several times. Thus, let $\mathcal{G}^k = \langle\tilde{\mathrm{P}},\tilde{\mathrm{I}}\rangle$ be equal to an iterated application of the collapse operator, starting with a graph $\mathcal{G}_N = \langle P, I\rangle$, that is $\mathcal{G}^0 = \mathcal{G}_N$ and $\mathcal{G}^k = col(\cdots col(\mathcal{G}_N, \widehat{P}_1, \widehat{p}_1)\cdots), \widehat{P}_k, \widehat{p}_k)$ for some k. Then, for each $k \geq 0$, nodes in $\mathcal{G}^k$ are called *objects* of $\mathcal{G}_N$, that are nodes of $\mathcal{G}_N$ itself, or subgraphs of $\mathcal{G}_N$ reduced to single nodes through (iterated) collapsing.

In the following, we assume that the graph representing the interaction network of a given organism is connected. This is in general reasonable. Moreover, if this condition is not met, the technique discussed below can be applied to connected components of the graph network by their own. Furthermore, suitable dictionaries will be exploited to store similarities values between pairs of proteins in different organisms. These dictionaries store triplets of the form $\langle p', p'', f\rangle$, where p' and p'' are nodes of the two input networks $\mathcal{G}'_N$ and $\mathcal{G}''_N$, and f is a similarity coefficient, usually in the real interval $[0, 1]$: the larger f the more similar p' and p''. For each of the considered dictionary, a cut-off value of similarity is always provided such that only triplets with f greater than the cut-off value will be considered in the analysis. Such triplets will be referred to as *significant* in the following.

7.3 Methods

7.3.1 Bi-Grappin

In this section, we briefly recall the Bi-Grappin algorithm, described in detail in Chapter 5. Assume a *Basic Knowledge Dictionary (BKD)* is given in input, which stores similarity values associated to protein structural properties (e.g., sequence similarity). The Bi-Grappin algorithm constructs a new *Refined Similarities Dictionary (RSD)* where similarities also encode network topology information, since they are refined via neighborhood analysis.

The Bi-Grappin algorithm starts by initializing the *RSD*, setting it equal to *BKD*. Then, each significant triplet $\langle p', p'', f\rangle$ in *BKD* is considered in order to refine the f value. To this end, the i-neighborhoods $\mathcal{N}'_i = \mathcal{N}(p', i)$ and $\mathcal{N}''_i = \mathcal{N}(p'', i)$ of p' and p'', resp., ($i > 0$) are iteratively generated. At the generic iteration i, $\mathcal{N}'_i$ and $\mathcal{N}''_i$ are compared in order to refine the f value. In particular, f will embed the following contributions:

- the objective function of a maximum weight matching [74] for the bipartite weighted graph consisting of the two i-neighborhoods $\mathcal{N}'_i$ and $\mathcal{N}''_i$ with weights $g_{hk} = C_{hk} \cdot f_{hk}$, where f_{hk} is the similarity between $p'_h \in \mathcal{N}'_i$ and $p''_k \in \mathcal{N}''_i$ as stored in the input dictionary *BKD* and $C_{hk} = \min\{C(short(p'_h, p')), C(short(p''_k, p''))\}$;

- the proportion of the unmatched nodes in the two *i*-neighborhoods w.r.t. matched ones, suitably weighted by the corresponding cumulative confidences;
- the value of f at iteration $i-1$.

Note that, while Bi-Grappin proceeds toward farthest neighborhoods, their influence on changing the f value becomes weaker.

Once that *RSD* is filled in and, thus, combined information about protein and network topology similarity is stored, a subgraph collapsing phase starts, which is accounted for in the next section.

7.3.2 Collapse

Our collapsing technique takes in input the dictionary *RSD*, and a threshold value $\overline{f}_{\text{msc}}$, denoting the minimum similarity value to be scored by a pair such that it is considered matchable (see below). The output of Collapse is the dictionary *OSD* storing triplets of the form $\langle \widehat{p}', \widehat{p}'', \widehat{f}_{\text{out}} \rangle$, where $\widehat{p}'$ and $\widehat{p}''$ are objects of $\mathcal{G}'_N$ and $\mathcal{G}''_N$, respectively (so, they might be derived from collapsing), and $\widehat{f}_{\text{out}}$ (also denoted in full as $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}''))$ is a coefficient in the real interval $[0, 1]$ expressing the value of the similarity between $\widehat{p}'$ and $\widehat{p}''$. As usual, the larger $\widehat{f}_{\text{out}}$ the more similar $\widehat{p}'$ and $\widehat{p}''$.

The algorithm starts by copying into *OSD* all the significant triplets of *RSD*, and ordering the entries of the *OSD* on the basis of the similarity values. Thus, the algorithm works according to the steps illustrated below.

1. Each triplet $\langle \widehat{p}', \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ in *OSD* for which $\widehat{f}_{\text{out}}$ is maximum is considered, and the two sets $\mathcal{N}(\widehat{p}', 1)$, $\mathcal{N}(\widehat{p}'', 1)$ from the graph networks $\mathcal{G}'_N$ and $\mathcal{G}''_N$ are analyzed.
2. If $\mathcal{N}(\widehat{p}', 1)$ (resp., $\mathcal{N}(\widehat{p}'', 1)$) is equal to the empty-set, or $\widehat{f}_{\text{out}} \leq \overline{f}_{\text{msc}}$, then $\widehat{p}'$ (resp., $\widehat{p}''$) will not be further involved in collapsing. Note that *OSD* does not contain *all* the entries of *RSD*. Otherwise, two nodes $\widehat{p}_h \in \mathcal{N}(\widehat{p}', 1)$ and $\widehat{p}_k \in \mathcal{N}(\widehat{p}'', 1)$ are chosen such that $\widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k) \cdot (C'_h + C''_k)$ is maximum.
3. $\mathcal{G}'_N$ and $\mathcal{G}''_N$ are collapsed by computing $\widehat{\mathcal{G}}'_N = col(\mathcal{G}'_N, \{\widehat{p}', \widehat{p}_h\}, \widehat{p}')$ and $\widehat{\mathcal{G}}''_N = col(\mathcal{G}''_N, \{\widehat{p}'', \widehat{p}_k\}, \widehat{p}'')$.
4. The *OSD* is updated as follows.
   a) The triplet $\langle \widehat{p}', \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ is changed by computing a new value for $\widehat{f}_{\text{out}}$ according to the following formula:

   $$\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'') = [(1 - \widehat{a} \cdot \widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'') + \widehat{a} \cdot \widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k)]$$

   where $\widehat{a}$ is a tuning parameter that we set in two different ways in our experimental campaign, in order to compare two different answers of the system. In particular, as we will explain in detail in Section 7.5, we exploited a first configuration in which both the cumulative confidences of $short(\widehat{p}_h, \widehat{p}')$, $short(\widehat{p}_k, \widehat{p}'')$ and the cardinality of the subgraphs (in the original graphs $\mathcal{G}'_N$ and $\mathcal{G}''_N$) associated to the four involved nodes have been taken into account. Then, we used a second configuration for the Collapse module, where the arithmetic means of $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'')$ and $\widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k)$ has been considered.

b) For those entries $\langle \widehat{p}', \widehat{p}_s, \widehat{f}_{\text{out}} \rangle$ where $\widehat{p}_s \neq \widehat{p}'', \widehat{p}_k$ $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}_s)$ is updated according to the following formula: $\widehat{f}_{\text{out}} = b \cdot \widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}_s) + (1 - b) \cdot \widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_s)$ where b is a tuning parameter used to weigh the two similarities.

c) The triplets $\langle \widehat{p}_t, \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ where $\widehat{p}_t \neq \widehat{p}', \widehat{p}_h$ are updated, analogously to the case (b), as:

$$\widehat{f}_{\text{out}} = b \cdot \widehat{f}_{\text{out}}(\widehat{p}_t, \widehat{p}'') + (1 - b) \cdot \widehat{f}_{\text{out}}(\widehat{p}_t, \widehat{p}_k).$$

d) All entries where one of $\widehat{p}_h$ and $\widehat{p}_k$ occurs are deleted from the dictionary.

5. The *OSD* is ordered according to the (new) values of $\widehat{f}_{\text{out}}$.
6. If there are any triplets $\langle \widehat{p}', \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ such that $\widehat{f}_{\text{out}}$ is maximum and $\widehat{p}'$ and $\widehat{p}''$ may be further collapsed, go to step (i), with input $\widehat{\mathcal{G}}'_N$ and $\widehat{\mathcal{G}}''_N$. Otherwise, stop and return the *OSD*.

Thus, the algorithm stops when no further collapsing is possible, that is, one of the nodes under consideration has empty neighborhood or the $\widehat{f}_{\text{out}}$ is less or equal to the threshold value $\overline{f}_{\text{msc}}$. Figure 7.1 shows the pseudocode of the algorithm.

```
Algorithm COLLAPSE
Input:
- an input protein similarity dictionary RSD
- a fixed threshold value $\overline{f}_{\text{msc}}$
- a real value $b$
Ouput: an object similarity dictionary OSD
    fill OSD with significant triplets in RSD
    order OSD according to $\widehat{f}_{\text{out}}$
    for each triplet $\langle \widehat{p}', \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ in OSD for which $\widehat{f}_{\text{out}}$ is maximum
    and $\widehat{p}''$ may be further collapsed
        generate $\mathcal{N}(\widehat{p}', 1)$ and $\mathcal{N}(\widehat{p}'', 1)$
        if ($\mathcal{N}(\widehat{p}', 1), \mathcal{N}(\widehat{p}'', 1) \neq \emptyset$ and $\widehat{f}_{\text{out}} > \overline{f}_{\text{msc}}$)
            choose two nodes $\widehat{p}_h \in \mathcal{N}(\widehat{p}', 1)$ and $\widehat{p}_k \in \mathcal{N}(\widehat{p}'', 1)$
            s.t. $\widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k) \cdot (C'_h + C''_k)$ is maximum
            $\widehat{\mathcal{G}}'_N = col(\mathcal{G}'_N, \{\widehat{p}', \widehat{p}_h\}, \widehat{p}')$
            $\widehat{\mathcal{G}}''_N = col(\mathcal{G}'_N, \{\widehat{p}'', \widehat{p}_k\}, \widehat{p}'')$
            update OSD as follows:
                $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'') = [(1 - \widehat{a} \cdot \widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'') + \widehat{a} \cdot \widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k)]$
                each triplet $\langle \widehat{p}', \widehat{p}_s, \widehat{f}_{\text{out}} \rangle$ s.t. $\widehat{p}_s \neq \widehat{p}'', \widehat{p}_k$ is updated according to:
                    $\widehat{f}_{\text{out}} = b \cdot \widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}_s) + (1 - b) \cdot \widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_s)$
                each triplet $\langle \widehat{p}_t, \widehat{p}'', \widehat{f}_{\text{out}} \rangle$ s.t. $\widehat{p}_t \neq \widehat{p}', \widehat{p}_h$ is updated according to:
                    $\widehat{f}_{\text{out}} = b \cdot \widehat{f}_{\text{out}}(\widehat{p}_t, \widehat{p}'') + (1 - b) \cdot \widehat{f}_{\text{out}}(\widehat{p}_t, \widehat{p}_k)$
                delete from OSD those entries where $\widehat{p}_h$ and $\widehat{p}_k$ are involved
            sort OSD w.r.t. $\widehat{f}_{\text{out}}$
        else $\widehat{p}'$ and $\widehat{p}''$ cannot be further collapsed
    return the dictionary OSD
```

Fig. 7.1. The COLLAPSE algorithm.

As for the worst case complexity of the collapsing technique, let $m = max\{m', m''\}$, where m' and m'' are the number of nodes of $\mathcal{G}'_N$ and $\mathcal{G}''_N$, resp., involved in significant triplets of *RSD*, and n is the maximum number of nodes of $\mathcal{G}'_N$ and $\mathcal{G}''_N$; then, in the worst case, the collapse algorithm runs in $O(max(m^3 log(m^2), n^2)$ time. In fact, the two predominant terms correspond to the filling of *OSD* with triplets in *RSD*, that costs $O(n^2)$, and ordering of the *OSD* for each of the triplets candidate for collapsing (at most $O(m)$), for an overall cost of $O(m^3 \cdot \log(m^2))$.

7.3.3 Sub-Grappin

The final procedure Sub-Grappin (SUB-GRAph extraction through PPI Networks) implementing the proposed approach consists in interleaving calls to the Bi-Grappin and Collapse procedures, so that the output dictionary produced by one call is taken in input by the following one. Figure 7.2 illustrates the pseudocode of the final algorithm Sub-Grappin.

```
Algorithm Sub-Grappin
Input:
- an input protein similarity dictionary BKD
- a threshold value $\overline{f}_{\mathrm{msc}}$
Ouput: an object similarity dictionary $D_{\mathrm{out}}$
    $D_{\mathrm{out}} = BKD$
    iterate
        call Bi-Grappin on $D_{\mathrm{out}}$ to obtain $D_{\mathrm{temp}}$
        call Collapse on $D_{\mathrm{temp}}$ with the threshold $\overline{f}_{\mathrm{msc}}$ to obtain $D_{\mathrm{out}}$
    until no nodes are collapsed by the last Collapse call
    return the dictionary $D_{\mathrm{out}}$
```

Fig. 7.2. The Sub-Grappin algorithm.

Before proceeding, we note that, in our algorithms, paths other than the shortest ones linking two nodes are disregarded and, in addition, we chose the cumulative confidence value to approximate reliability of interaction paths. In these respects, we argue that since proteins that interact are linked by an edge in the corresponding PPI network, the significance of the shortest path subsumes that of other paths potentially linking two nodes. Furthermore, the choice we performed to approximate path reliability by the Cumulative Confidence seems to work properly, as confirmed by our experimental results. We leave as future work a more detailed analysis of other approaches to combining interaction quality factors. The list of acronyms and abbreviations exploited in the chapter is reported in Table 7.1.

A Comprehensive Example

Consider the two networks $\mathcal{G}'_N$ and $\mathcal{G}''_N$ shown in Figure 7.3(b) (the reason why such networks are significant is explained below). Bi-Grappin is called first. Assume that the *RSD* dictionary returned by Bi-Grappin for $\mathcal{G}'_N$ and $\mathcal{G}''_N$ is as reported in Figure 7.3(a). The identity function has been used as $\mathcal{F}$, the threshold values 0.450 and 0.700 have been exploited to single out significant triplets in *RS D* and *OS D*, respectively, while $\overline{f}_{\mathrm{msc}}$ has been set equal to 0.450. Thus, at the beginning, the dictionary *OSD* is filled in using the first 10 entries of *RSD*. During the first call to the collapse algorithm, the triplet of *OSD* having maximum $\widehat{f}_{\mathrm{out}}$ is $\langle p'_1, p''_1, 0.900\rangle$, thus the

G_N	A Graph PPI Network
P	The set of proteins p of a Graph PPI Network
I	The set of interactions e of a Graph PPI Network
w_e	The weakness of the edge e
c_e	The confidence of the edge e
π	A path connecting a node p_i to a node p_j
$len(\pi)$	The length of the path π computed as the sum of the weaknesses of the involved edges
$short(i, j)$	A shortest path connecting the node p_i to a node p_j
$C(\pi)$	The Cumulative Confidence of the path π computed as the product of the confidences of the involved edges
$\mathcal{N}(p, i)$	The i-th neighborhood of the node p
$col(\mathcal{G}_N, \hat{P}, \hat{p})$	The collapsing operator building the collapsed node $\hat{p}$ from the set of proteins $\hat{P} \subseteq P$
$dec(\hat{p})$	The decollapsing operator that returns the set of nodes $\{p_1, \ldots, p_n\}$ previously collapsed into $\hat{p}$
BKD	The Basic Knowledge Dictionary
RSD	The Refined Similarities Dictionary
OSD	The Object Similarities Dictionary
$\widehat{f}_{\mathrm{out}}$	A similarity value computed after a Collapse step
$\overline{f}_{\mathrm{msc}}$	The minimum similarity collapsing threshold
$\textsc{sas}_X$	The Sub-graph Alignment Score w.r.t. the ontology x
f_{X}	The functional similarity w.r.t. the ontology x
f_0	The basic similarity of nodes
sbs	the basic similarity of subgraphs

Table 7.1. List of acronyms and abbreviations

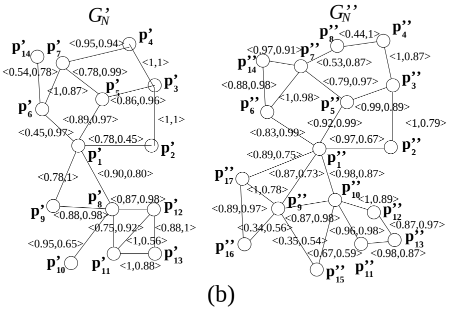
1-neighborhoods of p'_1 and p''_1 are analyzed. Nodes p'_5 and p''_5 are considered since they present the maximum value of $\widehat{f}_{\mathrm{out}}(\widehat{p_h}, \widehat{p_k}) \cdot (C'_h + C''_k) = 1.568$, and the networks $\widehat{\mathcal{G}}'_N$ and $\widehat{\mathcal{G}}''_N$ represented in Figure 7.4(b) are obtained as $\widehat{\mathcal{G}}'_N = col(\mathcal{G}'_N, \{p'_1, p'_5\}, \widehat{p_1}')$ and $\widehat{\mathcal{G}}''_N = col(\mathcal{G}''_N, \{p''_1, p''_5\}, \widehat{p_1}'')$, respectively. Figure 7.4(a) shows the new *OSD* obtained after the collapsing process. In particular, the similarity between $\widehat{p_1}'$ and $\widehat{p_1}''$ is computed as $\widehat{f}_{\mathrm{out}}(\widehat{p_1}', \widehat{p_1}'') = [(1 - 0.490) \cdot 0.900 + 0.490 \cdot 0.800] = 0.851$.

Following the same line of reasoning, during the second and third iterations, also nodes p'_7 and p'_6 (p''_7 and p''_6, resp.) have been englobed in $\widehat{p_1}'$ ($\widehat{p_1}''$, resp.). During the fourth iteration, the pair of nodes with maximum $\widehat{f}_{\mathrm{out}}$ are p'_{11} and p''_{11}, and they have p'_{13} and p''_{13} as neighbors satisfying the condition for collapsing; thus, at this iteration $\widehat{\mathcal{G}}'_N = col(\widehat{\mathcal{G}}'_N, \{p'_{11}, p'_{13}\}, \widehat{p_{11}}')$ and $\widehat{\mathcal{G}}''_N = col(\widehat{\mathcal{G}}''_N, \{p''_{11}, p''_{13}\}, \widehat{p_{11}}'')$.

During the next three iterations, the following subsets of nodes are collapsed: $\{\widehat{p}'_1, p'_4\}$ and $\{\widehat{p}''_1, p''_4\}$, $\{\widehat{p}'_1, p'_3\}$ and $\{\widehat{p}''_1, p''_3\}$, $\{\widehat{p}'_1, p'_2\}$ and $\{\widehat{p}''_1, p''_2\}$. In the last iteration our algorithm collapses $\{\widehat{p}'_{11}, p'_{12}\}$ and $\{\widehat{p}''_{11}, p''_{12}\}$, obtaining a similarity value $\widehat{f}_{\mathrm{out}}(\widehat{p}'_{11}, \widehat{p}''_{11}) = 0.710$. The networks obtained from this last iteration of Collapse are shown in Figure 7.5(b). Then, the collapse algorithm stops at this iteration. At the next iteration of Sub-Grappin, Bi-Grappin is called again and, since the following

RSD Dictionary		
$\widehat{\mathbf{p}}'_{\mathbf{h}}$	$\widehat{\mathbf{p}}''_{\mathbf{k}}$	$\widehat{\mathbf{f}}_{\text{out}}(\widehat{\mathbf{p}}'_{\mathbf{h}}, \widehat{\mathbf{p}}''_{\mathbf{k}})$
p'_1	p''_1	0.900
p'_2	p''_2	0.650
p'_3	p''_3	0.680
p'_4	p''_4	0.720
p'_5	p''_5	0.800
p'_6	p''_6	0.720
p'_7	p''_7	0.750
p'_{11}	p''_{11}	0.800
p'_{12}	p''_{12}	0.630
p'_{13}	p''_{13}	0.680
p'_{14}	p''_{14}	0.350
p'_8	p''_{17}	0.300
p'_8	p''_{10}	0.310

(a)

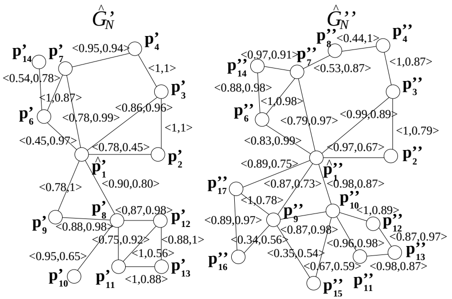


(b)

Fig. 7.3. (a) *RSD* for $\mathcal{G}'_N$ and $\mathcal{G}''_N$; (b) the two networks $\mathcal{G}'_N$ and $\mathcal{G}''_N$.

OSD Dictionary		
$\hat{\mathbf{p}}'_{\mathbf{h}}$	$\hat{\mathbf{p}}''_{\mathbf{k}}$	$\widehat{\mathbf{f}}_{\text{out}}(\hat{\mathbf{p}}'_{\mathbf{h}}, \hat{\mathbf{p}}''_{\mathbf{k}})$
$\widehat{p}'_1$	$\widehat{p}''_1$	0.851
p'_2	p''_2	0.650
p'_3	p''_3	0.680
p'_4	p''_4	0.720
p'_6	p''_6	0.720
p'_7	p''_7	0.750
p'_{11}	p''_{11}	0.800
p'_{12}	p''_{12}	0.630
p'_{13}	p''_{13}	0.680

(a)

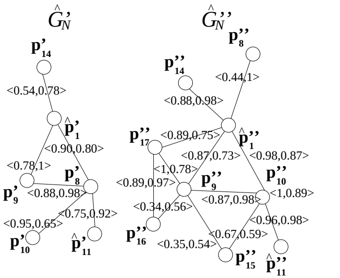


(b)

Fig. 7.4. $\widehat{\mathcal{G}}'_N$ and $\widehat{\mathcal{G}}''_N$ (a) *OSD* and (b) after the first iteration of Collapse.

OSD Dictionary		
$\hat{\mathbf{p}}'_{\mathbf{h}}$	$\hat{\mathbf{p}}''_{\mathbf{k}}$	$\widehat{\mathbf{f}}_{\text{out}}(\hat{\mathbf{p}}'_{\mathbf{h}}, \hat{\mathbf{p}}''_{\mathbf{k}})$
$\hat{p}'_1$	$\hat{p}''_1$	0.763
$\hat{p}'_{11}$	$\hat{p}''_{11}$	0.710

(a)



(b)

Fig. 7.5. (a) *OSD* for $\widehat{\mathcal{G}}'_N$ and $\widehat{\mathcal{G}}''_N$ and (b) $\widehat{\mathcal{G}}'_N$ and $\widehat{\mathcal{G}}''_N$.

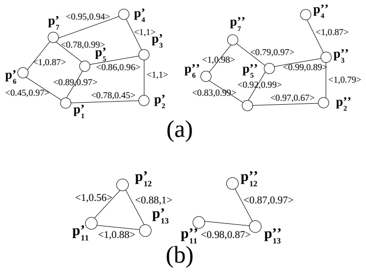


Fig. 7.6. The two pairs of subgraphs extracted by Sub-Grappin.

call to Collapse does not cause any further collapsing of nodes, the result obtained at this iteration is not modified and is as displayed in Figure 7.6.

The example illustrated above shows that our approach is able to grasp evolutionary mechanisms shaping the PPI networks. As pinpointed in [20], during evolution, two main processes may affect protein interaction networks, that are, *link attachment/detachment* and *gene duplication*. Link attachment/detachment corresponds to adding/deleting an edge involving a particular protein for which a nucleotide substitution occurred in the gene encoding for it, while gene duplication causes the addition of new nodes in the network. For instance, Figure 7.6(b) shows that the method is able to suitably cope with link detachments (look at nodes p''_{11}, p''_{12} and the edge missing in between).

7.4 Related Work

In this section a comparison with the methods proposed to align PPI networks and siscussed in Chapter 6 is provided.

The approaches that are less similar to Sub-Grappin are PATHBLAST [97] and the method proposed by Bandyopadhyay et al. [14].
The main difference between PATHBLAST [97] and our approach is that our technique does not limit itself to consider linear paths across the networks, but more generally considers subgraphs of arbitrary structure.
The approach presented in [14] is similar to Bi-Grappin, although the two methods are based on different strategies, but differently from Sub-Grappin, it does not extract connected subgraphs from the input networks.

The approaches that are more similar to Sub-Grappin are those also searching for subgraphs of general structure repeated in different networks [71, 184, 185, 92, 124].

Besides technical differences characterizing our algorithms with respect to the methods cited above, differently from our approach, all the techniques recalled above do not exploit neither reliability nor quantitative information. This two kind of information together can make conversely the analysis more accurate, as also confirmed by our experimental analysis.

7.5 Results

This section describes the evaluation of Sub-Grappin on PPI networks of well characterized organisms. Datasets have been collected from the MINT database [33] and, in particular, two PPI networks have been considered. The first one is the *Saccharomyces cerevisiae* (yeast) network, which contains $5,194$ nodes and $29,570$ interactions. The second one is the *Homo sapiens* (human) network, which contains $5,868$ nodes and $13,237$ interactions.

Roughly speaking, our system returned the alignment results in some hours, that is the time required also by other techniques performing the same task.

7.5.1 Validation Measures

Recently, some authors [216, 228] have proposed to assess the functional similarity between two proteins by exploiting Gene Ontology (GO) annotations [7].

At the same time, protein sequence similarity is often used in order to infer protein homology [207], showing to be a valuable indicator of how much proteins share similar features and behaviors.

We validated our results by exploiting a combination of both the GO annotations and the sequence similarities of pairs of proteins.

Basic similarity of nodes

We consider as *basic* similarity between two proteins their sequence similarities. We exploited the Blast 2 sequences algorithm [202], available at the Blast website[1], and referred to the BLAST *E-value* parameter to measure protein sequence similarity. In particular, after aligning two proteins p' and p'' of two different organisms, we computed the sequence similarity function f_0 according to the following transformation:

$$f_0(p', p'') = \begin{cases} 0, & \text{if } E \geq 10^{-2} \\ 2^{\frac{20}{\log E}}, & \text{if } E < 10^{-2} \end{cases}$$

where E is the BLAST E-value returned for p' and p''.

Note that the *E-value* can assume, in general, values greater than 1, and the lower it is, the more similar the protein sequences are. The formula for f_0 reported above serves the purpose of both normalizing the sequence similarity function (thus that it varies between 0 and 1) and obtaining significant variations when the E-value reaches very small values (corresponding to very similar sequences).

Given two nodes $\widehat{p'}$ and $\widehat{p''}$ in two collapsed networks, let p'_m and p''_m be two proteins belonging to the collapsed subgraphs identified by $\widehat{p'}$ and $\widehat{p''}$, resp., such that $f_0(p'_m, p''_m)$ is maximum. Then, p'_m and p''_m are fixed as representative elements of $\widehat{p'}$ and $\widehat{p''}$, and the basic similarity of two nodes is defined as:

$$f_0(\widehat{p'}, \widehat{p''}) = f_0(p'_m, p''_m).$$

[1] ftp://ftp.ncbi.nlm.nih.gov/blast/executables

Basic similarity of subgraphs

Let $\bar{S}'$ and $\bar{S}''$ be two subgraphs, identified by the corresponding collapsed nodes $\widehat{p'_h}$ and $\widehat{p''_k}$ resp., that have been associated during the alignment process. Note that $|dec(\widehat{p_h})| = |dec(\widehat{p_k})|$. The *Sub-graph Basic Score* is defined as:

$$\textsc{sbs} = \frac{\sum_{(\widehat{p'},\widehat{p''})} f_0(\widehat{p'},\widehat{p''})}{(|\bar{S}'|)},$$

where $(\widehat{p'},\widehat{p''})$ are pairs of nodes in $\bar{S}'$ and $\bar{S}'$, resp., that have been associated in the alignment process. Thus, the SBS denotes a cumulative measure of the sequence similarities between nodes in two aligned subgraphs.

Functional Similarity of Nodes

The Gene Ontology is a structured and controlled vocabulary that describes proteins based on their functions in the cell. We encoded the biological meanings of GO terms into a numeric value by using the notion of Intrinsic Information Content [179] and computed the similarity between two GO terms by exploiting the *P&S* similarity measure [166]. The GO annotations of *yeast* and *human* proteins used in our evaluation have been obtained from the GO website[2]. In particular, $9,646$ of the $11,062$ proteins belonging to the *human* and *yeast* networks were annotated at least to one GO term.

The Gene Ontology contains three kinds of terms (belonging to three independent ontologies): biological process (BP), molecular function (MF), and cellular component (CC). The GO terms of each of these ontologies are related to each other by inheritance or *is-a* relationships and form three directed acyclic graphs (DAGs).

Let X, with X ∈ {BP, MF or CC}, be a sub-ontology of the Gene Ontology and let $A_{\mathrm{X}}(p')$ ($A_{\mathrm{X}}(p'')$, resp.) be the set of annotations of the protein p' (p'', resp.) w.r.t. X. Let $sim_{\mathrm{X}}(a^i_{\mathrm{X}}, a^j_{\mathrm{X}})$ be the similarity between the two GO terms corresponding to the annotation $a^i_{\mathrm{X}} \in A_{\mathrm{X}}(p')$ and $a^j_{\mathrm{X}} \in A_{\mathrm{X}}(p'')$ computed by exploiting the *P&S* similarity measure.

The functional similarity $f_{\mathrm{X}}(p', p'')$ between two proteins p' and p'' w.r.t. X is computed according to the following formula:

$$f_{\mathrm{X}}(p', p'') = \max_{a^i_{\mathrm{X}} \in A_{\mathrm{X}}(p'), a^j_{\mathrm{X}} \in A_{\mathrm{X}}(p'')} sim(a^i_{\mathrm{X}}, a^j_{\mathrm{X}}).$$

Given two nodes $\widehat{p'}$ and $\widehat{p''}$ in two collapsed networks, let p'_m and p''_m be two proteins belonging to the collapsed subgraphs identified by $\widehat{p'}$ and $\widehat{p''}$, resp., such that $f_X(p'_m, p''_m)$ is maximum. Then, p'_m and p''_m are set as representative elements of $\widehat{p'}$ and $\widehat{p''}$, and the functional similarity of two nodes is defined as:

$$f_X(\widehat{p'}, \widehat{p''}) = f_X(p'_m, p''_m).$$

[2] http://www.geneontology.org

Furthermore, we build three different dictionaries, one for each ontologies, called D_{BP}, D_{MF} and D_{CC}, respectively, such that each D_X stores triplets $\langle \widehat{p'}, \widehat{p''}, f_{\mathrm{X}} \rangle$, where triplets are considered significant if the corresponding f_{X} is greater than a fixed cut-off value $\overline{f}_X$ (one cut-off value for each ontology out of BP, MF and CC).

Functional Similarity of Subgraphs

Let $\overline{S}'$ and $\overline{S}''$ be two subgraphs. Three *Sub-graph Alignment Scores (*SAS) can be defined, each of which refers to one of the three GO ontologies (SAS-BP for the Biological Process ontology, SAS-MF for the Molecular Function ontology and SAS-CC for the Cellular Component one).

Let D_X be one of the three dictionaries described above and let $\overline{f}_X$ the associated cut-off. The subgraph alignment score SAS-X for $\overline{S}'$ and $\overline{S}''$ is defined as:

$$\text{SAS-X} = \frac{\sum_{(\widehat{p'}, \widehat{p''}) \in N_X} f_{\mathrm{X}}(\widehat{p'}, \widehat{p''}) + |\overline{N}_X| \times \overline{f}_X}{(|N_X| + |\overline{N}_X|) \times \overline{f}_X},$$

where N_X is the set of significant triplets in D_X and $\overline{N} = D_X \setminus N_X$.

Normalized Sub-graph Alignment Score

In order to obtain our combined alignment score, we normalize the SASS w.r.t. SBS as follows:

$$\text{NSAS-BP} = \text{SAS-MP} \times \text{SBS}$$

$$\text{NSAS-MF} = \text{SAS-MF} \times \text{SBS}$$

$$\text{NSAS-CC} = \text{SAS-CC} \times \text{SBS}.$$

7.5.2 Settings and Configurations

In this section, we describe the different parameter settings and system configurations adopted in our validation campaign.

BI-GRAPPIN: Parameter Setting

Let $f_{\max}$ be the maximum similarity value in the dictionary D_{out} fed as input to BI-GRAPPIN during a generic iteration of SUB-GRAPPIN (see Figure 7.2). We exploited the reliability information on interaction data provided by the MINT database. In particular, we carried out two sets of experiments by setting the cut-off value associated to D_{out} in two different ways.

In the first set of experiments we set the D_{out} cut-off equal to $f_{\max}/2 \times 0.1$, and referred such cut-off to $g = C \cdot f$ rather than to f (see Section 7.3.1). In particular, 0.1 was assumed as the default reliability value for those interactions for which the MINT database does not provide any reliability information. Hence, a triplet $\langle p', p'', f \rangle$ is considered significant, and then involved in the maximum bipartite graph weighted matching, if $g = C \cdot f$ is greater than the cut-off.

In the second set of experiments, the D_{out} cut-off has been set equal to $f_{\max}/2$ and forced on f, as usual.

Finally, during the execution of Bi-Grappin, we stopped neighborhood analysis after looking at the 2-neighborhoods.

Collapse: Parameter Setting

Let $f_{\max}$ be the maximum similarity value in the input dictionary OSD. We fixed the minimum similarity collapsing threshold value $\overline{f}_{\text{msc}} = 0.8 \times f_{\max}$. Moreover, the tuning parameter b has been set equal to the value 0.5.

For collapsing of $\{\widehat{p}', \widehat{p}_h\}$ and $\{\widehat{p}'', \widehat{p}_k\}$, the tuning parameter $\widehat{a}$ in the resulting similarity $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'')$ (see Section 7.3) has been computed in two different ways, in order to understand how the system responded to different ways of weighting collapsed node cardinalities.

Let:

- $s' = |dec(\widehat{p}')|$,
- $s'' = |dec(\widehat{p}'')|$,
- $s_h = |dec(\widehat{p}_h)|$,
- $s_k = |dec(\widehat{p}_k)|$.

In the first Collapse configuration, we set $\widehat{a}$ as:

$$\widehat{a} = a \cdot \frac{(C'_h + C''_k)}{2},$$

where C_h and C_k are the cumulative confidences of $short(\widehat{p}_h, \widehat{p}')$ and $short(\widehat{p}_k, \widehat{p}'')$, resp., before the collapsing, while $a = \frac{s_h + s_k}{s' + s'' + s_h + s_k}$ is proportional to the cardinality of the subgraphs (in the original graphs $\mathcal{G}'_N$ and $\mathcal{G}''_N$) associated to the four involved nodes.

In the second Collapse configuration, we set $\widehat{a} = 0.5$, thus that the resulting $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'')$ has been obtained as the arithmetic means of $\widehat{f}_{\text{out}}(\widehat{p}', \widehat{p}'')$ and $\widehat{f}_{\text{out}}(\widehat{p}_h, \widehat{p}_k)$.

In the following, we refer to the $\widehat{f}_{\text{out}}$ computed according to the first configuration as $\widehat{f}'_{\text{out}}$, and to the $\widehat{f}_{\text{out}}$ computed according to the second configuration as $\widehat{f}''_{\text{out}}$.

Sub-Grappin: Configurations

Table 7.2 shows the different configurations of Sub-Grappin exploited in the experimental evaluation and obtained by using different parameter values For instance, Sub-Grappin(1-1) uses the cut-off on g for Bi-Grappin and the $\widehat{f}'_{\text{out}}$ function for Collapse. For each test we performed, Sub-Grappin was stopped after two consecutive iterations.

Table 7.3 summarizes the number of subgraph pairs discovered by running Sub-Grappin for each configuration.

		Bi-Grappin	
		cut-off on g	cut-off on f
Collapse	$\widehat{f'_{\text{out}}}$	Sub-Grappin(1-1)	Sub-Grappin(1-2)
	$\widehat{f''_{\text{out}}}$	Sub-Grappin(2-1)	Sub-Grappin(2-2)

Table 7.2. The Sub-Grappin system configurations.

		Bi-Grappin	
		cut-off on g	cut-off on f
Collapse	$\widehat{f'_{\text{out}}}$	22	37
	$\widehat{f''_{\text{out}}}$	17	76

Table 7.3. The total number of discovered subgraph pairs.

Table 7.4 shows the maximum sizes of the conserved subgraphs paired by running Sub-Grappin for each configuration. We note that Sub-Grappin(2-1) is the configuration returning the lowest number of common subgraphs with smallest sizes.

		Bi-Grappin	
		cut-off on g	cut-off on f
Collapse	$\widehat{f'_{\text{out}}}$	42	304
	$\widehat{f''_{\text{out}}}$	7	27

Table 7.4. The maximum sizes of the conserved subgraphs pairs discovered.

To validate the results obtained by Sub-Grappin we adopted our accuracy measure encompassing both protein functional information and sequence similarity, as described above in Section 7.5.1 These measures are exploited to compare Sub-Grappin to NetworkBlast-M [92]. Also, a biology-oriented discussion of the biologically most significant alignments identified by Sub-Grappin is provided.

7.5.3 Comparison with Existing Methods

Table 7.5 summarizes the comparison between the results obtained by running the four different configurations of Sub-Grappin and the results of *NetworkBlast-M* on the same interaction data. The validation measures taken into account for the comparison are the means, the maximum and the minimum value of sag-bp, nsas-mf and nsas-cc, which are computed on the set of subgraph alignments discovered by the tools under consideration. The firsts four columns of Table 7.5 correspond to the results obtained by running Sub-Grappin according to the four configurations defined above. The last column contains the values of the validation scores computed on the results returned by NetworkBlast-M. Table 7.5 highlights that Sub-Grappin outperforms NetworkBlast-M w.r.t all the considered parameters. The best configuration of Sub-Grappin is Sub-Grappin(2-1). It is also worth noting that the main differences obtained on the returned results are related to the mean values on all the three measures. In particular, Sub-Grappin(2-1) obtained 1.131, 1.240 and 1.185 as the means for SAG-BP, SAG-MF and SAG-CC while NetworkBlast-M obtained 0.630, 0.648 and 0.667.

parameter	Sub-Grappin	Sub-Grappin	Sub-Grappin	Sub-Grappin	NetworkBlast-M
	(1-1)	(1-2)	(2-1)	(2-2)	
mean of SAG-BP	0.998	0.981	1.131	1.079	0.630
maximum SAG-BP	1.446	1.548	1.446	1.548	1.079
minimum SAG-BP	0.371	0.371	0.371	0.371	0.206
mean of SAG-BP	0.371	0.371	0.371	0.371	0.206
maximum SAG-MF	1.667	1.667	1.667	1.667	1.061
minimum SAG-MF	0.371	0.371	0.371	0.371	0.225
mean of SAG-CC	1.048	0.983	1.185	1.091	0.667
maximum SAG-CC	1.637	1.667	1.637	1.667	1.122
minimum SAG-CC	0.371	0.350	0.371	0.350	0.216

Table 7.5. Comparison between Sub-Grappin and NetworkBlast-M

7.5.4 Discussion

This section presents a discussion about the most relevant alignments found by running the four different configurations of Sub-Grappin. For each of the discussed subgraph alignments, a table reporting the sequence and functional similarities between corresponding proteins or subgraphs is shown. In particular, proteins are identified by their *SWISS-PROT ids*, the sequence similarity is reported in terms of the Blast *E-value*, while the functional similarity is expressed in terms of f_{BP}, f_{MF} and f_{CC}. The functional similarity value is *na* for those pairs such that at least one of the proteins is not annotated. Moreover, a graphical representations of the alignments (for the sake of space the last one is not reported though), showing the interaction structures of the corresponding subgraphs, is provided. In the figures, the nodes filled with the same color represent aligned proteins. Note that, as resulting from our analysis, our system also allows for multiple pairings, that is, nodes in one network to be paired with different nodes in the other one (see e.g., protein *Q*14566 in Table 7.7). Table 7.6 reports the scores computed for the discussed alignments w.r.t the validation measures SAS-BP, NSAS-BP, SAS-MF, NSAS-MF, SAS-CC and NSAS-CC.

	SAS-BP	NSAS-BP	SAS-MF	NSAS-MF	SAS-CC	SASG-CC
First alignment	1.573	1.333	1.667	1.413	1.628	1.380
Second alignment	1.438	1.304	1.510	1.369	1.465	1.329
Third alignment	1.421	0.826	1.482	0.928	1.457	0.911

Table 7.6. Validation scores for the three discussed subgraph alignments

The Proteasome Complex

Table 7.7 illustrates the sequence and functional similarities of the pairs of corresponding nodes (proteins or smaller subgraphs) of the first alignment under consideration, that has been obtained using the configuration Sub-Grappin(1-1). Figure 7.7 reports a graphical representation of the two corresponding subgraphs discovered by Sub-Grappin. In particular, Figure 7.7(a) represents the subgraph identified on the PPI network of the *yeast* and Figure 7.7(b) represents the subgraph identified on the PPI network of the *human*.

The proteins aligned in the two subgraphs are components of a well preserved complex known as *proteasome*, which is a multicatalytic proteinase complex consisting of many different proteins, organized in a catalytic core and two regulative subunits. It is involved in the ubiquitin-mediated degradation of proteins, where the covalent, regulated attachment of ubiquitin to proteins target them for degradation, thus controlling the half-life of cell components. Two of the node pairs ($P53091/Q14566$ and $P29496/Q14566$) are instead proteins involved in dna replication, that functions as dna elicase. They are connected to the subgraph since they are probably regulated through the cell cycle by proteasome mediated degradation.

yeast protein	human protein	*E-value*	f_{BP}	f_{MF}	f_{CC}
P29496	Q14566	$2.0E-81$	1.000	1.000	1.000
P53091	Q14566	0.0	1.000	1.000	1.000
P40302	P25786	$1.0E-68$	1.000	1.000	1.000
P23638	P25789	$5.0E-74$	1.000	1.000	1.000
P23639	P25787	$1.0E-71$	1.000	1.000	1.000
P40303	O14818	$6.0E-79$	1.000	1.000	1.000
P21243	P60900	$5.0E-69$	1.000	1.000	1.000
P21242	P25788	$5.0E-70$	1.000	1.000	1.000

Table 7.7. Protein similarity scores for the proteasome complex

The PP2A Complex

Table 7.8 reports the sequence and functional similarities of the pairs of corresponding nodes in the second alignment. Figure 7.8 reports a graphical representation of

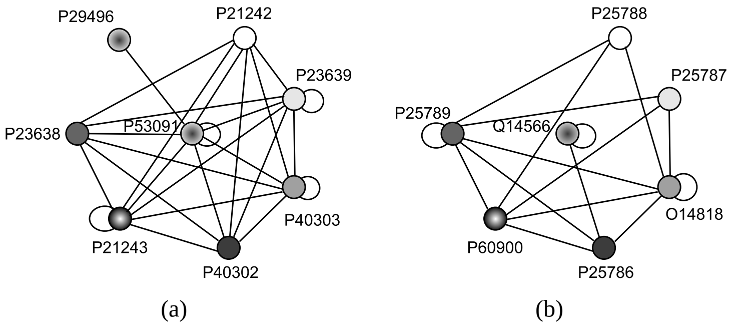


Fig. 7.7. The aligned *proteasome* subgraphs of (a) *yeast* and (b) *human*.

the two corresponding subgraphs discovered by Sub-Grappin in the configuration (2-1). In particular, Figure 7.8(a) represents the subgraph identified on the PPI network of *yeast* and Figure 7.8(b) represents the subgraph identified on the PPI network of *human*.

Most of the paired proteins are subunits of the serine phosphatase *PP*2*A* complex, composed of catalytic, structural and regulatory proteins. This crucial enzyme is conserved from yeast to human, acting on a broad range of substrates and being involved in diverse cellular processes. Akt (*P*31749) is instead a known substrate of *PP*2*A*.

yeast protein	human protein	*E-value*	f_{BP}	f_{MF}	f_{CC}
Q00362	P63151	0.000	0.964	1.000	1.000
P23594	P62714	0.000	0.964	1.000	1.000
P38903	Q7L7W2	0.000	*na*	*na*	*na*
P31383	P30153	0.000	0.964	0.744	1.000
P23595	P67775	0.000	0.9640	1.000	1.000
P08458	P31749	$7.0E-26$	0.791	1.000	0.955
Q12469	Q13043	$1.0E-53$	0.791	1.000	0.368

Table 7.8. Protein similarity scores for the *PP*2*A* complex

The Cytoskeleton Complex

Table 7.9 reports the sequence and functional similarities of the pairs of corresponding nodes in the third alignment, obtained via the configuration Sub-Grappin(1-1).

The two networks are composed by proteins that are component of the cytoskeleton structure (actins, myosins, cofilin etc), or implicated in cytoskeletal reorganization (*Rvs*167, *las*17, and their human counterparts *nebl* and *wasp*). Many regulative

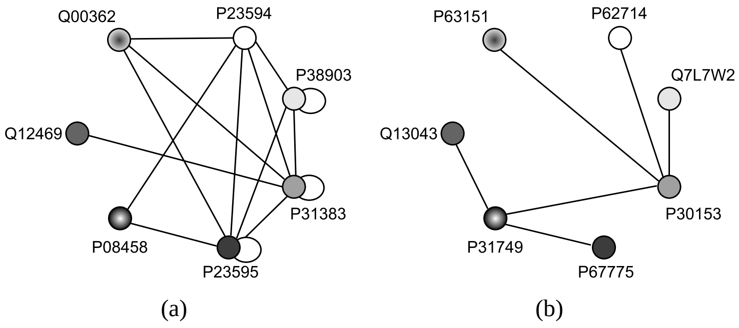


Fig. 7.8. The aligned *PP*2*A* subgraphs of (a) *yeast* and (b) *human*.

enzymes or enzymatic complexes are also included: phosphatases, kinases and GTPases are all known to regulate cytoskeleton assembly as well as cell morphology and polarization acting on cytoskeleton subrates.

7.6 Concluding Remarks

In this chapter, we dealt with the problem of discovering common modules in PPI networks. We presented a technique based on the exploitation of dictionaries storing similarities between pairs of nodes belonging to different networks. We presented an algorithm, called Sub-Grappin, based on the iterative exploitation of two different stages, that are, protein similarities computation and refining, and connected subgraphs extraction. The first stage is based on Sub-Grappin (see Chapter 5), while the second one consists in a node collapsing technique. Experimental evaluation on the yeast and human PPI networks showed the effectiveness of our approach, also validated by some suitable accuracy parameters we defined.

In the next part of the thesis, involving Chapter 8 and Chapter 9, the problem of protein-protein interaction networks querying will be faced. In particular, in Chapter 8 a new PPI network querying algorithm will be described.

yeast protein	human protein	*E-value*	f_{BP}	f_{MF}	f_{CC}
P39940	Q9HAU4	0.000	0.844	0.964	1.000
P25039	Q96RP9	0.000	0.948	1.000	1.000
P32381	P68133	$9.0E-95$	0.806	1.000	0.955
P47029	Q6ZNA4	0.0020	0.452	0.556	0.955
P39743	O76041	$1.0E-5$	0.302	0.955	0.000
Q12344	Q15797	0.0030	0.458	0.325	1.000
P52490	P61088	$2.0E-64$	0.964	1.000	0.955
P53152	Q13404	$7.0E-34$	1.000	0.925	0.964
P10862	Q9Y3C5	0.0050	0.839	1.000	0.908
P60010	P63261	0.000	0.525	1.000	0.955
Q03048	P23528	$3.0E-27$	0.955	1.0	1.000
Q04439	Q9UM54	0.000	0.914	1.000	0.955
Q01389	Q99759	$1.0E-56$	0.888	1.000	0.000
P36006	O94832	0.000	0.000	1.000	0.897
P06787	P62158	$9.0E-54$	0.412	1.000	0.560
P38903	Q7L7W2	0.000	*na*	*na*	*na*
P31383	P30153	0.000	0.964	0.744	1.000
P23594	P62714	0.000	0.964	1.000	1.000
Q00362	P63151	0.000	0.964	1.000	1.000
P53049	P13569	$3.0E-61$	1.000	1.000	0.974
P23595	P67775	0.000	0.964	1.000	1.000
P08458	P31749	$7.0E-26$	0.791	1.000	0.955
Q12469	Q13043	$1.0E-53$	0.791	1.000	0.368
Q12163	Q96EX0	$1.0E-7$	*na*	*na*	*na*
Q03497	Q13153	0.000	0.739	1.000	0.955
P19073	P63000	$3.0E-82$	0.922	1.000	1.000
P39083	Q53QZ3	$1.0E-19$	0.816	0.925	1.000
P06780	P60953	$3.0E-55$	0.922	1.000	1.000
Q12434	P52565	$5.0E-37$	0.867	1.000	0.955
P48562	P42685	$2.0E-19$	0.870	1.000	0.610
P08018	O15530	$8.0E-18$	0.840	1.000	0.955
P08018	P23443	$1.0E-20$	0.870	1.000	0.955
P19524	P46940	$4.0E-5$	0.458	1.000	0.696
Q12446	P42768	$2.0E-14$	0.910	0.505	0.955
P38822	O60592	$2.0E-8$	*na*	*na*	*na*
P32793	O60593	$1.0E-11$	*na*	*na*	*na*
P32790	P06241	$6.0E-9$	0.504	0.542	0.978
Q08581	Q99816	$4.0E-9$	0.917	0.634	0.908
P13186	Q13464	$6.0E-25$	0.870	1.000	0.712
P53281	Q99962	$2.0E-11$	0.554	0.547	0.955

Table 7.9. Protein similarity scores for the *cytoskeleton* complex

Part IV

Network Querying

8

PInG-Q: a Tool for Protein Interaction Graph Querying

Summary. This chapter describes a novel method for querying protein-protein interaction networks. In particular, in Section 8.1 some background information on protein-protein interaction network querying is recalled. Section 8.2 illustrates in detail the proposed approach. In Section 8.3, a brief comparison with existing methods is provided. Note that, a detailed comparison among PPI network querying techniques is discussed in the following chapter. Section 8.4 discusses some preliminary experimental results obtained and, finally, in Section 8.5 some conclusions are drawn.

8.1 Introduction

As already discussed in Chapter 2, one of the main modes to compare biological networks is *network querying* that has the aim of transferring biological knowledge within and across species, as also stated by Sharan and Ideker [181]. In fact, PPI subnetworks may correspond to functional modules made of proteins involved in the same biological processes. Unfortunately, as subgraph isomorphism checking is involved, the applicability of exact approaches to solve network querying is rather limited due to the NP-completeness of the problem [75]. Thus, approaches have been proposed where the search is constrained to simple structures, such as paths and trees [97, 165, 183], some heuristic methods have been presented to deal with true subgraph queries [205], whereas only a few techniques have been proposed based on exact algorithms, so that their practical applicability is limited to queries that are sparse graphs or containing a small number of nodes [231].

This chapter provides a contribution in this setting, by proposing a new technique to network querying, called PInG-Q. The main characteristics of PInG-Q are as follows: *(i)* it allows to manage arbitrary topology networks, *(ii)* it allows to take into account reliability values associated with interactions and *(iii)* it is capable of singling out also approximated answers to the query graph, as corresponding to evolution determined variations in the sets of nodes and edges. To the best of our knowledge, this is the first technique that comprises all those three characteristics, as also pointed out in the following Section 8.3.

To illustrate, given a target protein-protein interaction network $\mathcal{G}^{\mathrm{T}}$ and a (typically much smaller) query network $\mathcal{G}^{\mathrm{Q}}$, we are interested in finding a (possibly approximated) occurrence of $\mathcal{G}^{\mathrm{Q}}$ in $\mathcal{G}^{\mathrm{T}}$. To this end, PInG-Q first *focuses* a portion of the target network being relevant to the query as resulting by aligning the two networks. To do that, a minimum bipartite graph weighted matching [74] is used, which work by relying on protein sequence similarities. This initial "global" alignment produces a preliminary solution, whose topology may, however, significantly disagree with that of the query network. Therefore, our algorithm "zooms" toward a suitable solution, that matches with a sufficiently large extent the query topology. This is obtained by *refining* similarity values associated with pairs of proteins in $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$ taking into account topology constraints, and then looking for a new alignment of the networks. The process is iterated, going through a number of alignments, until one is obtained that satisfies both protein similarities and topological constraints.

Note that repeatedly computing such global alignments provides some guarantees that the resulting solution remains close to the globally optimum match. Furthermore, differently from other network querying techniques, which are typically based on a oil-stain visiting strategy, our global alignment strategy permits to naturally deal with missing edges (possibly corresponding to information missing in the database): this case corresponds to producing an alignment of the query network with a generally unconnected subgraph of the target one.

The rest of the chapter is organized as follows. The next section discusses in detail the proposed approach. In Section 8.3, PInG-Q is compared with some related work. Section 8.4 discusses some preliminary experimental results and, finally, in Section 8.5 some conclusions are drawn.

8.2 The Proposed Approach

Before explaining the technique in detail, we give two preliminary definitions useful to formulate the problem under consideration.

Definition 8.1. (Protein Interaction Graph) A *Protein Interaction Graph* is a weighted (undirected) graph $\mathcal{G} = \langle P, I \rangle$, such that:

- $P = \{p_1, p_2, \ldots, p_n\}$ is the set of nodes, each of which represents a protein;
- $I = \{\langle \{p_i, p_j\}, c_{i,j} \rangle\}$ is the set of weighted edges, each denoting an interaction between proteins, and the label $c_{i,j}$ is the *reliability factor* associated to that interaction.

Definition 8.2. (Distance Dictionary) Given a query protein interaction graph $\mathcal{G}^{\mathrm{Q}}$ and a target protein interaction graph $\mathcal{G}^{\mathrm{T}}$, the *Distance Dictionary DD* is a set of triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j} \rangle$, where p_i^{Q} belongs to $\mathcal{G}^{\mathrm{Q}}$, p_j^{T} belongs to $\mathcal{G}^{\mathrm{T}}$ and $d_{i,j}$ is the distance value associated to the pair p_i^{Q} and p_j^{T}.

Thus, let $\mathcal{G}^{\mathrm{Q}} = \langle P^{\mathrm{Q}}, I^{\mathrm{Q}} \rangle$ and $\mathcal{G}^{\mathrm{T}} = \langle P^{\mathrm{T}}, I^{\mathrm{T}} \rangle$ be two protein interaction graphs. In particular, $\mathcal{G}^{\mathrm{Q}}$ denotes the query network to search for in the target network $\mathcal{G}^{\mathrm{T}}$.

Assume that a distance dictionary $DD^{(0)}$ is available, which stores information about protein sequence similarities of $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$ (details about the computation of $DD^{(0)}$ will be given in Section 8.2.1).

At step 0, the algorithm first aligns $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$ by exploiting a minimum bipartite graph weighted matching procedure [74] applied to the bipartite weighted graph $\mathcal{G}^{\mathrm{QT}} = \langle P^{\mathrm{QT}}, I^{\mathrm{QT}} \rangle$ such that:

- $P^{\mathrm{QT}} = P^{\mathrm{Q}} \cup P^{\mathrm{T}}$,
- $I^{\mathrm{QT}} = \{\langle \{p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}\}, d_{i,j}^{(0)} \rangle\}$ is the set of weighted edges, where the label $d_{i,j}^{(0)}$ is the distance score between p_i^{Q} and p_j^{T} as stored in the distance dictionary $DD^{(0)}$.

The result of running the weighted matching algorithm on $\mathcal{G}^{\mathrm{QT}}$ is returned in a dictionary $DD^{\mathrm{S}(0)} \subset DD^{(0)}$ storing the triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(0)} \rangle$ corresponding to computed node pairings.

Algorithm for protein interaction graph querying

```
Input:
    a basic distance dictionary $DD^{(0)}$;
    a query protein interaction graph $\mathcal{G}^{Q} = \langle P^{Q}, I^{Q} \rangle$;
    a target protein interaction graph $\mathcal{G}^{T} = \langle P^{T}, I^{T} \rangle$;
    real values $\pi_{ins}, \pi_{del}, \pi_{egd}, \pi_{cm}, I_{MAX}, \alpha, \beta, \gamma$;
    an integer value MaxIteration
    a threshold value $D_{th}$;
Ouput:
    an approximate occurrence $\sigma^*$ of $\mathcal{G}^{Q}$ on $\mathcal{G}^{T}$ s.t. $D^{\sigma^*} \leq D_{th}$;

$h = 0$;
for $k = 1$ to MaxIteration do
    compute $\sigma^{(h)} = \langle \mathcal{G}^{S}, DD^{S(h)} + \rangle$ solving minimum bipartite weighted
    matching problem on $\mathcal{G}^{QT} = \langle P^{QT}, I^{QT} \rangle$ s.t.
        $P^{QT} = P^{Q} \cup P^{T}$,
        $I = \{\langle p_i^{Q}, p_j^{T}, d_{i,j}^{(h)} \rangle\}$ if $\langle p_i^{Q}, p_j^{T}, d_{i,j}^{(h)} \rangle \in DD^{(h)}$;
    compute $D^{\sigma(h)}$;
    if $(D^{\sigma(h)} > D_{th})$
        $h = h + 1$;
        for each $\langle p_i^{Q}, p_j^{T}, d_{i,j}^{(h-1)} \rangle \in DD^{S(h-1)}$
            refine $DD^{(h-1)}$ to obtain $DD^{(h)}$ using:
                $d_{i,j}^{(h)} = d_{i,j}^{(h-1)} + \alpha \cdot \mu'_{ins} \cdot d_{ins} + \beta \cdot \mu'_{del} \cdot d_{del} + \gamma \cdot \mu'_{egd} \cdot d_{egd} +$
                $-\mu'_{cm} \cdot d_{cm}$;
    else stop and return $\sigma^* = \sigma^{(h)}$;
return "No solution found."
```

Fig. 8.1. The PInG-Q algorithm.

Before going on with illustrating our algorithm, we need to introduce some further concepts. Thus, define *unmatched*$^{\mathrm{T}}(DD^{\mathrm{S(0)}})$ the set of nodes $p^{\mathrm{T}}_{j} \in P^{\mathrm{T}}$ such that *(i)* p^{T}_{j} is on the shortest path connecting two nodes p^{T}_{j1} and p^{T}_{j2} in P^{T}, *(ii)* the triplets $\langle p^{\mathrm{Q}}_{i1}, p^{\mathrm{T}}_{j1}, d^{(0)}_{i1,j1}\rangle$ and $\langle p^{\mathrm{Q}}_{i2}, p^{\mathrm{T}}_{j2}, d^{(0)}_{i2,j2}\rangle$ belong to $DD^{\mathrm{S(0)}}$, and *(iii)* p^{Q}_{i1} and p^{Q}_{i2} are directly linked by an edge in $\mathcal{G}^{\mathrm{Q}}$. Moreover, define *unmatched*$^{\mathrm{Q}}(DD^{\mathrm{S(0)}})$ the set of nodes $p^{\mathrm{Q}}_{i} \in P^{\mathrm{Q}}$ that have not been paired with any node of $\mathcal{G}^{\mathrm{T}}$ in $DD^{\mathrm{S(0)}}$.

Define the *extended dictionary* $DD^{\mathrm{S(0)}}+ = DD^{\mathrm{S(0)}} \cup DD^{\mathrm{S(0)}}_{\mathrm{in}} \cup DD^{\mathrm{S(0)}}_{\mathrm{del}}$, where $DD^{\mathrm{S(0)}}_{\mathrm{in}} = \{\langle \bullet, p^{\mathrm{T}}_{j}, -\rangle\}$ for p^{T}_{j} a node in *unmatched*$^{\mathrm{T}}(DD^{\mathrm{S(0)}})$, and $DD^{\mathrm{S(0)}}_{\mathrm{del}} = \{\langle p^{\mathrm{Q}}_{i}, \bullet, -\rangle\}$ for p^{Q}_{i} a node in *unmatched*$^{\mathrm{Q}}(DD^{\mathrm{S(0)}})$. Let $\mathcal{G}^{\mathrm{S}} = \langle P^{\mathrm{S}}, I^{\mathrm{S}}\rangle$ be the subgraph of $\mathcal{G}^{\mathrm{T}}$ such that P^{S} is the set of nodes of $\mathcal{G}^{\mathrm{T}}$ occurring in $DD^{\mathrm{S(0)}}+$, and the set of edges I^{S} is as follows. An edge is added in $\mathcal{G}^{\mathrm{S}}$ between proteins p^{T}_{h} and p^{T}_{k} if *(i)* there is an edge $\langle p^{\mathrm{T}}_{h}, p^{\mathrm{T}}_{k}, c_{h,k}\rangle$ in $\mathcal{G}^{\mathrm{T}}$, *(ii)* there is an edge $\langle p^{\mathrm{Q}}_{i}, p^{\mathrm{Q}}_{j}, c_{i,j}\rangle$ in $\mathcal{G}^{\mathrm{Q}}$, and *(iii)* the triplets $\langle p^{\mathrm{Q}}_{i}, p^{\mathrm{T}}_{h}, d^{(0)}_{i,h}\rangle$ and $\langle p^{\mathrm{Q}}_{j}, p^{\mathrm{T}}_{k}, d^{(0)}_{j,k}\rangle$ belong to $DD^{\mathrm{S(0)}}$. Moreover, for those pairs of proteins p^{T}_{h} and p^{T}_{k} for which conditions *(ii)* and *(iii)* above hold, but condition *(i)* does not, all the edges in the shortest path connecting p^{T}_{h} and p^{T}_{k} in $\mathcal{G}^{\mathrm{T}}$ are added to $\mathcal{G}^{\mathrm{S}}$. The edge labels of $\mathcal{G}^{\mathrm{S}}$ are those of $\mathcal{G}^{\mathrm{T}}$. We refer to $\sigma^{(0)} = \langle \mathcal{G}^{\mathrm{S}}, DD^{\mathrm{S(0)}}+\rangle$ as an *approximate occurrence* of $\mathcal{G}^{\mathrm{Q}}$ in $\mathcal{G}^{\mathrm{T}}$.

Note that $\sigma^{(0)}$ may well encode a suitable matching for $\mathcal{G}^{\mathrm{Q}}$ in $\mathcal{G}^{\mathrm{T}}$ or, otherwise, some relevant topological differences might be there significantly distinguishing $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{S}}$. In order to evaluate the “quality” of $\sigma^{(0)}$, we introduce a measure of “distance” between subgraphs, which is encoded in a *distance score* $D^{\sigma^{(0)}}$ that, for the sake of the readability, will be detailed in Section 8.2.1. For the moment being, let us just state that the larger $D^{\sigma^{(0)}}$ the more $\mathcal{G}^{\mathrm{S}}$ differs from $\mathcal{G}^{\mathrm{Q}}$. Thus, we are going to consider $\sigma^{(0)}$ an acceptable solution if the corresponding $D^{\sigma^{(0)}}$ is less than a given fixed quality threshold D_{th}.

Hence, we can summarize PInG-Q algorithm. Its next step is to evaluate $D^{\sigma^{(0)}}$ for $\sigma^{(0)}$ and compare it to D_{th}. If $D^{\sigma^{(0)}} \leq D_{th}$, then $\sigma^{(0)}$ is returned as the output. Otherwise, a further minimum bipartite graph weighted matching step is performed as explained below. Let $\sigma^{(h)} = \langle \mathcal{G}^{\mathrm{S}}, DD^{\mathrm{S(h)}}+\rangle$ be the approximate occurrence computed at the generic step h of the algorithm using the dictionary $DD^{(h)}$ such that $D^{\sigma^{(h)}} > D_{th}$. The next run of the bipartite weighted matching algorithm uses an updated dictionary $DD^{(h+1)}$ obtained from $DD^{(h)}$ and $DD^{\mathrm{S(h)}}+$ as explained next. Initially, $DD^{(h+1)}$ is set equal to $DD^{(h)}$, then some of its entries are *refined*, using $DD^{\mathrm{S(h)}}+$ as follows. Let:

$$d^{(h+1)}_{i,j} = d^{(h)}_{i,j} + \alpha\cdot \mu_{\mathrm{ins}}\cdot (\frac{1 - d^{(h)}_{i,j}}{C_i\cdot I_{\mathrm{MAX}}}) + \beta\cdot \mu_{\mathrm{del}}\cdot (1 - d^{(h)}_{i,j}) + \gamma\cdot \mu_{\mathrm{egd}}\cdot (\frac{1 - d^{(h)}_{i,j}}{C_i}) + \tag{8.1}$$

$$-\mu_{\mathrm{cm}}\cdot (\frac{d^{(h)}_{i,j}}{C_i})$$

where:

- the triplet $\langle p^{\mathrm{Q}}_{i}, p^{\mathrm{T}}_{j}, d^{(h)}_{i,j}\rangle$ belongs to $DD^{\mathrm{S(h)}}$,
- the term C_i is defined as the sum of the reliability factors of the edges incident on p^{Q}_{i},

- I_{MAX} serves the purpose of bounding from above the number of insertions per single edge that we use in the computation,
- μ_{ins} is the penalty score for node insertions, μ_{del} that for node deletions, μ_{egd} that for edge deletions,
- μ_{cm} is a bonus score that rewards correct matches of edges,
- α, β, γ are real values used to weigh the penalty factors μ_{ins}, μ_{del} and μ_{egd} so that $\alpha + \beta + \gamma = 1$.

The rationale of the formula, whose terms will be detailed in the following Section 8.2.1, is that of modifying the original values of protein similarity in such a way as to take into account information about the topology mismatches of the current solution. By the virtue of this update, the following run of the bipartite weighted matching produces a new solution $\sigma^{(h+1)}$.

Iterations proceed until to either a good approximate solution σ^* is found (that is, $D^{\sigma*} \leq D_{th}$) or, otherwise, a maximum number of iterations (MaxIteration) is reached, in which case no solution is returned.

The pseudocode of PInG-Q is shown in Figure 8.1.
The following result holds.

Proposition 8.3. *Let n and m be the number of nodes in the target and query networks, respectively. In the worst case the algorithm runs in* $O(\text{MaxIteration} \cdot n^3)$ *time.*

Proof. The shortest path between each pair of nodes in the target network can be pre-computed by the Floyd-Warshall algorithm in $O(n^3)$. During each iteration, two steps are performed. The first one is the computation of a potential solution, obtained by solving a bipartite graph maximum weight matching. The second step is the refinement of the similarity values associated with matching nodes. The time required to compute the maximum weight matching of a bipartite graph made of $\overline{n}$ nodes is $O(\overline{n}^3)$ [74]. Since n is always larger than m, the maximum number of nodes in the bipartite graph is $O(n)$, thus the first step costs $O(n^3)$. The refinement step costs $O(m^2)$ because the number of the edges in the query graph is at most m^2 and all the edges (interactions) have to be explored once to refine the similarities of corresponding nodes. The maximum number of iterations is MaxIteration, thus, overall, the algorithm runs in $O(\text{MaxIteration} \cdot n^3)$ time.

8.2.1 Technical details

This section is devoted to illustrate some technical details regarding the parameters and other concepts we have used above.

Basic distance dictionary

As already pointed out, a preprocessing of the protein interaction graphs $\mathcal{G}^{\text{Q}}$ and $\mathcal{G}^{\text{T}}$ in input is necessary in order to evaluate the sequence similarity of pairs of proteins $(p^{\text{Q}}, p^{\text{T}})$ such that p^{Q} belongs to $\mathcal{G}^{\text{Q}}$ and p^{T} belongs to $\mathcal{G}^{\text{T}}$. All information obtained during the preprocessing stage are stored in the basic distance dictionary $DD^{(0)}$, that is computed as follows. The *Blast 2 sequences* algorithm [202] is executed to align the amino acid sequences of pairs of proteins from $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$, respectively. The resulting BLAST E-values are used to compute a distance score $d^{(0)}_{i,j}$ for each pair of nodes p^{Q}_i of $\mathcal{G}^{\mathrm{Q}}$ and p^{T}_j of $\mathcal{G}^{\mathrm{T}}$, according to the following formula:

$$d^{(0)}_{i,j} = \begin{cases} 1, & \text{if } E \geq 10^{-2} \\ 1 - 2^{\frac{20}{\log E}}, & \text{if } E < 10^{-2} \end{cases}$$

where E is the BLAST E-value as returned by Blast 2 on input p^{Q}_i and p^{T}_j.

Note that the E-value can assume, in general, values greater than 1, and the lower it is, the more similar the protein sequences are. The formula reported above serves the purpose of both normalizing the distance score thus that it varies between 0 and 1 and obtaining more significant variations when the E-value reaches very small values (corresponding to very similar sequences).

Node insertion/deletion

As pointed out in Section 8.1, given a query graph in input, our approach aims at searching for its *approximate* occurrences in the target network. In fact, as discussed in [20], during the evolution of an organism, some events may occur that modify the associated network structure. Those events are *gene duplication*, that causes the addition of new nodes, and *link dynamics*, corresponding to gain and loss of interactions through mutations in existing proteins. In its turn, a gene duplication may be associate to both *node insertions* and *node deletions* [51, 183].

Thus, a node insertion event may be associated to the presence of one or more surplus nodes in the path connecting two nodes p^{T}_i and p^{T}_j in the target network, when they are recognized to correspond to two nodes p^{Q}_i and p^{Q}_j in the query network, connected by just one edge. Figure 8.2(a) clarifies this issue, where the case of a single node insertion is represented.

To take into account node insertions, we define the *number of node insertions* between each pair of nodes p^{T}_i and p^{T}_j belonging to a connected subgraph of $\mathcal{G}^{\mathrm{T}}$ w.r.t. the query network $\mathcal{G}^{\mathrm{Q}}$ as the number of nodes in the shortest path linking p^{T}_i and p^{T}_j in $\mathcal{G}^{\mathrm{T}}$.

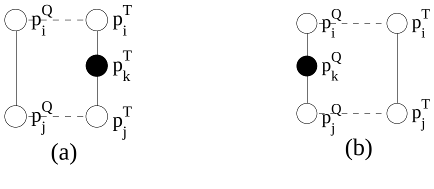


Fig. 8.2. (a) Node insertion; (b) node deletion.

A node deletion event occurs when there is a node in the query graph that does not correspond to any node in the target network (see Figure 8.2(b)). This is taken into account using scores, as will be detailed below.

Distance Score

The distance score D^{σ} for an approximate occurrence σ is obtained by evaluating: *(i)* protein sequence similarity, *(ii)* network topology, *(iii)* number of node insertions, *(iv)* number of node deletions and *(v)* number of edge deletions, where edge deletions are intended in terms of edges that occur in the query but not in the target graph, and that are interpreted as lack or incorrectness of information.

Thus, let $\mathcal{G}^{\mathrm{Q}}$ be the query protein interaction graph, $\mathcal{G}^{\mathrm{T}}$ be the target protein interaction graph and $\sigma^{(\mathrm{h})} = \langle \mathcal{G}^{\mathrm{S}}, DD^{\mathrm{S(h)}}+\rangle$ $(DD^{\mathrm{S(h)}}+ = DD^{\mathrm{S(h)}} \cup DD^{\mathrm{S(h)}}_{\mathrm{in}} \cup DD^{\mathrm{S(h)}}_{\mathrm{del}})$ be an approximate occurrence of $\mathcal{G}^{\mathrm{Q}}$ on $\mathcal{G}^{\mathrm{T}}$. The distance score $D^{\sigma^{(\mathrm{h})}}$ associated to $\sigma^{(\mathrm{h})}$ is computed according to the following formula:

$$D^{\sigma^{(\mathrm{h})}} = \sum_{\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle \in DD^{\mathrm{S(h)}}} d_{i,j}^{(\mathrm{h})} + \mu_{\mathrm{ins}}^{\mathrm{S}} + \mu_{\mathrm{del}}^{\mathrm{S}} + \mu_{\mathrm{egd}}^{\mathrm{S}} - \mu_{\mathrm{cm}}^{\mathrm{S}} \tag{8.2}$$

where $d_{i,j}^{(\mathrm{h})}$ is the distance score of nodes p_i^{Q} and p_j^{T} as stored in $DD^{\mathrm{S(h)}}$ (if such a triplet exists), $\mu_{\mathrm{ins}}^{\mathrm{S}}$ is the penalty score for node insertions, $\mu_{\mathrm{del}}^{\mathrm{S}}$ is the penalty score associated to node deletions, $\mu_{\mathrm{egd}}^{\mathrm{S}}$ is the penalty score associated to edge deletions and $\mu_{\mathrm{cm}}^{\mathrm{S}}$ is a bonus score to reward presumably correct matches. In particular, the three penalty scores are computed as follows:

- Let $E = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle\}$ be the set of edges in $\mathcal{G}^{\mathrm{Q}}$, each of which corresponding to a pair of triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$.
  Then:

  $$\mu_{\mathrm{ins}}^{\mathrm{S}} = \sum_{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle \in E} \pi_{\mathrm{ins}} \cdot n_{\mathrm{ins}} \cdot c_{i,l}$$

  where π_{ins} is a fixed given penalty associated to a single node insertion and n_{ins} is the number of nodes on the shortest path between p_j^{T} and p_k^{T} (if any).
- $\mu_{\mathrm{del}}^{\mathrm{S}} = |DD^{\mathrm{S(h)}}_{\mathrm{del}}| \cdot \pi_{\mathrm{del}}$ where π_{del} is the penalty associated to a single node deletion.
- Let $F = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle\}$ be the set of edges in $\mathcal{G}^{\mathrm{Q}}$, each of which corresponding to a pair of triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$ such that p_j^{T} and p_k^{T} are non connected in $\mathcal{G}^{\mathrm{T}}$. Then:

  $$\mu_{\mathrm{egd}}^{\mathrm{S}} = \sum_{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle \in F} \pi_{\mathrm{egd}} \cdot c_{i,l}$$

  where π_{egd} is a fixed given penalty associated to a single edge deletion w.r.t. $\mathcal{G}^{\mathrm{Q}}$.

  The bonus score $\mu_{\mathrm{cm}}^{\mathrm{S}}$, is computed as follows:

- Let $G = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,h}\rangle\}$ be the set of edges in $\mathcal{G}^{\mathrm{Q}}$, each of which corresponds to a pair of triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$, such that the edge $\langle\{p_j^{\mathrm{T}}, p_k^{\mathrm{T}}\}, c_{j,k}\rangle$ is in $\mathcal{G}^{\mathrm{T}}$. Then:

  $$\mu_{\mathrm{cm}}^{\mathrm{S}} = \sum_{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle \in G} \pi_{\mathrm{cm}} \cdot \frac{c_{i,l} + c_{j,k}}{2}$$

  where π_{cm} is a fixed given score associated to the correct match between the two edges in $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$.

Note that, in the formulae above, reliability factors c_{il} are exploited in order to weigh penalty and bonus scores between proteins by the probabilities that the corresponding interactions actually hold.

Refined similarity scores

Let $\mathcal{G}^{\mathrm{Q}}$ be the query protein interaction graph, $\mathcal{G}^{\mathrm{T}}$ be the target protein interaction graph, $DD^{(\mathrm{h})}$ be a distance dictionary involving all the pairs of proteins of $\mathcal{G}^{\mathrm{Q}}$ and $\mathcal{G}^{\mathrm{T}}$. Furthermore, let $\sigma^{(\mathrm{h})} = \langle \mathcal{G}^{\mathrm{S}}, DD^{\mathrm{S(h)}}+\rangle$, s.t. $DD^{\mathrm{S(h)}}+ = DD^{\mathrm{S(h)}} \cup DD_{\mathrm{in}}^{\mathrm{S(h)}} \cup DD_{\mathrm{del}}^{\mathrm{S(h)}}$ and $DD^{\mathrm{S(h)}} \subset DD^{(\mathrm{h})}$, be an approximate occurrence of $\mathcal{G}^{\mathrm{Q}}$ in $\mathcal{G}^{\mathrm{T}}$. The penalty scores μ_{ins}, μ_{del} and μ_{egd}, necessary to compute the refined similarities according to formula (8.1), are evaluated as follows:

- Let $E_i = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle\}$ be the set of edges incident onto p_i^{Q} in $\mathcal{G}^{\mathrm{Q}}$, each of which corresponding to a pair of triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$. Then:

  $$\mu_{\mathrm{ins}} = \sum_{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle \in E_i} \min\{n_{\mathrm{ins}}, I_{\mathrm{MAX}}\} \cdot c_{i,l}.$$

  where I_{MAX}, n_{ins} and I_{MAX} are as explained in the previous section.
- Let $DD_{\mathrm{del},i}$ be a subset of $DD^{\mathrm{S(h)}}+$ that contains the triplets $\langle p_l^{\mathrm{Q}}, \bullet, -\rangle$ such that the nodes p_l^{Q} are connected by an edge to p_i^{Q} in $\mathcal{G}^{\mathrm{Q}}$, and $n_{\mathrm{adj},i}$ be the number of nodes directly linked by an edge to p_i^{Q} in $\mathcal{G}^{\mathrm{Q}}$. Then:

  $$\mu_{\mathrm{del}} = \frac{|DD_{\mathrm{del},i}|}{n_{\mathrm{adj},i}}$$

- Let $F_i = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle\}$ be the set of edges incident on p_i^{Q} in $\mathcal{G}^{\mathrm{Q}}$, each corresponding to the triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$ such that there does not exist any path in $\mathcal{G}^{\mathrm{T}}$ connecting p_j^{T} and p_k^{T}. Then:

  $$\mu_{\mathrm{egd}} = \sum_{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle \in F_i} c_{i,l}$$

  The bonus score μ_{cm} of formula (8.1) is computed as follows:

- let $G_i = \{\langle\{p_i^{\mathrm{Q}}, p_l^{\mathrm{Q}}\}, c_{i,l}\rangle\}$ be the set of edges incident on p_i^{Q} in $\mathcal{G}^{\mathrm{Q}}$, each corresponding to the triplets $\langle p_i^{\mathrm{Q}}, p_j^{\mathrm{T}}, d_{i,j}^{(\mathrm{h})}\rangle$ and $\langle p_l^{\mathrm{Q}}, p_k^{\mathrm{T}}, d_{l,k}^{(\mathrm{h})}\rangle$ in $DD^{\mathrm{S(h)}}$ such that there exists the edge $\langle\{p_j^{\mathrm{T}}, p_k^{\mathrm{T}}\}, c_{j,k}\rangle$ in $\mathcal{G}^{\mathrm{T}}$. Then:

$$\mu_{\mathrm{cm}} = \sum_{\langle\{p_i^{\mathrm{Q}},p_l^{\mathrm{Q}}\},c_{i,l}\rangle\in G} \frac{c_{i,l} + c_{j,k}}{2}$$

8.3 Related Work

Network querying techniques, briefly surveyed below, as applied to biological networks, can be divided in two main categories: those searching for efficient solutions under particular topological constraints imposed on the query graph, e.g., the query is required to be a path, and those more general that, like PInG-Q, manage arbitrary query topologies.

Specific query topologies

Kelley et al. developed *PathBLAST* [97], a procedure to align two PPI networks in order to identify conserved interaction pathways and complexes. It searches for high scoring alignments involving two paths, one for each network, in which proteins of the first path are paired with putative homologs occurring in the same order in the second path.

The algorithm *MetaPathwayHunter*, presented in [165] solves the problem of querying metabolic networks, where the queries are multi-source trees. MetaPathwayHunter searches the networks for approximated matching, allowing node insertions (limited to one node), whereas deletions are not allowed.

The references [183] and [51] illustrate two techniques for network querying, called *QPath* and *QNet*. In particular, QPath queries a PPI network by a query pathway consisting of a linear chain of interacting proteins. The algorithm works similarly to sequence alignment, so that proteins in analogous positions have similar sequences. Interactions reliability scores of PPI networks are considered, and insertions and deletions are allowed. QNet is an extension of QPath in which queries can take the form of trees or graphs with limited tree-width.

As already stated, differently from the approaches surveyed above, our technique deals with arbitrary topologies in both the query and the target networks. In that, it is more closely related to the approaches described below.

General query topologies

The system *GenoLink* presented in [53] is able to integrate data from different sources (e.g., databases of proteins, genes, organisms, chromosomes) and query the resulting data graph by graph patterns with constraints attached to both vertices and edges. A query result is the set of all subgraphs of the target graph that are similar to the query pattern and satisfy the imposed constraints. The goals of [53] are clearly different from our own, since the aim here is that of comparing heterogeneous graphs via constrained network querying.

Ferro et al. in [67] presented a tecnique called *NetMatch*, a Cytoscape plug-in for network querying allowing for approximated querying. A query in NetMatch is a graph in which some nodes are specified and others are wildcards (which can match an unspecified number of elements). Although dealing, as we do, with approximate network querying, the technique in [67] mainly focuses on topological similarity, whereas our results are deeply influenced by information about *node similarities* as well. We argue that this information is essential for the analysis of PPI networks.

In [205], a tool for querying large graph datasets, called *SAGA*, is described. The tool allows for searching for all the subgraphs contained in a graph data-set that are similar to a query graph. The authors define a score of similarity between subgraphs based on the structural distances of the match, the number of mismatches and the number of gaps. An index-based heuristic is exploited for the purposes of query processing. SAGA has been successfully exploited to query biological pathways and literature data-sets, although it shows some limitations in dealing with dense and large query graphs.

In [231] the problems of path matching and graph matching are considered. An exact algorithm is presented to search for subgraphs of arbitrary structure in a large graph, grouping related vertices in the target network for each vertex in the query. Being an exact matcher, it is only practical for queries having a number of nodes as large as 20 in the query network, though its performances improve if the query is a sparse graph. However, for the same reason of being an exact matcher, this is the reference technique we chose in our comparative experiments (see, below, Section 8.4.1).

The techniques presented in [53, 67, 205, 231], are closely related to PInG-Q, but with the following differences: (*i*) none of them exploits edge labels to manage interaction reliability factors which, considered the diverse trustability of methods used to establish the various protein interactions to hold, are practically very relevant to correctly single out, in the target network, highly-probable matchers of the query network; (*ii*) our technique does not imply any constraint on the number of involved nodes or the density of the query subgraph; (*iii*) as far as we know, our technique is the first one naturally dealing with *edge deletions*. This way possible incompleteness in the available information about interactions is dealt with.

8.4 Experimental Results

In this section, we illustrate some preliminary results we obtained by running our algorithm on the four PPI networks of *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens*. In particular, as described in detail in Sections 8.4.1, we exploited *S. cerevisiae*, *D. melanogaster* and *C. elegans* networks to compare our results with those presented in [231] for the same organisms. Section 8.4.2 illustrates how the *S. cerevisiae* and *H. sapiens* networks have been queried to further assess the ability of our technique in recognizing conservations across species. To this end, we considered some well characterized groups of both *yeast* and *human* proteins for which the biological processes they are involved in are well known.

In the experiments, we fix the parameter values as follows: Factors π_{ins}, π_{del}, π_{egd} and π_{cm} have been all set to 1, α has been set to 0.3, β to 0.1, γ to 0.6 and I_{MAX} to 5.

Note that, within the preliminary test experiments discussed below, we did not perform a fine tuning on such parameters, which is deferred to further experimental work. The algorithm was implemented on a 3.4 GHz Pentium *IV* with 4 GB RAM. The resulting running times of the experiments vary from a minimum of 53 seconds to a maximum of about 17 minutes.

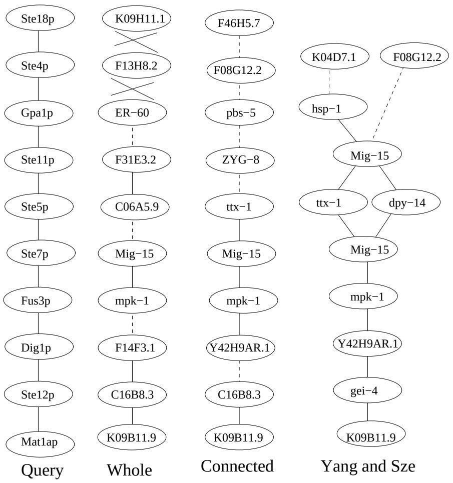
8.4.1 Querying *D. melanogaster* and *C. elegans* by *S. cerevisiae*

We compared our method with the one presented in [231]. In particular, we focused first on a path of the *S. cerevisiae* network to query the *C. elegans* network. This path, denoted by “Query” in Figure 8.3(b), corresponds to the longer mating-pheromone response pathway from the protein interaction network of *S. cerevisiae* [79]. The same figure also shows the output returned by the approach of Yang and Sze [231], and the outputs returned by our algorithm when run on “whole” the *C. elegans* network and on a “connected” part of it, resp. In the figures, graph nodes are labeled by protein names, dashed edges correspond to node insertions, whereas cross edges represent edge deletions. In particular, we obtained at most two node insertions per edge on this example, whereas Yang and Sze fixed a priori the maximum number of node insertions per edge to be equal to one. Note that PInG-Q does not require any such a limitation about the maximum number of node insertions to be fixed. The table in Figure 8.3(a) reports the E-values corresponding to the solutions returned by the considered approaches.

Looking at the results shown in Figure 8.3(b), the first important observation is that our algorithm is able to associate the MAP kinases *Fus3p* of *S. cerevisiae* with *mpk-1* of *C. elegans*. In this case, the associated E-value is equals zero, which agrees with the results reported in [231]. The results of both our executions agree with Yang and Sze also for the *S. cerevisiae* and *C. elegans* proteins *Ste7p/Mig-15* and *Mat1ap/K09B11.9*. Furthermore, the result on the connected subnetwork of *C. elegans* is the same of Yang and Sze also for *Ste4p/F08G12.2, Ste5p/ttx-1* and *Dig1p/Y42H9AR.1*. On the contrary, both executions of our algorithm returned a different result for the protein *Ste11p* of *S. cerevisiae* that, in [231], is paired again with *Mig-15* (which was paired with *Ste7p* as well). This incongruence might be caused for Yang and Sze admit multiple pairings of proteins; on the contrary, our approach search for one-to-one pairings. In any case, the result our approach returns is significant from the biological standpoint, since proteins *Ste11p* and *F31E3.2* both belong to putative serine/threonine-protein kinase family (as well as *Ste7p* and *Mig-15*). Results returned in both executions of our algorithm are slightly better, in terms of E-values, than those reported in [231] for the two proteins *Ste12p* and *Gpa1p*. Furthermore, we are able to pair also protein *Ste18p*, that Yang and Sze do not associate to any protein of *C. elegans*.

p_i^Q	E-value Yang and Sze	E-value Whole net	E-value subnet
Mat1ap	$3 \cdot 10^{-3}$	$3 \cdot 10^{-3}$	$3 \cdot 10^{-3}$
Ste12p	$1 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
Dig1p	$2 \cdot 10^{-5}$	$3 \cdot 10^{-4}$	$2 \cdot 10^{-5}$
Fus3p	0	0	0
Ste7p	$7 \cdot 10^{-34}$	$7 \cdot 10^{-34}$	$7 \cdot 10^{-34}$
Ste5p	$5 \cdot 10^{-3}/$ $6 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
Ste11p	$2 \cdot 10^{-35}$	$2 \cdot 10^{-17}$	$3 \cdot 10^{-14}$
Gpa1p	$1 \cdot 10^{-2}/-$	$4 \cdot 10^{-3}$	$2 \cdot 10^{-3}$
Ste4p	$2 \cdot 10^{-14}/$ $4 \cdot 10^{-15}$	$7 \cdot 10^{-10}$	$4 \cdot 10^{-15}$
Ste18p	–	$1 \cdot 10^{-3}$	$2 \cdot 10^{-3}$

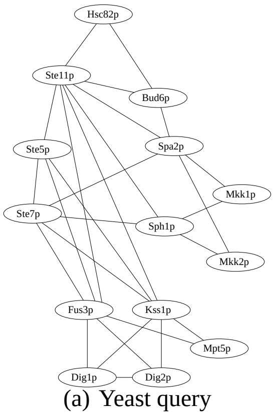
(a)



(b)

Fig. 8.3. Comparison on the longer *mating-pheromone response* pathway.

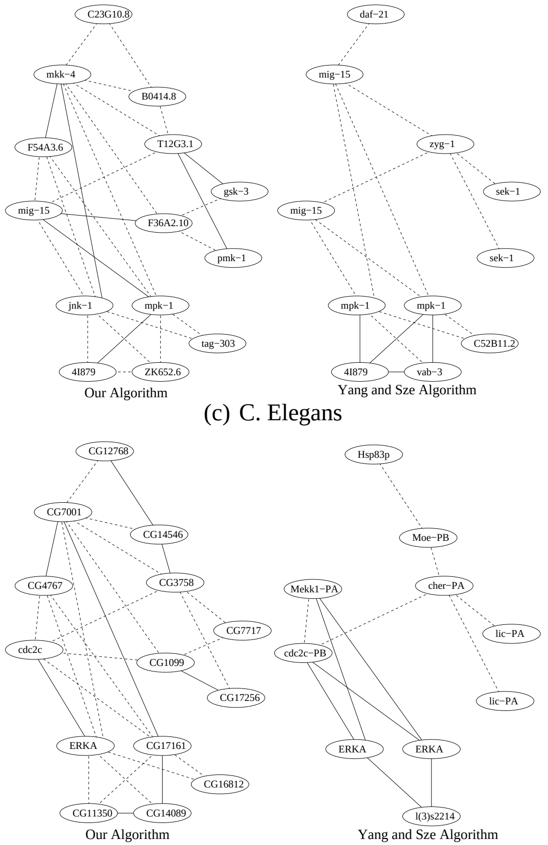
In the second example, the query is a yeast graph with general topology representing a related functional module from Spirin and Mirny [192]. Figure 8.4(a) illustrates the yeast query, Figure 8.4(b) shows a table containing the E-values corresponding to the results returned by our algorithm (applied on connected subnetworks) and by Yang and Sze, resp., when applied on both *C. elegans* and *D. melanogaster*. Figure 8.4(c) and Figure 8.4(d) illustrate the corresponding result subgraphs. In this experiment, the bait used to query *C. elegans* and *D. melanogaster* networks is a well characterized yeast signalling cascade. This yeast pathway controls peculiar yeast processes that are pheromone response (via $Fus3p$) and pseudohyphal invasive groth pathway (via $Kss1p$) through a so-called *MAPK* pathway (Mitoge activated protein kinase). The *MAPK* signalling cascades are likely to be found in all eukaryotic organism although the substrates phosphorylated by these kinases and the final response can be different in different organisms. Thus, in response to the query network, our technique retrives two *C. elegans* and *D. melanogaster MAPK* cascades (Figure 8.4(c)-left an Figure 8.4(d)-left), as suggested by the presence of several *MAPK* (i.e. proteins $mkk-4$, $pmk-1$, $mpk-1$, $jnk-1$ in *C. elegans*, and proteins *ERKA*, $CG7717$ in *D. melanogaster*, resp.) and other S/T kinases (i.e. proteins $mig-15$, $gsk-3$ in *C. elegans* and $CG7001$, $cdc2c$, $CG17161$ in *D. melanogaster*, resp.). This example illustrates well a peculiarity of our approach, that is, trying to find a good compromise between node similarity and network topology. In fact, the solution of [231] presents, in some cases, lower E-values than the correspondent ones in our solution, but our algorithm is able to pair all the proteins of the query network, which the technique in [231] does not, where three node deletions in *C. elegans* and four ones in *D. melanogaster*, respectively, can be observed.



(a) Yeast query

	C. Elegans		D. Melanogaster	
p_i^Q	E-val Our Alg.	E-val Yang and Sze	E-val Our Alg.	E-val Yang and Sze
Hsc82p	$1 \cdot 10^{-3}$	0	$2 \cdot 10^{-3}$	0
Ste11p	$4 \cdot 10^{-20}$	$2 \cdot 10^{-35}$	$2 \cdot 10^{-18}$	-
Bud6p	$1 \cdot 10^{-3}$	-	$3 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
Ste5p	$8 \cdot 10^{-4}$	–	$6 \cdot 10^{-3}$	$6 \cdot 10^{-3}$
Spa2p	$1 \cdot 10^{-4}$	$3 \cdot 10^{-5}$	$7 \cdot 10^{-4}$	$6 \cdot 10^{-3}$
Ste7p	$7 \cdot 10^{-34}$	$7 \cdot 10^{-34}$	$6 \cdot 10^{-21}$	$6 \cdot 10^{-21}$
Sph1p	$4 \cdot 10^{-3}$	-	$1 \cdot 10^{-3}$	-
Mkk1	$8 \cdot 10^{-19}$	$2 \cdot 10^{-44}$	$7 \cdot 10^{-25}$	$5 \cdot 10^{-49}$
Mkk2	$8 \cdot 10^{-24}$	$1 \cdot 10^{-43}$	$3 \cdot 10^{-31}$	$4 \cdot 10^{-48}$
Fus3p	$6 \cdot 10^{-61}$	0	0	0
Kss1p	$4 \cdot 10^{-97}$	$4 \cdot 10^{-97}$	$3 \cdot 10^{-19}$	$7 \cdot 10^{-94}$
Dig1p	$2 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	$8 \cdot 10^{-4}$	–
Dig2p	$1 \cdot 10^{-3}$	$2 \cdot 10^{-5}$	$4 \cdot 10^{-4}$	$1 \cdot 10^{-3}$
Mpt5p	$8 \cdot 10^{-3}$	$8 \cdot 10^{-3}$	$3 \cdot 10^{-3}$	–

(b)



(c) C. Elegans

(d) D. Melanogaster

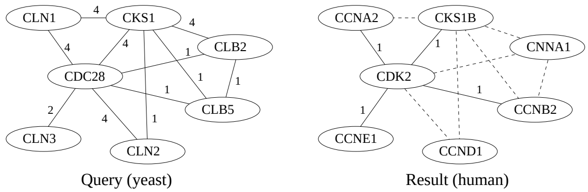
Fig. 8.4. Comparison on the functional module from Spirin and Mirny [192].

8.4.2 Querying *H. sapiens* by *S. cerevisiae*

In Figure 5 and Figure 6 two yeast queries are shown that are matched to the target human network. In particular, the query network in Figure 5 concerns proteins that control cell-cycle transitions. The progression through the cell-division cycle in eukaryotes is driven by particular protein kinases (*CDK*), which trigger the transition to the following phase of the cycle. These enzymes are serine/threonine kinases that require for their activation to be associated with regulative subunits known as cyclins. The query is composed of the budding yeast *S. cerevisiae* cyclin dependent kinase (*CDC*28) which associates with all different cyclins (*CLN*1, *CLB*2, *CLB*5, *CLN*2, *CLN*3). In yeast, different cyclins work in different phases of the cell cycle binding the same *CDK*. Mammalian cells, instead, have evolved multiple *CDK*s, each one working only with some cyclins. Consequently, in the human *CDK* network retrieved by applying our algorithm, some yeast interactions correspond to multiple-edge interactions in the human. For example, human cyclin *D* (*CCND*1) does not interact directly with *CDK*2 (*CDK*2) because it binds the homologs *CDK*4 and *CDK*6, but they have as a common partner the inhibitory protein *p*21 (*CDKN*1*A*) that is found as a node insertion in our approach (not explicitly shown in Figure 5). Instead cyclin *A*2 (*CCNA*2) and cyclin *E* (*CCNE*1) are directly connected to *CDK*2.

p_i^{Q}	p_j^{T}	E-value
CDC28	CDK2	0
CLN1	CCNA2	$2 \cdot 10^{-10}$
CKS1	CKS1B	$1 \cdot 10^{-29}$
CLB2	CCNA1	$3 \cdot 10^{-39}$
CLB5	CCNB2	$7 \cdot 10^{-43}$
CLN2	CCND1	$7 \cdot 10^{-6}$
CLN3	CCNE1	$2 \cdot 10^{-4}$

(a)



(b)

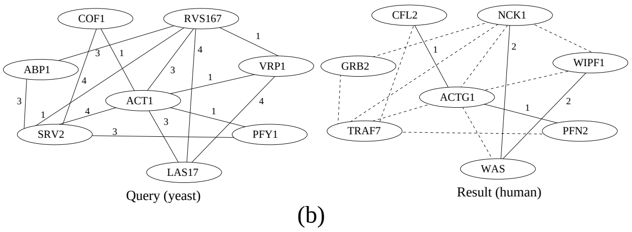
Fig. 8.5. Querying *H. sapiens* by *S. cerevisiae*: example 1

In the second experiment, we queried the human network with the yeast actin-related-proteins graph. Results are illustrated in Figure 6. Actin is well conserved among eukaryotes being a main component of the cytoskeleton. In yeast, it binds several proteins which regulate its polymerization/depolymerization and which are presented in the graph. Human homologs of the yeast proteins have been correctly paired (i.e., *ACT*1/*ACTG*1, *COF*1/*CFL*2, *VRP*1/*WIPF*1, *PFY*1/*PFN*2, *LAS*17/*WAS* in yeast and human, respectively). Furthermore, as in the previous example, the network has increased its complexity moving from yeast to human. Thus, while *PFN*2 and *CFL*2 are still directly linked to actin, an insertion node, not shown in Figure 6, divides the regulators *WIPF*1 and *WAS* from it.

This latter set of experiments has preliminarily confirmed that our technique is indeed able of retrieving biologically meaningful subgraphs matching the query network in the target one.

p_i^Q	p_j^T	E-value
ACT1	ACTG1	0
COF1	CFL2	$6 \cdot 10^{-27}$
RVS167	NCK1	$9 \cdot 10^{-10}$
VRP1	WIPF1	$4 \cdot 10^{-9}$
PFY1	PFN2	$2 \cdot 10^{-7}$
LAS17	WAS	$2 \cdot 10^{-14}$
SRV2	TRAF7	$4 \cdot 10^{-3}$
ABP1	GRB2	$4 \cdot 10^{-5}$

(a)



Query (yeast)

Result (human)

(b)

Fig. 8.6. Querying *H. sapiens* by *S. cerevisiae*: example 2

8.5 Concluding Remarks

In this chapter a novel approach to search for approximate occurrences of a query module in protein-protein interaction networks, based on bipartite graph weighted matching, has been presented. To summarize, the technique presents the following characteristics: *(i)* it manages graphs of arbitrary topology, both as query and as target networks, *(ii)* edge labels are used to represent and manage the reliability of involved interactions and *(iii)* node insertions, node deletions and edge deletions are dealt with. The preliminary experimental results are encouraging, since the approach is able to find significant results from a biological point of view while having a polynomial running time.

In next chapter an analysis and comparison of protein-protein interaction network querying techniques is provided.

9

Biological Network Querying Systems: Analysis and Comparison

Summary. This chapter analyzes and compares some recently proposed techniques to query biological networks, including the PInG-Q approach described in Chapter 8. In particular, the analysis performed in this chapter is meant to provide a comparative overview, which will be useful to understand problems and research issues, state of the art and opportunities for researchers working in this area.

Section 9.1 recalls the problem under consideration. Section 9.2 provides a basic comparison of the network querying techniques, based on: *(i)* the adopted network model, *(ii)* biological information exploited, *(iii)* exact versus approximate results and *(iv)* types of approximation supported. Section 9.3 describes the methods and systems by focusing on the types of networks they can handle. In Section 9.4, a further comparison is carried out by considering *(i)* the structures of the queries, *(ii)* exact versus heuristic algorithms, *(iii)* computational complexity and *(iv)* data used for the evaluation. Section 9.5 discusses the strengths and weaknesses of the considered approaches and, finally, Section 9.6 draws some conclusions.

9.1 Introduction

Network querying techniques search a whole biological network to identify conserved occurrences of a given query module, which can be used for transferring biological knowledge from one species to another (or possibly within the same species). Indeed, since the query generally encodes a well-characterized functional module (e.g., the MAPK cascade in yeast), its occurrences in the queried network (e.g., the MAPK cascade in human) suggest that the latter (and then the corresponding organism) features the function encoded by the former.

This chapter focuses on some techniques devised to query biological networks. In this respect, two important issues must be taken into account. The first one is that sub-graph isomorphism checking, which is a sub-problem of network querying, is a well-known NP-complete problem [75], thus limiting the applicability of exact techniques. The second one is that any effective approach should look for approximated, rather than exact, occurrences of the query sub-network. This way, the possible modifications of functional modules, determined by evolutive processes, can be taken into the right account [20].

In the last few years, the problem of querying biological networks has been studied by several researchers [51, 53, 67, 97, 165, 183, 205, 219, 231, 25, 170]. However, computational techniques for network querying are still at an early stage, thus making this research area still open and worth to investigate.

In this context, the goal of this chapter is to analyze and compare various facets of network querying algorithms, including the PInG-Q approach described in Chapter 8. In particular, the following specific aspects will be considered: *(a)* adopted network model; *(b)* biological information exploited (e.g., sequence similarity, interaction reliabilities, etc.); *(c)* delivery of exact versus approximate results; *(d)* types of approximation supported (e.g., node insertions and deletions); *(e)* handling of general versus specific types of network; *(f)* supported query structures; *(g)* adoption of exact versus heuristic algorithms; *(h)* computational complexity and *(i)* data used for the evaluation. Some relevant data pertaining the comparison carried out in this paper are listed in Table 9.2 (concerning points *(b)* - *(h)*) and Table 9.3 (concerning point *(i)*).

The analysis performed in this chapter is meant to provide a comparative overview on the network querying techniques developed in the last few years. This will help to understand problems and research issues, state of the art and opportunities for researchers working in this area.

The remainder of this chapter is organized as follows. The next section starts by providing some background information. Moreover, a basic comparison of the network querying techniques, focusing on points (a)-(d), is performed. Section 9.3 briefly describes the methods and systems and compares them w.r.t. point (e). In Section 9.4, a coarse-grain comparison is carried out w.r.t. points (f)-(i). Finally, Section 9.5 discusses the strengths and weaknesses of the considered approaches and 9.6 draws some conclusions.

9.2 Preliminaries

This section starts by recalling some background information about the network querying problem. Hence, network querying algorithms will be compared along the following directions:

(a) adopted network model;
(b) biological information exploited (e.g., sequence similarity, interaction reliabilities, etc.);
(c) delivery of exact versus approximate results;
(d) types of approximation supported (e.g., node insertions and deletions);

Some relevant data pertaining the comparison carried out in this section are listed in Table 9.2.

9.2.1 Biological Network Modeling

Biological networks, which store information about molecular relationships and interactions, as already discussed in Chapter 2, can be conveniently represented as graphs. A graph is built from of a set of nodes or vertices, representing cellular building blocks (e.g, proteins or genes), and a set of edges (directed or undirected), representing interactions (see Figure 9.1). A graph is a pair $G = (V, E)$, where V is the set of nodes and E is the set of edges, so that the elements from E are pairs of elements of V. In an undirected graph, an edge linking nodes A and B represents a mutual interaction. Conversely, in a directed graph, each edge represents the flow of material or information from a source node to a target node.

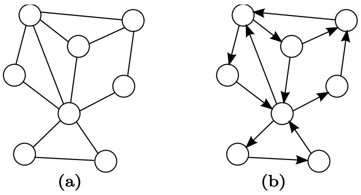


Fig. 9.1. An example of (a) undirected and (b) directed graph

As discussed in detail in Chapter 2. different types of graphs are used to represent different types of biological networks, each of which stores information about interactions related to specific entities or molecules [1]. Relevant kinds of networks for the scope of this chapter include metabolic networks and protein-protein interaction networks.

Some techniques[165, 205] proposed to query metabolic networks, represent the networks as directed graphs in which nodes represent enzymes and directed edges connect pairs of enzymes for which the product of the source enzyme is a substrate of the sink enzyme. Another reviewed technique[219] uses a directed graph in which nodes represent metabolites and directed edges represent enzymes that catalyze a reaction having the source metabolite as the reactant and the sink metabolite as the product. A slightly more complicated model is used in the last reviewed technique that handle metabolic networks [231], which considers two types of nodes, chemical compounds and enzymes. For each enzyme node, an incoming edge occurs with each of its substrate nodes and an outgoing edge occurs with each of its product nodes.

All the techniques proposed to query protein-protein interaction networks [97, 183, 231, 51, 170, 25], and analyzed in this chapter (encompassed PING-Q), model PPI networks as undirected graph in which the nodes represent proteins and the edges, that are possibly weighted, connect two proteins if they bind. However, only some of the analyzed techniques [183, 51, 170, 25] incorporate reliability information encoded as edge weights.

As already discussed in chapter 2, a biological network N is commonly represented by a graph $G^N = \langle V^N, E^N \rangle$, directed or undirected (see Figure 9.2), in which the set of nodes (or vertices) V^N denotes a set of cell building blocks (proteins, enzymes, metabolites, genes, etc.) and the set of edges E^N encodes the interactions between pairs of nodes.

In the most general definition, each edge $e_{ij} \in E^N$ takes the form of a triplet

$e^N_{ij} = \langle v_i, v_j, l_{i,j} \rangle$ where $v_i, v_j \in V^N$ are the interacting cell components and $l_{i,j}$ is the label associated to that edge (in PINs, for example, the edge label may encode the reliability of that interaction to actually occur).

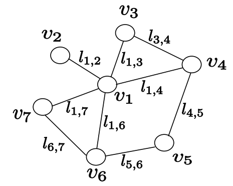


Fig. 9.2. An example of biological network graph G^N.

Biological networks can be queried in order to extract useful biological information. Let $G^Q = \langle V^Q, E^Q \rangle$ and $G^T \langle V^T, E^T \rangle$ be a pair of biological networks, corresponding to the sub-network used as query and the network to be queried, respectively. The goal of a network querying algorithm is to retrieve the sub-networks of G^T similar to G^Q.

9.2.2 Node Similarity Computation

Usually, the similarity between the nodes of the query network and the nodes of the target network is computed and exploited by querying algorithms. In our analysis we noted that only two techniques [53, 67] do not consider similarity between nodes. Similarity values, if exploited, are computed in different ways depending on the kind of the biological networks under inspection.

For example, in protein-protein interaction networks, similarity between proteins is often computed by exploiting the score obtained by aligning their amino acid sequences by exploiting existing tools such as BLAST (Basic Local Alignment Search Tool) [202] and the PRSS routine of the FASTA package [161]. The output of a BLAST and PRSS alignment is accompanied by an expectation value (the so called $E - value$). The lower the $E - value$, the more significant the alignment.
Among the analyzed techniques only one [170] exploits the PRSS routine, whereas all the others [97, 183, 51, 231, 25] (including PInG-Q) use BLAST. Another interesting remark is that the analyzed techniques differ from one another in the threshold value used to assess if two proteins are similar. As an example, PATHBLAST [97] considers two proteins similar if they are characterized by a BLAST $E - value$ small than or equal to 10^{-2}, whereas Torque [25] considers two proteins similar if their $E - value$ is less than 10^{-7}. Finally, differently from the other approaches, the PInG-Q software, discussed in the previous chapter, exploits the $E - value$ to compute a distance value rather than a similarity value (in particular, the lower the BLAST E-value, the lower the node distance).
Another way to assess protein similarity is by exploiting some databases like COG (Clusters of Orthologous Groups) [200] or KEGG (Kyoto Encyclopedia of Genes

and Genomes) [94]. These databases organize proteins into orthologous groups, so that two proteins are similar if they belong to the same group.

On the other hand, in dealing with metabolic networks, the similarity between pairs of enzymes is measured according to the EC (Enzyme Commission) classification, that is, a numbering system, consisting of four sets of numbers, that categorize the type of the catalyzed chemical reaction [206]. Note that the EC numbers give a functional classification that does not necessarily reflect sequence similarity. All the techniques for metabolic networks analyzed in this paper [165, 219, 231, 205] exploit EC-numbers to compute enzyme similarity.

9.2.3 Approximation Handling

Given a query network $G^Q = \langle V^Q, E^Q \rangle$ and a target network $G^T = \langle V^T, E^T \rangle$, a potential solution of the querying problem is a sub-graph of G^T, hereafter denoted by σ, which represents a (possibly approximated) occurrence of G^Q in G^T (see Figure 9.1 for a summary on the notation used in this paper). Approximation handling is needed for dealing with possible occurrences of evolution events modifying a network structure. This also allows to suitably take into account the significant number of both false negative and false positive interactions found when looking up existing databases. Overall, different types of approximation should be taken into account: *(i) node insertions*, corresponding to the addition of nodes in the target network; *(ii) node deletions*, corresponding to the additions of nodes in the query network; and *(iii) node mismatches*, corresponding to pairs of nodes characterized by a low similarity, but sharing similar biological characteristics (e.g., proteins performing the same function). Examples of evolution events that may affect protein-protein interaction networks are gene duplication, that causes the addition of new nodes (proteins), and link dynamics, corresponding to gain or loss of interactions through mutations in proteins [20].

Using approximate matching allows to obtain a solution σ in which: *(i)* some nodes belonging to G^Q may not correspond to any node of σ (node deletions); *(ii)* some nodes belonging to σ may not correspond to any node of G^Q (node insertions), and *(iii)* some corresponding pairs of nodes $\langle v^Q, v^T \rangle$ may have low similarity (mismatches), but the retrieved (approximated) occurrence σ of G^Q within G^T is still biologically meaningful. Figure 9.3 shows an example of a query network G^Q (Figure 9.3(a)) and a target network G^T (Figure 9.3(b)). A potential solution of the querying problem σ is shown in Figure 9.3(c). Note that σ is an approximate solution since it contains node insertions, node mismatches and node deletions.

It is important to point out that approximation occurrences should penalize the ranking of a given potential solution within the overall set of solutions. However, not all the approaches developed for network querying take into account the same types of approximations (Table 9.2). Rather, some of them [53, 67] search for sub-graphs that satisfy all the structural constraints imposed by the query. However, the analyzed network querying techniques use a scoring schema to rank the potential solutions.
For instance, as for PIN querying techniques, PATHBLAST detects the best solutions by computing a score that takes into account the probability of an actual homology

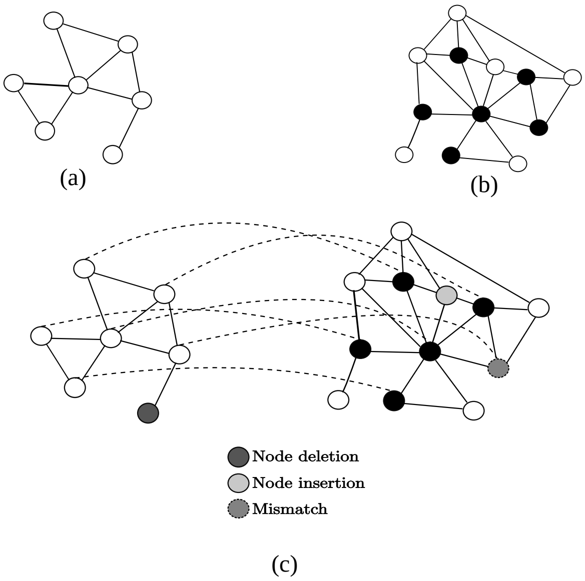


Fig. 9.3. (a) The query sub-network; (b) the target network; (c) a solution.

to occur within corresponding pairs of proteins (given their similarity value) and the probability that the interactions are true (and not false-positive). In other systems [183, 51, 231, 205, 170, 25] solutions are ranked according to the sequence similarity of matching nodes and penalties for introduced approximations (only node insertions and deletions [183, 231, 51, 25] or also mismatches [97, 231, 205, 170]). Besides, when applicable [183, 51, 170, 25], the ranking scores include also edge reliabilities. In this respect, PInG-Q is able to handle node insertions, node deletions and edge reliabilities.

On the other hand, all the techniques developed to query metabolic networks [165, 219, 205, 231] rank the potential solutions on the basis of matched enzyme similarities and penalties for approximations (only node insertions [165] or both node insertions and deletions [219, 231, 205]). Finally, one of the proposed technique [205] also takes into account graph structural differences, that is differences in node connectivity relationships.

9.2.4 Problem Statement

The *biological network querying problem* can be stated as follows:

Given a query sub-network G^Q and a target network G^T, the biological network querying problem consist in finding the solutions σ corresponding to matching G^T onto G^Q attaining the maximum scores, according to a given scoring schema.

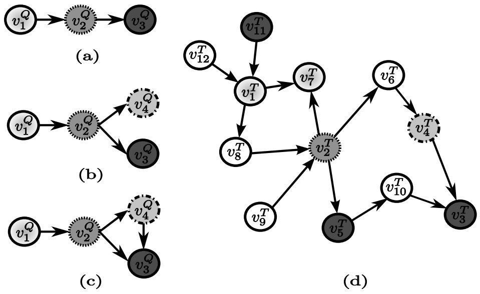
Symbol	Meaning
G^Q	The query sub-network
V^Q	The set of nodes of the query sub-network
E^Q	The set of edges of the query sub-network
G^T	The target queried network
V^T	The set of nodes of the target network
E^T	The set of edges of the target network
σ	A (possibly approximated) occurrence of G^Q within G^T

Table 9.1. Notation used in the chapter.

9.3 Methods

In the last few years, the problem of querying biological networks has been studied by several researchers. Hence, several tools [51, 53, 67, 97, 165, 183, 205, 219, 231, 25, 170] have been made available. Some of these tools were developed with particular focus on specific types of networks (e.g., protein-protein interaction networks [51, 97, 183, 231, 25, 170], or metabolic networks [165, 205, 231, 219]), while others were designed to be generally applicable, being these able to query any type of biological graph [53, 67].

In order to evaluate the above mentioned tools and PInG-Q in this chapter, a synthetic example shown in Figure 9.4 will be used throughout. Note that, all edge weights are assumed to be equal to one. Moreover, we do not use numerical similarity values, but we use ”high” or ”low” to denote high or low node similarity, respectively. If no similarity value is indicated, no relevant similarity is assumed to hold between the corresponding nodes. For techniques dealing with undirected graphs, undirected graphs underlining those shown in Figure 9.4 are considered. Note that in the following figures similar filling tones denote similar nodes.



Query node	Target node	Similarity value
v_1^Q	v_1^T	High
v_1^Q	v_7^T	Low
v_2^Q	v_2^T	High
v_3^Q	v_3^T	High
v_3^Q	v_5^T	Low
v_3^Q	v_{11}^T	Low
v_4^Q	v_4^T	High

(e)

Fig. 9.4. (a)-(c) Query examples; (d) target network and (e) similarity ratings.

9.3.1 Methods Developed to Query PPI Networks

Some of the approaches developed in the last few years to deal with the network query problem, as well as PInG-Q, are oriented to protein-protein interaction network analysis.

The first approach proposed in this context is *PATHBLAST* [97]. PATHBLAST in its original formulation identifies the conserved pathways across a pair of input networks. However, it has been subsequently extended to identify protein interaction complexes and pathways by aligning more than two networks [182, 93]. Nevertheless PATHBLAST was conceived to align the whole networks of two organisms, it can also be exploited to query a whole network against a specific pathway by merely using that pathway as one of the two input networks. The method starts by building a global alignment graph, where each node v represents a pair of similar proteins $\langle v^Q, v^T \rangle$, one from each of the input networks. Moreover, each edge represents either a conserved interaction, a gap (corresponding to both node insertions and deletions) or a mismatch. Each pathway in the global alignment graph corresponds to a sequence of conserved interactions across the two input PINs. The problem of finding the highest scoring path of length m in acyclic graphs can be solved in linear time in the number of edges. Nevertheless, the global alignment graph may contain some cycles. To overcome this difficulty, PATHBLAST generates $5 \cdot m!$ random acyclic sub-graphs by randomly deleting some of the edges, where m is the length of the query pathway. Then, it collects and combines the results discovered from each of those acyclic graphs. Note that the same protein pair cannot occur more than once in a resulting pathway and neither gaps nor mismatches can occur consecutively.

Example 1. As an example, assume that the query pathway and the target network shown in Figure 9.5 (a) and 9.5 (b) are given to PATHBLAST as input. The resulting global alignment graph is shown in Figure 9.5 (c). Each path of such a graph is a potential solution of the querying problem, thus the solution paths found by PATHBLAST are $\langle v_1^T, v_2^T, v_5^T \rangle$ and $\langle v_7^T, v_2^T, v_5^T \rangle$.

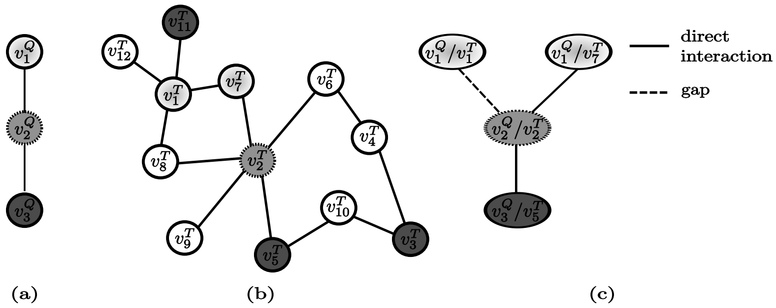


Fig. 9.5. (a) Query; (b) target network; and (c) *PATHBLAST* alignment graph.

QPath [183] and *QNet* [51] are other two techniques to query PINs, both based on the color coding technique. QPath limits itself to path-structured queries, while QNet is able to deal with queries shaped as trees or graphs having bounded treewidth. In both methods, the number of node insertions and deletions in the potential solutions are bounded by two threshold values, called N_{ins} and N_{del}, respectively. In a preprocessing phase, according to the color coding technique, QPath and QNet assign to each node a randomly chosen color from $\{1,\ldots,k+N_{ins}\}$ ($k+N_{ins}$ distinct colors are used to take into account the N_{ins} allowed node insertions). Several random coloring trials of the graph are to be executed since any particular query structure may be assigned non-distinct colors and, hence, may fail to be discovered. Both approaches exploit dynamic programming techniques to search for the best alignment. In particular, for each coloring, QPath searches for a path of length k that spans distinct colors. Similarly, QNet starts by rooting G^Q at a generic node r and proceeds by searching for the optimal colorful alignment. The algorithm used to handle tree queries can be easily extended to handle graph queries with bounded tree-width as well. In this case, a tree-decomposition $\langle X,T\rangle$ of G^Q is computed and the coloring method is extended to be applied to T, taking into account that: *(i)* a set of query nodes, representing a super-node of the tree-decomposition, may have an arbitrary topology (e.g., forming a clique) and *(ii)* a query node may appear in more than one super-node. However, in the current system release, only tree-shaped queries are handled. It should be finally noted that the two algorithms search for solutions involving at most N_{del} node deletions and both of them guarantee that each resulting solution includes distinct proteins.

Example 2. As an example, suppose $N_{ins} = N_{del} = 1$ and consider for QPath the query pathway and the target network represented in Figure 9.4(a) and Figure 9.4(d) and for Qnet the query tree and the target network represented Figure 9.6(a) and Figure 9.6(b). For this example, QPath finds the same result pathways discovered by PATHBLAST (see Figure 9.5 (c)) whereas QNet is able to retrieve the solution trees reported in Figure 9.6(c).

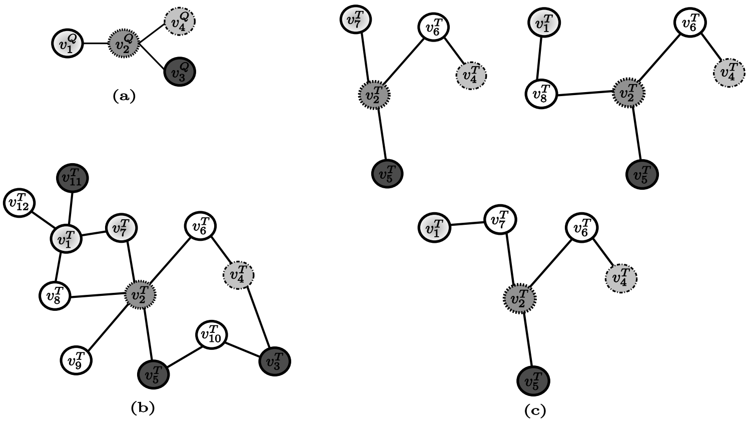


Fig. 9.6. (a) Query; (b) target network; and (c) *QNet* solutions.

Torque (TOpology-free netwoRk QUErying) [25] is another approach focused on querying PINs, where a bound on both the maximum number of node insertions and node deletions is imposed. Torque is a topology-free querying algorithm, that is, the query ”network” solely specifies the set of involved proteins, and does not carry out any information about the interactions among them. The observation underlying this choice is that most of the protein complexes reported in the literature are not correlated with any information about their interaction pattern. Thus, the goal of Torque is to find a connected set of proteins in the target network matching the query proteins. Torque has been implemented using several fixed-parameter algorithms based on dynamic programming. Each vertex in the target network is associated to a subset of colors, on the basis of the similarity scored to the query proteins. In a preprocessing phase, Torque assigns a different color to each query node. To handle node insertions, the algorithm is not applied to G^T, but it uses a new graph $G' = \langle V', E' \rangle$, such that for each node $v_i^T \in V^T$, a non-colored copy v_i' of v_i^T is added to V'. Moreover, an edge (v_i', v_j^T) and an edge (v_i', v_j'), such that the edge $(v_i^T, v_j^T) \in E^T$, are added to E'. Torque tries to find a solution to the querying problem by searching for a colorful tree. Note that each sub-graph has a spanning tree, so it is fair to search for colorful trees in lieu of colorful sub-graphs. The authors also provide an integer programming formulation of the querying problem to allow commercial solvers to be exploited.

Example 3. By applying Torque to the query proteins reported in Figure 9.7(a) and the target network in Figure 9.7(b), the solutions returned by Torque are shown in Figure 9.7(c). Note that only one node insertion and one node deletion are allowed in the solution sub-graphs and recall that Torque considers no query structural information.

Along the same line, another approach [170] has been developed. This approach, similarly to PInG-Q, imposes no simultaneous bound on the number of node insertions and deletions. However, while in PInG-Q neither the number of node insertions nor node deletions is ”ex-ante” bounded, the algorithm by Qian et al. [170], hereafter denoted *Qian*, imposes a bound only on the maximum number of node insertions. Qian is based on computing hidden Markov models (HMMs) and, as in PATHBLAST, the query stucture is constrained to pathways. In this framework, PPI are modelled using the HMM formalism that embeds into its probabilistic framework both protein similarities and interaction reliabilities. In particular, an hidden state v_i^T in the HMM corresponds to each protein $v_i^T \in V^T$ and the HMM has the same edge structure as G^T. On the one hand, in order to take into account node deletions, for each state v_j^T, a new state u_j^T is added to the HMM, and an outgoing edge from u_j^T to each state v_i^T in the neighborhood of v_j^T is added. Varying the transition probability $t(u_j^T|v_j^T)$, the probability to have deletions occurring can be controlled. Moreover, a self-transition at u_j^T is added and, suitably setting $t(u_j^T|u_j^T)$, the probability to have consecutive deletions is set up. On the other hand, to model node insertions, each state v_i^T in the HMM may emit a gap symbol ϕ. Setting the gap emission probability $e(\phi|v_j^T)$, the probability (and penalty) to have node insertions can be also tuned. Using the above construction, the problem is reduced to the one of finding the most prob-

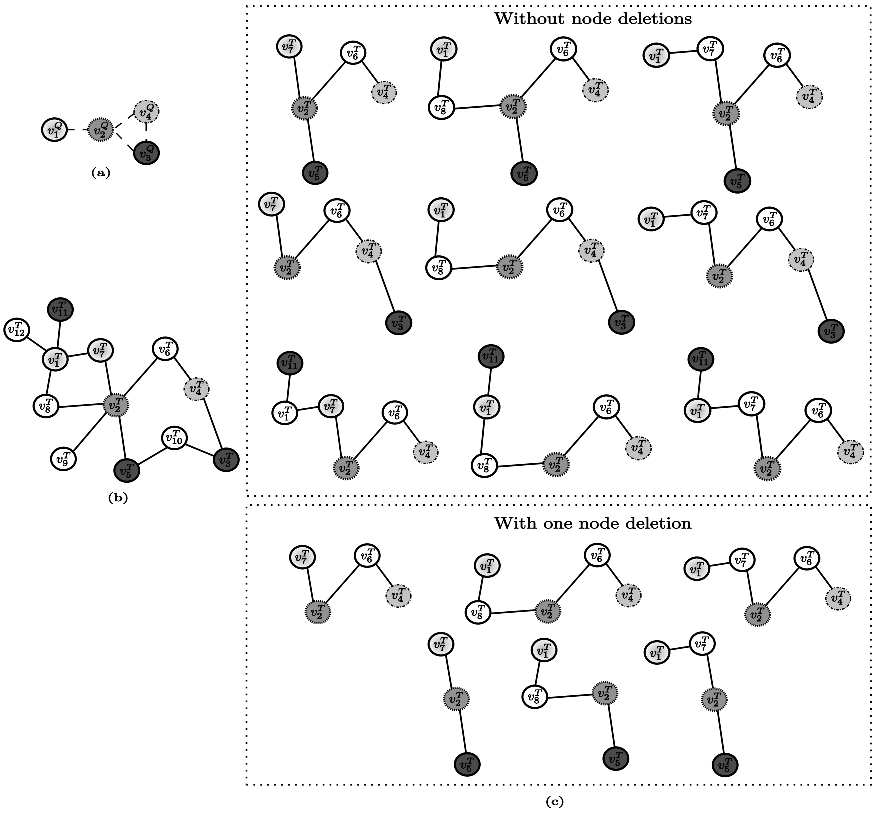


Fig. 9.7. (a) Query proteins; (b) target network; and (c) *Torque* solutions.

able path within the so constructed HMM. To retrieve the top *k* similar pathways, instead of just one, the *k* most probable paths are searched for.

Example 4. By considering PInG-Q as applied to the query graph reported in Figure 9.8(a) and the target network reported in Figure 9.8(b), the algorithm is able to find the solutions shown in Figure 9.8(c).

Example 5. As an example, by assuming that the maximum number of allowed node insertions is equal to 1, Qian applied to the query pathway represented in Figure 9.4(a) and the target network reported in Figure 9.4(b), is able to discover the same result pathways as those identified by PATHBLAST and QPath, and reported in Figure 9.5(c).

9.3.2 Methods Developed to Query Metabolic Networks

In this section, an overview on the techniques developed to query graphs encoding metabolic networks is given.

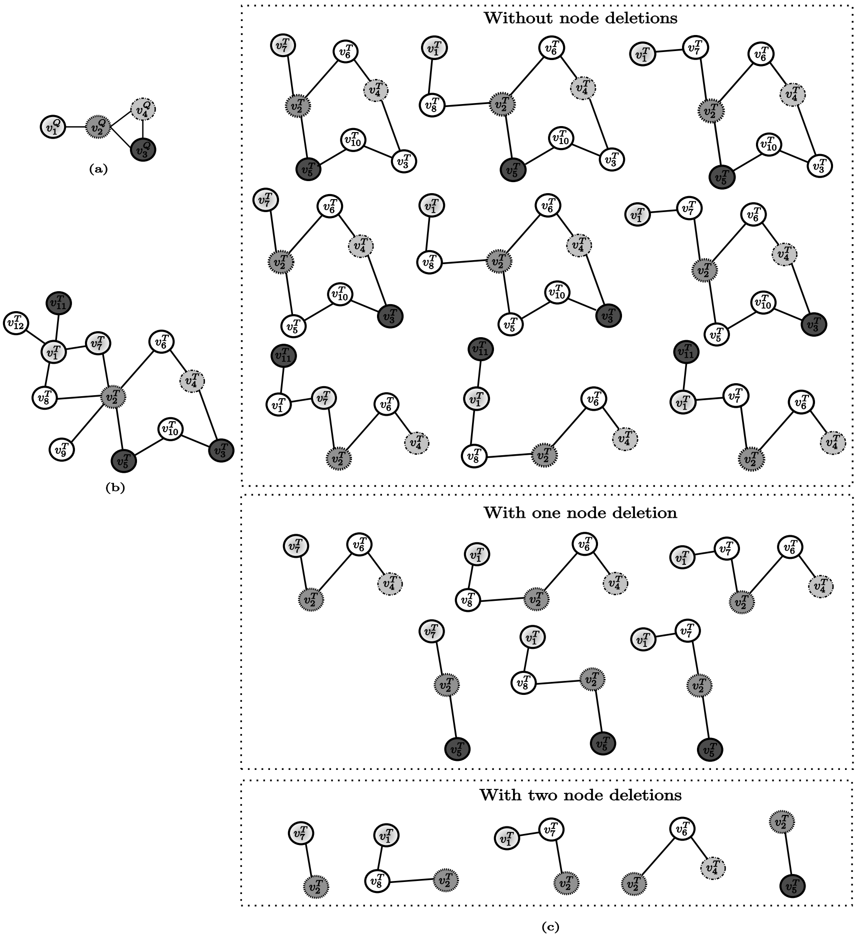


Fig. 9.8. (a) Query; (b) target network; and (c) *PInG-Q* solutions.

MetaPathwayHunter [165] is probably the first tool designed to work on metabolic networks. The system takes advantage of the particular topology of most metabolic pathways, usually shaped as multi-source trees (i.e., directed acyclic graphs whose underlying undirected graphs are trees). In fact, this tool only deals with queries and target networks shaped as multi-source trees. Moreover, it does not handle node deletions from the query module, but only node insertions in the retrieved target submodules (that can be also viewed as deletions from the target trees). The method exhaustively computes both all optimal solutions and several suboptimal solutions (up to a predefined threshold score), which are ranked by their statistical significance. All of the query and target nodes are labeled by the *EC-numbers* of the enzymes they encode. Moreover, a label scoring table, reporting the similarity scores between the target labels and the query labels, is built. The tradeoff between an insertion and a mismatch, in the retrieved solution, is established by tuning the node insertion score. This system exploits a bottom-up dynamic programming approach based on a subtree homeomorphism computation, which is based on the close relationship holding for subtree homeomorphism and weighted assignments in bipartite graphs. In particular, MetaPathwayHunter is based on the computation of the subtree of G^T for which, given a scoring table ad a node insertion penalty, the similarity score with G^Q is maximal. This problem is recursively translated into a collection of smaller problems, which are solved using weighted assignment algorithms.

Example 6. To apply MetaPathwayHunter to the example of Figure 9.9(a) it is necessary to modify the target network illustrated in Figure 9.4(d) as shown in Figure 9.9(b), since the tool requires a forest of multi-source trees. The solutions discovered by MetaPathwayHunter are shown in Figure 9.9(c) (note that at most one consecutive node insertion is allowed by the algorithm).

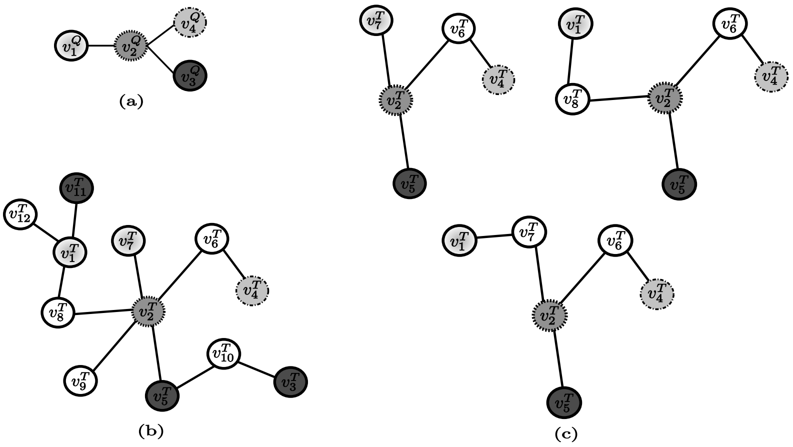


Fig. 9.9. (a) Query; (b) target network; and (c) *MetaPathwayHunter* solutions.

MetaPAT [219] deals with metabolic network querying as well. The underlying technique partitions the query vertices into two set: (a) path vertices, that are those vertices having exactly one incoming edge and one outgoing edge, and (b) branch vertices, that are all the other vertices. The authors of the system observed that branch vertices must be conserved, whereas paths can be elongated or shortened. The approach exhaustively examines all the sub-graphs of the target network that are homeomorphic to the query sub-graph. Two graphs are homeomorphic if their edges can be split (i.e., edges can be replaced by paths of arbitrary length in the same direction) in a way that the resulting graphs are isomorphic. The algorithm starts out by aligning a branch vertex of the query pattern with a branch vertex of the target graph, and then it uses a recursive sub-procedure to deal with all possible extensions of the attained partial solution. To reduce the search space, MetaPAT exploits the principle of *local diversity*. Given a real number $0 \leq f \leq 1$, a gap score g, a path

$p1$ with x vertices and a path $p2$ with y vertices, $p1$ and $p2$ fit if a maximum-score alignment between them aligns at most $\min\{\lceil(1-f)\cdot x\rceil, \lceil(1-f)\cdot y\rceil\}$ vertices to a gap. An extension of a partial solution fits if every simple path between two branch query vertices fits the corresponding simple path in the target network.

Example 7. If MetaPAT is applied to the query graph and target network reported in Figure 9.10(a) and Figure 9.10(b), it returns the solutions shown in Figure 9.10(c).

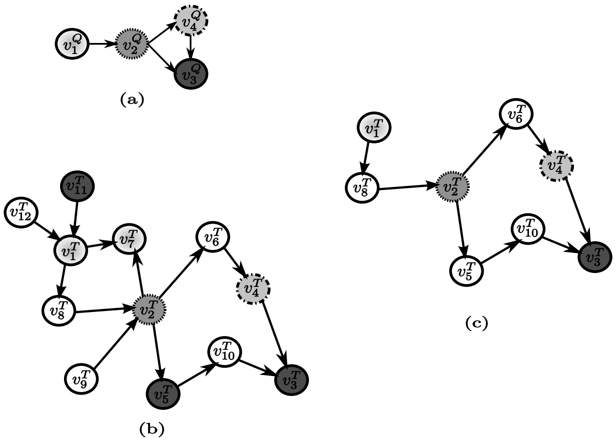


Fig. 9.10. (a) Query; (b) target network; and (c) *MetaPAT* solutions.

9.3.3 Methods Developed to Query Varied Biological Networks

This section illustrates techniques developed to query more than one kind of biological network (e.g., both protein interaction and metabolic networks).

SAGA (Substructure Index-based Approximate Graph Alignment) [205] is a general system to search for a query sub-graph in a database of graphs. A label is associated with each node of the query and each node of the graphs in the database with the aim to identify node mismatches. Indeed, if a node from the query and a node from a target graph have different labels, they correspond to a mismatch. The search is based on the construction of an index, called *Fragment Index*, containing substructures of size k extracted from the graphs in the database. In particular, for a subset of k nodes $v_1, \ldots, v_k$ extracted from a target graph, a pseudo edge between each pair of nodes $(v_i, v_j), i, j \in 1, \ldots, k$ is added if their distance $d(v_i, v_j)$ is less then a predefined threshold d_{max} (in order for node insertions to be allowed). This fragment is then added to the *Fragment Index* if the resulting subgraph is connected. During the search process, the query sub-graph is divided into fragments (i.e., sets of k nodes) in the same way as done for the database graphs. The fragments extracted from the query are then used to probe the *Fragment Index*. The matching fragments, retrieved from the index, are then combined into larger matches.

Example 8. Consider SAGA as applied to the query graph shown in Figure 9.4(c) and the target network reported in Figure 9.4(d) and recall that only one consecutive node insertion and only one node deletion is allowed. The resulting subgraphs retrieved by this approach are the same as those discovered by Torque (see Figure 9.7(c)).

PathMatch and *GraphMatch* [231] are two other examples of tools suitable for querying different biological networks. PathMatch has been proposed to search for paths, whereas GraphMatch to look for general graphs. A peculiarity of these two approaches is that each node of the target network may correspond to more that one node of the query sub-network. In a first phase, for each node $v_i^Q \in V^Q$, both algorithms build a set of correspondences $V_i = \{v_{i,1}, \ldots, v_{i,t}\}$, where $v_{i,1}, \ldots, v_{i,t} \in V^T$. In particular, $v_{i,1}, \ldots, v_{i,t}$ correspond to those nodes of G^T sharing a significant similarity with v_i^Q. Moreover, both algorithms fix the maximum number of allowed node insertions and mismatches for each direct edge in G^Q by a threshold value N_{ins}. While PathMatch takes advantage of the linearity of the query module, thus reducing the query problem to that of finding the longest weighted path in a directed acyclic graph, GraphMatch exploits an exact algorithm. In particular, PathMatch builds a directed graph $G' = \langle V', E' \rangle$, where $V' = \bigcup_{i=1}^{n} V_i \cup \{s, t\}$ and s and t are two additional nodes representing the source and the sink of paths in G'. Each vertex $v_{i,j}$ has associated a weight $s_{i,j}$, that encodes the similarity score between v_i^Q and the node of G^T associated to $v_{i,j}$. The weights for s and t are set to 0. An edge between the nodes $v_{i,j}, v_{i+d,l} \in G'$ is added to E' if (a) $0 < d \leq m$ (m bounds the number of node deletions) and (b) the number of nodes in the shortest path connecting the nodes corresponding to $v_{i,j}$ and $v_{i+d,l}$ in G^T is smaller than N_{ins}. Moreover, each node $v_{i,j}$ is connected by an edge to the source and sink node. Finally, each edge e has associated a negative weight proportional to the number of mismatches and gaps in the path it denoted. Clearly, the above construction reduces the path querying problem to that of finding a path P' in G' with the maximum sum of vertex and edge weights.

Example 9. By applying PathMatch to the query path shown in Figure 9.11(a) and the target network reported in 9.11(b) and allowing only one consecutive node insertion and one consecutive node deletion, the graph G', built by PathMatch, is shown in Figure 9.11 (c). Note that the *ids* used to identified the nodes of G' are the same *ids* of the nodes of G^T.

On the other hand, GraphMatch enumerates all the potential solutions so that the query process turns out to be effective only if the query network and the correspondence lists are small enough. To handle node deletions, the algorithm partitions V^Q into two sets V^- and V^+; the first set represents the set of nodes deleted in the result subgraph and the second one the set of nodes for which a corresponding node in σ exists. To solve the graph matching problem, all the connected induced subgraphs of G^Q are enumerated, to obtain all the possible partitioning way of V^Q into V^- and V^+. To enumerate all solutions, GraphMatch builds a graph $G' = \langle V', E' \rangle$, where $V' = \bigcup_{i=1}^{n} V_i$ and an edge between the pair of nodes $v_{i,j}$ and $v_{k,l}$ is added to E' if: *(a)* there is an edge in G^Q connecting v_i^Q and v_k^Q and *(b)* the number of nodes in the shortest path connecting the nodes corresponding to $v_{i,j}$ and $v_{k,l}$ in G^T is less than or

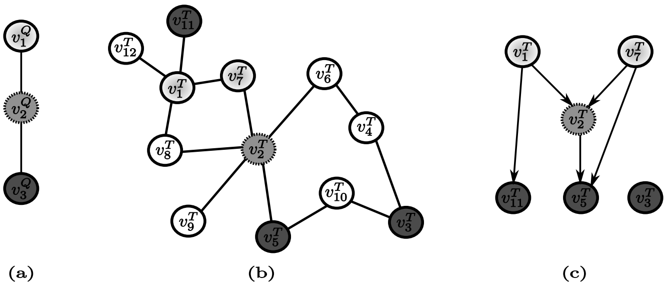


Fig. 9.11. (a) Query; (b) target network; and (c) *PathMatch* graph G'.

equal to N_{ins}. Given the set $V^+ \subseteq V^Q$, a valid solution is represented as a set of nodes $\{v_{i,j}, \ldots, v_{k,l}\}$, such that for each edge $\langle v_i^Q, v_k^Q \rangle \in E^Q$, the nodes $v_{i,j}$ and $v_{k,l}$ must be connected by an edge in G'.

Example 10. As an example, let N_{ins} be equal to 1, and suppose to apply GraphMatch to the query and target graphs shown in Figure 9.12(a) and 9.12(b), respectively. The subgraphs found out by GraphMatch are shown in Figure 9.12(c).

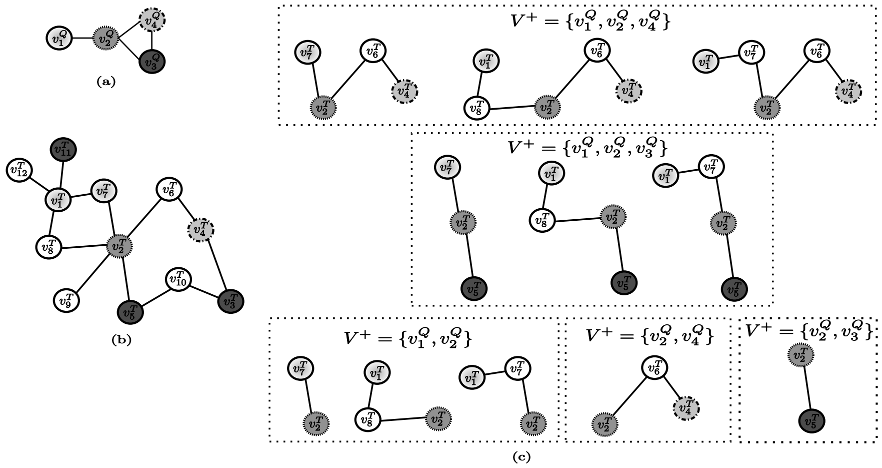


Fig. 9.12. (a) Query; (b) target network; and (c) *GraphMatch* solutions.

9.3.4 Methods Developed to Query General Biological Graphs

This section surveys on techniques developed for querying general graphs, the nodes of which may possibly denote biological entities.

GenoLink [53] is a software platform developed for graph querying and exploration. A query consists in a graph pattern in which nodes and edges are constrained.

The nodes of the graphs may represent biological objects (e.g., Organism, Gene, Chromosome, Protein) with the edges modeling the relationships holding among the nodes (e.g., Chromosome *BelongsTo* Organism, Gene *IstranslatedTo* Protein). In more detail, a GenoLink query is a graph pattern where nodes and edges are marked with data types. Moreover, nodes and edges may carry some algebraic expression constraints defined on the node or edge attributes. Finally, a query may define global algebraic expression constraints involving attributes of different vertices or edges. An occurrence of the query graph in the target graph is a subgraph of the target graph that must feature: *(a)* the same topology as the query graph, *(b)* all its nodes and edges must have the same data types (or subtypes) of corresponding query nodes and edges, *(c)* all the query constraints on attributes must be satisfied. In building the result set, the algorithm performs a depth-first search, which guarantees to find all matching sub-graphs.

Example 11. GenoLink as applied to the example queries shown in figures 9.4(a), 9.4(b) and 9.4(c) and the target network reported in Figure 9.4(d), is not able to return any solution, since there does not exist any subgraph of the target network that satisfy all the structural constraint imposed by the queries.

NetMatch [67] is another tool devised along the same ideas, which was built as a Cytoscape plugin[1]. NetMatch queries may be *structurally* approximated in the sense that some of their parts may be left unspecified. Each node and edge may have associated a list of attributes specifying query constraints. Thus, some elements of the query sub-graph are marked as constants, whereas others are unspecified. In particular, a node or an edge labeled with a wild card symbol '?' may correspond to any single value of a node or edge attribute, whereas an unspecified path (identified by a dashed edge in the query graph) may correspond to a path of length bounded by *n*, where *n* is a positive integer. The resulting sub-graphs are connected according to the same structure as the query graph. The query process starts by independently handling all maximal specified subparts and then combining the results of the sub-queries in all the possible ways. The combination process tries to connect the partial sub-graphs through all paths satisfying the approximate query paths. NetMatch is able to handle query and target graphs with more than one edge between a pair of nodes, loops (that are, edges starting and ending at the same node) and lists of attributes for each node and edge.

Example 12. Similarly to GenoLink, NetMatch as applied to the example queries shown in figures 9.4(a), 9.4(b) and 9.4(c) and the target network reported in Figure 9.4(d) does not return any solution. However, suppose to apply the algorithm to the query and target graphs shown in figures 9.13(a) and 9.13(b), respectively. The subgraphs found out by NetMatch are, in this case, shown in Figure 9.13(c).

9.4 Coarse-Grain Comparison

In the previous sections of this chapter, network querying tools have been analyzed with respect to: *(a)* adopted network model; *(b)* biological information exploited

[1] http://www.cytoscape.org

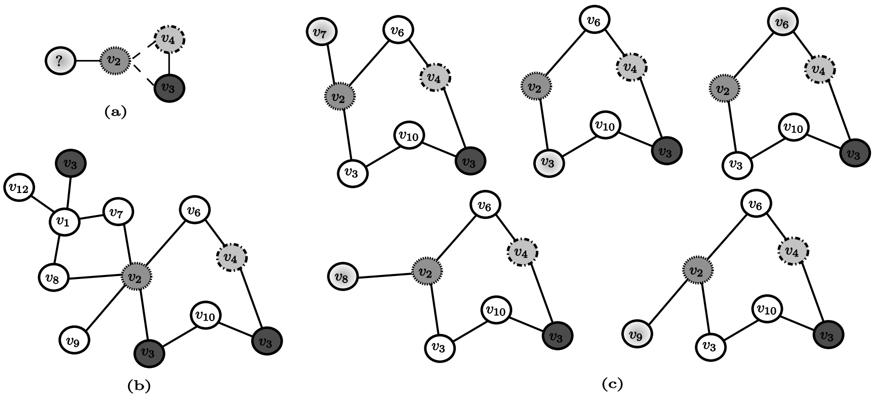


Fig. 9.13. (a) Query; (b) target network; and (c) *NetMatch* solutions.

(e.g., sequence similarity, interaction reliabilities, etc.); *(c)* delivery of exact versus approximate results; *(d)* types of approximation supported (e.g., node insertions and deletions); *(e)* handling of general versus specific types of network. In this section a comparison will be carried out for the given tools along the following directions:

(f) supported query structures;
(g) adoption of exact versus heuristic algorithms;
(h) computational complexity;
(i) data used for the evaluation.

Some relevant data pertaining the comparison carried out in this section are listed in Table 9.2 (concerning points *(f)* - *(h)*) and Table 9.3 (concerning point *(i)*).

9.4.1 Supported Query Structure

Network querying techniques can be classified with reference to the structural constraints imposed on the query networks. Some of the techniques here discussed (i.e., PATHBLAST, QPath, PathMatch and Qian) only handle path-shaped queries. Other techniques, such as MetaPathwayHunter and QNet, were developed to manage tree queries. Actually, QNet might also be used to search for graph queries (of bounded treewidth), but in the current system release this latter feature is not available. The most general techniques (i.e., GenoLink, NetMatch, SAGA, GraphMatch, MetaPAT and PInG-Q) can handle queries shaped as general graphs. Finally, Torque is a topology-free querying technique, where no information about the interaction pattern as encoded in the query graph is taken into account.

Moreover, the constraints in some cases imposed in the algorithms allow for the use of heuristic techniques to efficiently perform the search (for example, the color coding technique [3]). In some cases (e.g., for MetaPathwayHunter) the restrictions imposed on the query structure are dictated by the particular topology of the most interesting biological substructures in the biological networks of interest (e.g., metabolic networks).

9.4.2 Adoption of Exact versus Heuristic Algorithms

Because of the typical size of the graph structures encoding biological networks, the adoption of exact vs. heuristic search techniques can produce significant differences in performances. In fact, only five of the eleven approaches under analysis (i.e., GenoLink, NetMatch, MetaPathwayHunter, GraphMatch and Torque) implement exact algorithms. Thus, on the one hand, due to the complexity of the sub-graph matching problem (recall that sub-graph isomorphism is NP-complete [75]), exact algorithms can be applied only to small problem instances. On the other hand, since other methods (i.e., QPath, QNet, PATHBLAST, SAGA, MetaPAT, PathMatch, PInG-Q and Qian) exploit heuristic algorithms, they do not guarantee optimal solutions to be necessarily returned.

9.4.3 Computational Complexity

A further analysis dimension regards the computational complexity of the considered approaches. In this respect the following parameters are introduced:

- n is the number of nodes of the target network;
- m is the number of edges of the target network;
- q is the number of nodes of the query sub-network;
- N_{ins} is the maximum number of allowed node insertions;
- N_{del} is the maximum number of allowed node deletions.

Note that, for three of the analyzed techniques (i.e., GenoLink, SAGA and MetaPAT) complexity figures are not reported since complexity results are not available.

Polynomial Time Techniques

Some the analyzed techniques, that is, MetaPathwayHunter, PathMatch, PInG-Q and Qian, run in polynomial time. In particular, MetaPathwayHunter [165] has a time complexity of $O(\frac{q^2 n}{\log q} + qn \log n)$. Therefore, its running time is polynomial both in n and in q. PInG-Q's running time is $O(MAXITERATION \cdot n^3)$, where $MAXITERATION$ is a constant denoting the maximum number of iterations the algorithm is allowed to perform, and the factor n^3 is implied by the computation of the minimum bipartite weighted matching problem. The time complexity of PathMatch [231] is $O(m + n + k)$, where k is the number of best scoring hits returned by the algorithm. Finally, the time complexity of Qian [170] is $O(k \cdot q \cdot N_{ins} \cdot m)$ where, also in this case, k is the number of highest scoring pathways retrieved by the algorithm.

Exponential Time Techniques

Five of the discussed systems, that are, PATHBLAST, NetMatch, QPath, QNet, GraphMatch and Torque, run in exponential time in the number of nodes of the query sub-network. Therefore, these techniques are applicable only to relatively small problems. In detail, PATHBLAST [97] runs in time $O(q!l)$, where l is the number of edges of the global alignment graph. NetMatch's time complexity [67] is $O(q!q)$, while QPath [183] runs in $ln\frac{n}{\epsilon} \cdot 2^{O(q+N_{ins})} \cdot mN_{del}$, where ϵ is the probability that the algorithm does not find an optimal solution. The time complexity of QNet [51] is $ln\frac{1}{\epsilon} \cdot 2^{O(q+N_{ins})} \cdot m$ for tree queries and $2^{O(q)} \cdot n^{t+1}$ for bounded treewidth graph queries, where ϵ has the same meaning as above, and t is the maximum allowed treewidth of the query graph. The initial phase of GraphMatch [231], where all the connected subgraphs of G^Q are enumerated for the construction of all the potential solutions, runs in $O(2^q q^2)$. Finally, the computational complexity for Torque is $O(q!3^q mN_{ins}^2)$.

Note that the exponential trend in q might not be as much problematic as an exponential trend in n, since in real applications q is expected to be relatively small as compared to n.

9.4.4 Data Used for the Evaluation

The approaches developed for querying biological graphs were tested by their developers on different organism networks. The data used for the evaluation have been extracted from several databases, as reported in Table 9.3. All the techniques proposed to querying PINs [51, 97, 183, 231, 25, 170] (including PInG-Q) were evaluated on networks downloaded from *DIP* (Database of Interacting Proteins) [175], though some of them also used other databases to obtain additional information (e.g., functional classification) or to perform evaluations on different data. For example, QPath [183] and QNet [51] used *FlyGrid* (the section of *BioGRID* containing the interaction data pertaining the fly) [195]; PInG-Q also exploits data downloaded from *MINT* (Molecular INTeraction database) [33]; and Torque [25] downloaded the interaction data also from *Flybase* (a database of Drosophila genes and genomes) [72], *SGD* (Saccharomyces Genome Database) [37], *AmiGo* (Gene Ontology database) [31], *CORUM* (the Comprehensive Resource of Mammalian protein complexes) [174] and *HPRD* (Human Protein Reference Database) [168].

Similarly to the approaches working on PINs, the approaches proposed to querying metabolic networks [165, 231, 205, 219] were evaluated on datasets downloaded from several databases. In detail, PathMatch, Graphmatch and *SAGA* were evaluated on the data downloaded from *KEGG* (Kyoto Encyclopedia of Genes and Genomes) [94]. The information stored in *EcoCyc* [99] was used in PathMatch, Graphmatch and MetaPathwayHunter. Furthermore, this latter system also exploited the *SGD* [37] data, SAGA used *Reactome* [136] data and MetaPAT downloaded information from *BioCyc* [96]. Finally, GenoLink was evaluated on the data downloaded from *COG* [201], *InterPro* [87] and *BRENDA* [32]. The systems were evaluated on networks of different organisms (see Table 9.3). In detail, PATHBLAST was tested on the PIN of *S. cerevisiae* (yeast); QPath and QNet were evaluated using the networks of *S. cerevisiae*, *D. melanogaster* (fly) and *H. sapiens* (human); PathMatch was run on the networks of *S. cerevisiae*, *D. melanogaster*, *C. elegans* (worm), *H. pylori* (bacteria) and *E. coli* (bacteria); for GraphMatch, the networks of *S. cerevisiae*, *D. melanogaster*

Table 9.2. Comparison summary

	Types of networks *(e)*	**Query structure *(f)***	**biological information exploited *(b)***
PATHBLAST [97]	Tested on PPI networks	Pathways	BLAST E-values
MetaPathwayHunter [165]	Tested on metabolic pathways	Trees in a forest	Functional classification
QPath [183]	Tested on PPI networks	Pathways	Interaction reliability, BLAST E-values
GenoLink [53]	General	General graphs	None
QNet [51]	Tested on PPI networks	Trees or graphs with bounded treewidth	Interaction reliability, BLAST E-values
NetMatch [67]	General	General graphs	None
SAGA [205]	Tested on metabolic pathways	General graphs	Functional classification
PathMatch [231]	Tested both on PPI networks and metabolic pathways	Pathways	BLAST E-values, Functional classification
GraphMatch [231]	Tested both on PPI networks and metabolic pathways	General graphs	BLAST E-values, Functional classification
MetaPAT [219]	Tested on metabolic networks	General graphs	Functional classification
PInG-Q	Tested on PPI networks	General graphs	Interaction reliability, BLAST E-values
Torque [25]	Tested on PPI networks	topology-free	Interaction reliability, BLAST E-values
Qian [170]	Tested on PPI networks	Pathways	Interaction reliability, FASTA E-values

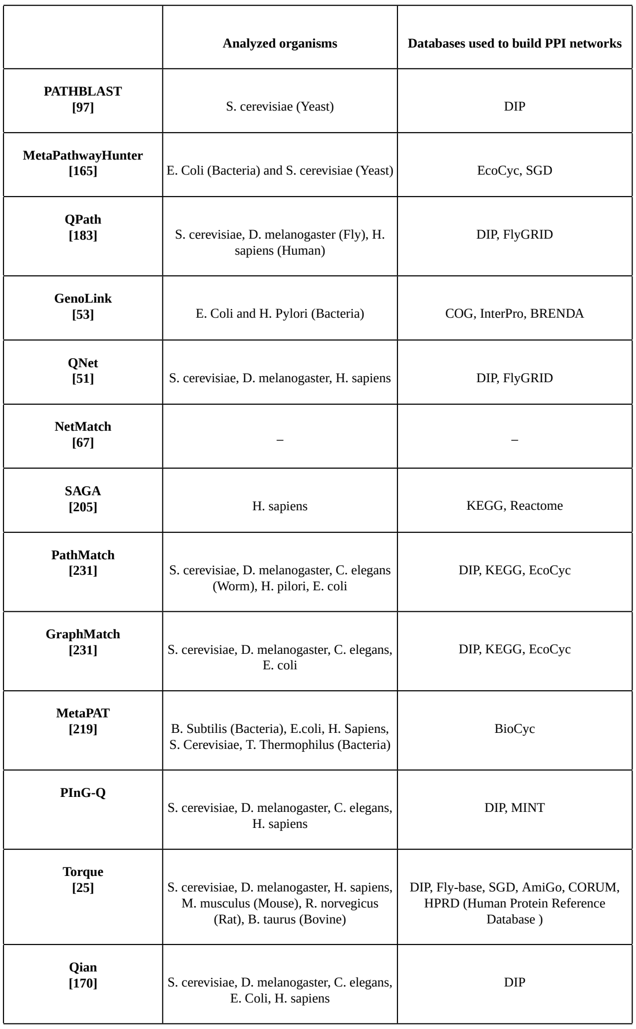
	Exact vs approximate results *(c)*	**Types of approximation *(d)***	**Exact vs heuristic algorithm *(g)***	**Time complexity *(h)***
PATHBLAST [97]	Approximate	Node insertions, node deletions, mismatches	Heuristic	$O(q!l)$
MetaPathwayHunter [165]	Approximate	Node insertions	Exact	$O(\frac{q^2 n}{\log q} + qn \log n)$
QPath [183]	Approximate	Node insertions, node deletions	Heuristic	$\ln \frac{n}{\epsilon} \cdot 2^{O(q+N_{ins})} \cdot mN_{del}$
GenoLink [53]	Exact	None	Exact	Not evaluated
QNet [51]	Approximate	Node insertions, node deletions	Heuristic	$\ln \frac{1}{\epsilon} \cdot 2^{O(q+N_{ins})} \cdot m \cdot N_{del}$ (trees) $2^{O(q)} \cdot n^{t+1}$ (bounded treewidth graphs)
NetMatch [67]	Exact (but wildcards allowed in the query)	None	Exact	$O(q!q)$
SAGA [205]	Approximate	Node insertions, node deletions, mismatches	Heuristic	Not evaluated
PathMatch [231]	Approximate	Node insertions, node deletions, mismatches	Heuristic	$O(m + n + k)$
GraphMatch [231]	Approximate	Node insertions, node deletions	Exact	$O(2^q q^2)$
MetaPAT [219]	Approximate	Node insertions, node deletions	Heuristic	Not evaluated
PInG-Q	Approximate	Node insertions, node deletions	Heuristic	$O(MAXITERAION \cdot n^3)$
Torque [25]	Approximate	Node insertions, node deletions	Heuristic	$O(q!3^q mN_{ins}^2)$
Qian [170]	Approximate	Node insertions, node deletions, mismatches	Heuristic	$O(kqN_{ins}m)$

and *C. elegans* were used; PInG-Q was tested on the networks of *S. cerevisiae*, *D. melanogaster*, *C. elegans* and *H. sapiens*; Torque was evaluated on the networks of *S. cerevisiae*, *D. melanogaster*, *H. sapiens*, *M. musculus* (mouse), *R. norvegicus* (rat) and *B. taurus* (bovine); Qian was tested on the networks of *S. cerevisiae*, *D. melanogaster*, *H. sapiens*, *C.elegans* and *E. Coli*.

With regard to the systems tested on metabolic networks, MetaPathwayHunter was evaluated using the networks of *E. Coli* and *S. Cerevisiae*; SAGA on the networks of *H. Sapiens* and MetaPAT on the networks of *B. Subtilis* (bacteria), *E.coli*, *H. Sapiens*, *S. Cerevisiae* and *T. Thermophilus* (bacteria). Finally, Genolink was evaluated on the networks of *E. Coli* and *H. Pylori*.

The quality of the results obtained by the different systems were evaluated from a biological point of view, manually or using some well known information stored in biological databases, such as Swiss-Prot [11]. In particular, some well-known query modules characterizing some model organisms (e.g., the MAP kinase cascade of yeast) were used as benchmark. Also the capabilities of the techniques were stressed by using less characterized modules and organisms (e.g., the fly).

Table 9.3. Exploited organisms and data sets



	Analyzed organisms	**Databases used to build PPI networks**
PATHBLAST [97]	S. cerevisiae (Yeast)	DIP
MetaPathwayHunter [165]	E. Coli (Bacteria) and S. cerevisiae (Yeast)	EcoCyc, SGD
QPath [183]	S. cerevisiae, D. melanogaster (Fly), H. sapiens (Human)	DIP, FlyGRID
GenoLink [53]	E. Coli and H. Pylori (Bacteria)	COG, InterPro, BRENDA
QNet [51]	S. cerevisiae, D. melanogaster, H. sapiens	DIP, FlyGRID
NetMatch [67]	–	–
SAGA [205]	H. sapiens	KEGG, Reactome
PathMatch [231]	S. cerevisiae, D. melanogaster, C. elegans (Worm), H. pilori, E. coli	DIP, KEGG, EcoCyc
GraphMatch [231]	S. cerevisiae, D. melanogaster, C. elegans, E. coli	DIP, KEGG, EcoCyc
MetaPAT [219]	B. Subtilis (Bacteria), E.coli, H. Sapiens, S. Cerevisiae, T. Thermophilus (Bacteria)	BioCyc
PInG-Q	S. cerevisiae, D. melanogaster, C. elegans, H. sapiens	DIP, MINT
Torque [25]	S. cerevisiae, D. melanogaster, H. sapiens, M. musculus (Mouse), R. norvegicus (Rat), B. taurus (Bovine)	DIP, Fly-base, SGD, AmiGo, CORUM, HPRD (Human Protein Reference Database)
Qian [170]	S. cerevisiae, D. melanogaster, C. elegans, E. Coli, H. sapiens	DIP

9.5 Discussion

In this paper, a comparative survey of the methods developed to query biological networks has been carried out. As implied by the previous descriptions, those techniques are rather different from one another. There are methods that, due to the adoption of exact algorithms (e.g., GraphMatch [231]), or becacause they handle generic graph queries (such as NetMatch [67]), result in rather high time complexity. Improvements in execution times are obtained by exploiting heuristic algorithms, like in PInG-Q, by restricting query structure (e.g., as done in PathMatch [231]), or by only allowing few types of approximations in the result sub-graphs (e.g., in MetaPathwayHunter [165]).

However, all the tools seem able to find biologically significant results. Since all methods are accurate with respect to the ”biological” quality of the returned results, it is sensible to look at both the application domains and the complexity in order to find the best method to use.

In this respect, for pathway queries, the best choices seem to be PathMatch [231] and Qian [170], which are able to deal with node mismatches, insertions and deletions and have the lowest time complexity among the considered systems (linear time complexity in target and query network size).

For queries shaped as general graphs the most promising tools seem to be PInG-Q and GraphMatch [231]. PInG-Q has the advantage of exploiting a heuristic technique that keeps the time complexity low, while at the same time considering both node insertions and deletions in the result sub-graph. GraphMatch, on the other hand, since exploiting an exact search algorithm, guarantees to find the best solution according to the adopted scoring schema. Finally, Torque [25] proves itself to be quite appropriate for it opening an appealing view on the topology-free querying issue.

As a general trend, most of the tools do not yet exploit all kinds of biological additional information (e.g. GO terms or interaction reliability coefficients) that might improve the quality of the returned result. Besides, the most part do not take into account all the possible biological diversities (e.g., approximations in resulting subgraphs) that might permit to obtain more accurate results.

Despite these limitations, the efforts within this research area have been steadily increasing in the last few years. As such, this area seems to be a promising research domain in the quest toward improving the knowledge about biological data and mechanisms at the basis of life processes.

9.6 Concluding Remarks

In this chapter the analysis and comparison of some techniques proposed to query biological networks has been carried out. This analysis considered the biological network querying problem from different perspectives, ranging from structural properties of the networks (e.g., query subnetwork shape constraints) to computational complexity of network querying algorithms. Despite the performed comparison, it has not been possible to identify the “best” method in the absolute sense, since the most performant algorithms, in terms of running time, produce approximated results, whereas exact algorithms are very time consuming. This analysis has been useful for better understanding research issues and future directions to improve the quality of solutions to the biological network query problem.

Part V

Conclusions and Future Trends

10

Conclusions and Future Trends

Summary. The content of this thesis concerned three main strands of research. The first investigated the problem of predicting protein functions. The second one studied the problem of aligning protein-protein interaction (PPI) networks. Finally, the last one dealt with the biological network querying problem, with particular emphasis on querying PPI networks. In this chapter, the content of this work will be summarized by remarking its main contributions. Besides, here a brief overview on future trends in the fields of research related to this thesis will be provided.

10.1 Content Summary

This section recaps the content of this thesis, briefly discussing the various research issues that have been investigated.

A road map of this work has been given in Chapter 1. The motivations of this thesis have been laid out in Chapter 2 by analyzing simple (i.e., proteins) and complex (i.e., biological networks) biological structures. This allowed to identify and investigate some relevant problems concerning these structures. Both the fundamental role played by proteins in living organisms and the complex set of molecular interactions regulating cell life cycle have been described. Besides, an overview on the most important bioinformatics tasks related to these simple and complex biological structures has been provided. This has been useful to understand the open perspectives in this research field, which have been tackled in the subsequent parts of the thesis.

In Part II, the problem of predicting protein functions has been analyzed. In particular, Chapter 3 charted the state of the art in this research area, which helped to motivate the two novel approaches proposed in Chapter 4 and Chapter 5. In particular, in Chapter 4, an approach for predicting protein quaternary structures, called PQSC-FCNN, has been illustrated. PQSC-FCNN exploits protein functional domain information and the Fast Condensed Nearest Neighbor (FCNN) rule [6]. PQSC-FCNN is able to reduce both the portion of the dataset to be used and the number of comparisons to carry out at classification time. This allows sensible space and time savings, while achieving very good accuracy. In Chapter 5, an approach called Bi-Grappin, for annotating proteins with functional information by comparing PPI networks, has been presented. The algorithm is based on the exploration and comparison of proteins neighborhoods (interaction profiles). The basic idea is that proteins with similar neighborhoods are probably involved in similar biological processes. One peculiarity of this approach is its capability of incorporating both quantitative (i.e., interaction strengths) and reliability information about interactions. The quantitative information is used to distinguish nodes belonging to different neighborhoods. The reliability, that is determined by the experimental method used to detect the interaction, is taken into account in the computation of neighborhood similarity.

Part III has covered issues regarding the alignment of protein-protein interaction networks. In particular, in Chapter 6 the state of the art on PPI networks alignment has been analyzed to unearth the open research paths in this context. This analysis was essential to motivate the Sub-Grappin tool that has been introduced in Chapter 7. In particular, the goal of this approach is that of discovering common modules in PPI networks by exploiting the similarities between pairs of nodes belonging to different networks. The algorithm is based on the iterative alternation of two sub-stages: protein similarity refining, and connected sub-graphs extraction. The first stage is based on Bi-Grappin, while the second one consists in a node collapsing technique, called Collapse.

Finally, Part IV has dealt with the problem of querying biological networks. In particular, in Chapter 8, a novel approach, called PInG-Q, has been proposed. PInG-Q searches for approximated occurrences of a query module in protein-protein interaction networks by iteratively computing a minimum weighted bipartite matching. This technique has the following peculiarities: *(i)* query and target networks of arbitrary topology can be handled *(ii)* interaction reliability information is taken into account by incorporating it in edge labels *(iii)* node insertions, node deletions and edge deletions are allowed. Finally, in Chapter 9, an analysis and comparison of biological network querying algorithms, including PInG-Q, has been carried out. This analysis considered the biological network querying problem from different perspectives to provide the reader with a rich overview on the existing techniques. The comparison ranges from structural properties of the input networks (e.g., query subnetwork shape constraints) to computational complexity. This analysis has been useful for highlighting open problems and research opportunities in this field.

10.2 Contributions

The research developed in thesis has been motivated by identifying a set of issues and requirements in the bioinformatics research area (see Chapter 1). In the following, we summarize the contributions of this thesis, which tackled relevant bioinformatics tasks. Focus is given to both simple and complex biological structures.

10.2.1 Simple Biological Structures

By looking at proteins as independent macromolecules, a relevant task is that of predicting protein functions, with the aim of properly understanding the role of uncharacterized proteins within living cells. To this purpose, two approaches have been devised: PQSC-FCNN and Bi-Grappin.

PQSC-FCNN

PQSC-FCNN is a novel method for protein quaternary structure classification, which is able to exploit protein functional domain information and the Fast Condensed Nearest Neighbor (FCNN) rule. Most of the approaches for protein quaternary structure prediction, previously proposed in the literature, were only tested on homo-oligomeric proteins. Besides, all of them need an entire dataset (training set) of proteins with known quaternary structure to be exploited for classifying an unclassified protein. In particular, each unclassified protein has to be compared to each protein belonging to the dataset. Differently from all previous methods, PQSC-FCNN has been tested on both homo-oligomers and hetero-oligomers and has been proved to be more efficient than other techniques. Indeed, PQSC-FCNN extracts a representative subset of the training set and uses this subset (instead of the whole training set) during the classification. This enables to reduce the total number of comparisons to be carried out without any significant loss in precision.

Bi-Grappin

Bi-Grappin is a novel method for transferring biological knowledge about protein functions, from characterized to uncharacterized proteins, by comparing PPI networks. In particular, this tool is useful for discovering the biological process in which the uncharacterized proteins of a given organism are involved. Given two PPI networks of two different organisms, Bi-Grappin identifies the most similar characterized proteins in the second network starting from the set of uncharacterized proteins of the first network. In particular, the most similar protein pair is determined by considering both sequence and interaction profile similarities. The advantage of Bi-Grappin, as compared to other techniques, is its ability of incorporating both quantitative (interaction strengths) and qualitative (interaction reliabilities) information in the analysis of the input networks.

10.2.2 Complex Biological Structures

The observation that proteins, and macromolecules in general, can be better characterized by analyzing their interaction patterns suggests the development of graph-based techniques to analyze and compare biological networks. This allows to infer new information about cellular activity and evolutive processes of the species. In this context, two techniques have been devised, that are Sub-Grappin and PInG-Q.

Sub-Grappin

Sub-Grappin is a novel method for discovering similar sub-graphs, possibly representing similar functional modules, across the PPI networks of two different species.

Sub-Grappin exploits Bi-Grappin as a submodule during the sub-graph extraction phase together with the Collapse technique. The iterative alternation of these two submodules led to the final achievement of two collapsed networks (corresponding to the two different organisms under consideration) in which corresponding macronodes identify similar subgraphs. Also in this case, the main benefit of Sub-Grappin is the possibility of exploiting both reliability and quantitative information, which can make the sub-graph search more accurate, as also confirmed by experimental analysis.

PInG-Q

PInG-Q is a novel method for querying PPI networks based on a two phases: *(i)* global alignment and *(ii)* similarity refinement. PInG-Q, starts by globally aligning the query and the target networks by considering node pairs similarities. Then, it refines the similarities of the corresponding nodes on the basis of how much the alignment satisfies the structural constraints imposed by the query (i.e., how much the query interactions are conserved). The main advantages of PInG-Q, w.r.t. previously existing tools, are: *(i)* its ability to handle query and target networks shaped as general graphs and *(ii)* to take into account reliability information. While having such good properties, PInG-Q runs in polynomial time in the size of the target network.

10.3 Future Trends

This section outlines possible future research directions related to the main topics discussed in this thesis.

As for the protein quaternary structure classification, some work is required on protein representation. Indeed, by enriching the set of features used to represent a protein (currently only the protein functional domain composition is exploited), the classification accuracy might be improved. For instance, the representation may take into account also protein sequence information (e.g., amino acid composition) or some knowledge about the protein secondary structure.

Concerning Bi-Grappin, an immediate extension is the adaptation of this technique to other types of biological networks (e.g., metabolic pathways or gene networks). In this respect, efforts should mostly involve the initial phase of node similarity computation. Furthermore, Bi-Grappin can be extended to search for similarities in multi-aligned networks instead of just one pair of networks. Supposedly, such an extension would be easily achieved since it only requires to exploit a multipartite graphs maximum weight matching algorithm instead of a bipartite one.

Also for the similar sub-graphs extraction problem, a future research direction is the extension of Sub-Grappin to align multiple networks and deal with other types of biological networks instead of only PPI networks.

Finally, as for the network querying problem, possible extensions of PInG-Q can concern the “fixing” of some pairs of corresponding nodes, for which the homologous of the query proteins in the target network are known. This can help the biologists to guide the algorithm toward better solutions where known correspondences between proteins are imposed. Also in this case, a desirable extension would be the adaptation of PInG-Q to query other types of biological networks.

10.4 Concluding Remarks

The work presented in this thesis has discussed some main strands of research in bioinformatics. In particular, a few hints and some real solutions to interesting bioinformatics tasks have been given. Finally, some tips regarding possible improvements of the proposed techniques have also been sketched.

Bioinformatics is a very active field of research and widely investigated. Indeed, many are the contributions still to be provided, the topics to be analyzed and the discoveries to be attained.

Life conceals the deepest knowledge about the universe. At the same time, the biggest and challenging mysteries are about life. To fill this gap, bioinformatics tries to shed light on the mechanisms that regulate life processes. Therefore, this research field becomes attractive for all the researchers yearning for this mysterious knowledge. However, one may wonder if there are some mysteries that have not to be revealed; perhaps, tasting the absolute knowledge will remain only a dream.

References

1. R. Albert. Scale-free networks in cell biology. *Journal of Cell Science*, 118(Pt 21):4947–4957, November 2005.
2. W. Ali and C. M. Deane. Functionally guided alignment of protein interaction networks for module detection. *Bioinformatics*, pages btp569+, October 2009.
3. N. Alon, R. Yuster, and U. Zwick. Color-coding. *Journal of ACM*, 42(4):844–856, 1995.
4. S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Reserch*, 25(17):33893402, 1997.
5. S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
6. F. Angiulli. Fast condensend nearest neighbor rule. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 25–32, 2005.
7. M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, and et al. Gene Ontology: Tool for the Unification of Biology. The Gene Ontology Consortium. *Nature Genetics*, 25:25–29, 2000.
8. L. Badea. Functional discrimination of gene expression patterns in terms of the gene ontology. In *Pacific Symposium on Biocomputing*, 2003.
9. J. Bader, A. Chaudhuri, J. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology*, 22:78–85, 2004.
10. A. Bairoch and R. Apweiler. The swiss-prot protein sequence data bank and its new supplement trembl. *Nucleic Acids Research*, 24(1):21–25, 1996.
11. A. Bairoch, B. Boeckmann, S. Ferro, and E. Gasteiger. Swiss-prot: Juggling between evolution and stability. *Briefings in Bioinformatics*, 5(1):39–58, 2004.
12. C. A. Ball, I. A. Awad, J. Demeter, J. Gollub, J. M. Hebert, T. Hernandez-Boussard, H. Jin, J. C. Matese, M. Nitzberg, F. Wymore, Z. K. Zachariah, P. O. Brown, and G. Sherlock. The stanford microarray database accommodates additional microarray platforms and data formats. *Nucleic Acids Research*, 33(Database issue), 2005.
13. D. Bandyopadhyay, J. Huan, J. Liu, J. Prins, J. Snoeyink, W. Wang, and A. Tropsha. Structure-based function inference using protein family-specific fingerprints. *Protein Sciences*, 15:1537–1543, 2006.
14. S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome Research*, 16(3):428–435, 2006.

15. Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20(16):2493–2503, November 2004.
16. T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, and R. Edgar. Ncbi geo: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Research*, 35(Database issue), 2007.
17. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. Eddy, S. Griffiths-Jones, K. Howe, M. Marshall, and E. Sonnhammer. The pfam protein families database. *Nucleic Acids Reserch*, 30(1):276–280, 2002.
18. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6:281297, 1999.
19. D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucleic acids research*, 36(Database issue), 2008.
20. J. Berg, M. Lässig, and A. Wagner. Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications. *BMC Evolutionary Biology*, 4(1), 2004.
21. A. T. Binkowski, L. Adamian, and J. Liang. Inferring functional relationships of proteins from local sequence and spatial surface patterns. *Journal of Molecular Biology*, 332(2):505–526, 2003.
22. K. Blekas, D. I. Fotiadis, and A. Likas. Motif-based protein sequence classification using neural networks. *Journal of Computational Biology*, 12:64–82, 2005.
23. P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, N. Huynen, and Y. Yuan. Predicting function: from genes to genomes and back. *Journal of Molecular Biology*, 283:707–725, 1998.
24. M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the USA*, 97(1):262–267, 2000.
25. S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan. Topology-free querying of protein interaction networks. In *Proceedings of the 13th Annual International Conference on Research in Computational Molecular Biology (RECOMB '09)*, Lecture Notes in Bioinformatics. Springer, 2009. To appear.
26. C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5(1), 2003.
27. K. Bryan, P. Cunningham, and N. Bolshakova. Biclustering of expression data using simulated annealing. In *CBMS '05: Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems*, pages 383–388, Washington, DC, USA, 2005. IEEE Computer Society.
28. A. J. Butte, L. Bao, B. Y. Reis, T. W. Watkins, and I. S. Kohane. Comparing the similarity of time-series gene expression using signal processing metrics. *Journal of Biomedical Informatics*, 34(6):396–405, December 2001.
29. C. Z. Cai, L. Y. Han, Z. L. Ji, X. Chen, and Y. Z. Chen. Svm-prot: Web-based support vector machine software for functional classification of a protein from its primary sequence. *Nucleic Acids Research*, 31:36923697, 2003.
30. Y. D. Cai and A. J. Doig. Prediction of saccharomyces cerevisiae protein functional class from functional domain composition. *Bioinformatics*, 20(8):1292–1300, 2004.
31. S. Carbon, A. Ireland, C. J. J. Mungall, S. Shu, B. Marshall, S. Lewis, and and. Amigo: online access to ontology and annotation data. *Bioinformatics*, 25(2):288289, 2008.

32. A. Chang, M. Scheer, A. Grote, I. Schomburg, and D. Schomburg. Brenda, amenda and frenda the enzyme information system: new content and tools in 2009. *Nucleic Acids Research*, (Database issue):D588–D592, November 2008.
33. A. Chatr-aryamontri, A. Ceol, L. Montecchi-Palazzi, G. Nardelli, M. V. Schneider, L. Castagnoli, and G. Cesareni. Mint: the molecular interaction database. *Nucleic Acids Research*, 35(Database issue):572–574, 2007.
34. J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng. Labeling network motifs in protein interactomes for protein function prediction. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 546–555, 2007.
35. B. Y. Cheng, J. G. Carbonell, and J. Klein-Seetharaman. Protein classification based on text document classification techniques. *Proteins*, 58:955970, 2005.
36. Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.
37. J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. Sgd: Saccharomyces genome database. *Nucleic Acids Research*, 26(1):73–79, January 1998.
38. K. C. Chou and Y. D. Cai. Predicting protein quaternary structure by pseudo amino acid composition. *Proteins: Structure, Function, and Genetics*, 53(2):282–289, 2003.
39. K. C. Chou and Y. D. Cai. Predicting protein structural class by functional domain composition. *Biochemical and biophysical research communications*, 321(4):1007–1009, 2004.
40. H. N. Chua, W.-K. Sung, and L. Wong. Exploiting indirect neighbours and topological weight to predict protein function from proteinprotein interactions. *Bioinformatics*, 22(13):16231630, 2006.
41. I. H. G. S. Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.
42. T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Inform. Th.*, 13(1):21–27, 1967.
43. F. Crick. On protein synthesis. *The Symposia of the Society for Experimental Biology*, 12:138–163, 1958.
44. F. Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.
45. T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends in Biochemical Sciences*, 23(9):324–328, 1998.
46. S. V. Date and E. M. Marcotte. Protein function prediction using the protein link explorer (plex). *Bioinformatics*, 21(10):2558–2559, 2005.
47. M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein protein interactions and protein function prediction. In *Pacific Symposium on Biocomputing (PSB 2003)*, 2003.
48. L. Devroye. On the inequality of cover and hart in nearest neighbor discrimination. *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 3:75–78, 1981.
49. P. D. Dobson and A. J. Doig. Predicting enzyme class from protein structure without alignments. *Journal of Molecular Biology*, 345(1):187–199, 2005.
50. T. Doerks, A. Bairoch, and P. Bork. Protein annotation: detective work for function prediction. *Trends in Genetics*, 14:248–250, 1998.
51. B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. Qnet: A tool for querying protein interaction networks. In *Proceedings of The 11th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2007)*, pages 1–15, 2007.

52. R. Dunn, F. Dudbridge, and C. M. Sanderson. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinformatics*, 6(1), 2005.
53. P. Durand, L. Labarre, A. Meil, J. Divol, Y. Vandenbrouck, A. Viari, and J. Wojcik. Genolink: a graph-based querying and browsing system for investigating the function of genes and proteins. *BMC Bioinformatics*, 21(7), 2006.
54. J. Dutkowski and J. Tiuryn. Identification of functional modules from conserved ancestral protein protein interactions. *Bioinformatics*, 23(13):i149–i158, July 2007.
55. J. A. Eisen. Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome research*, 8(3):163–167, 1998.
56. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the USA*, 95(25):14863–14868, 1998.
57. R. Eisner, B. Poulin, D. Szafron, P. Lu, and R. Greiner. Improving protein function prediction using the hierarchical structure of the gene ontology. In *Computational Intelligence in Bioinformatics and Computational Biology*, 2005.
58. F. Enault, K. Suhre, C. Abergel, O. Poirot, and J. M. Claverie. Annotation of bacterial genomes using improved phylogenomic profiles. *Bioinformatics*, 19 Suppl 1:i105i107, 2003.
59. B. E. Engelhardt, M. I. Jordan, K. E. Muratore, and S. E. Brenner. Protein molecular function prediction by bayesian phylogenomics. *PLoS computational biology*, 1(5), 2005.
60. A. J. Enright and C. A. Ouzounis. Functional associations of proteins in entire genomes by means of exhaustive detection of gene fusions. *Genome Biology*, 2, 2002.
61. J. Ernst, G. J. Nau, and Z. Bar-Joseph. Clustering short time series gene expression data. *Bioinformatics*, 21(suppl_1):i159–168, 2005.
62. J. Espadaler, R. Aragues, N. Eswar, M. Marti-Renom, E. Querol, F. Aviles, A. Sali, and B. Oliva. Detecting remotely related proteins by their interactions and sequence similarity. *Proceedings of the National Academy of Sciences of the USA*, 102(20):7151–7156, May 2005.
63. J. Espadaler, E. Querol, F. X. Aviles, and B. Oliva. Identification of function-associated loop motifs and application to protein function prediction. *Bioinformatics*, 22(18):2237–2243, 2006.
64. P. Evans, T. Sandler, and L. Ungar. Protein-protein interaction network alignment by quantitative simulation. In *Proceedings of the 2008 IEEE International Conference on Bioinformatics and Biomedicine*, pages 325–328, Washington, DC, USA, 2008. IEEE Computer Society.
65. F. Ferre, G. Ausiello, A. Zanzoni, and M. Helmer-Citterich. Surface: a database of protein surface regions for functional annotation. *Nucleic Acids Research*, 32(Database issue), 2004.
66. S. Ferre and R. D. King. Finding motifs in protein secondary structure for use in function prediction. *Journal of Computational Biology*, 13:719–731, 2006.
67. A. Ferro, R. Giugno, G. Pigola, A. Pulvirenti, D. Skripin, G. D. Bader, and D. Shasha. Netmatch: a cytoscape plugin for searching biological networks. *Bioinformatics*, 2007.
68. J. S. Fetrow, N. Siew, J. A. Di Gennaro, M. Martinez-Yamout, H. J. Dyson, and J. Skolnick. Genomic-scale comparison of sequence- and structure-based methods of function prediction: does structure provide additional insight? *Protein science*, 10(5):1005–1014, 2001.

69. J. S. Fetrow and J. Skolnick. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. *Journal of Molecular Biology*, 281(5):949–968, 1998.
70. J. Flannick, A. Novak, C. B. Do, B. S. Srinivasan, and S. Batzoglou. Automatic parameter learning for multiple network alignment. In *Research in Computational Molecular Biology (RECOMB 2008)*, pages 214–231, 2008.
71. J. Flannick, A. Novak, B. Srinivasan, H. McAdams, and S. Batzoglou. Graemlin: General and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, 2006.
72. C. Flybase. The flybase database of the drosophila genome projects and community literature. the flybase consortium. *Nucleic Acids Research*, 27(1):85–88, January 1999.
73. K. Fukunaga and L. Hostetler. *k*-nearest-neighbor bayes-risk estimation. *IEEE Transactions on Information Theory*, 21:285–293, 1975.
74. Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18(1):23–38, 1986.
75. M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. Freeman, New York, 1979.
76. R. Garian. Prediction of quaternary structure from primary structure. *Bioinformatics*, 17(6):551–556, 2001.
77. C. Gille, A. Goede, C. Schloetelburg, R. Preibner, P. Kloetze, U. Gobel, and C. Frommel. A comprehensive view on proteasomal sequences: Implications for the evolution of the proteasome. *Journal of Molecular Biology*, 326(5):1437–1448, 2003.
78. X. Guo and A. J. Hartemink. Domain-oriented edge-based alignment of protein interaction networks. *Bioinformatics*, 25(12):i240–1246, 2009.
79. M. Gustin, J. Albertyn, M. Alexander, and et al. Map kinase pathways in the yeast saccharomyces cerevisiae. *Microbiol. Mol. Biol. Rev.*, 62:1264–1300, 1998.
80. S. S. Hannenhalli and R. B. Russell. Analysis and prediction of functional sub-types from protein sequence alignments. *Journal of Molecular Biology*, 303(1):61–76, 2000.
81. N. A. Heard, C. C. Holmes, D. A. Stephens, D. J. Hand, and G. Dimopoulos. Bayesian coclustering of anopheles gene expression time series: study of immune defense response to multiple experimental challenges. *Proceedings of the National Academy of Sciences of the USA*, 102(47):16939–16944, 2005.
82. H. Hegyi and M. Gerstein. The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. *Journal of molecular biology*, 288(1):147–164, 1999.
83. S. Hennig, D. Groth, and H. Lehrac. Automated gene ontology annotation for anonymous sequence data. *Nucleic Acids Research*, 13:3712–3715, 2003.
84. H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from proteinprotein interaction data. *Yeast*, 18:523–531, 2001.
85. J. Hou, S.-R. Jun, C. Zhang, and S.-H. Kim. Global mapping of the protein structure space and application in structure-based inference of protein function. *Proceedings of the National Academy of Sciences of the USA*, 102(10):3651–3656, 2005.
86. H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl_1):i213–221, 2005.
87. S. Hunter, R. Apweiler, T. K. K. Attwood, A. Bairoch, A. Bateman, D. Binns, P. Bork, U. Das, L. Daugherty, L. Duquenne, R. D. D. Finn, J. Gough, D. Haft, N. Hulo, D. Kahn,

E. Kelly, A. Laugraud, I. Letunic, D. Lonsdale, R. Lopez, M. Madera, J. Maslen, C. McAnulla, J. McDowall, J. Mistry, A. Mitchell, N. Mulder, D. Natale, C. Orengo, A. F. F. Quinn, J. D. D. Selengut, C. J. A. J. Sigrist, M. Thimma, P. D. D. Thomas, F. Valentin, D. Wilson, C. H. H. Wu, and C. Yeats. Interpro: the integrative protein signature database. *Nucleic acids research*, (Database issue):D211–D215, 2008.

88. T. R. Hvidsten, J. Komorowski, A. K. Sandvik, and A. Laegreid. Predicting gene function from gene expressions and ontologies. *Pacific Symposium on Biocomputing*, pages 299–310, 2001.
89. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the USA*, 98(8):4569–4574, 2001.
90. L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Staerfeldt, K. Rapacki, C. Workman, C. A. Andersen, S. Knudsen, A. Krogh, A. Valencia, and S. Brunak. Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology*, 319(5):1257–1265, 2002.
91. L. J. Jensen, R. Gupta, H. H. Staerfeldt, and S. Brunak. Prediction of human protein function according to gene ontology categories. *Bioinformatics*, 19(5):635–642, 2003.
92. M. Kalaev, V. Bafna, and R. Sharan. Fast and accurate alignment of multiple protein networks. *Journal of Computational Biology*, 16(8), 2009.
93. M. Kalaev, M. Smoot, T. Ideker, and R. Sharan. Networkblast: comparative analysis of protein networks. *Bioinformatics*, 24(4):594–596, 2008.
94. M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.
95. U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the USA*, 101(9):2888–2893, 2004.
96. P. D. Karp, C. A. Ouzounis, C. Moore-kochlacs, L. Goldovsky, P. Kaipa, D. Ahrn, S. Tsoka, N. Darzentas, and V. Kunin. Expansion of the biocyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Research*, 33:6083–6089, 2005.
97. 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 of the USA*, 100(20):11394–11399, 2003.
98. R. Kelley and T. Ideker. Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology*, 23(5):561–566, May 2005.
99. I. M. Keseler, C. Bonavides-Martinez, J. Collado-Vides, S. Gama-Castro, R. P. Gunsalus, D. A. Johnson, M. Krummenacker, L. M. Nolan, S. Paley, I. T. Paulsen, M. Peralta-Gil, A. Santos-Zavaleta, A. G. Shearer, and P. D. Karp. Ecocyc: A comprehensive view of escherichia coli biology. *Nucleic Acids Research*, 37(Database issue):D464–D470, 2009.
100. A. Kidera, Y. Konishi, T. Ooi, and H. A. Scheraga. Relation between sequence similarity and structural similarity in proteins. role of important properties of amino acids. *Journal of Protein Chemistry*, 4:265–297, 1985.
101. W. K. Kim, J. Park, and J. K. Suh. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (pid) pair. *Genome informatics. International Conference on Genome Informatics*, 13:42–50, 2002.
102. T. Kin, T. Kato, K. Tsuda, B. Schoelkopf, K. Tsuda, and J. Vert. *Protein Classification via Kernel Matrix Completion*. MIT Press, 2004.

103. R. D. King, A. Karwath, A. Clare, and L. Dehaspe. Accurate prediction of protein functional class from sequence in the m. tuberculosis and e. coli genomes using data mining. *Yeast*, 17:283–293, 2000.
104. R. D. King, A. Karwath, A. Clare, and L. Dehaspe. The utility of different representations of protein sequence for predicting functional class. *Bioinformatics*, 17:445–454, 2001.
105. K. Kinoshita, J. Furui, and H. Nakamura. Identification of protein functions from a molecular surface database and ef-site. *Journal of Structural and Functional Genomics*, 2:9–22, 2002.
106. M. Kirac, G. Ozsoyoglu, and J. Yang. Annotating proteins by mining protein interaction networks. *Bioinformatics*, 22(14), 2006.
107. G. J. Kleywegt. Recognition of spatial motifs in protein structures. *Journal of Molecular Biology*, 285(4):1887–1897, 1999.
108. I. M. Klotz, N. R. Langerman, and D. W. Darnall. Quaternary structure of proteins. *Annual review of biochemistry*, 39:25–62, 1970.
109. G. Kolesov, H.-W. Mewes, and D. Frishman. Snapper: gene order predicts gene function. *Bioinformatics*, 18:1017–1019, 2002.
110. R. Kolodny and N. Linial. Approximate protein structural alignment in polynomial time. *Proceedings of the National Academy of Sciences of the USA*, 101(33):12201–12206, 2004.
111. J. O. Korbel, L. J. Jensen, C. von Mering, and P. Bork. Analysis of genomic context: prediction of functional associations from conserved bidirectionally transcribed gene pairs. *Nature Biotechnology*, 22(7):911–917, 2004.
112. M. Koyuturk, 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, 2006.
113. E. Krissinel and K. Henrick. Secondary-structure matching (ssm), a new tool for fast protein structure alignment in three dimensions. *Acta Crystallographica Section D: Biological Crystallography*, 60:2256–2268, 2004.
114. N. Krogan, G. Cagney, and *et al*. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.
115. M. Kuramochi and G. Karypis. Gene classification using expression profiles: A feasibility study. In *2nd IEEE International Symposium on Bioinformatics and Bioengineering (BIBE'01)*, 2001.
116. A. Lagreid, T. R. Hvidsten, H. Midelfart, J. Komorowski, and A. K. Sandvik. Predicting gene ontology biological process from temporal gene expression patterns. *Genome Research*, 13(5):965–979, 2003.
117. R. A. Laskowski, J. D. Watson, and J. M. Thornton. ProFunc: a server for predicting protein function from 3D structure. *Nucleic Acids Research*, 33(Web Server issue), 2005.
118. R. A. Laskowski, J. D. Watson, and J. M. Thornton. Protein function prediction using local 3D templates. *Journal of Molecular Biology*, 351:614–626, 2005.
119. H. Lee, Z. Tu, M. Deng, F. Sun, and T. Chen. Diffusion kernel-based logistic regression models for protein function prediction. *OMICS: Integrative Biology*, 10(1):40–55, 2006.
120. A. Lehninger, D. Nelson, and M. Cox. *Principles of Biochemistry*. W. H. Freeman Company, 2004.
121. S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19 Suppl 1, 2003.
122. E. D. Levy, C. A. Ouzounis, W. R. Gilks, and B. Audit. Probabilistic annotation of protein sequences based on functional classifications. *BMC Bioinformatics*, 6, 2005.

123. J. Li, S. K. Halgamuge, C. I. Kells, and S. L. Tang. Gene function prediction based on genomic context clustering and discriminative learning: an application to bacteriophages. *BMC Bioinformatics*, 8, 2007.

124. C.-S. Liao and et al. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25:i253–i258, 2009.

125. D. Liberles, A. Thoren, G. von Heijne, and A. Elofsson. The use of phylogenetic profiles for gene predictions. *Current Genomics*, 3:131137, 2002.

126. C. Lin, D. Jiang, and A. Zhang. Prediction of protein function using common-neighbors in protein-protein interaction networks. In *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering*, pages 251–260, Washington, DC, USA, 2006. IEEE Computer Society.

127. A. H. Liu and A. Califano. Functional classification of proteins by pattern discovery and top-down clustering of primary sequences. *IBM System Journal*, 40(2):379–393, 2001.

128. J. Liu, W. Wang, and J. Yang. Gene ontology friendly biclustering of expression profiles. In *CSB '04: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, pages 436–447, Washington, DC, USA, 2004. IEEE Computer Society.

129. N. M. Luscombe, S. E. Austin, H. M. Berman, and J. M. Thornton. An overview of the structures of protein-dna complexes. *Genome Biology*, 1(1), 2000.

130. C. J. V. Marcotte and E. M. Marcotte. Predicting functional linkages from gene fusions with confidence. *Applied Bioinformatics*, 1:93–100, 2002.

131. E. Marcotte. Computational genetics: finding protein function by nonhomology methods. *Current Opinion in Structural Biology*, 10(3):359–365, 2000.

132. E. M. Marcotte, M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.

133. A. C. Martin, C. A. Orengo, E. G. Hutchinson, S. Jones, M. Karmirantzou, R. A. Laskowski, J. B. Mitchell, C. Taroni, and J. M. Thornton. Protein folds and functions. *Structure*, 6(7):875–884, 1998.

134. D. M. Martin, M. Berriman, and G. J. Barton. Gotcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, 5, 2004.

135. A. Mateos, J. Dopazo, R. Jansen, Y. Tu, M. Gerstein, and G. Stolovitzky. Systematic learning of gene functional classes from dna array expression data by using multilayer perceptrons. *Genome Research*, 12(11):1703–1715, 2002.

136. L. Matthews, G. Gopinath, M. Gillespie, M. Caudy, D. Croft, B. de Bono, P. Garapati, J. Hemish, H. Hermjakob, B. Jassal, A. Kanapin, S. Lewis, S. Mahajan, B. May, E. Schmidt, I. Vastrik, G. Wu, E. Birney, L. Stein, and P. D'Eustachio. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Research*, (Database issue):D619–D622, November 2008.

137. J. McDermott, R. Bumgarner, and R. Samudrala. Functional annotation from predicted protein interaction networks. *Bioinformatics*, 21(15):3217–3226, 2005.

138. J. Meiler and D. Baker. Coupled prediction of protein secondary and tertiary structure. *Proceedings of the National Academy of Sciences of the USA*, 100(21):12105–12110, 2003.

139. H. Midelfart, A. Laegreid, and H. J. Komorowski. Classification of gene expression data in an ontology. In *ISMDA '01: Proceedings of the Second International Symposium on Medical Data Analysis*, pages 186–194, London, UK, 2001. Springer-Verlag.

140. J. P. Miller, R. S. Lo, A. Ben-Hur, C. Desmarais, I. Stagljar, W. Stafford Noble, and S. Fields. Large-scale identification of yeast integral membrane protein interactions.

Proceedings of the National Academy of Sciences of the USA, 102(34):12123–12128, 2005.

141. J. Moult and E. Melamud. From fold to function. *Current Opinion in Structural Biology*, 10(3):384–389, 2000.
142. 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, 2005.
143. R. J. Najmanovich, J. W. Torrance, and J. M. Thornton. Prediction of protein function from structure: insights from methods for the detection of local structural similarities. *Biotechniques*, 38:847, 849, 851, 2005.
144. M. Narayanan and R. Karp. Comparing protein interaction networks via a graph match-and-split algorithm. *Journal of Computational Biology*, 14(7):892–907, 2007.
145. K. Narra and L. Liao. Use of extended phylogenetic profiles with e-values and support vector machines for protein family classification. *International Journal of Computer and Information Sciences*, 6, 2005.
146. S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.
147. S. K. Ng, S. H. Tan, and V. S. Sundararajan. On combining multiple microarray studies for improved functional classification by whole-dataset feature selection. *Genome Informatics*, 14:44–53, 2003.
148. S.-K. Ng, Z. Zhu, and Y.-S. Ong. Whole-genome functional classification of genes by latent semantic analysis on microarray data. In *APBC '04: Proceedings of the second conference on Asia-Pacific bioinformatics*, pages 123–129, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
149. K. P. O'Brien, M. Remm, and E. L. L. Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acid Research*, 33(Database Issue):D476–D480, 2005.
150. C. A. Orengo, A. E. Todd, and J. M. Thornton. From protein structure to function. *Current Opinion in Structural Biology*, 9:374–382, 1999.
151. N. Orlev, R. Shamir, and Y. Shiloh. Pivot: Protein interaction visualization tool. *Bioinformatics*, 20:424–425, 2004.
152. R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. Use of contiguity on the chromosome to predict functional coupling. *In silico biology*, 2:93108, 1999.
153. R. Overbeek, M. Fonstein, M. D'Souza, G. D. Pusch, and N. Maltsev. The use of gene clusters to infer functional coupling. *Proceedings of the National Academy of Sciences of the USA*, 96(6):2896–2901, 1999.
154. L. Palopoli, D. Sacca, G. Terracina, and D. Ursino. Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294, 2003.
155. W. Pan. Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7):795–801, 2006.
156. H. Parkinson, U. Sarkans, M. Shojatalab, N. Abeygunawardena, S. Contrino, R. Coulson, A. Farne, G. G. Lara, E. Holloway, M. Kapushesky, P. Lilja, G. Mukherjee, A. Oezcimen, T. Rayner, P. Rocca-Serra, A. Sharma, S. Sansone, and A. Brazma. Arrayexpress–a public repository for microarray gene expression data at the ebi. *Nucleic Acids Research*, 33(Database issue), 2005.
157. C. Pasquier, V. J. Promponas, and S. J. Hamodrakas. Pred-class: cascading neural networks for generalized protein classification and genome-wide applications. *Proteins*, 44:361–369, 2001.

158. F. Pazos and M. J. E. Sternberg. Automated prediction of protein function and detection of functional sites from structure. *Proceedings of the National Academy of Sciences of the USA*, 101:14754–14759, 2004.
159. F. Pazos and A. Valencia. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Engineering*, 14:609–614, 2001.
160. W. R. Pearson. Rapid and sensitive sequence comparison with fastp and fasta. *Methods in enzymology*, 183:63–98, 1990.
161. W. R. Pearson and D. J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences of the USA*, 85(8):2444–2448, 1988.
162. M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles.
163. A. J. Perez, A. Rodriguez, O. Trelles, and G. Thode. A computational strategy for protein function assignment which addresses the multidomain problem. *Comparative and Functional Genomics*, 3:423–440, 2002.
164. PFAM. http://www.sanger.ac.uk/software/pfam/.
165. R. Y. Pinter, O. Rokhlenko, E. Y. Lotem, and M. Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005.
166. G. Pirró. A semantic similarity metric combining features and intrinsic information content. *Data & Knowledge Engineering*, In press, 2009.
167. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47:228–235, 2002.
168. K. T. S. Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, L. Balakrishnan, A. Marimuthu, S. Banerjee, D. S. Somanathan, A. Sebastian, S. Rani, S. Ray, H. C. J. Kishore, S. Kanth, M. Ahmed, M. K. Kashyap, R. Mohmood, Y. L. Ramachandra, V. Krishna, A. B. Rahiman, S. Mohan, P. Ranganathan, S. Ramabadran, R. Chaerkady, and A. Pandey. Human protein reference database–2009 update. *Nucleic Acids Research*, (Database issue):D767–D772, 2008.
169. B. Qian and R. A. Goldstein. Detecting distant homologs using phylogenetic tree-based hmms. *Proteins: Structure, Function, and Genetics*, 52(3):446–453, 2003.
170. 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.
171. M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314:10411052, 2001.
172. A. W. Rives and T. Galitski. Modular organization of cellular networks. *Proceedings of the National Academy of Sciences of the USA*, 100:1128–1133, 2003.
173. B. Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12:85–94, 1999.
174. A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. N. Doudieu, V. Stümpflen, and H. W. Mewes. Corum: the comprehensive resource of mammalian protein complexes. *Nucleic Acids Research*, 36(Database issue):D646–D650, January 2008.
175. L. Salwinski, C. Miller, A. Smith, F. Pettit, J. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Reserch*, 32(Database issue):D449–D451, 2004.

176. M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences of the USA*, 100(22):12579–12583, 2003.
177. J. Schug, S. Diskin, J. Mazzarelli, B. P. Brunk, and C. J. Stoeckert. Predicting gene ontology functions from prodom and cdd protein domains. *Genome Research*, 12(4):648–655, 2002.
178. B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18(12):1257–1261, 2000.
179. N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, pages 1089–1090, 2004.
180. B. E. Shakhnovich. Improving the precision of the structure-function relationship by considering phylogenetic context. *PLoS Computational Biology*, 1:e9, 2005.
181. R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, 2006.
182. 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 USA*, 102(6):1974–1979, 2005.
183. 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.
184. R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Annual International Conference on Research in Computational Molecular Biology (RECOMB 2007)*, 2007.
185. R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks. In *Pacific Symposium on Biocomputing (PSB 2008)*, 2008.
186. R. Singh, J. Xu, and B. Berger. Isorank: Global alignment of multiple protein interaction networks with applications to functional orthology detection. *Proceedings of the National Academy of Sciences of the USA*, 105(35):12763–12768, 2008.
187. K. Sjölander. Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics*, 20(2):170–179, 2004.
188. J. Skolnick and J. S. Fetrow. From genes to protein structure and function: novel applications of computational approaches in the genomic era. *Trends in Biotechnology*, 18(1):34–39, 2000.
189. T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
190. B. Snel, P. Bork, and M. A. Huynen. The identification of functional modules from the genomic association of genes. *Proceedings of the National Academy of Sciences of the USA*, 99(9):5890–5895, April 2002.
191. J. Song and H. Tang. Accurate classification of homodimeric vs other homooligomeric proteinsusing a new measure of information discrepancy. *Journal of chemical information and computer sciences*, 44(4):1324–1327, 2004.
192. V. Spirin and L. Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the National Academy of Sciences of the USA*, 100:12123–12128, 2003.
193. E. Sprinzak, S. Sattath, and H. Margalit. How reliable are experimental protein-protein interaction data? *Journal of Molecular Biology*, 327(5):919–923, 2003.
194. T. Stangler, T. Tran, S. Hoffmann, H. Schmidt, E. Jonas, and D. Willbold. Competitive displacement of full-length hiv-1 nef from the hck sh3 domain by a high-affinity artificial peptide. *Journal of Biological Chemistry*, 338(6):611–615, 2007.

195. C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(Database issue):D535–D539, 2006.
196. C. Stone. Consistent nonparametric regression. *Annals of Statistics*, 8:1348–1360, 1977.
197. S. Sun, Y. Zhao, Y. Jiao, Y. Yin, L. Cai, Y. Zhang, H. Lu, R. Chen, and D. Bu. Faster and more accurate global protein function assignment from protein interaction networks using the mfgo algorithm. *FEBS Letters*, 580:1891–1896, 2006.
198. H. Sund and K. Weber. The quaternary structure of proteins. *Angewandte Chemie (International ed. in English)*, 5(2):231–245, 1966.
199. S. Swift, A. Tucker, V. Vinciotti, N. Martin, C. Orengo, X. Liu, and P. Kellam. Consensus clustering and functional interpretation of gene-expression data. *Genome Biology*, 5(11), 2004.
200. R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A genomic perspective on protein families. *Science*, 278(5338):631–637, 1997.
201. R. L. Tatusov, D. A. Natale, I. V. Garkavtsev, T. A. Tatusova, U. T. Shankavaram, B. S. Rao, B. Kiryutin, M. Y. Galperin, N. D. Fedorova, and E. V. Koonin. The cog database: new developments in phylogenetic classification of proteins from complete genomes. *Nucleic Acids Research*, 29(1):22–28, 2001.
202. T. Tatusova and T. L. Madden. Blast 2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS Microbiology Letters*, 174(2):247–250, 1999.
203. J. M. Thornton, C. A. Orengo, A. E. Todd, and F. M. Pearl. Protein folds, functions and evolution. *Journal of Molecular Biology*, 293(2):333–342, 1999.
204. W. Tian and N. F. Samatova. Pairwise alignment of interaction networks by fast identification of maximal conserved patterns. In *In proceeding of Pacific Symposyum on Biocomputing*, 2009.
205. Y. Tian, R. C. Mceachin, C. Santos, D. J. States, and J. M. Patel. Saga: a subgraph matching tool for biological graphs. *Bioinformatics*, 23(2):232–239, 2007.
206. Y. Tohsato, H. Matsuda, and A. Hashimoto. A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, pages 376–383, 2000.
207. A. Tramontano. *Introduction to Bioinformatics*. Chapman & Hall/CRC, 2007.
208. V. Vannoort, B. Snel, and M. Huynen. Predicting gene function by conserved co-expression. *Trends in Genetics*, 19(5):238–242, 2003.
209. A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21(6):697700, 2003.
210. J. P. Vert. A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, 18 Suppl 1, 2002.
211. D. von Mering, C. Krause, and *et al*. Comparative assessment of a large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.
212. M. G. Walker, W. Volkmuth, E. Sprinzak, D. Hodgson, and T. Klingler. Prediction of gene function by genome-scale expression analysis: prostate cancer-associated genes. *Genome research*, 9(12):1198–1203, 1999.
213. A. C. Wallace, N. Borkakoti, and J. M. Thornton. Tess: a geometric hashing algorithm for deriving 3d coordinate templates for searching structural databases. application to enzyme active sites. *Protein Science*, 6(11):2308–2323, 1997.
214. C. Wang and S. D. Scott. New kernels for protein structural motif discovery and function classification. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 940–947. ACM, 2005.

215. J. T. L. Wang, Q. Ma, D. Shasha, and C. H. Wu. New techniques for extracting features from protein sequences. *IBM Systems Journal*, 40(2):426–441, 2001.
216. J. Z. Wang, Z. Du, R. Payattakool, P. S. Yu, and C.-F. Chen. A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):12741281, 2007.
217. K. Wang and R. Samudrala. Fssa: a novel method for identifying functional signatures from structural alignments. *Bioinformatics*, 21:2969–2977, 2005.
218. X. Wang, D. Schroeder, D. Dobbs, and V. G. Honavar. Automated data-driven discovery of motif-based protein function classifiers. *Information Sciences*, 155(1-2):1–18, 2003.
219. S. Wernicke and F. Rasche. Simple and fast alignment of metabolic pathways by exploiting local diversity. *Bioinformatics*, 23(15):1978–1985, 2007.
220. D. L. Wild and M. A. S. Saqi. Structural proteomics: Inferring function from protein structure. *Current Proteomics*, 1:59–65, 2004.
221. D. Wilson and T. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.
222. J. Wojcik and V. Schachter. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, 17(1):296–305, 2001.
223. C. Wu, M. Berry, S. Shivakumar, and J. McLarty. Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition. *Machine Learning*, 21(1-2):177–193, 1995.
224. C. Wu, G. Whitson, J. McLarty, A. Ermongkonchai, and T. C. Chang. Protein classification artificial neural system. *Protein Sciences*, 1:667–677, 1992.
225. J. Wu, S. Kasif, and C. DeLisi. Identification of functional links between genes using phylogenetic profiles. *Bioinformatics*, 19(12):1524–1530, 2003.
226. L. F. Wu, T. R. Hughes, A. P. Davierwala, M. D. Robinson, R. Stoughton, and S. J. Altschuler. Large-scale prediction of saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nature Genetics*, 31:255–265, 2002.
227. H. Xiong, X. He, C. H. Q. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In R. B. Altman, T. A. Jung, T. E. Klein, A. K. Dunker, and L. Hunter, editors, *Pacific Symposium on Biocomputing*. World Scientific, 2005.
228. T. Xu, L. Du, and Y. Zhou. Evaluation of go-based functional similarity measures using s. cerevisiae protein interaction and expression profile data. *BMC Bioinformatics*, 9(472), 2008.
229. I. Yanai, A. Derti, and C. DeLisi. Genes linked by fusion events are generally of the same functional category: a systematic analysis of 30 microbial genomes. *Proceedings of the National Academy of Sciences of the USA*, 98:7940–7945, 2001.
230. J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced biclustering on expression data. In *BIBE '03: Proceedings of the 3rd IEEE Symposium on BioInformatics and BioEngineering*, pages 321–327, Washington, DC, USA, 2003. IEEE Computer Society.
231. Q. Yang and S.-H. Sze. Path matching and graph matching in biological networks. *Journal of Computational Biology*, 14(1):56–67, 2007.
232. X. Yu, J. Lin, T. Shi, and Y. Li. A novel domain-based method for predicting the functional classes of proteins. *Chinese Science Bullettin - English Edition-*, 49(22):2379–2384, 2004.
233. X. Yu, C. Wang, and Y. Li. Classification of protein quaternary structure by functional domain composition. *BMC Bioinformatics*, 7(187), 2006.
234. M. Zaslavskiy, F. Bach, and J.-P. Vert. Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–i1267, June 2009.
235. G. Zehetner. Ontoblast function: From sequence similarities directly to potential functional annotations by ontology terms. *Nucleic Acids Res*, 31(13):3799–3803, 2003.

236. L. V. Zhang, O. D. King, S. L. Wong, D. S. Goldberg, A. H. Y. Tong, G. Lesage, B. Andrews, H. Bussey, C. Boone, and F. P. Roth. Motifs, themes and thematic maps of an integrated saccharomyces cerevisiae interaction network. *Journal of Biology*, 4(2), 2005.
237. S. Zhang, X.-S. Zhang, and L. Chen. Biomolecular network querying: a promising approach in system biology. *BMC System Biology*, 5(2), 2008.
238. S. W. Zhang, Q. Pan, H. C. Zhang, Y. L. Zhang, and H. Y. Wang. Classification of protein quaternary structure with support vector machine. *Bioinformatics*, 19(18):2390–2396, 2003.
239. W. Zhang, Q. D. Morris, R. Chang, O. Shai, M. A. Bakowski, N. Mitsakakis, N. Mohammad, M. D. Robinson, R. Zirngibl, E. Somogyi, N. Laurin, E. Eftekharpour, E. Sat, J. Grigull, Q. Pan, W. T. Peng, N. Krogan, J. Greenblatt, M. Fehlings, D. van der Kooy, J. Aubin, B. G. Bruneau, J. Rossant, B. J. Blencowe, B. J. Frey, and T. R. Hughes. The functional landscape of mouse gene expression. *Journal of Biology*, 3(5), 2004.
240. Y. Zheng, R. J. Roberts, and S. Kasif. Genomic functional annotation using co-evolution profiles of gene clusters. *Genome Biology*, 3:research0060.10060.9, 2002.
241. X. Zhou, M.-C. J. Kao, and W. H. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proceedings of the National Academy of Sciences of the USA*, 99(20):12783–12788, October 2002.
242. Y. Zhou, J. A. Young, A. Santrosyan, K. Chen, S. F. Yan, and E. A. Winzeler. In silico gene function prediction using ontology-based pattern identification. *Bioinformatics*, 21(7):1237–1245, April 2005.