Università degli Studi di Pisa

Dipartimento di Informatica
Dottorato di Ricerca in Informatica

Settore Scientifico Disciplinare: INF/01

Ph.D. Thesis

A Gibbs sampling strategy for mining of protein-protein interaction networks and protein structures

Giovanni Micale

Supervisor
Paolo Ferragina
University of Pisa

Supervisor
Alfredo Ferro
University of Catania

Abstract

Complex networks are general and can be used to model phenomena that belongs to different fields of research, from biochemical applications to social networks. However, due to the intrinsic complexity of real networks, their analysis can be computationally demanding. Recently, several statistic and probabilistic analysis approaches have been designed, resulting to be much faster, flexible and effective than deterministic algorithms. Among statistical methods, Gibbs sampling is one of the simplest and most powerful algorithms for solving complex optimization problems and it has been applied in different contexts. It has shown its effectiveness in computational biology but in sequence analysis rather than in network analysis. One approach to analyze complex networks is to compare them, in order to identify similar patterns of interconnections and predict the function or the role of some unknown nodes. Thus, this motivated the main goal of the thesis: designing and implementing novel graph mining techniques based on Gibbs sampling to compare two or more complex networks. The methodology is domain-independent and can work on any complex system of interacting entities with associated attributes. However, in this thesis we focus our attention on protein analysis overcoming the strong current limitations in this area. Proteins can be analyzed from two different points of view: (i) an internal perspective, i.e. the 3D structure of the protein, (ii) an external perspective, i.e. the interactions with other macromolecules. In both cases, a comparative analysis with other proteins of the same or distinct species can reveal important clues for the function of the protein and evolutionary convergences or divergences between different organisms in the way a specific function or process is carried out. First, we present two methods based on Gibbs sampling for the comparative analysis of protein-protein interaction networks: GASOLINE and SPECTRA. GASOLINE is a stochastic and greedy algorithm to find similar groups of interacting proteins in two or more networks. It can align many networks and more quickly than the state-of-the-art methods. SPECTRA is a framework to retrieve and compare networks of proteins that interact with one another in specific healthy or tumor tissues. The aim in this case is to identify changes in protein concentration or protein "behaviour" across different tissues. SPECTRA is an adaptation of GASOLINE for weighted protein-protein interaction networks with gene expressions as node weights. It is the first algorithm proposed for multiple comparison of tissue-specific interaction networks. We also describe a Gibbs sampling based algorithm for 3D protein structure comparison, called PROPOSAL, which finds local structural similarities across two or more protein structures. Experimental results confirm our computational predictions and show that the proposed algorithms are much faster and in most cases more accurate than existing methods.

Acknowledgements

First of all, I would like to thank my supervisor, Paolo Ferragina, who followed me in my thesis and provided me with important suggestions on how to carry on my work as well as many interesting ideas to develop my methodology.

The work of this thesis starts from University of Catania, when I was a master student in Computer Science and I was going to graduate. So, I also need to thank the research group of Catania (Alfredo Ferro, Alfredo Pulvirenti and Rosalba Giugno) for their continuous support. They always gave me the right advice, whenever I was stuck or discouraged with my work.

A special thank goes to Pierpaolo Degano, the Director of the PhD program in Computer Science in Pisa, for his great willingness and his endless patience towards me. I have always had all I needed to work at my best.

In Pisa I met many colleagues and friends who were always willing to help me. In particular, I would like to mention the following people: Davide Basile, who helped me a lot to find an accomodation when I had just arrived in Pisa and gave me a great support in many PhD activities, my 'next-office neighbour' Simone Zenzaro, Leonardo Bartoloni, Matteo Sammartino and Letterio (Lillo) Galletta.

I would also like to thank my parents for their endless moral and economic support. They were always ready to help me in every situation.

Finally, I need to thank Francesca, the love of my life. Her presence is fundamental in all my activities, including my studies.

Contents

Chapter 1

Introduction

A complex network is a model representing entities with one or more associated attributes which interact one another (physically or virtually). It is called 'complex' because it exhibits non-trivial topological features that are not observed at random, such as the tendency of entities to form communities or link to other important entities. Recently, an increasing attention has been devoted to complex networks for many reasons. First, they offer a simple framework to analyze real word phenomena. Second, they are general and can be used to model events belonging to different fields of research, from biology to social networks.

The analysis of such networks, especially in a comparative framework, can reveal the existence of recurrent patterns of interactions, such as communities or structural motifs, and highlight the role of some entities and the functioning of certain processes.

Data mining techniques can be applied to find and extract these useful informations from a network or a collection of networks. However, due to the intrinsic complexity of real networks, the process of finding and extracting structural patterns can be time consuming. Therefore, in the last few years many approaches relying on statistics and probability have been designed, especially in classification and prediction problems. In many cases, they have been proved to be much faster, more flexible and effective than deterministic algorithms.

Among all statistical methods, Gibbs sampling [50] is one of the simplest and most powerful algorithms. Gibbs sampling belongs to the class of Markov Chain Monte Carlo (MCMC) methods and can be used to find approximate solutions to complex combinatorial problems, whose solution depends on many different parameters, such as optimization problems.

All these considerations motivate the main goal of the thesis: designing and implementing novel graph mining techniques based on Gibbs sampling to compare two or more complex networks. Such techniques have been developed for the mining of general networks, but they have been exstensively applied to the bioinformatics context, where the network comparison problem is widely studied [141], for the functional and structural comparative analysis of proteins.

Proteins are biological macromolecules made out of small compounds, called amino acids. They perform many essential operations to ensure the survival of living organisms, including catalyzing metabolic reactions, replicating DNA, responding to external stimuli, and transporting molecules inside the cell.

Proteins tend to fold in space due to the interactions between amino acids, giving rise to different levels of structures (primary, secondary, tertiary and sometimes quaternary). The tertiary or 3D structure of a protein mainly determines its function. However, proteins do not act in isolation, instead they tend to cooperate with other macromolecules (DNA, RNA or other proteins) to achieve a specific goal. In this work we will focus on protein-protein interactions (PPIs) within a cell, which can be modeled by a network.

Consequently, proteins can be analyzed from two different points of view: (i) an internal perspective, i.e. the 3D structure of the protein, (ii) an external perspective, i.e. the interactions with other macromolecules. In both cases, a comparative analysis with other proteins of the same or distinct species can reveal important clues for the function of the protein. It can also shed light

on possible evolutionary convergences or divergences between different organisms, which allows to understand how functions or processes evolve through time.

In computational biology, Gibbs sampling has been successfully applied by Lawrence et al. [87] for comparative analysis of protein sequences. However, it has never been used for network comparison. Thus, we first investigate the most natural extension of this algorithm to the comparison of PPI networks of different species. The first objective of the thesis is the identification of similarities in protein interaction patterns across different organisms. This is done through a stochastic and greedy algorithm, called GASOLINE. We show that GASOLINE outperforms the state-of-the-art in both speed and accuracy, especially in the simultaneous comparison of many real PPI networks.

Motivated by the good performance of GASOLINE, we next present a framework to graphically visualize and analyze the results of such comparative network analysis. This provides a new useful tool for the graphical representation of PPI networks comparison results. The software is implemented as an app for Cytoscape, which is the most popular platform for analyzing biological networks in the Bioinformatics community.

A PPI network represents a static snapshot of the global set of physical associations between proteins in a cell of a specific tissue and in certain conditions (health, temperature, etc.). So, it ignores the role of proteins in specific human tissues. Usually, proteins are predominantly expressed in one or few tissues. Morover, they tend to interact with different proteins and carry out different functions, depending on the tissue where they are present. Finally, the function of a protein can change from healthy to diseased tissues.

Therefore, a second result presented through the thesis is the development of a framework, named SPECTRA, to build and compare tissue and tumor specific PPI networks. An adapted version of GASOLINE is used for comparative analysis of these networks, in order to identify changes in proteins or protein interactions across different tissues, or between the normal and pathological states in the same tissue. To our knowledge, this is the first tool for querying and comparing PPI networks of normal and tumor tissues.

We conclude the thesis with a third objective, the design and implementation of an algorithm for 3D protein structure comparison, called PROPOSAL. Like GASOLINE, it is based on Gibbs sampling, but it is developed for comparing 3D models, instead of networks. The aim is to find similar sub-regions in multiple protein structures of the same species or different species. PROPOSAL is the first algorithm to compare multiple protein structures for finding common sub-regions. We show that PROPOSAL is accurate and scalable with respect to the number of compared structures.

1.1 Outline

The rest of the thesis is structured as follows.

Chapter 2 presents some preliminary concepts on biology and statistics. Biological background includes the description of genes and proteins and their structure, the central dogma of molecular biology, that is the set of all the processes that lead to the production of a protein from the corresponding gene, and the description of biological networks. Statistical background starts with some basic concepts of probability theory, which are necessary to define Monte Carlo methods and Markov chains. Then, Monte Carlo Markov Chains (MCMC) methods will be described, with particular emphasis on Gibbs sampling and its application to local sequence alignment [87], upon which the algorithms presented in the thesis are based.

Chapter 3 illustrates the state-of-the-art of all the datasets and problems described in the thesis. We first present the main datasets of PPI and gene expression data and we review stochastic methods for biological network analysis. Then, we focus on the three topics of the thesis: network alignment, protein structure comparison and the analysis of tissue-specific PPI networks.

Chapters 4 and 5 describe the solutions proposed for mining of PPI networks (GASOLINE and SPECTRA) and 3D protein structures (PROPOSAL).

In Chapter 6 we give the conclusions and sketch the future research developments in the field of multiple comparison of interaction networks and protein structures.

Chapter 2

Background

2.1 Biology

In this section we introduce basic notions of molecular biology, concerning the structure of genes and proteins. We then illustrate the processes that lead to protein formation starting from genes. Finally, we define biological networks and describe the most important classes of biological networks.

2.1.1 DNA

Cells are the basic structural and functional units of living organisms. They are formed by a protoplasm, containing many biomolecules, including proteins and nucleic acids, enclosed by a membrane. There are two types of cells: eukaryotic, which contain a central unit called *nucleus*, and prokaryotic, which do not have a nucleus. Eukaryotic cells are formed by subunits or components, such as the ribosomes, the vacuoles and the mitochondria. Each cellular component executes a specific task to ensure the survival of the cell.

Similar cells from the same origin that together carry out a specific function form a *tissue* (e.g. nerve tissue, bone tissue, adipose tissue). In turn, functional groupings of multiple tissues form *organs* (e.g. heart, liver, lung).

All the genetic instructions required to perform cell functions are encoded in a single molecule, called *DNA* (Fig. 2.1). DNA is a nucleic acid composed by two strands coiled around to form a double helix. DNA strands are chains of nucleotides, which are macromolecule formed by a phospate group, a monosaccharide sugar called deoxyribose and a nitrogenous base (or nucleobase). Depending on the specific nucleobase, there are four different nucleotides: adenine (A), guanine (G), cytosine (C) or thymine (T).

Consecutive nucleotides are bind by covalent bonds between the sugar of one nucleotide and the phosphate group of the next (Fig. 2.1). Pairs of nucleotides of opposite strands form hydrogen bounds, according to the following base pair: adenine with thymine (A-T) and cytosine with guanine (C-G). This structure leads to a stable conformation of DNA molecule and makes the two DNA strands complementary. DNA can be simply represented as a long string of characters in the alphabet formed by A, C, G, T symbols.

DNA can be divided into physical units, called *chromosomes*, each formed by a set of functional units called *genes*.

2.1.2 Genes

A gene is a portion of DNA which contains the information for a specific function. Fig. 2.2 depicts the structure of a gene in an eukaryotic cell. A gene contains a coding part, which is formed by exons separated by introns, and a non coding region. The sequences that immediately preceed and follow the coding part are special non coding exons, called UnTranslated Regions (UTR).

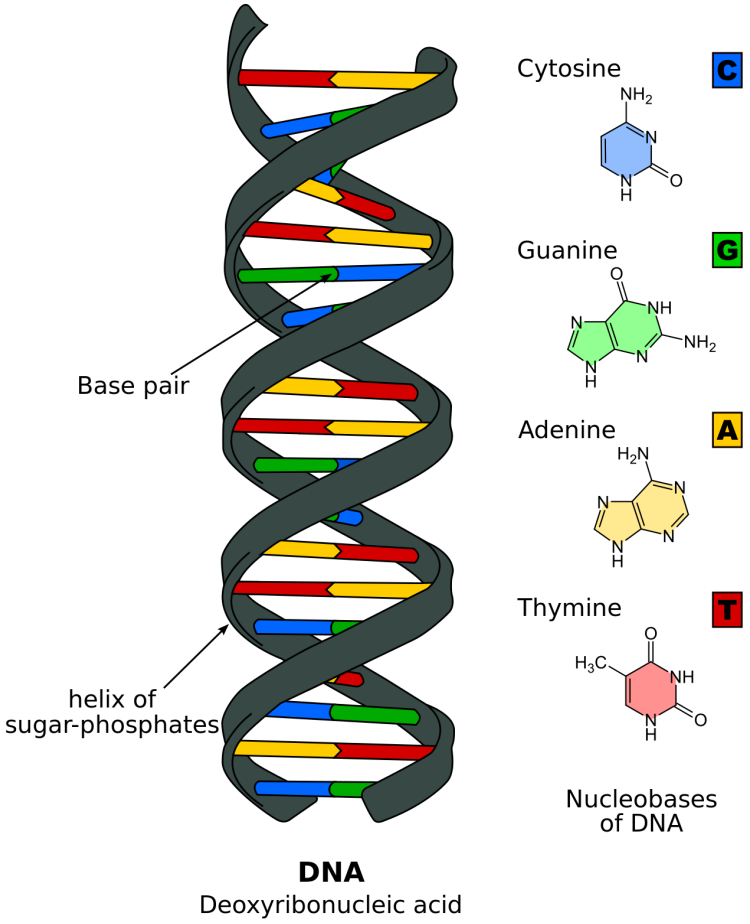


Figure 2.1: Structure of DNA molecule. Nucleotides are paired according to the base pair rule: adenine with thymine (A-T) and cytosine with guanine (C-G). [Source: http://commons.wikimedia.org/wiki/File:Difference_DNA_RNA-EN.svg].

The non coding region, which is located at the beginning of the gene, contains different regulatory sequences, such as promoters, enhancers and silencers. They perform a key role in the regulation of gene expression (see Subsection 2.1.4). Promoters, such as the TATA box, are close to the coding region, while enhancers and silencers are usually farther.

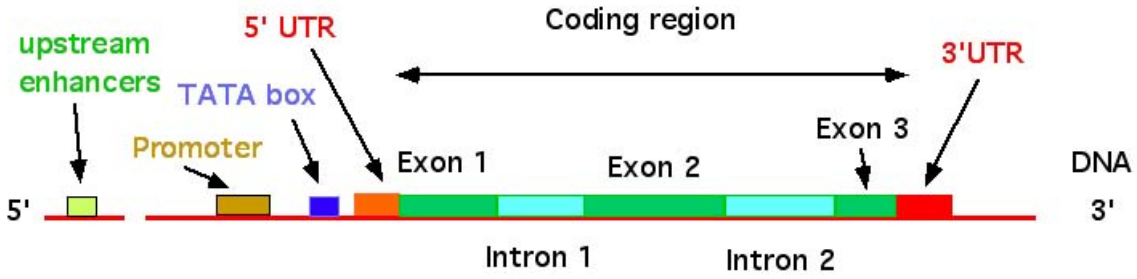


Figure 2.2: Structure of an eukaryotic gene [Source: http://nitro.biosci.arizona.edu/courses/EEB600A-2003/lectures/lecture24/figs/euk.jpg].

2.1.3 Proteins

The gene content has to be decoded before being used to perform the associated function. Usually, the result of decoding is the production of a macromolecule, called *protein*, which is the final executor of the function encoded by the corresponding gene. A protein is formed by one or more polypetidic chains of *amino acids* or *residues*. There are 20 distinct aminoacides, identified by a specific letter (e.g. L for Leucine, P for Proline)

Fig. 2.3 shows the general structure of an amino acid. Each amino acid is composed by an amine group (NH_2), a carboxylic acid group (COOH), an alpha carbon atom and a radical (R) group. The R group is specific to each amino acid. The set of all amine and carboxylic groups forms the *backbone* of the protein, while the set of all R groups forms the *side chain*.

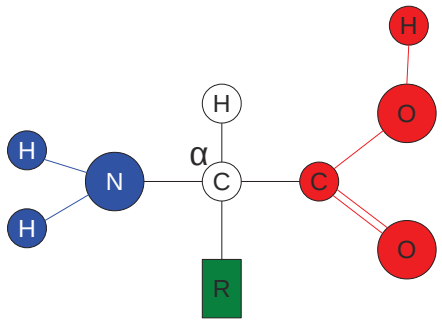


Figure 2.3: Generic structure of an amino acid. Blue atoms identify the amino group, red atoms form the carboxylic group, the green box is the radical group, specific for each amino acid.

Amino acids can be classified into different categories, according to their charge (positive, negative or neutral), their tendency to interact with water (hydrophobic or hydrophilic), their size (small or tiny) and the structure of their R groups (aromatic or aliphatic).

Adjacent amino acids form peptide bonds, caused by the reaction between the carboxyl group of one residue and the amine group of the other by the loss of a water molecule (Fig. 2.4).

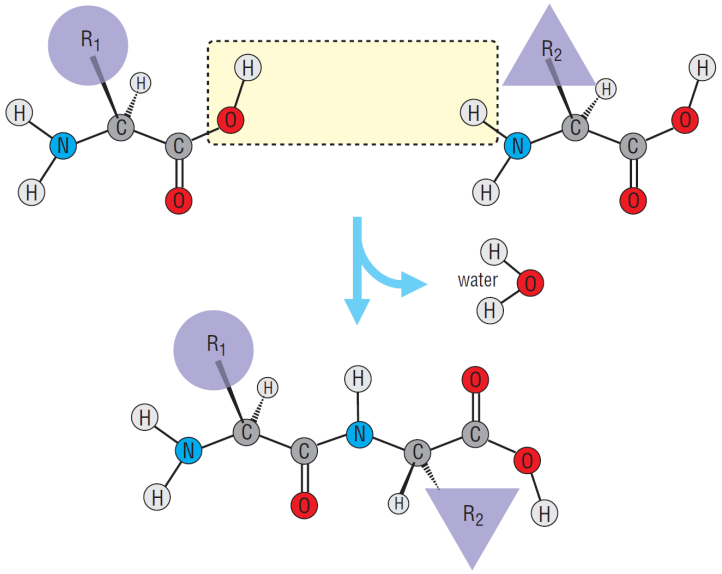


Figure 2.4: Two amino acids react to form a peptide bond. This reaction produces a molecule of water and a polypeptide/protein chain [Source: "Protein Structure and Function", G.A. Petsko and D. Ringe, 2004].

Proteins tend to fold in space, in order to achieve stable conformations. This is due to the interactions between non-adjacent aminoacids. There are four levels of protein structures (Fig. 2.5).

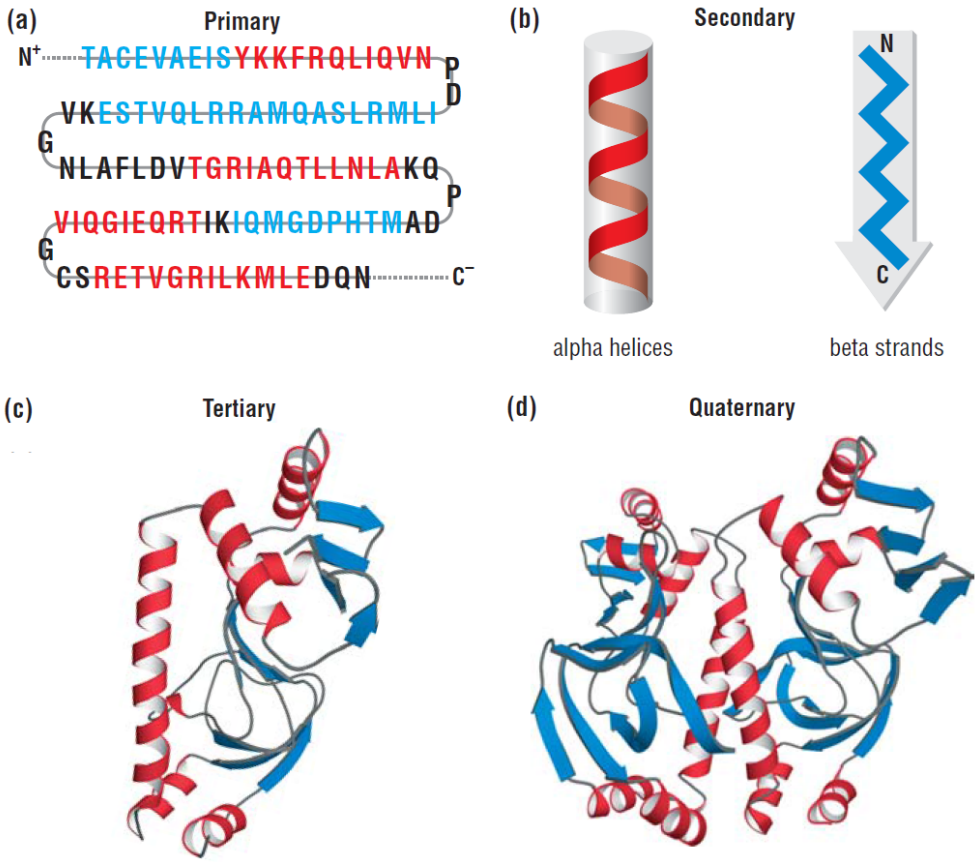


Figure 2.5: The four levels of protein structure: a) Primary structure, b) Secondary structures, c) Tertiary structure, d) Quaternary structure [Source: "Protein Structure and Function", G.A. Petsko and D. Ringe, 2004].

The sequence of the different amino acids in a protein form its *primary structure* (Fig. 2.5, a). Local foldings are called *secondary structures* and are determined by patterns of hydrogen bonds between amine groups and carboxyl groups (Fig. 2.5, b). Secondary structures include α-*helices* and β-*sheets*. α-*helices* arise when the polypetidic chain envelops to form a rigid cylinder. β-*sheets* are determined by hydrogen bonds between groups of adjacent amino acids of different polypeptidic chains. In turn, helices and sheets can combine to form more complex foldings, called *super secondary structures* (e.g. greek key, β-meander). The three-dimensional (3D) structure of a protein, as defined by the atomic coordinates of amino acids atoms form the *tertiary structure* (Fig. 2.5, c). The 3D structure of a protein is largely determined by the primary structure and determines its function, since each function requires a specific tertiary structure. If a protein is formed by two ore more polypetidic chains, it can also assume a *quaternary structure*, arising from non-covalent interactions between 3D structures of different chains (Fig. 2.5, d).

2.1.4 Central dogma of molecular biology

The flow of operations needed to produce a protein from the corresponding gene is the so-called *central dogma of molecular biology* (Fig. 2.6).

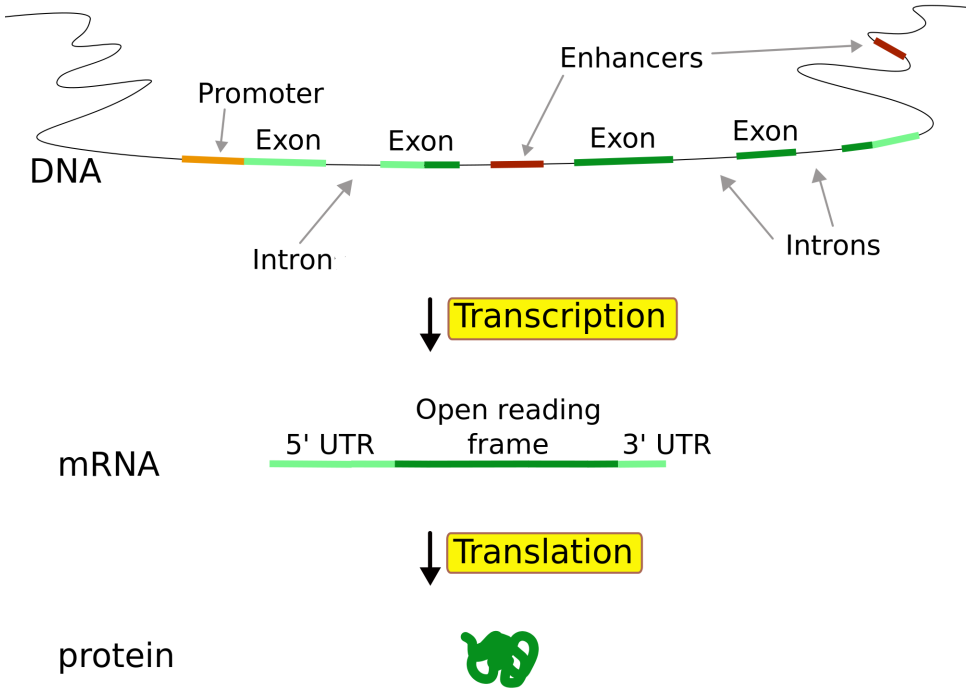


Figure 2.6: The central dogma of molecular biology [Source: http://commons.wikimedia.org/wiki/File:Gene2-plain.svg].

A protein is produced in two different steps:

1. Transcription
2. Translation

In the transcription process, one of the two replicated DNA strands is used as template to produce a new molecule, called *messenger RNA* (mRNA), which is identical to the other strand, except for thymine (T) nucleotides, which are replaced by uracile (U). The process is carried out by a specific enzyme, called RNA polymerase. The *expression value* for a gene is a measure of the amount of mRNA molecules produced from that gene in the cell.

The mRNA molecule migrates from the nucleus to the endoplasmic reticulum, which contains rybosomes, where it is translated into the corresponding protein. The mRNA sequence to translate is called Open Reading Frame (ORF) and it is formed by a series of triplets of nucleotides, called codons. The translation process reads the codons in succession and maps each codon to a specific amino acid. The mapping can be described by a table, called *genetic code*. When a codon is identified, the corresponding amino acid is transported by special RNA molecules, called transfer RNAs (tRNAs), and appended to the peptidic sequence.

In some cases the result of transcription process is not unique, meaning that different mRNAs and consequently different proteins (called *isoforms*) can be produced from the same gene. This process is called *alternative splicing* and it depends on which exons of the gene are used to synthesize the mRNA molecule.

Exceptions to the central dogma are possible. In some cases, the transcription process produces a different RNA molecule, called non-coding RNA (ncRNA). A ncRNA is not translated into a protein, so the translation process is skipped. Non-coding RNAs include microRNAs (miRNAs), short interfering RNAs (siRNAs), Piwi-interacting RNAs (piRNAs), small nucleolar RNAs (snoRNAs) and long non-coding RNAs (lncRNAs) (Fig. 2.7).

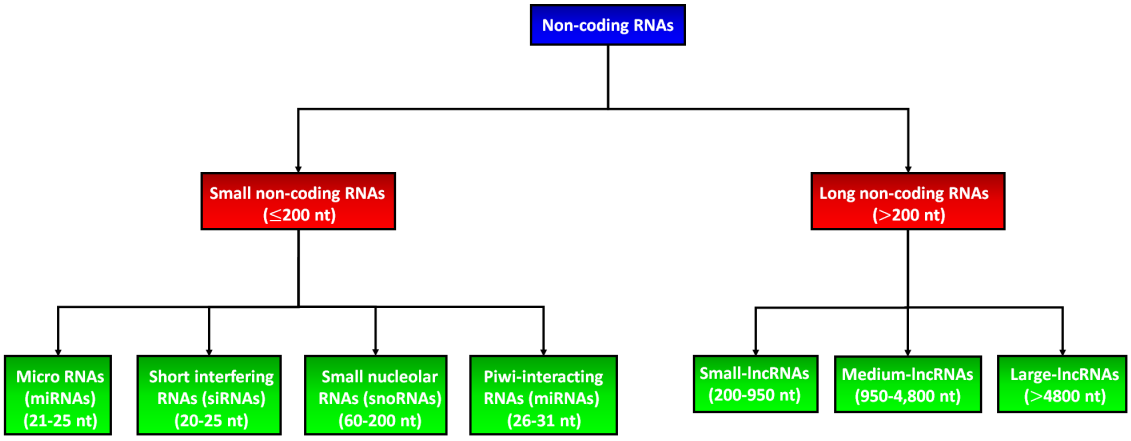


Figure 2.7: A taxonomy of the most important non-coding RNAs (ncRNAs). For each ncRNA, the average length in terms of nucleotides is reported.

MicroRNAs are small non-coding RNAs (about 22 nucleotides) which can act as post-transcriptional regulator of gene expression. miRNAs perform their function via partial or complete base-pairing with complementary sequences of the targeted mRNA molecules. siRNAs are so named because they interfere with the expression of specific genes and cause mRNA to be broken down after transcription. snoRNAs are involved in the processing and maturation of ribosomal RNAs and other types of RNAs, increasing their activity. piRNAs interact with Piwi proteins to form RNA-protein complexes for the post-transcriptional silencing of retrotransposons and other genetic elements in germ line cells. lncRNAs are a broad class of ncRNAs longer than 200 nucleotides. Recent studies indicate that there are tens of thousands of lncRNAs in mammals, but few of them have been demonstrated to be biologically relevant. They are mainly involved in the regulation of gene transcription, by acting on the specific gene or the whole transcription machinery. Some lncRNAs are also involved in aging, human neurological diseases and methastasis.

2.1.5 Biological networks

Real phenomena are characterized by entities (e.g. people, machines, molecules) that interact (virtually or physically) with one another. The set of relationships between entities can be described by a model called *network* or *graph*.

Formally, a network $G = (V, E)$ is a data structure defined by two sets:

- V: the set of *nodes*, representing the entities of the network;
- E: the set of *edges*, representing the relations between pairs of entities. Given two nodes i and j, an edge can be described by an ordered pair of nodes (i, j).

If $(i, j) \in E$, we say that j is *adjacent* to i or equivalently, that j is a neighbor of i. The *degree* of a node i is the number of neighbors of i. A graph is *undirected* if $\forall\,(i, j) \in E$, $(j, i) \in E$, otherwise it is called *directed*.

Networks offer a powerful representation of complex structures and processes in many domains such as: computer science, sociology, communication, and more.

In computational biology, networks enable the characterization of biological processes at different levels within the cell. They can be used to describe the set of interactions and chemical reactions between various types of molecules, such as proteins, RNAs, genes and metabolites.

Examples of biological networks are protein-protein interaction (PPI) networks, metabolic networks and transcriptional regulatory networks (Fig. 2.8).

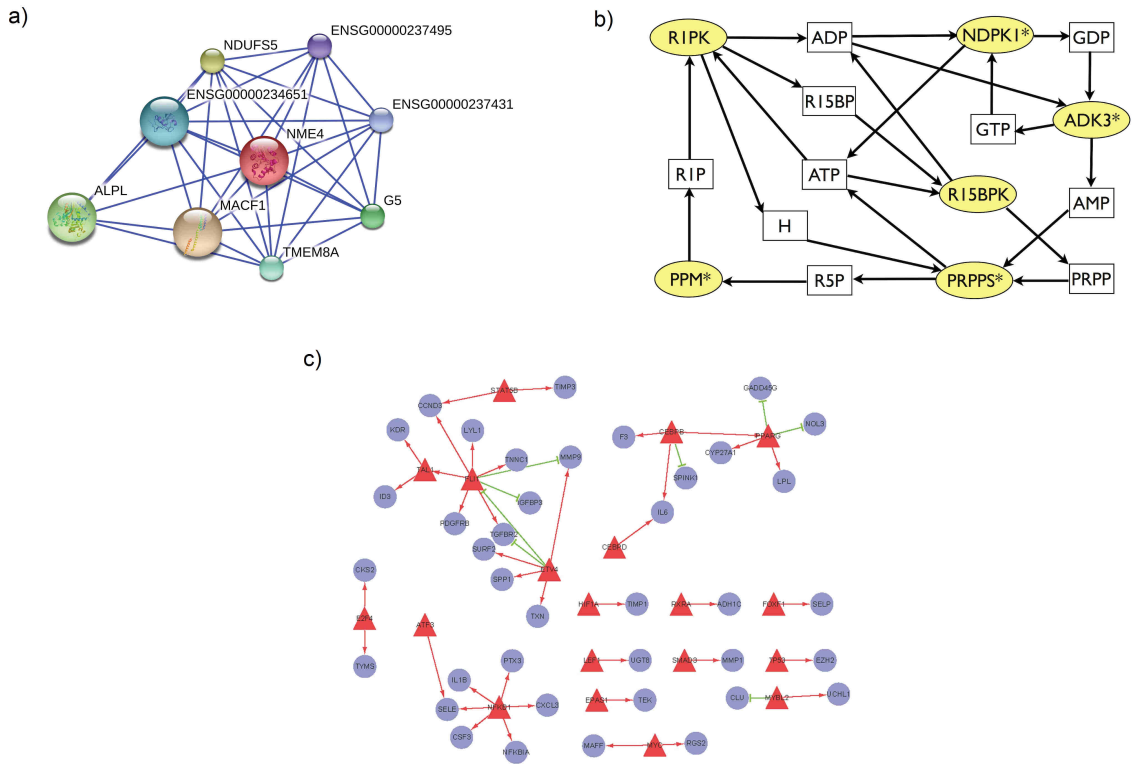


Figure 2.8: Three examples of biological networks: a) Predicted protein interactions of TMEM8A in human [Source: http://commons.wikimedia.org/wiki/File:Protein_Interaction_Network_for_TMEM8A.png]; b) A portion of the metabolic network of *Escherichia coli* [32]. Rectangles denote metabolites and ellipses represent enzymes; c) Reconstruction of a transcriptional network of genes involved in human adenocarcinoma [99]. Red triangular nodes are transcription factors, while blue circular nodes represent target genes. Red and green edges represent upregulation and downregulation, respectively

Protein-protein interaction networks

The term protein-protein interaction (PPI) refers to intentional physical contacts established between two or more proteins as a result of biochemical events and electrostatic forces.

There are two kinds of processes in which proteins interact: cellular signaling and complex assembly. In the first process, an extracellular signal or stimulus is transduced into the nucleus in order to initiate gene transcription. Signal transduction involves ordered sequence of biochemical reactions in which proteins activate other proteins usually by phosphorylation. In complex assembly, a set of proteins is assembled together to build a larger cellular machinery.

PPIs can be predicted or experimentally validated [148, 147]. Both methods are not perfect, because they can miss true interactions or produce interactions that are not present in nature (high false positive and false negative rates). Experimental methods to detect PPIs can be divided into low-throughput (LT) techniques (such as X-ray crystallography or NMR spectroscopy [158]) and high-throughput (HT) techniques (such as yeast two-hybrid screening [160, 66] and affinity purification coupled to mass spectrometry [132]). LT methods study interactions individually at the atomic level. They have higher quality but lower coverage than HT methods, since they only focus on a subset of proteins.

The collection of all PPIs in the cell form the *PPI network* (Fig. 2.8, a), where nodes are

proteins and edges represent physical interactions between proteins. Edges can be weighted with probabilities, indicating the reliability of the interaction. Probabilities can be evaluated either experimentally, as a measure of the strength of the observed interaction, or computationally, e.g. considering the duration of the interaction or the number of publications reporting it [162].

Metabolic networks

Metabolic networks (Fig. 2.8, b) are used to study and model the metabolism, that is the set of biochemical reactions in the cell that allow an organism to grow, reproduce and respond to the environment.

Nodes of metabolic networks are *metabolites* and *enzymes*. Metabolites are small molecules resulting from intermediate and final products of metabolism. They include, for instance, glucose, amino acids and polysaccharides. Enzymes are proteins that catalyze chemical reactions. Edges are directed and correspond to metabolic reactions. Each reaction convert one metabolite into another, with the support of one or more enzymes.

Transcriptional regulatory networks

The transcription of a gene can be activated or repressed by *transcription factors*, which are proteins that bind to a regulatory region of the gene to regulate the production rate, during the transcription process.

Transcriptional regulatory networks (Fig. 2.8, c) describe a set of transcriptional interactions through which transcription factors affect the production of other genes. Nodes correspond to genes, which can be linked by two different kinds of directed edges, indicating either stimulation or repression of gene transcription.

Transcriptional networks can also contain self edges, i.e. edges starting from and ending to the same node. Self edges indicate that the product of a gene can regulate the transcription rate of the gene itself.

2.2 Statistics

In this section we first introduce some basic concepts of probability theory. Then, we describe Monte Carlo methods and Markov chains. Finally, we present Monte Carlo Markov chain methods, with particular emphasis on Gibbs sampling, which is used by the algorithms presented in this thesis.

2.2.1 Introduction to probability

Probability theory is the branch of mathematics concerning the analysis of random phenomena or events. *Probability* is the measure of the likelihood that such events will occur. It is a real number between 0 and 1, where 0 denote an event that will never occur and 1 represent an event that will certainly happen. The higher is the probability, the higher is the chance that the event will occur.

Formally, an event can be represented a propositional formula which is true or false with some probability and involves one or more random variables. A *random variable* is a numerical quantity that can take a value in a set of possible continuous or discrete values, called *range*.

By convention, random variables are represented with uppercase symbols (e.g. X, Y, Z) and their values are denoted with lowercase letters (e.g. x, y, z). Notation $P(X = x)$ indicates the probability that variable X takes value x.

Let R be the range of a discrete random variable X. The set $P(X) = \{P(X = x), \forall\, x \in R\}$ is the *mass probability distribution* of X. If X is continuous, then $P(X)$ is called *density distribution*. Common examples of distributions include the uniform, the Poisson, the geometric, the binomial and the gaussian or normal distributions.

Given n random variables X_1, X_2, ..., X_n, the *joint probability distribution* $P(X_1, X_2, ..., X_n)$ is the probability that all the n events represented by the random variables occur. The computation of the joint probability distribution is related to the concept of independence of random variables.

Two or more random variables are *independent* if knowing the value of one of them does not affect in any way the probabilities associated with the possible values of any of the other random variables. A common example of independent random variables is given by variables representing the outcomes of two different rolls of a dice. Given two independent random variables X and Y, the joint probability distribution is:

$$P(X, Y) = P(X) \times P(Y). \tag{2.1}$$

In the case of two or more *dependent* variables, the probability that a variable assume a certain value depends on the outcomes of the other variables. For instance, if we roll a dice twice and variables X and Y represent the probability of getting a number less or equals than 3 in the first roll and an odd number in the second one, respectively, then X and Y are dependent. In fact, if $X > 3$, then $P(Y) = 1/3$, otherwise $P(Y) = 2/3$.

For n dependent random variables X_1, X_2, ..., X_n, the joint probability distribution is:

$$P(X_1, X_2, ..., X_n) = P(X_1, ..., X_i) \times P(X_{i+1}, ..., X_n | X_1, ..., X_i) \tag{2.2}$$

for $1 \leq i \leq n-1$. $P(X_{i+1}, ..., X_n | X_1, ..., X_i)$ is the *conditional probability distribution* of $X_{i+1}, ..., X_n$ given $X_1, ..., X_i$ and represent the probability of observing some values for $X_{i+1}, ..., X_n$ variables, given values for $X_1, ..., X_i$.

In the case of two dependent random variables X and Y:

$$P(X, Y) = P(X) \times P(Y|X) = P(Y) \times P(X|Y). \tag{2.3}$$

From Eq. 2.3 it follows:

$$P(Y|X) = \frac{P(Y) \times P(X|Y)}{P(X)}. \tag{2.4}$$

Eq. 2.4 is known as the *Bayes' theorem*, which is widely applied in statistical inference.

2.2.2 Monte Carlo methods

Monte Carlo methods are a broad class of algorithms based on repeated random sampling to obtain numerical results. Their name comes from the resemblance of the technique to the act of playing and recording results in a real gambling casino. The modern version of the Monte Carlo method was invented in the late 1940s by Stanislaw Ulam [102].

Given a problem P to solve, let D be its domain and S its exact solution, which is actually unknown. The goal is to find an approximation of S for P. A standard Monte Carlo method performs the following steps:

- Generate inputs randomly from a probability distribution over D;
- Perform a deterministic computation on the inputs;
- Aggregate the results.

The deterministic computation is usually performed on a specific parameter of interest, upon which the solution of the problem depends.

An example of Monte Carlo simulation is the approximation of π value (Fig 2.9). Consider a circle c inscribed in a unit square s. Since the ratio between the area of circle and the area of square is $\pi/4$ we can approximate π using the following Monte Carlo method:

- Choose randomly n objects over the square;
- Count the number n_c of objects inside the circle;

- Compute the ratio n/n_c and multiply it by 4;

Indeed, the ratio n/n_c is an estimate of the ratio between the two areas, which is $\pi/4$. Multiplying the final result by 4, we can yield an approximation of π. In order to obtain a good estimation of π, n should be large enough and objects have to be uniformly distributed over the square. In Fig. 2.9 the simulation with $n = 3000$ objects yields an estimate of π within 0.007% of the actual value.

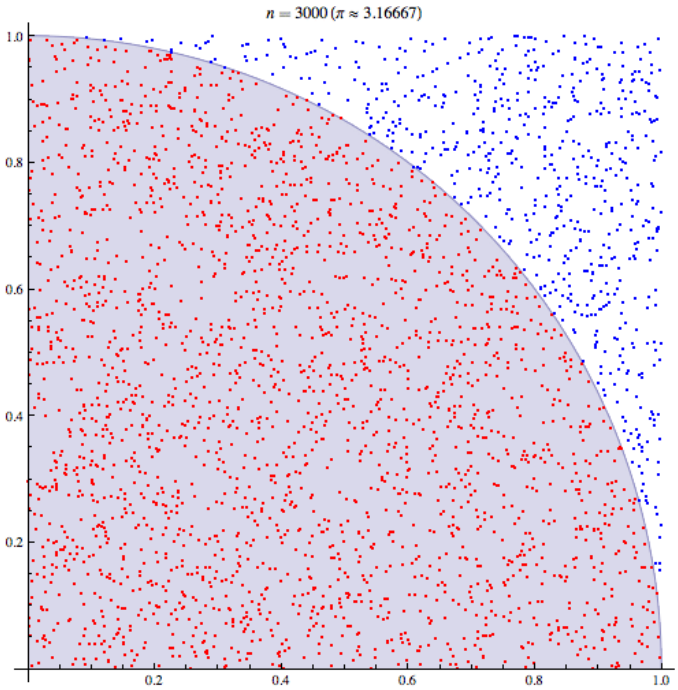


Figure 2.9: An example of Monte Carlo simulation: the approximation of π value. n is the total number of sampled objects [Source: http://commons.wikimedia.org/wiki/File:Pi_30K.gif].

Monte Carlo methods are often used in physical and mathematical problems, where the solution is difficult to express in an exact mathematical form, or applying a deterministic algorithm is unfeasible. Typical applications of Monte Carlo methods include simulation of complex systems (i.e. fluids and cellular structures), optimization, multidimensional numerical integration, prediction and generation of samples from a probability distribution.

2.2.3 Markov chains

Many real processes are characterized by a sequence of events, where the outcome of an event depends on the previous outcomes. In the simplest case, the process is "memoryless", meaning that the outcome of the next event is influenced only by the current state of the process and not by past events. The last property is called *Markov property*.

A *Markov chain* is a discrete-time stochastic process modeled as a set of states, representing events. At each time point, the chain is characterized by a transition from one state to another.

Formally, a Markov chain is a triple $(Q, P(\pi_1), A)$ where:

- Q is a finite set of n states (or events);
- π_i for $i = 1, 2, \ldots$ is the state of the chain on time i. $P(\pi_1)$ is the initial probability distribution of states;

- $A = [a_{ij}]$ is a matrix of transition probabilities from generic state i to state j, with $\sum_i a_{ij} = 1 \, \forall \, i \in Q$.

A Markov chain is called finite if it has a finite number of states and irreducible if all transition probabilities in A are non-zero.

The *period* of a state s in a Markov chain is defined as the minimum number of temporal steps that are necessary to have a non-zero probability to go back to s, starting from s itself. A state is aperiodic if it has period equals to 1. A Markov chain is aperiodic if all its states are aperiodic, otherwise it is periodic.

A Markov chain of order m is a Markov chain where the future state depends on the last m states. A *first order Markov chain* follows the Markov property, since the probability of observing a future event only depends on the currently observed one. Formally:

$$\forall i = 1, \, 2, \, ... \, P(\pi_{i+1}|\pi_1, ..., \pi_i) = P(\pi_{i+1}|\pi_i) \tag{2.5}$$

So, given n events π_1, π_2, ..., π_n, the joint probability distribution of the n states in a first-order Markov chain, i.e. the probability of observing events π_1, π_2, ..., π_n in n consecutive time points is:

$$P(\pi_1, ..., \pi_n) = P(\pi_1) \times \prod_i P(\pi_{i+1}|\pi_1, ..., \pi_i) = P(\pi_1) \times \prod_i P(\pi_{i+1}|\pi_i) \tag{2.6}$$

In a finite, aperiodic and irreducible Markov chain, the probability of being in state i at time k varies with k, until reaching a constant value. In other words, there exists a probability distribution on states, $\varphi = (\varphi_1, \varphi_2, ..., \varphi_n)$, called *stationary distribution*, that do not vary in consecutive time points of Markov chain. Such a result is independent from the initial probability distribution $P(\pi_1)$.

Formally, φ is a stationary distribution iff:

$$\varphi \times A = \varphi. \tag{2.7}$$

A Markov chain is *reversible* if the probability of observing an event at time k (with k large enough) is the same if we run forward or backward on the chain. More precisely, a Markov chain is reversible if there exists a probability distribution π such that:

$$\pi_i a_{ij} = \pi_j a_{ji} \tag{2.8}$$

for all states i and j. Equation 2.8 is also called *detailed balance condition*. Summing the last equation over i gives:

$$\sum_i \pi_i a_{ij} = \sum_i \pi_j a_{ji} = \pi_j \sum_i a_{ji} = \pi_j \tag{2.9}$$

So, for reversible Markov chains, π is also a stationary distribution. In other words, a reversible Markov chain has also a stationary distribution. Vice versa is not true in general: there are Markov chains with stationary distributions that are not reversible.

2.2.4 Markov Chain Monte Carlo methods

Given a finite, aperiodic and irreducible Markov chain, its stationary distribution can be obtained by solving Equation 2.7. Conversely, building a Markov chain following a certain stationary distribution is a hard problem, that can be solved with a special class of Monte Carlo simulation methods, called *Markov Chain Monte Carlo (MCMC) methods*.

MCMC methods build a Markov chain where states represent different assignments of values to n variables $X_1, X_2, ..., X_n$. The joint probability distribution P of the n variables is the stationary distribution of the chain.

MCMC methods can be used to sample from joint probability distributions which are unknown or too hard to compute analitically using Eq. 2.2. The idea is to sample from a probability distribution which comes closer and closer to P, as long as we generate new samples, using a finite,

aperiodic and irreducible Markov chain. Building a new sample implies a new assignment of values and a transition from one state to another one of the chain.

If the Markov chain models a problem and we iterate sampling enough, starting from a random state, we can finally build samples representing a probable solution. Even though we can start from any feasible state, the number of steps needed to converge highly depends on the starting point.

The most widely used MCMC methods are the Hastings-Metropolis algorithm [101, 59] and Gibbs sampling [50].

2.2.5 Hastings-Metropolis algorithm

Hastings-Metropolis [101, 59] (HM) is the most famous random walk Monte Carlo method. In HM transition probabilities are defined according to a density distribution q, called *proposal distribution*. If s is the number of states of the Markov chain whose stationary distribution must be φ, q is defined such that $q_{ij} > 0$ and $\sum_{j=1}^{s} q_{ij} = 1 \; \forall 1 \leq i, j \leq s$. The transition probability p_{ij} from state i to state j is defined as $p_{ij} = q_{ij} a_{ij}$, where $a_{ij} = min\left(1, \frac{\varphi_j q_{ji}}{\varphi_i q_{ij}}\right)$.

If x is a state of Markov chain, corresponding to the assignment of values $x_1, x_2, ..., x_n$ to variables $X_1, X_2, ..., X_n$, a common choice for the proposal distribution is a Gaussian distribution G centered at x. However, the variance of G, σ^2, requires reasonable tuning for ensuring the convergence to the stationary distribution. If σ^2 is too large, almost all transition steps in the HM algorithm will be rejected and the convergence will be very slow. On the other hand, if σ^2 is too small, almost all transition steps will be accepted but the Markov chain will result in a pure random walk process with no converge to φ.

An example of application of HM is the approximation of the *Beta distribution*. A Beta distribution is defined by the following density function:

$$B_{\alpha,\beta}(X = x) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1 - x)^{(\beta-1)} \tag{2.10}$$

where α and β are two positive real parameters, $0 \leq x \leq 1$ and function Γ is defined as:

$$\Gamma(u) = \int_0^\infty e^{-t} t^{u-1} \, dt \tag{2.11}$$

We can simulate a Beta distribution $B_{\alpha,\beta}$ using a HM algorithm with a uniform distribution U in range [0,1] as a proposal distribution. The iterative HM algorithm starts generating N random samples $X_1, X_2, ..., X_N$ between 0 and 1, where N is a user-defined parameter. Such assignment corresponds to a random choice of the initial state of a Markov chain having $B_{\alpha,\beta}$ as stationary distribution. At the i-th iteration ($2 \leq i \leq N$), we replace X_i with a new sample Y generated according to distribution U with probability $\rho(X_i, Y) = B_{\alpha,\beta}(Y)/B_{\alpha,\beta}(X_{i-1})$. If the substitution is accepted, the chain undergoes a transition from one state to another.

The final state of the chain at the end of the HM algorithm corresponds to a sequence of samples generated according to a distribution D, which resembles the Beta distribution. The higher is N, the closer is D to $B_{\alpha,\beta}$. Fig. 2.10 depicts the histograms of relative frequencies of samples generated with a Beta distribution B with $\alpha = 2.7$ and $\beta = 6.3$ ('Direct Generation' plot) and a HM algorithm with a uniform proposal distribution in [0,1] and $N = 10000$ ('Metropolis-Hastings' plot). In both plots the density function of B is drawn in brown, to show the similarity between the two distributions.

In high-dimensional spaces, the convergence of HM can be very slow even if σ^2 is well-tuned. This is due to the so-called *curse of dimensionality*: when dimensionality increases, the volume of the space increases so fast that objects appear to be sparse and dissimilar in many ways. In this scenario HM needs to perform large steps, which will almost always rejected because they would land in regions of low probability. Multiple-try Metropolis [91] is a variant of HM with multiple trials at each point, allowing increased acceptance rate and step size.

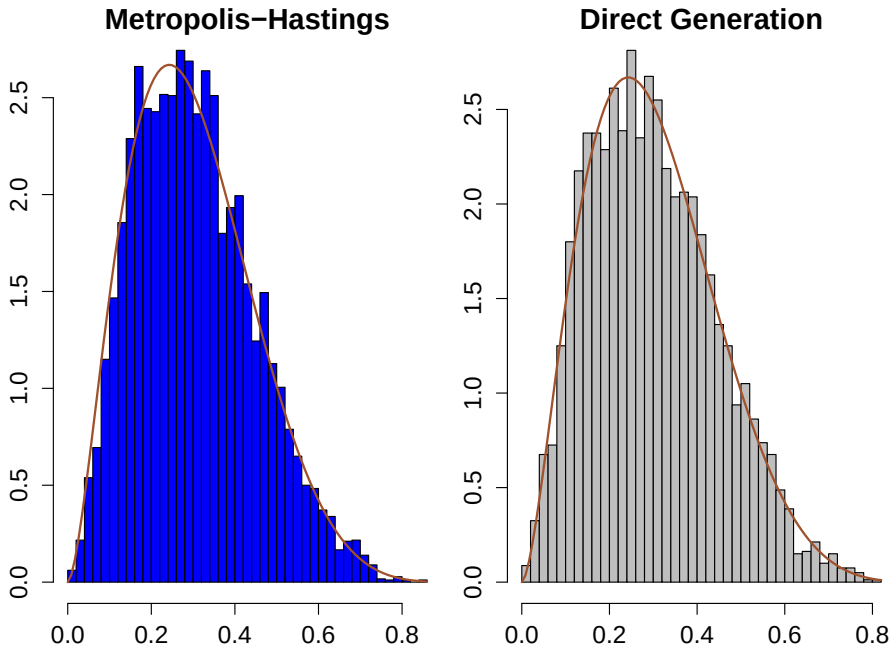


Figure 2.10: Histograms of relative frequencies of samples generated with a Beta distribution B with $\alpha = 2.7$ and $\beta = 6.3$ (right plot) and a distribution generated with HM algorithm with a uniform proposal distribution in [0,1] and $N = 10000$ (left plot). Sample values are represented in X axis, while Y axis correspond to relative frequencies. In both plots the density function of B is drawn in brown.

Reversible jump method [56] is a HM algorithm based on proposal distributions that change the dimensionality of the space. This method is useful when the simulation is performed on models where the number of parameters is unknown or should be inferred from the data.

An application of HM algorithm is simulated annealing [78, 22]. The goal is to find a minimum of some positive "energy" function f. Each value $f(i)$ is associated to a particular state E_i. In this case a new state E_j is accepted with probability $p_{ij} = q_{ij}a_{ij}$, where $a_{ij} = min\left(1, e^{\frac{(f(i)-f(j))}{T}}\right)$ where T is a fixed parameter.

2.2.6 Gibbs sampling

Let $X_1, X_2, ..., X_n$ be discrete random variables and $P(X_1, X_2, ..., X_n)$ the joint probability distribution of the n variables. Gibbs sampling [50] is a method to build a finite, irreducible and aperiodic Markov chain, whose stationary distribution is $\varphi = P(X_1, ..., X_n)$.

We first define a Markov chain whose states are all possible combinations of values for the n variables. Each state i represents a vector V_i of value assignments for variables $X_1, X_2, ..., X_n$.

Let a_{ij} be the transition probability from state i to state j. If V_i and V_j differ in more than one component, we put $a_{ij} = 0$. Otherwise, suppose for simplicity that they only differ in their first component, that is $V_i = (x_1, x_2, ..., x_n)$ and $V_j = (x_1^*, x_2, ..., x_n)$. Then, we set:

$$a_{ij} = P(X_1 = x_1^* | X_2 = x_2, X_3 = x_3, ..., X_n = x_n) \tag{2.12}$$

From Equation 2.2, it follows:

$$a_{ij} = \frac{P(X_1 = x_1^*, X_2 = x_2, X_3 = x_3, ..., X_n = x_n)}{P(X_2 = x_2, X_3 = x_3, ..., X_n = x_n)} \tag{2.13}$$

This Markov chain is finite, irreducible and aperiodic, and furthermore has stationary distribution $P(X_1, ..., X_n)$.

The chain is finite since the set of states is finite. Aperiodicity follows from the fact that $a_{ii} > 0$ $\forall i$. Since each state can be reached, after a finite number of steps, from every other state, the chain is also irreducible. Finally, $P(X_1, ..., X_n)$ is a stationary distribution because the chain is reversible and the detailed balance condition (Eq. 2.8) holds:

$$\varphi_i a_{ij} = \frac{P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) \times P(X_1 = x_1^*, X_2 = x_2, ..., X_n = x_n)}{P(X_2 = x_2, X_3 = x_3, ..., X_n = x_n)} \tag{2.14}$$

$$\varphi_j a_{ji} = \frac{P(X_1 = x_1^*, X_2 = x_2, ..., X_n = x_n) \times P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n)}{P(X_2 = x_2, X_3 = x_3, ..., X_n = x_n)} \tag{2.15}$$

A Gibbs sampling algorithm for local sequence alignment

In this subsection, we describe an application of Gibbs sampling to local sequence alignment [87], which inspired our design and implementation of GASOLINE algorithm for local network alignment presented in section 4.1.

Let $S = \{S_1, S_2, ..., S_N\}$ a set of N sequences, where characters are chosen from an alphabet Σ, and w a positive integer value. The *local sequence alignment problem* (Fig. 2.11) aims at finding a set of N subsequences, one for each sequence, of w symbols, such that the similarity between the N sequences is maximized in correspondence to the substrings and input sequences can be aligned through these common patterns.

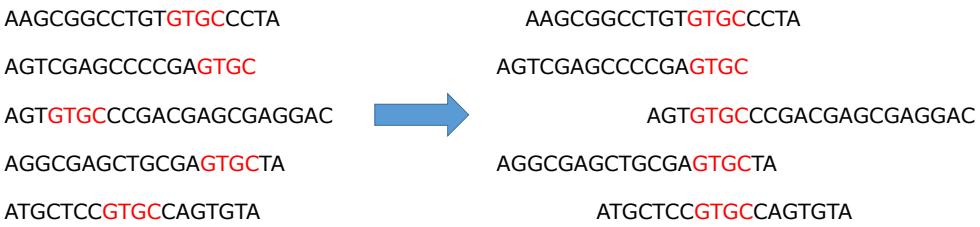


Figure 2.11: An example of local sequence alignments with N=5 nucleotidic sequences and W=4. Characters of aligned subsequences are highlighted in red.

Gibbs sampling can be used to find an approximate solution to the problem, which is NP-hard [165]. Each state of the Markov chain is a possible combination of N segments (one for each sequence) of w symbols, that is a possible alignment. A transition represents a shift from one alignment to another.

The initial state of the chain is randomly selected, by choosing one of the possible subsequences of w characters in each input sequence. The method is iterative: at each step a randomly chosen segment of w symbols in the current alignment is replaced with another segment of w characters of the same sequence according to a properly defined probability. This results in a transition in the corresponding Markov chain. After a sufficient number of steps, the chain will converge to a stationary distribution, leading to states associated to optimal alignments.

Transition probabilities are defined starting from the frequencies of characters of aligned subsequences. The idea is to reward local alignments formed by characters with low frequency in the population of input sequences, because alignments with highly frequent characters are less significant and have more chances to be found at random. We can represent the set of currently aligned segments of length w as a matrix M of characters with N rows and w columns, where row i (with $1 \leq i \leq N$) contains the segment of i-th sequence. Suppose that at a certain iteration of Gibbs sampling we remove the i-th aligned segment and let x the candidate segment of length w to replace it. We first compute two values for x, P_x and Q_x, called *background probability* and *pattern probability*, respectively.

Background probability P_x is defined as follows:

$$P_x = \prod_{j=1}^{w} p_{x_j} \tag{2.16}$$

where p_{x_j} is the relative frequency of the j-th character of x in the input sequences, excluding the whole i-th sequence and all the currently aligned segments. If a character is not present, then $p_{x_j} = 1$.

Pattern probability Q_x is computed as:

$$Q_x = \prod_{j=1}^{w} q_{jx_j} \tag{2.17}$$

where q_{jx_j} is the relative frequency of the j-th character of x in the j-th column of matrix M, excluding the i-th aligned segment. In particular, q_{jx_j} is defined as:

$$q_{jx_j} = \frac{c_{jx_j} + b_j}{N + B} \tag{2.18}$$

where c_{jx_j} is the number of occurrencies if character x_j in the j-th column of M, excluding the i-th aligned segment, $B = \sum_{j=1}^{w} b_j$ and b_j are pseudocounters. An optimal experimentally derived value for b_j is $b_j = \sqrt{N} \times p_j$.

We call Likelihood Ratio of x, $LR(x)$, the ratio between Q_x and P_x. The probability of selecting x and performing a transition from a state to another in the Markov chain is the normalization of $LR(x)$ in range [0,1]:

$$TP(x) = \frac{LR(x)}{\sum_{h \in Subseq_h(S_i)} LR(h)} \tag{2.19}$$

where $Subseq_h(S_i)$ is the set of all possible subsequences of w characters in the sequence S_i.

States of the chain corresponding to good alignments are those for which the relative entropy between the pattern and the background probabilities is high. Given a state s, the relative entropy $E(s)$ is:

$$E(s) = \sum_{j=1}^{w} \sum_{k=1}^{|\Sigma|} q_{jk}(s) \log\left(\frac{q_{jk}(s)}{p_k}\right) \tag{2.20}$$

One can prove that the Markov chain converges to a stationary distribution λ, with:

$$\lambda(s) = const \prod_{j=1}^{w} \prod_{k=1}^{|\Sigma|} \left(\frac{q_{jk}(s)}{p_k}\right)^{c_{jk}(s)} \tag{2.21}$$

where the constant is chosen such that $\sum_{s \in S} \lambda_s = 1$, where S is the set of states in the Markov chain.

Equations 2.20 and 2.20 imply that states with high entropy are also states with high stationary probability, so they are frequently visited, leading to an optimal solution.

In some cases, the above Gibbs sampling procedure can be trapped in local maxima. To avoid this and improve accuracy, aligned subsequences can be sometimes slightly shifted.

Chapter 3

Literature review

3.1 Sources of PPI and gene expression data

Nowadays, protein-protein interactions and gene expression data of various species are collected and stored in many authoritative databases, which are usually weekly or monthly updated.

Primary sources of PPI data include BioGRID [154], DIP [171], HPRD [123], IntAct [114] and MINT [90].

DIP [171] was the first database which combined information from multiple observations and experimental techniques into networks of interacting proteins for different species. HPRD [123] contains manually curated proteomic information regarding human proteins, which are annotated and linked to OMIM database [20]. BioGRID [154] collects protein-protein and genetic interactions for all major model organisms trying to remove redundancy and create a single mapping of interactions. The IntAct database [114] provides tools for both textual and graphical representations of protein interactions. Interacting proteins can be annotated with GO terms for functional analysis. MINT [90], which is based on the IntAct database infrastructure, collects experimentally verified PPIs by extracting experimental evidences from the scientific literature.

Some databases integrate PPIs data of human and other organisms from primary sources, by removing redundancies and assigning a unique reliability score. These include STRING [46], IRefIndex [131], ConsensusPathDB [73] and HitPredict [119].

STRING [46] combines physical interaction data and curated pathways of different organisms with predicted interactions from text mining, genomic features and interactions transferred from model organisms based on orthology. IRefIndex [131] is a set of tools to index and retrieve proteins and interactions from major public databases. Indexes are built according to protein sequences and taxonomy identifiers and mapping scores evaluate the quality of the mapping. ConsensusPathDB [73] integrates human protein-protein interactions, biochemical pathways, gene regulatory and drug-target interactions into a global network, containing genes, proteins and metabolites, which can be visualized, analyzed and annotated. HitPredict [119] combines PPI data from IntAct [114], BIOGRID [154] and HPRD [123], by assigning a confidence score based on sequence, structure and functional annotations of the interacting proteins. The reliability score is calculated using the Bayesian networks.

Repositories of gene expression data [133, 10, 161, 21, 48] contain information about the concentration of RNAs, genes and proteins in normal or cancer tissues and at different diseases stages. These data are available and freely downloadable.

ArrayExpress [133] and GEO [10] include gene expression data from microarray and high throughput sequencing experiments, which can be easily queried or downloaded. Users can also submit data directly by using the standard MIAME format. More recently, new projects have started with the aim of cataloging tissue or tumor sequencing data. The Cancer Genome Atlas (TCGA) (http://cancergenome.nih.gov) collects complete high-throughput genome data (clinical information, expressions data, methylations, mutations) for specific cancer tissues, with the pur-

pose of helping the diagnosis and the treatment of cancers. The Human Protein Atlas [161] is a database with histological images showing the spatial distribution of proteins in normal and cancer tissues. Protein Atlas contains also transcription expression levels, protein expression profiles and subcellular localization data. The cBio Cancer Genomics Portal [21, 48] integrates TCGA and published cancer genomics datasets and provides a tool for the visualization and analysis of gene mutations and altered pathways in different cancer types across a set of patients.

3.2 Stochastic methods for biological network analysis

Most applications of Markov chains and Monte Carlo methods concern the functional classification and prediction of nodes in biological networks.

In prediction problems, Monte Carlo methods can be used for maximum likelihood estimation (MLE), expecially for complex stochastic processes, such as Markov Random Fields (MRF), where the exact MLE is not feasible. MRFs are undirected graphs which describe a set of variables satisfying the Markov property and their dependencies.

Deng et al. [34] use a MRF to infer the function of a protein using PPI data and the functional annotation of its interacting partners. Posterior probabilities for the function of a protein are estimated through Gibbs sampling, starting from the annotation of neighbor proteins. Later, the method was extended [33] by including genetic interactions, correlated networks and domain protein structures for prediction. Letovsky et al. [88] assume that the number of neighbors of a protein that are annotated with a given function is binomially distributed. They combine their binomial model with a MRF propagation algorithm.

In [168], Wei et al. use local discrete MRFs to identify differentially expressed genes and pathways linked to diseases. The MRF model is used to capture the dependencies of differential expression patterns for genes and their upstream regulators in the networks. In [169] the same authors define a method based on hidden spatial-temporal Markov random field to identify pathways which are modified or activated in a biological process, combining the pathway structure with time course gene expression data.

Sanguinetti et al. [137] describe a probabilistic model to find differentially expressed submodules in metabolic pathways. Differentially expressed enzymes are identified using Gibbs sampling, then the submodule is grown by iteratively adding all the neighbouring nodes in the same class. Plata et al. [124] annotate enzymes of metabolic networks through Gibbs sampling, by integrating sequence homology and context-based correlations using a unified probabilistic framework, called GLOBUS. Their method also suggests possible alternative functions for each metabolite.

Very recently, Chen et al. [23] propose a multiple data integration method similar to [33] for predicting candidate disease genes in human. To deal with unlabeled data, they assign prior labels to unknown vertices and then perform a pseudolikelihood parameter estimation method on all vertices. Wu et al. [170] employ Markov chains with restart to improve the prediction of a protein's function whenever the number of labeled protein data is limited or expensive to obtain.

Most application of Markov chains for clustering of biological networks are based on the Markov Cluster (MCL) algorithm [39], an iterative clustering algorithm on general graphs that simulates random walks using Markov chains.

In [18], MCL algorithm was applied for the first time to biological networks. Compared to other state-of-the-art graph clustering algorithms, MCL is more noise-tolerant and accurate in discovering high-quality protein complexes. Such results are confirmed in [164].

Satuluri et al. [139] propose an enhanced version of MCL, called Regularized MCL (RMCL), which penalizes large clusters at each iteration of MCL to obtain more balanced clusters. Their method show better accuracy in identifying clusters with potential functional specificity. Shih et al. [145] describe a soft variation of RMCL, in order to find overlapping clusters. The resulting algorithm consists in iterative executions of RMCL to ensure that multiple executions do not always converge to the same clustering result.

Wang et al. [167] propose a joint clustering algorithm to reduce the effects of the noise and the poor data quality of biological networks. They first build an integrated network, combining

topology and homology information from two PPI networks. Then, they perform a Markov random walk on the integrated network using both topology and homology data for computing the transition matrix.

Very recently, stochastic methods have been applied to comparative analysis of biological networks.

In [128] Qian et al. use a Hidden Markov Model (HMM) to find the best matching of a query path in a biological network. HMMs [14] are Markov chains where some states are hidden or unobservable. The querying problem is transformed into the problem of finding the sequence of states in the HMM that maximizes the observation probability of the query path. In their HMMs they also define accompanying states, in order to model insertions and deletions. The method was then extended [129] to identify conserved paths across two or more biological networks of different species. They consider a HMM for each compared network and look for an optimal set of state sequences in the HMMs that jointly maximize the observation probability of the conserved path.

The efficiency and accuracy of both methods have been improved [134, 127], by including global correspondence scores between matching proteins in the HMMs. An iterative procedure based on a semi-Markov random walk procedure is proposed to compute such scores and prune the search space. Sahraeian et al. [136] use such correspondence scores for multiple global comparison of biological networks, in order to estimate matching probabilities between nodes.

Mina et al. [109] apply the MCL algorithm to identify dense and conserved modules in two PPI networks of different species. They run MCL on the alignment graph, a graph where edges weights represent the likelihood of ortholog pairs of proteins to interact.

3.3 Network alignment

Given N networks, network alignment problem consists in finding a set of N subnetworks of W nodes, one for each network, such that their subgraph similarity is maximized, according to a properly defined similarity score, defined on top of label and topology similarities between corresponding nodes. W is also called the size of the alignment.

The final goal of network alignment is to produce a mapping between nodes of the different networks. Correspondence between nodes can be one-to-one or many-to-many. In one-to-one mapping, each node is matched with exactly one node of each remaining network. In many-to-many mapping, one or more nodes in a network can be mapped to one or more nodes in every other network.

Network alignment can be local or global. In local alignment, W is very small compared to the average number of nodes in the aligned networks. The goal of local alignment is to find local conserved subnetworks and align the N networks through these common patterns. Global alignment methods compare the global structure of the networks and it is often equivalent to graph isomorphism. Network alignment can be pairwise or multiple, whether it involves two or more networks, respectively.

Fig. 3.1 depicts an example of network pairwise alignment of size $W = 5$. The final mapping of the alignment is represented by dashed lines connecting nodes of different networks. In the example of Fig. 3.1 the mapping is one-to-one.

Alignment methods usually follow a seed-extend approach: they start from promising seeds (alignments of edges or very small subgraphs) and greedily extend them. A classic search strategy involves the construction of a merged representation of the aligning networks, called network alignment graph, in which nodes represent sets of similar proteins, one or more for each network, and links represent conserved protein interactions.

Network alignment has been extensively studied in the field of computational biology to compare biological networks.

Local alignment of biological networks aims at identifying conserved small sets of proteins, called functional modules, in two or more different species. Conserved functional modules can represent proteins which perform similar biological functions or are involved in similar processes across distinct species.

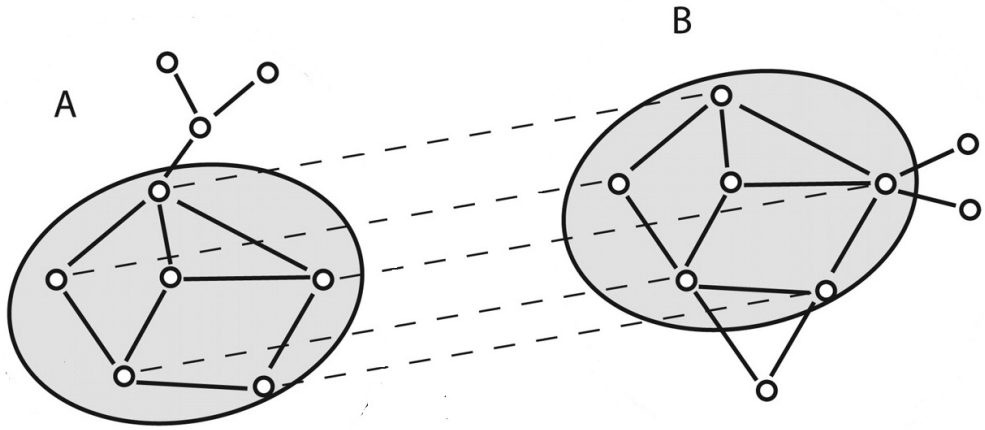


Figure 3.1: Example of pairwise network alignment of size $W = 5$. Dashed lines connect correspondent nodes in the final mapping of the alignment.

Functional modules include pathways or complexes. Pathways are chains of proteins that activate one another, usually by phosphorilation, to transduce an extra-cellular signal into the nucleus. Complexes are set of proteins assembled together to build a larger complex machinery in order to perform a specific function.

Methods for local alignment of biological networks usually produce a many-to-many mapping, considering the presence of two or more (possibly diverging) copies of a protein in the same species, called homologs, which perform a similar function. However, many-to-many mapping can be sometimes ambigouos, since two proteins of the same organism which map to the same protein in another species are not necessarily homologs. So, one-to-one mapping is a plausible alternative for local alignment, despite its limitations.

The first attempt to perform local PPI network alignment was proposed by Kelley et al. [76]. They describe PathBLAST, a method to identify common pathways in two different PPI networks, by taking into account the probabilities that PPIs in the path are true PPIs and not false positives.

Later, PathBLAST was generalized to identify protein complexes [142]. The algorithm, called NetworkBLAST, can align up to three networks and use a likelihood-based scoring scheme that measures the density of a complex versus the chance of observing such substructure at random. In [71] the authors extended NetworkBlast, by introducing a new data structure, called k-layered alignment graph, in order to align more than three networks in few minutes.

Koyuturk et al. [81] present an evolutionary-based scoring function relying on the duplication/divergence model, which is used to explain the evolution of PPI networks via preferential attachment. Their algorithm, called MaWISH, performs pairwise local alignment.

Graemlin [45] scores a module by computing a likelihood ratio which takes into account evolutionary constraints and uses phylogenetic relationships between species to perform local alignment.

AlignNemo [26] proposes a novel approach to weight edges in the alignment graph, based on the number, reliability and local significance of paths in two input networks. Connected small subgraphs are then extracted from the alignment graph and iteratively expanded through a merging strategy.

NetAligner [117] uses the differences of the evolutionary distances to predict the probabilities for likely conserved interactions between the protein pairs. Conserved subnetworks are identified as connected components in the alignment graph through a DFS visit, allowing gaps and mismatches.

In AlignMCL [109] the alignment graph is built in two steps. The initial graph only contains conserved and likely-conserved interactions and it is used to find promising seed. The graph is then extended with predicted potentially conserved interactions, to reduce false negatives. Finally, Markov clustering is applied to extract conserved modules from the alignment graph.

In analogy to genomic sequence alignment, several global network alignment methods have

been proposed to investigate phylogenetic relationships between species, considering the global interactome of different organisms. The quality of a global alignment is usually measured in terms of edge conservation.

In [83, 107, 84, 98], authors propose different global pairwise alignment methods based only on network topology. All algorithms match nodes that share a similar distribution of graphlets, which are small connected non-isomorphic induced subgraphs. MI-GRAAL [84] is the most general one and can integrate multiple types of similarity measures between network nodes, including degree difference and local clustering coefficient difference.

Very recently, new pairwise global alignment methods have been proposed to directly optimize an edge cost function and improve accuracy. MAGNA [138] optimizes edge conservation by using a genetic algorithm, which simulates a population of alignments that evolve over time. The alignments that best fit a given function can survive and proceed to next generation, until the alignments cannot be further refined. Crawford et al. [27] design an edge similarity function based on graphlets. Their method, GREAT, starts building an optimal edge alignment through a greedy strategy, then it defines a node cost function and an optimal node alignment on top of the edge alignment.

IsoRank [149] is the first proposed multiple global alignment method, where nodes with similar neighborhoods are matched. Their method, called IsoRank, uses spectral graph theory to find pairwise alignment scores, that are finally used in a greedy algorithm to compute the final alignment. IsoRankN [89] improves IsoRank, by using a different spectral clustering method similar to the PageRank-Nibble algorithm developed by Andersen et al. [6].

SMETANA [136] deploys a semi-Markov random walk model to compute probabilistic node similarity measures. The latter are combined with local and cross-species network similarity information to produce alignment probabilities, which are used to predict the final alignment.

BEAMS [4] decomposes the alignment problem into two subproblems, backbone extraction and backbone merging, where a backbone is defined as a small set of protein pairs of different networks with high sequence similarity. Both extraction and merging phases are performed in a greedy fashion.

Very recently, Gligorijevic et al. [53] proposed FUSE, which combines topology and sequence similarities by using a non-negative matrix tri-factorization method. Global alignment is then modeled as a maximum weight k-partite matching problem.

Few tools are available for the visualization and analysis of network alignments. In [108] Mina et al. present a plugin for Cytoscape [140] based on the AlignMCL algorithm [109], which offers a graphical interface for the method. NetCompare [173] is a tool developed for Galaxy platform [55] to visualize networks alignments produced by different alignment methods. It uses a multi-hierarchical layout to display alignments at various levels of resolution. However, both tools are limited to pairwise alignments.

Current local and global alignment methods have two important limitations. They are not scalable with the number of aligning networks and have low accuracy, especially in the multiple-network case.

3.4 Comparison of protein structures

Protein function is largely determined by its 3D structure. Since the structure of many proteins is still unknown and proteins with similar structural motifs often exhibit similar biological properties, even when they are distantly related, 3D structure alignment can help to characterize the role of many proteins, similarly to protein sequence alignment.

The ultimate goal of protein structure alignment is to find an optimal rigid-body superposition between structures, which is the one that minimizes the RMSD (Root Mean Square Deviation) distance between representative atoms of the aligned or matched residues [70]. However, the problem is NP-hard [42] and there is a huge variety of plausible strategies and models that can be used to compare protein structures [43].

First of all, protein structure alignment can be global or local. Local alignment can help to identify conserved structural motifs or binding sites within a family or different families of proteins. Local alignment methods can be sometimes more sensitive than global ones, since even proteins with dissimilar folds may share common binding sites or interfaces.

Moreover, there is no clear notion of common ancestor, so similarity scoring schemes can be defined starting from different features, such as euclidean distances between residues, chemical properties or geometrical conformations.

Finally, structural alignment can be rigid, flexible or elastic, depending on how structural variations (e.g. plastic deformations, shifts and rotations of the secondary structure elements) are treated.

Some methods perform the comparison task directly on the 3D protein structures. Pure 3D structure global aligners include DALI [62], VAST [52], SSAP [116], CE [146], SSM [82], MultiProt [143], TM-Align [177], MATT [100] and DeepAlign [166].

DALI [62] uses intramolecular distances to compute similarity scores. Distance matrices are first splitted into contact patterns, which are then combined into larger consistent sets of pairs. A Monte Carlo method is used to optimize alignments in parallel. DALI was first developed as a web server, then it was implemented as a standalone program, DaliLite [61], with a web interface for visualizing results.

VAST [52] starts matching secondary structure elements that have similar type, orientation and connectivity. Then, it builds a graph where nodes are these matches and edges connect corresponding pairs of elements whose angle and distance is within a threshold. An alignment is found by solving a maximal clique detection problem and extended through Gibbs sampling. SSM [82] follows a similar strategy: an iterative 3D graph matching procedure based on singular value decomposition is applied to build the optimal alignment starting from a labeled graph of similar secondary structures elements.

SSAP [116] starts with the construction of a series of inter-residue distance vectors between each residue and its nearest non-contiguous neighbors, considering beta carbon atoms. Double dynamic programming is applied to a series of matrices containing the vector differences to find the overall multiple structural alignment. SSAP has been used to produce the CATH hierarchical fold classification scheme [115].

In CE method [146], aligned fragments in two protein structures are represented as alignment paths, which are selectively extended or discarded through a combinatorial procedure.

MultiProt [143] finds common geometrical cores and does not require that all the input structures are aligned, allowing high scoring partial alignments. Spatial similarities of amino acids are searched disregarding the order of the residues on the chain.

In [177], authors define a new score, TM-score, which weights close aligned atom pairs stronger than distant matches and is more robust to local variations than RMSD. Their algorithm, TM-Align, iterates a dynamic programming procedure to compute alignments, resulting much faster than the previous methods.

MATT [100] introduce geometric flexibility to refine a multiple structure alignment. A dynamic programming algorithm assemblies fragments of aligned residues, by allowing small translations and rotations. Geometric consistency is then used to output the final alignment.

DeepAlign [166] includes amino acid and local substructure substitution matrices in the scoring function, which is iteratively optimized in order to build pairwise structural alignments that are also meaningful from an evolutionary perspective.

An alternative approach proposed for protein structure comparison is based on 2D models, called contact maps, which are graphs where nodes are amino acids and edges connect residues having a distance lower than a fixed cutoff, usually 7-12 Å. In this case, an optimization problem on contact maps, called Maximum Contact Map Overlap (MCMO), is solved [54]. MCMO is the sequence alignment that maximizes the number of corresponding contacts between pairs of aligned residues is computed.

The first exact algorithm for MCMO was developed by Lancia et al. [86]. The problem is formulated with integer linear programming and solved using lagrangian relaxation and branch-and-bound reduction techniques. However, their algorithm is exponential in the worst case. A

similar technique is used by Andonov et al. [7], which present a faster exact algorithm, Apurva, with better upper and lower bounds than those found in [86].

In [176, 2] two polynomial-time approximation schemes for contact map alignment are proposed. Xu et al. [176] decomposes protein structures using tree decomposition and discretizing the rigid-body transformation space. Their algorithm is polynomial in the protein sequence length but exponential with respect to some constant parameters. The method developed by Agarwal et al. in [2] is based on a decomposition procedure on the input graphs. It is a six-approximation algorithm with a polynomial running time.

Di Lena et al. [37] map the MCMO problem to a one-dimensional global alignment of eigenvectors, based on the property that a contact map can be well approximated by few eigenvectors of its nodes. They propose an heuristic eigendecomposition method, called Al-Eigen.

MSVNS [122] uses a variable neighborhood search strategy to compute an approximate solution to MCMO problem. The algorithm is based on a local search approach including dynamic changes in the neighborhood of the solutions.

Very recently, Dognin et al. [96] proposed GR-Align, a fast heuristic method which implements a matching cost function based on graphlet degree similarities. The matching task is performed through a Needleman-Wunch algorithm. GR-Align is several orders of magnitude faster than Apurva, MSVNS and Al-Eigen.

Local structure alignment methods are all pairwise and include MolLoc [8], MultiBind [144, 121], MAPPIS [120, 121], LabelHash [110], ProBiS [79, 80] and SMAP [174, 175].

MolLoc [8] is a web server for comparing known binding sites, cavities or user-defined sets of residues of two or more molecular surfaces. The algorithm builds a structural alignment maximizing the extension of surfaces superimposition.

MultiBind [144, 121] recognizes common spatial chemical binding patterns in a set of proteins, solving a 3D k-partite matching problem through efficient geometric hashing techniques. MAPPIS [120, 121] relies on a similar algorithm and performs multiple alignment of protein-protein interfaces, predicting hot spot residues that contribute to the conserved patterns of interactions.

LabelHash [110] pre-computes reference hash sets to guarantee instant lookup of partial matches to motifs. Then, partial matches are expanded using a variant of the match augmentation algorithm [24].

Unlike the prevoius methods, ProBiS [79, 80] and SMAP [174, 175] can align two protein structures with no information about the location of potentially conserved binding sites.

ProBiS [79, 80] identifies in both structures sets of solvent accessible surface atoms and represent them as graphs where nodes are the functional groups associated to these surface residues. A product graph is then built considering pairs of vertices with identical chemical properties. Local alignments are found using a maximum clique algorithm on the product graph.

SMAP [174, 175] characterizes protein structures using geometric potential and it is based on a sequence order independent profile-profile alignment tool, called SOIPPA. SOIPPA integrates geometric, evolutionary and physical information into a unified similarity score and produce local alignments which are indipendent from sequence order.

All methods discussed above are limited to the comparison of two protein structures and they are generally slow. Moreover, they do not perform well with proteins with dissimilar sequences, which could still share local regions of high structural similarity.

3.5 Analysis of tissue-specific PPI networks

In the last few years, the annotation of PPI networks with external data (i.e. diseases, expression data, phenotypes) has helped to classify genes according to the expression profiles [31], predict new gene-disease associations [64, 178] and discover new drugs [64, 29, 3].

Tissue-Specific PPI (TS-PPI) networks [17] represent a powerful model to highlight the implication of some genes in specific disease or tumors. In fact, human diseases often occur in specific tissues [85]. Some genes can be predominantly expressed in one or few tissues and can control the formation of protein complexes [44]. Furthermore, genes can use alternative splicing as a powerful

mechanism to enlarge the number of their interactors and perform distinct functions in different tissues [44]. Therefore, the integration of PPI networks with tissue-specific gene expression data can help to highlight the role of some genes in specific disease or tumors. A TS-PPI network is a subgraph of a PPI network where the genes corresponding to both interacting proteins are expressed in one or more tissues.

Some studies focus on the analysis of global and local properties of TS-PPI networks. In [17] authors prove that most housekeeping proteins form highly tissue-specific protein interactions, suggesting a key role of those proteins in tissue-specific biological processes. Emig et al. [44] show that the number of tissue-specific proteins is very low and the receptor-activated signaling processes and the transcriptional regulation are two key factors for tissue specificity. In [153] a gradient model is used to describe the structure of TS-PPI networks, containing interactions of regulatory and developmental functions at the core of the TS-PPI network and physiological functions at the periphery.

Several recent works highlight the advantages of using TS-PPI networks. In [93], a set of proteins related to the response of viral infection in a TS-PPI network lead to a more reliable functional enrichment. Magger et al. [95] use TS-PPI networks to improve the prioritization of candidate disease-causing genes with respect to a generic PPI network. In [25], authors identify functional modules in TS-PPI networks using CFinder [1] and show that they exhibit more biological meaning than modules in a PPI network. Xiao et al. [172] propose a new method for the identification of multi-tissue gene co-expression networks associated with specific functional processes relevant for phenotype variation and disease in humans. Barshir et al. [12] show that genes causing hereditary diseases tend to have higher transcript levels and more interacting partners in the TS-PPI network of disease tissues than in the TS-PPI network of unaffected tissues.

Very recently, some tools have been proposed for querying and analyzing TS-PPI networks [11, 113]. CyKeggParser [113] is a Cytoscape app for generating and analyzing tissue-specific KEGG pathways. Pathways can be checked for inconsistencies and modified based on gene expression data from normal and cancer tissues. TissueNet [11] is a dataset of TS-PPIs in humans, which integrates a collection of four PPI networks (BioGRID, DIP, IntAct and MINT) with three expression datasets (GEO, Human Protein Atlas and Illumina Body Map 2.0). The database provides a web interface for retrieving tissue-specific interactions of a query protein. However, it handles only 16 normal tissues and does not provide any tool for the analyses of TS-PPI networks.

The analysis of TS-PPI networks is an emerging research field. As far as we are concerned, no method has been proposed to compare two or more TS-PPI networks of different tissues or tumors and there is no unique framework for retrieving and comparing TS-PPI networks.

Chapter 4

Mining of protein networks

In this chapter, two applications of Gibbs sampling for the comparison of protein-protein interaction (PPI) networks are described [105, 103, 104]. The first one, GASOLINE, is a method for the multiple local alignment of PPI networks [105]. We present a detailed description of the algorithm and describe a Cytoscape app based on the namesake algorithm for the computation, visualization and analysis of alignments within Cytoscape framework [103]. The second application, SPECTRA, is a knowledge base for retrieving and analyzing tissue-specific PPI (TS-PPI) networks, which uses an adapted version of GASOLINE to identify differentially expressed genes across multiple TS-PPI networks [104]. We finally illustrate and discuss experimental results, where we compare GASOLINE with state-of-the art tools for local alignment of PPI networks and we present a case study of adapted GASOLINE for local differential alignment of TS-PPI networks of different healthy and tumor tissues.

4.1 GASOLINE

GASOLINE (Greedy And Stochastic algorithm for Optimal Local multiple alignment of Interaction NEtworks) is an algorithm for local alignment of two or more PPI networks. It is based on iterative sampling [50] in connection with a greedy strategy. GASOLINE is inspired by the work of [87] on multiple sequence alignment and implements a seed-and-extend approach to extract conserved complexes among PPI networks of different species.

4.1.1 Description of the algorithm

Given N PPI networks of different species, GASOLINE computes a set of local alignments of the N networks. It produces an approximate solution to the problem through a stochastic-greedy strategy consisting of two phases.

During the first step called *bootstrap phase*, we look for orthologous proteins across the networks. These proteins are the so-called *seeds* of a candidate local alignment and represent the starting nodes of the suboptimal local network alignment we are searching for.

The second step, called *iterative phase*, repeatedly adds (extension step) and removes (removal step) nodes in the network alignment, trying to maximize the final alignment score. Each extension step adds, in each network, a single node to the corresponding seed. Therefore, during the extension step the seeds iteratively increase and become a set N subgraphs, one from each network.

The extension process is regulated by a properly defined *degree ratio*, which measures the average density of the aligned subgraphs with respect to their neighbors in the remaining parts of the networks. The extension is performed until the degree ratio increases.

Each removal step replaces from the current alignment the set of proteins (one from each network) providing the minimum score.

The initial phase and each extension step are performed through an iterative Gibbs sampling procedure. Consequently, different iterations of the algorithm may produce different local alignments. GASOLINE iterates the above steps and gives as result a set of local network alignments. Those local alignments are ranked according to an Index of Structural Conservation (ISC) score relying on topology and sequence similarity.

GASOLINE also implements preprocessing and post-processing steps. During preprocessing, the search space for potential seeds is reduced. This is obtained by marking only proteins having orthologs in all aligning networks and with a significant interaction degree in each network. All marked nodes in each network G_i $(1 \leq i \leq N)$ are added to a set called S_i. These sets will be used in the initial phase and will be updated at each iteration. More precisely, at the end of the iterative phase the aligned seeds are removed from the sets S_i (in order to guarantee termination) and the process starts again from the bootstrap phase with new seeds proteins chosen in $S_1, ..., S_N$.

Finally, during post-processing, the final set of local alignments returned by GASOLINE is filtered by removing highly overlapping complexes.

Flowchart in Figure 4.1 provides a general description of GASOLINE.

The bootstrap phase

The search for an initial set of seeds is performed by a Monte Carlo Markov Chain in connection with a Gibbs Sampling algorithm [50]. The Gibbs sampling builds a chain, where each state represents a combination (i.e. alignment) of N proteins, one from each network. First, a random initial state is selected. Then, the sampling method iteratively performs a transition from a state to another, by replacing a randomly chosen protein of the current alignment with a protein of the same network, according to a properly defined transition probability distribution. By iterating this sampling procedure a sufficient number of times, we eventually achieve a good alignment of seeds.

The transition probability is defined on top of a *Similarity Score*. Given two proteins a and b, we define their similarity score $S(a, b)$ as either their Bit Score or the inverse of their BLAST E-value [5].

Let $A^i = \{A^i_1, ..., A^i_N\}$ be the alignment of proteins at the i-th iteration of Gibbs sampling and suppose we remove the node A^i_k from it. Let p be a candidate protein replacing A^i_k. The similarity score of p is defined as the product of all similarity scores between p and the proteins still belonging to the alignment: $SIM(p) = \prod_{j=1,\, j\neq k}^{N} S(A^i_j, p)$.

The transition probability in p is then computed by using such the similarity scores as follows:

$$P(p|A^i_1, \ldots, A^i_{k-1}, A^i_{k+1}, \ldots, A^i_N) = \frac{SIM(p)}{\sum_{n\in S_k} SIM(n)} \tag{4.1}$$

Finally, the alignment score is defined as the sum-of-pairs of similarity scores between the aligned proteins:

$$\textsc{ScoreSeed} = \sum_{j=1}^{N}\sum_{k=1}^{N} S(A^i_j, A^i_k). \tag{4.2}$$

At the end of the bootstrap phase the alignment of seeds that maximizes the sum-of-pairs score over all the iterations of Gibbs sampling is chosen.

The extension of current seeds

Let $SG = \{SG_1, SG_2, ..., SG_N\}$ be an alignment of N subgraphs, one for each network and Adj_i the set of nodes adjacent to one or more nodes in SG_i. The goal of each extension step is to find an alignment $A = \{A_1, A_2, ..., A_N\}$ of N proteins where $A_i \in Adj_i$, and extend each SG_j with A_j and the edges connecting A_j with the remaining nodes in SG_j.

Fig. 4.2 shows a demo with two aligning networks. In Fig. 4.2 (a) the current alignment $SG = \{SG_1, SG_2\}$, consisting of two subgraphs composed by three nodes, is highlighted in green, with dashed lines connecting aligned proteins. Fig. 4.2 (b) highlights in red all the nodes in Adj_1

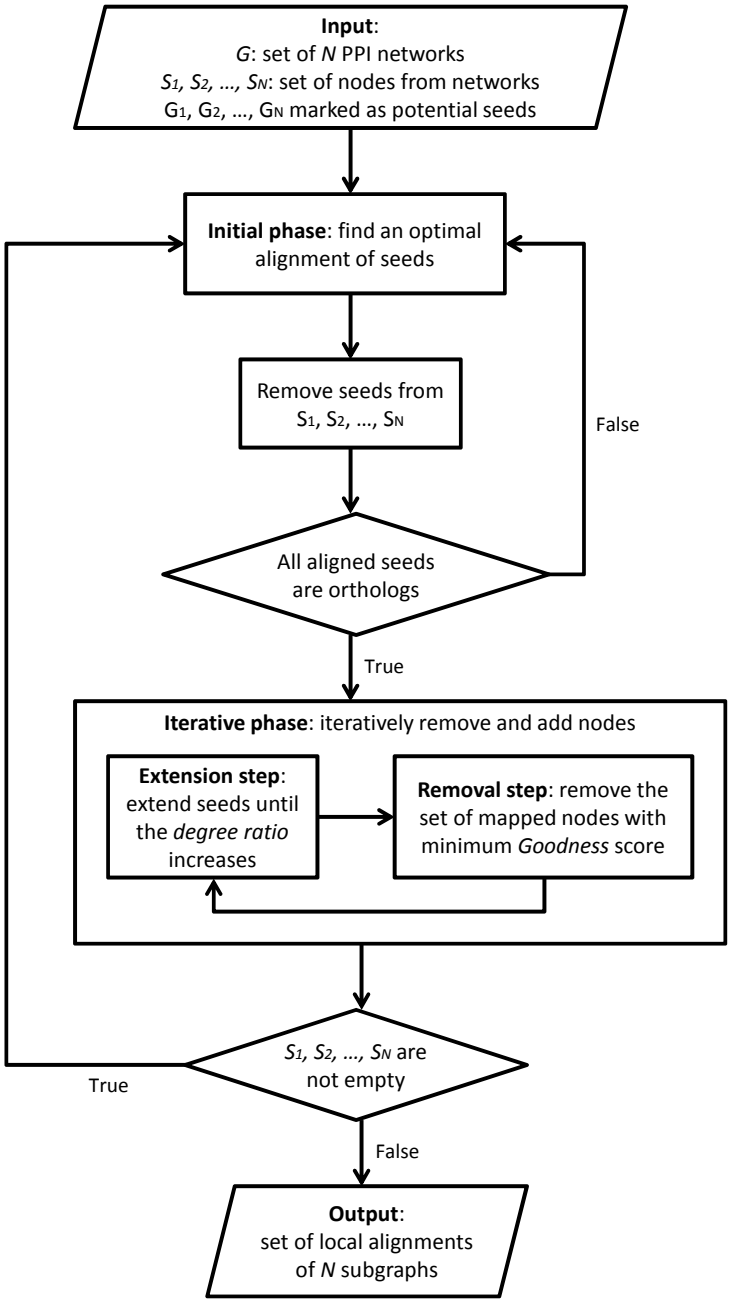


Figure 4.1: General description of GASOLINE

and Adj_2. In Fig. 4.2 (c) the new alignment of subgraphs yielded after a single extension step is shown in green.

Each extension step is performed through an iterative sampling similar to the one described above, where a state of the Markov chain represents an alignment of N nodes, one for each set Adj_i. Again, the initial state of the chain is randomly selected. Then, a series of transitions from a state to another one is made, by replacing a randomly chosen protein of the current alignment with a node of the same network in the corresponding adjacent set.

The transition probabilities are computed by considering sequence similarity in connection with

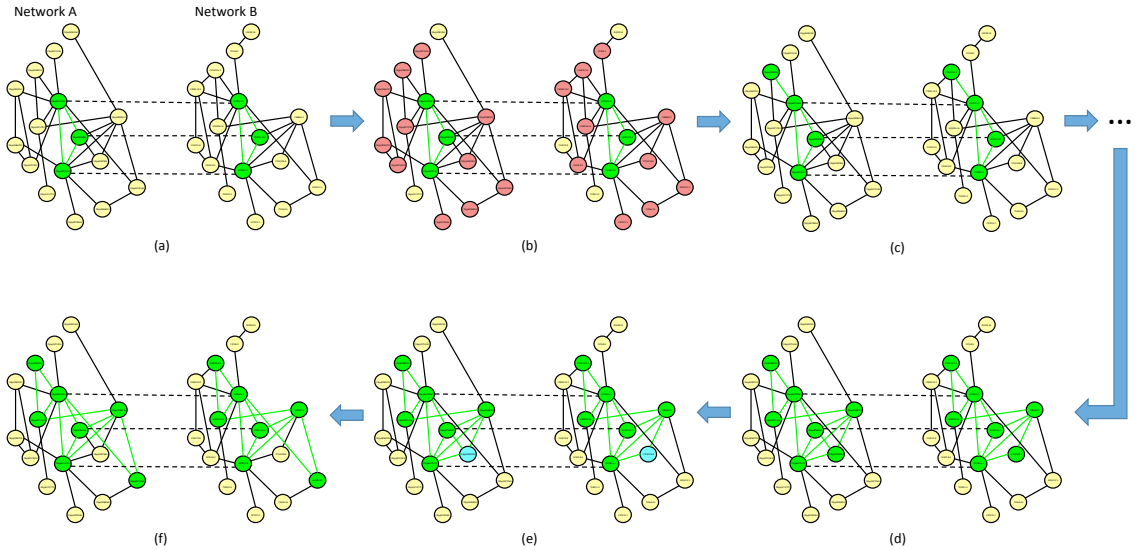


Figure 4.2: A toy example of extension and removal phases of GASOLINE algorithm in a pairwise alignment instance.

neighborhood similarity.

Let $A^i = \{A^i_1, A^i_2 \cdots, A^i_N\}$ be the alignment of proteins at the i-th iteration of Gibbs sampling. Suppose we remove protein A^i_k from A^i and let p be a protein of the same network candidated to replace it.

The Similarity Score of p takes into account both the orthology relation and the topological similarity between p and proteins in $A^i \setminus \{A^i_k\}$.

The orthology score of p which takes into account the sequence similarity, is defined as in the bootstrap phase: $SIM_O(p) = \prod_{j=1,\, i\neq k}^{N} S(p, A^i_j)$.

Concerning the topology similarity, we build a vector V, called *topology vector*, storing the weights of the edges linking p to the nodes of SG_k. If there is no link between two proteins the weight is set to 0. Likewise, we build a topology vector for all the proteins in $A^i \setminus \{A^i_k\}$. Given two proteins a and b, and their topology vectors V_a and V_b, the topology similarity score is the scalar product of the two vectors: $TOP(a, b) = \langle V_a, V_b \rangle$. The topology similarity score of p is then defined as: $SIM_T(p) = \prod_{j=1, j\neq k}^{N} TOP(p, A^i_j)$.

The overall similarity Score of p can the be computed as: $SIM(p) = SIM_O(p) \times SIM_T(p)$. By normalizing in the range $[0, 1]$ we obtain the transition probability of p:

$$P(p|A^i_1, \ldots, A^i_{k-1}, A^i_{k+1}, \ldots, A^i_N) = \frac{SIM(p)}{\sum_{n \in Adj_k} SIM(n)} \tag{4.3}$$

The alignment score is calculated in terms of orthology and topology similarity by the sum-of-pair of the pairwise alignments:

$$\textsc{ScoreExtend}(A) = \left[\sum_{j=1}^{N}\sum_{k=1}^{N} S\left(A^i_j, A^i_k\right)\right] \times \left[\sum_{j=1}^{N}\sum_{k=1}^{N} TOP\left(A^i_j, A^i_k\right)\right]. \tag{4.4}$$

At the end of Gibbs sampling, the alignment with highest sum-of-pair score is selected for the extension of subgraphs in SG. The extension of subgraphs mainly depends on a degree ratio of the alignment which evaluates the local density and the modularity of aligned subgraphs with respect to their neighborhood.

Given an aligned subgraph SG_i, the degree ratio of SG_i is the number of edges linking nodes within SG_i over the sum of the degrees of nodes in SG_i. Then, the degree ratio of a subgraph alignment SG is the average degree ratio of aligned subgraphs in SG.

The extension process is repeated until the following two properties hold: (i) all mapped proteins are in orthology relation (w.r.t. BLAST E-values or Bit Scores); (ii) the degree ratio of SG strictly increases.

The removal step

In the removal step, we discard from the current alignment a set of mapped proteins which give a minimal contribution to alignment quality. Such a step tries to refine the topology of the aligned subgraphs and therefore does not take into account the sequence similarity. The reason behind this choice is that during the extension steps the subgraph topology conservation intrinsically decreases since no backtracking is performed. This step deals with such an issue by making use of a measure called Goodness score.

Let SG be the current subgraph alignment and let W be the number of proteins in each aligned subgraph. We can represent SG as a $N \times W$ matrix, where each column contains mapped proteins across all the networks. The goal of this step is to delete the column minimizing Goodness. We define the Goodness of a generic protein $SG[i, j]$ of alignment SG as the ratio between the internal degree of $SG[i, j]$, i.e. the number of links connecting $SG[i, j]$ to the remaining nodes in the aligned subgraph, and its node degree. The Goodness of column j is the product of the Goodness scores of all its proteins:

$$\textsc{Goodness}(j) = \prod_{i=1}^{N} \textsc{Goodness}(SG[i, j]) \tag{4.5}$$

Each removal step deletes from the current alignment the nodes corresponding to the column with the minimum Goodness score. However, such proteins could be added again to the alignment, in some future extension steps.

In Fig. 4.2d-f) we report a toy example of the removal step. Fig. 4.2d) consists of the current local alingment identified through the iterative step; in Fig. 4.2e) the removal step identifies two cyan nodes as those giving a minimal contribution to the alignment score; in 4.2f) the nodes are replaced with to different nodes increasing the alignment score.

Notice that, the topology similarity score between proteins in the Gibbs sampling algorithm of the extension process is defined in order to reward structural conservation, edge weights and density of the aligning subgraphs. So, as long as the extension process continues, the degree ratio increases. However, it tends to reach local maxima, so the goal of the refinement phase is to try to shift from these local maxima, in order to reach a better approximation of the global maximum.

The final alignments ranking

Once the algorithm completes the extraction of conserved subgraphs, GASOLINE ranks all the alignments through a score called *Index of Structural Conservation* (ISC) which measures its quality in terms of topology and sequence similarity.

Let SG be the current subgraph alignment and W the number of nodes in each aligned subnetwork. SG can be represented as a matrix with N rows and W columns, where the i-th row stores proteins of the aligned subgraph SG_i. The structural similarity score between two aligned subgraphs, P and Q (i.e. two rows of the above matrix), measures the similarity between the topology vectors of the corresponding proteins in the current mapping.

Let x and y be two nodes and V_x and V_y their topology vectors. $\textsc{cInteraction}(x, y)$ denotes the percentage of entries in V_x and V_y that are either both null or both different from zero (consisting of conserved links in both species):

$$\textsc{cInteraction}(x, y) = \frac{|\{1 \leq i \leq W : (V_x[i] \neq 0 \wedge V_y[i] \neq 0) \vee (V_x[i] = 0 \wedge V_y[i] = 0)\}|}{W - 1} \tag{4.6}$$

The pairwise structural similarity score PairSim between P and Q is given by:

$$\textsc{PairSim}(P, Q) = \sum_{i=1}^{W} \textsc{cInteraction}(P[i], Q[i]) \tag{4.7}$$

where $P[i]$ and $Q[i]$ are the matched nodes in P and Q respectively.

The structural similarity score of alignment A, StructSim(A), can be defined as the sum-of-pair of all pairwise structural similarity scores:

$$\textsc{StructSim}(A) = \sum_{i=1}^{N}\sum_{j=1}^{N} \textsc{PairSim}\,(A_i, A_j) \tag{4.8}$$

According to this definition the maximum StructSim value is $N \times W$, achieved by N perfectly aligned cliques. Finally, the ISC of an alignment A can be defined as the normalization of StructSim(A) in the $[0, 1]$ interval:

$$ISC(A) = \frac{\textsc{StructSim}(A)}{N \times W} \tag{4.9}$$

The postprocessing

The final set of local alignments returned by GASOLINE is post-processed to filter out highly overlapping complexes. Alignments are sorted according to their size and ISC score.

Let $SG^j = \{SG^j_1, SG^j_2, ..., SG^j_N\}$ the local alignment of rank i in the sorted list. For each subnetwork SG^j_k of the alignment SG^j, $Perc(SG^j_k)$ denotes the percentage of proteins in SG^i_k observed in the previous $i-1$ alignments. Let $Perc(SG^j)$ the average value of $Perc(SG^j_k)$ across all the networks. If $Perc(SG^j)$ is above a given threshold (between 0 and 1) the alignment is discarded. The threshold is arbitrarily chosen: a conservative value (e.g. 0.5) gives a good tradeoff between accuracy of and interpretation of final results.

4.1.2 Computational complexity

In order to analyze the computational complexity of GASOLINE, we assume for simplicity that the size of the complexes returned at the end of each execution of GASOLINE is W. We also suppose to have N input networks with the same number of nodes, n, and m edges.

We first define the following variables:

- γ_s: the number of Gibbs sampling iterations of in the bootstrap phase;
- γ_e: the number of Gibbs sampling iterations in each extension step of the *iterative phase*;
- k: the average degree of a node ($k = \frac{m}{n}$);
- γ_i: the number of iterations of the iterative phase;
- γ_x: the number of executions of GASOLINE.

The time complexity will be expressed as a function of n and W. Through the analysis, we will assume that the generation of random numbers and the computation of the orthology score between two proteins are done in constant time $O(1)$.

First, let's analyze the bootstrap initial phase, whose goal is to find an optimal alignment of protein seeds. The generation of the initial alignment requires $O(N)$ time. The computation of transition probabilities for all the proteins of the selected network at each iteration of Gibbs sampling costs $O(nN)$, while the computation of the alignment score requires $O(N)$.

Therefore, the time complexity of the initial phase is:

$$T_{boot}(n) = O(N) + \gamma_s \times O(nN) = O(\gamma_s nN) \tag{4.10}$$

Since, in practice, $N << n$ we can write:

$$T_{boot}(n) = O(\gamma_s n) \tag{4.11}$$

Now, let's consider the iterative phase, which removes and adds nodes to the current local alignment iteratively. Suppose, for simplicity, that the alignment grows up in the following way: at the beginning, the size of aligned complexes increases from 1 up to W, then in the following extension and removal steps it switches from W to W-1 and viceversa. The last assumption fits quite well the behavior of GASOLINE in the context of real biological networks, since our algorithm yields an alignment of complexes of a certain size and then tries to adjust it by replacing bad parts according to a goodness score.

The extension step can be divided into three phases:

1. The computation of seeds' adjacent nodes;
2. The execution of Gibbs sampling;
3. The extension of seeds.

Let's suppose that networks are represented through adjacency lists. Under this assumption, the adjacent of a node can be found in $O(k)$ time. As regards Gibbs sampling, the generation of the initial alignment costs $O(N)$.

The computation of the transition probabilities and the alignment score depends on the size of seeds. Let L the current size of the aligned complexes. The transition probability of a protein is computed as the product of two components: orthology similarity score and topology similarity score. Computing the orthology score for a protein of the selected network at each iteration of Gibbs sampling costs $O(N)$ as in the bootstrap phase. In order to compute topology scores efficiently, topology vectors are built before starting Gibbs sampling, for all seed's adjacent nodes of all aligning networks. The construction of topology vector of a single protein can be done in $O(L)$ time, assuming that adjacent lists are implemented by using hash tables with buckets, thus providing constant-time access (in average) to an element of the list. So, the overall cost of building topology vectors is $N \times O(kL^2)$, supposing that the total number of a seed's adjacent nodes is $O(kL)$. Under this assumptions, the orthology score of a protein can be computed in $O(NL)$ time and the transition probability for all the proteins of the selected network requires $O(nNL)$ time. Finally, the computation of the alignment score requires $O(NL)$ time.

Summing up, the overall cost of the extension step we obtain:

$$T_{ext}(n) = N \times O(kL^2) + O(N) + \gamma_e \times O(nNL) \tag{4.12}$$

Assuming $N << n$ and $k << n$, we can rewrite the equation as:

$$T_{ext}(n) = O(L^2 + \gamma_e nL) \tag{4.13}$$

Since in the worst case $L = O(n)$ we can deduce that:

$$T_{ext}(n) = O(\gamma_x nL) \tag{4.14}$$

The removal step simply consists in computing the minimum value of a function (*Goodness* score) overall the L sets of aligned proteins in the current alignment. For each set, the *Goodness* score can be evaluated in $O(NL)$, which is the time required to compute the internal degree of all the proteins within the set. So, the cost of removal step is:

$$T_{rem}(n) = O(NL^2) = O(L^2) \tag{4.15}$$

Assuming that the extension step is performed $W - 1$ times at the beginning and $\gamma_i - 1$ times later on and the removal step is executed γ_i times, the overall cost of the iterative phase is:

$$T_{iter}(n) = \sum_{i=1}^{W-1} O(\gamma_e nL) + (\gamma_i - 1) \times O(\gamma_e n(W-1)) + (\gamma_i - 1) \times O((W-1)^2) \tag{4.16}$$

We can assume, without loss of generality, that $W = O(\gamma_i)$ and $W << n$, so:

$$T_{iter}(n) = O(\gamma_i \gamma_e nW) \tag{4.17}$$

Finally, all preprocessing steps can be performed in linear time, by considering the degree and the number of orthologous proteins for the proteins of all networks. Post-processing phase consists in filtering highly overlapping complexes and can be done in constant time. By combining equations 4.11 and 4.17 and considering preprocessing operations, the overall cost of γ_e executions of GASOLINE is:

$$T(n) = O(n) + \gamma_x \times [T_{boot}(n) + T_{iter}(n)] = O(n) + \gamma_x \times [O(\gamma_s n) + O(\gamma_i \gamma_e n W)] \quad (4.18)$$

From the results of the analysis, it follows that the running time of GASOLINE is polynomial in n. In fact, γ_x is at most equal to n since at each execution of the algorithm different protein seeds are considered. Moreover, in all applications, γ_s and γ_e are in the range 200-400 and can be considered constant. Therefore, the final complexity is $O(n^2 W)$.

We can distinguish three cases:

1. If networks are very similar, then the average size W of complexes found in each execution is high, so $W = O(n)$ and the algorithm requires $O(n^3)$ time. This is the worst case;
2. If networks are very distantly related, then $W = O(\sqrt{n})$ and the running time is $O(n^{2.5})$ this is the average case.
3. If W is independent to the size of networks we can suppose its size constant $W = O(1)$. Therefore, the running time will be $O(n^2)$ that is, the best case of our algorithm.

4.1.3 GASOLINE app for Cytoscape

Cytoscape [140] is an open source platform for visualizing and analyzing molecular interaction networks and integrating data with gene expression profiles and other kind of knowledge. It is the most popular framework for the analysis of biological networks in the Bioinformatics community. Cytoscape is multiplatform since it is written in Java, and its functionalities can be easily extended through Java apps.

Most network alignment algorithms have been provided with their implementations, however none of them is fully integrated within Cytoscape. For this reason, we implemented a Cytoscape app [103] based on GASOLINE algorithm [105].

The app enables the 2D visualization of local alignments in a user-friendly way, without requiring any post-processing operations by the user. Moreover, aligned proteins can be associated with GO annotations for further functional analysis.

General implementation

GASOLINE app has been written in Java version 7 and designed following a classic Model-View-Controller (MVC) model. The Model part is represented by the classes implementing the algorithm and the auxiliary data structures. The View part is composed by two Java Panels; one for setting all the input and output parameters, and one for listing local alignments and handling their visualization.

The Controller part ensures the communication between the Model and the View and is implemented by different Cytoscape Task classes, one for each process performed by GASOLINE (i.e. checking file format, computing alignments, importing networks, protein description and GO annotations, building the alignment graphs). Each Task class properly notifies the corresponding view class when a task has been completed.

Input networks are imported as text files and then internally represented in two different ways to optimize the performance of our algorithm. We used *CyNetwork* and *CyNetworkView* objects for network alignment visualization and custom classes for computing alignments. For all the imported networks, the corresponding Cytoscape view is initially disabled to reduce the memory consumption.

The main component of GASOLINE is represented by a tabbed panel named "GASOLINE" located in the Control Panel of Cytoscape (Fig. 4.3, panel A). Through the interface the users can provide the following information:

- "Similarity information", to upload orthology similarity scores between proteins of different species;
- "Networks", for selecting two or more networks to align;
- "Parameters setting", to modify the default GASOLINE parameters;
- "Optional parameters setting", for setting other advanced input parameters;
- "Ontologies", to upload GO terms linked to the proteins of the aligned networks;
- "Output", to specify the folder where the final alignments will be saved.

The button labeled ?, when present, explains the meaning of a specific function or parameter of GASOLINE, whenever the mouse arrow hovers it.

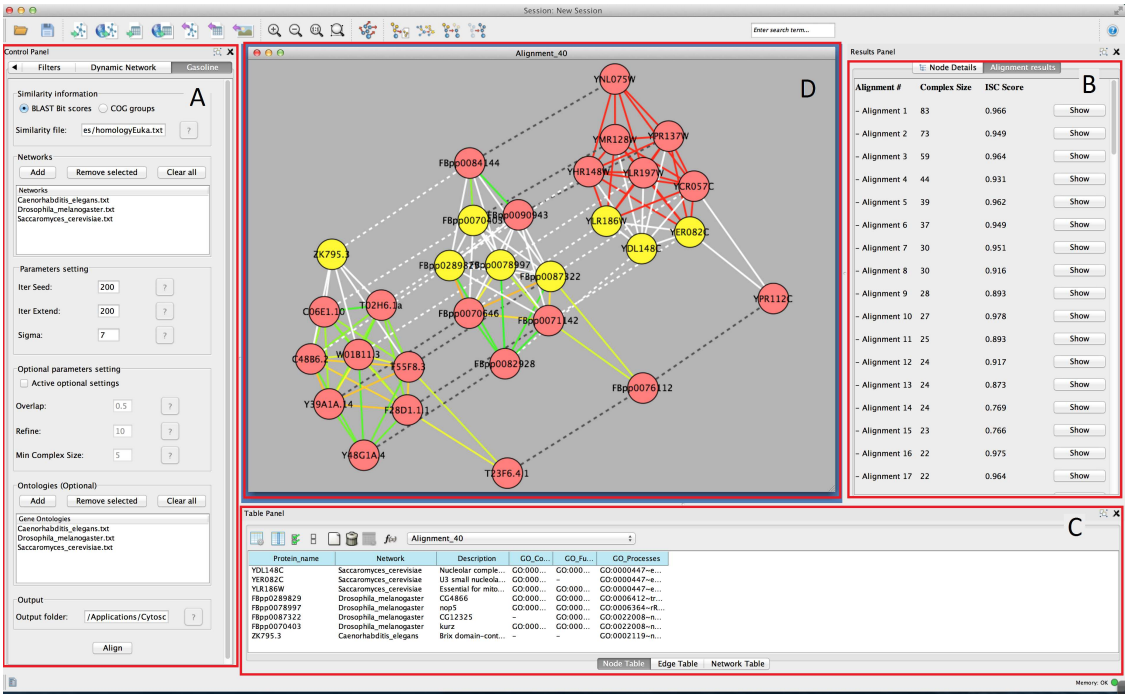


Figure 4.3: GASOLINE Cytoscape panels: A) GASOLINE parameters; B) Alignment results; C) Description of selected proteins and associated GO terms; D) Alignment visualization: intra-edges are represented with solid lines and coloured according to their weight (green for low values, yellow for medium values and red for high values); inter-edges are drawn with dashed lines.

Loading input data

Before running GASOLINE, the user needs to upload input data, including:

a) Two or more networks to be aligned;

b) A file of orthology BLAST bit scores between proteins of different species;

c) A set of GO terms linked to the proteins of each network.

The GO terms file is not mandatory and can be omitted.

Networks are given as a list of weighted edges and can be uploaded from the "Networks" panel. There is no theoretical limitation on the size and the number of uploadable networks.

Orthology data can be uploaded through the "Similarity information" panel. They can be supplied in two different formats: "BLAST Bit scores" or "COG groups".

The "BLAST Bit scores" format is a text file where each row has a couple of proteins of different species followed by their corresponding BLAST bit score.

Files in the "COG groups" associate a list of orthology groups to the proteins of aligning networks. Groups include KOG (euKaryotic Orthologous Groups), NOG (Non-supervised Orthologous Group) and COG (Cluster of Orthologous Groups) terms. When aligning many networks, the "COG groups" format can be more suitable, since the number of possible pairwise bit scores grows exponentially with the number of aligning networks.

GO categories can be optionally uploaded from "Ontologies" panel. They are provided as text files, where a list of GO cellular components, processes and functions is associated to each protein. Note that GO terms are not used for computing the alignments and can only facilitate the understanding of the alignments returned by GASOLINE.

Whenever the GO categories are provided as input, the list of GO terms for a specific protein is added as node attribute in Cytoscape. GO terms are accessible from the "Node Browser" tabbed panel once GASOLINE ends the computation and the local alignments are ready to be visualized.

Setting the parameters

The main input GASOLINE parameters are specified in the "Parameters setting" panel. These include:

- "Iter Seed": the number of iterations of Gibbs sampling in the bootstrap phase;
- "Iter Extend": the number of iterations of Gibbs sampling in each extension step of the iterative phase;
- "Sigma": minimum network degree of nodes that can be selected as seeds in the bootstrap phase.

Values for "Iter Seed" and "Iter Extend" depend on the number of aligning networks: the more networks we have, the higher these values should be. However, based on the experiments performed on real PPI networks and reported in [105], we empirically established that 200 iterations of Gibbs sampling in both phases are enough to produce reliable results for up to 25 networks.

The choice of "Sigma" implies a trade-off between speed and accuracy of GASOLINE: the higher the σ, the faster is GASOLINE but the lower is accuracy. If networks are very sparse (like most of the existing PPI networks), low values of sigma (1 or 2) are recommended.

The "Optional parameters setting" panel contains three more input parameters:

- "Overlap": a value between 0 and 1, denoting the maximum allowed fraction of common nodes between two alignments, in order to be considered distinct. If two alignments have many nodes in common, the one with the least number of nodes is discarded from the final set;
- "Refine": the number of iterations of GASOLINE in the refinement phase;
- "Min Complex Size": the minimum size of conserved complexes in the final set of local alignments.

These parameters can be modified by checking the box "Active optional settings", otherwise the default values will be used.

A high value of the "Refine" parameter can be used to increase the accuracy of the local alignments, but the algorithm will be more time consuming. In our tests [105], we experienced that a value of 10 guarantees the best trade-off between speed and accuracy of GASOLINE. For the "Overlap" and "Min Complex Size" parameters, we suggest 0.5 and 5 as default values, respectively.

Finally, the user can specify an output folder for the final alignments, by clicking on the text field next to the "Output folder" label. Each local alignment will be stored in a separate text

file into the specified folder, containing the list of aligned sub-graphs and the one-to-one mapping between aligned nodes.

Once all the required input files are provided and all the parameters are set up, GASOLINE can be executed by clicking on the "Align" button. Then, a progress bar will appear.

Visualizing local alignments

When GASOLINE ends, a table containing all the computed local alignments is shown on the right side of the "Results panel" of Cytoscape (Fig. 4.3, panel B). The table reports, for each alignment, the size of the aligned complexes and the ISC (Index of Structural Conservation) score. ISC index measures the average degree of structural conservation of an alignment and can be used to evaluate its quality.

Each row of the table contains a "Show" button, for the visualization of the corresponding alignment graph on the left side of the "Results Panel" of Cytoscape (Fig. 4.3, panel D).

In the alignment graph, each node is labeled with the ID of the corresponding protein. If the GO annotations have been provided, the user can select one or more nodes and view the description of the proteins and their corresponding GO terms (Fig. 4.3, panel C).

Two kinds of edges are shown (Fig. 4.3, panel D):

- Intra-edges, linking proteins of the same network, which are represented with solid colored lines;
- Inter-edges, linking proteins of different networks in the local alignment, which are drawn with dashed lines.

Colors of intra-edges depend on the probability p of the corresponding protein-protein interaction: for low values of p colors range from green to yellow, for high values of p colors range from yellow to red. Weights are automatically associated to edges as attributes, therefore the user can select an edge and retrieve its weight from the "Edge Attribute Browser" (Fig. 4.3, panel C).

Layout visualization of the alignment graph follows the "Kamada-Kawai" force-directed graph drawing algorithm [72]. The layout algorithm assign forces among the set of edges and the set of nodes, based on their relative positions, and then use these forces either to simulate the motion of the edges and nodes or to minimize their energy.

4.2 SPECTRA

SPECTRA (SPECific Tissue/Tumor Related PPI networks Analyzer) is a framework for retrieving and analyzing protein-protein interaction data specific for a given set of normal or cancer tissues. The underlying graph model in SPECTRA is the Tissues-Specific PPI network (or TS-PPI network), in which the genes of corresponding interacting proteins are both expressed in one or more tissues.

SPECTRA integrates tissue and tumor specific gene expression data from various authoritative online repositories with high-quality PPI data. Additionally, it provides a web interface for constructing, visualizing and comparing TS-PPI networks, with the aim of identifying differential interaction/expression patterns in TS-PPI networks (i.e. distinct tissues, or normal and pathological states of the same tissue). The TS-PPI networks together with the results of differential analysis can be easily visualized by using Cytoscape facilities [140] and downloaded as text files for further investigations.

4.2.1 SPECTRA database

SPECTRA combines protein-protein interactions in human with gene expressions, by integrating 13 authoritative resources. The final integrated SPECTRA database contains 16,435 protein coding genes and 175,841 gene interactions (GIs), 1,350,637 tissue-specific gene expression data entries

covering 107 normal tissues, and 2,171,808 tumor-specific expression data entries covering 160 different tumors.

Interaction datasets

Human protein interaction data were taken from BioGRID [154], DIP [171], a recent work by Havugimana et al. [60], HPRD [123], IntAct [114] and MINT [90].

Table 4.1 describes the features of the PPI networks integrated in SPECTRA. Networks taken from [60], IntAct and MINT are weighted with edge weights ranging in [0,1], while the other PPI networks are unweighted. Proteins of the considered PPI networks, including splicing isoforms, were first mapped to the corresponding gene. Next, a global GI network was built, by collecting all interactions reported in at least one dataset. We assigned to each edge a pair consisting of the average value of weights across the datasets that report that interaction and the percentage of datasets giving the interaction (dataset coverage). Average edge weights range from 0.131 to 1.

Table 4.1: Features of PPI networks integrated in SPECTRA

NETWORK	NODES	EDGES	TYPE
BioGRID	15,290	135,677	Unweighted
DIP	2,338	3,427	Unweighted
Havugimana et al. [60]	3,003	13,989	Weighted
HPRD	9,506	37,054	Unweighted
IntAct	11,637	63,030	Weighted
MINT	6,551	18,478	Weighted

Fig. 4.4 depicts a Venn diagram of common gene interactions between PPI datasets. Interaction databases generally show low overlap, with only 25 interactions shared by all datasets and only 7,783 interactions in common between MINT, BioGRID, IntAct and HPRD, which are the biggest ones.

The final integrated network has 16,435 nodes, 175,841 edges and 17 connected components, with a high average diameter (9) and low clustering coefficient (0.289). The average degree is 21.398 and the degree distribution follows a power law (Fig. 4.5).

Expression datasets

Gene expression data for various tissues and tumors were downloaded from ArrayExpress [133], GEO [10], ProteinAtlas [161] and TCGA (http://cancergenome.nih.gov). Table 4.2 lists the gene expression datasets integrated within SPECTRA, the platform used to detect the expressions and the number of covered tissues and tumors.

Table 4.2: Features of expression datasets integrated in SPECTRA

DATASET	PLATFORM	TISSUES	TUMORS
E-MTAB-62 [94]	GPL96	46	110
GDS181 [155]	GPL91	29	6
GDS596 [156]	GPL96	57	5
GDS1096 [49]	GPL96	36	0
GDS3113 [36]	GPL2986	32	0
ProteinAtlas	GPL11154	28	33
TCGA	Agilent G4502A-07-3	0	27

Fig. 4.6 depicts a Venn diagram of common tissues and tumors across expression datasets. While tissue names are generally shared, tumor names are much differentiated, resulting in a poor

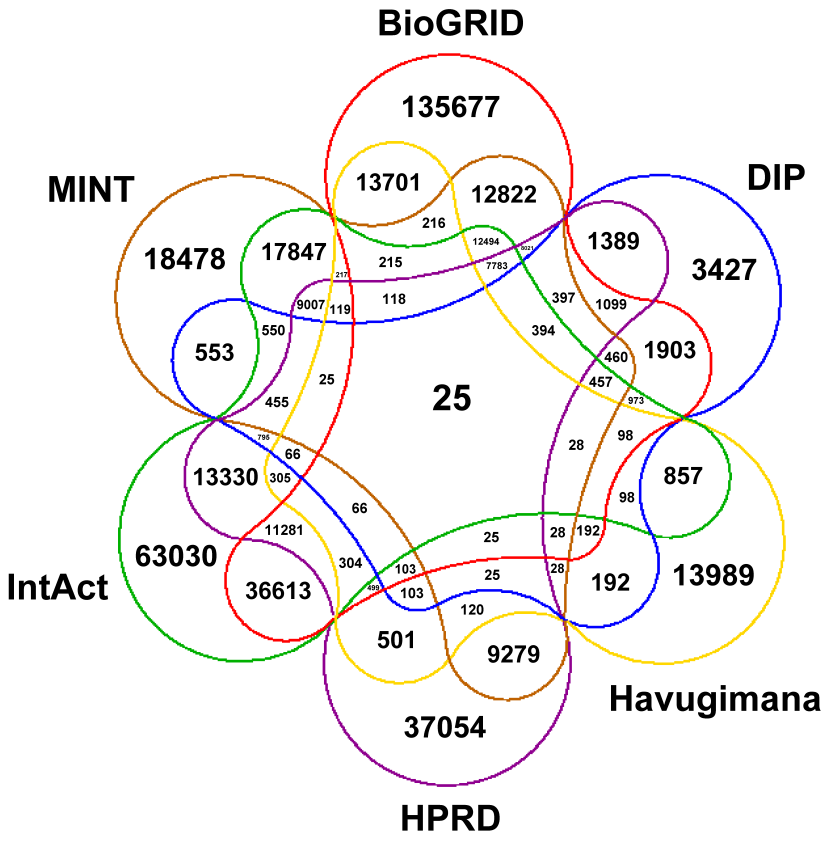


Figure 4.4: Venn diagram showing the number of common interactions across PPI datasets in SPECTRA.

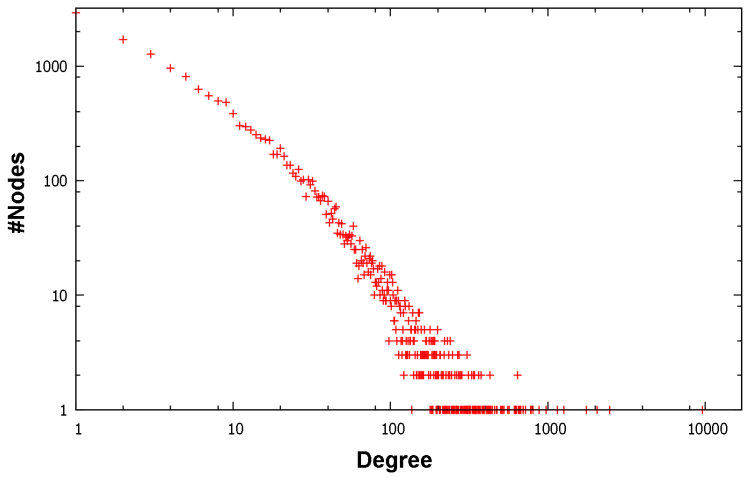


Figure 4.5: Log-log plot of degree distribution of the final integrated gene interaction network in SPECTRA.

overlap between datasets. In particular, TCGA contains data for very specific tumors and partially overlap only with E-MTAB-62 dataset, which is the richest one. Note that the numbers reported in Fig. 4.6b only refer to specific tumors and not to tumor classes. So, for instance, "breast carcinoma" and "breast adenocarcinoma" are considered distinct tumors, even though they belong

to the same class of tumors, "breast cancer".

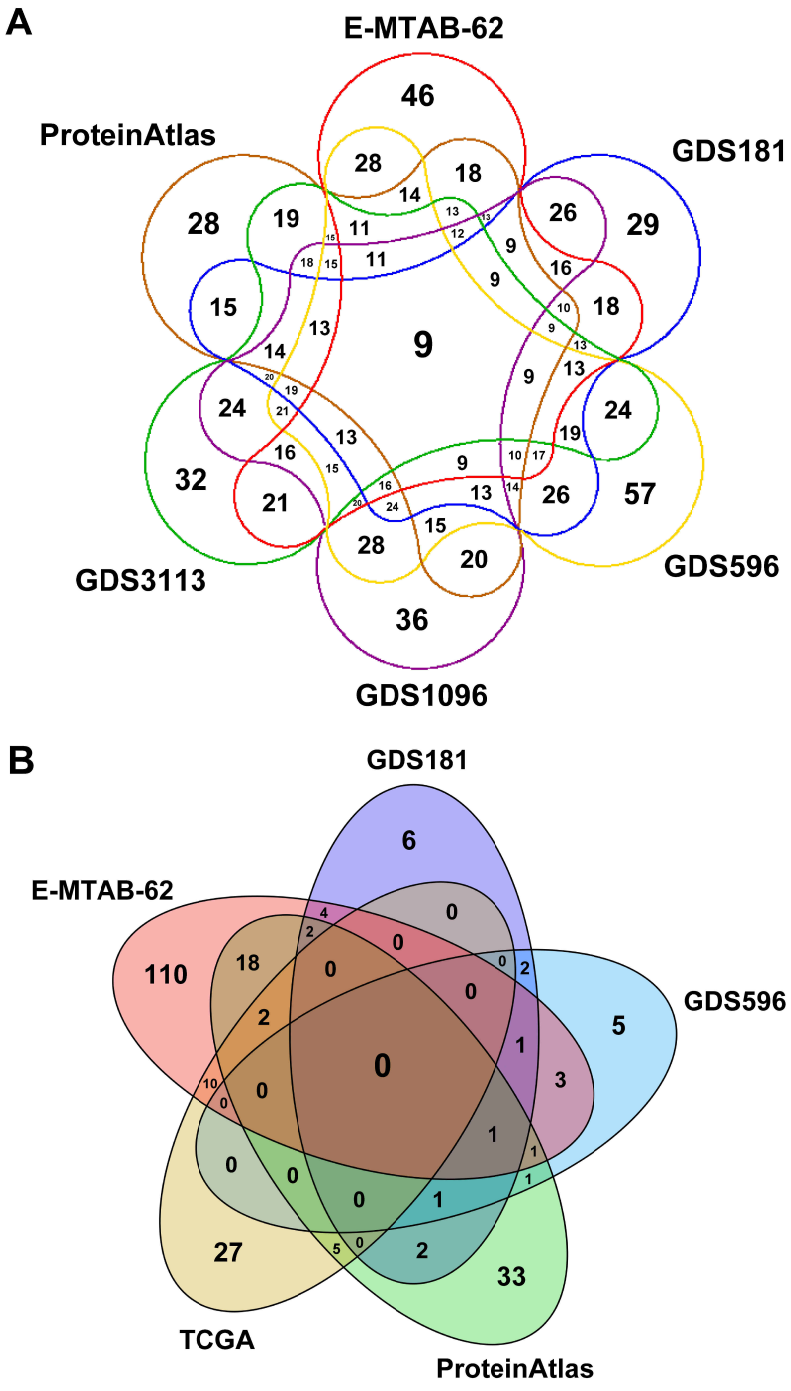


Figure 4.6: Venn diagrams depicting: A) the number of common tissues and B) the number of common cancer types across expression datasets in SPECTRA. The zero overlap among the cancer datasets is due to the fact that cancers are considered according to their specific names and not to their class (breast cancer, prostate cancer, etc.).

As regards the integration of expression data, we followed the work of [57], where authors show that there is positive correlation between normalized Affymetrix and RNA-Seq data. We performed RMA normalization [97] for datasets based on Affymetrix platforms (GDS181, GDS596

and GDS1096), using the corresponding R Bioconductor package [51]. For GDS3113, ProteinAtlas and TCGA we first computed the $\log_2$ of the number of fragments and then we normalized values using the quantile normalization method of [16]. For GDS181, GDS596 and GDS1096 datasets, normalized values were computed from raw data. Then, probes which were present in a particular microarray dataset were mapped to the corresponding genes. The expression of a gene for a specific tissue was computed as the average expression value of probes mapping to that gene in the tissue. For GDS3113, ProteinAtlas and TCGA, instead, we directly normalized gene expression values for different tissues. We did not normalize EMTAB-62 data, since its source values were already normalized with RMA.

Finally we assigned to each pair gene-tissue a unique positive expression score, given by the average normalized expression value of the gene in that tissue, according to the different datasets. Expression scores in SPECTRA range from 3.566 to 17.366 for tissues and from 0.01 to 17.343 for tumors.

Database schema

SPECTRA database is structured as a MySQL relational database with six tables: *Genes*, *Tissues*, *Tumors*, *Interactions*, *Expr_normal* and *Expr_tumor*.

The *Genes* table contains the list of all expressed and interacting genes. Each entry is identified by the gene symbol and contains associated data, including a description string, aliases and cross references to Entrez Gene (if available).

The *Tissues* and *Tumors* tables have the same structure. Tissues and tumors are associated to different classes, depending on the organism part they refer to. Each entry is identified by a unique number and contains a description and the corresponding class. SPECTRA contains 26 distinct classes of tissues and 32 distinct classes of tumors.

The *Interactions* table lists all the PPIs integrated in SPECTRA. Interactions are identified by a couple of gene symbols and the edge weight for each integrated dataset (when available) is stored, together with the average interaction weight across dataset reporting that interaction and the dataset coverage.

Expr_normal and *Expr_tumor* contain all the gene expressions in normal and cancer tissues. The unique identifier of *Expr_normal* is a couple gene-tissue, while entries in *Expr_tumor* are uniquely identified by the couple gene-tumor. In both tables, the normalized expression value for each integrated dataset (where available) and the average expression score are included as associated data.

4.2.2 Adapted GASOLINE for differential local alignment of TS-PPI networks

TS-PPI networks can be compared in SPECTRA for identifying patterns of differential gene expressions between multiple TS-PPI networks. The goal is to find conserved sub-regions in the TS-PPI networks which maximize the difference of expression values of aligned genes. These regions can reveal important differences in the way a process is carried out in healty and tumor tissues and drive the design of new drugs for targeting specific genes involved in a particular disease.

The problem is related to that of finding maximal-scoring connected subgraphs, which is NP-hard, even in a common simpler setting where the aligning TS-PPI networks have the same set of nodes and edges (e.g. TS-PPI networks built starting from different expression data and the same interaction datasets) [65].

In the case of two TS-PPI networks with the same set of nodes and edges (representing for instance case and control expression data) heuristic [65, 150, 130, 19, 58] and exact [38] solutions have been proposed. However, as far as we are concerned, no solutions are known for the multiple case.

We propose an approximate solution to the multiple differential alignment problem, based on a modified version of GASOLINE, where gene expressions are included as node weights. For simplicity, we consider TS-PPI networks with no multiple edges between two nodes.

Let A and B two genes and $Expr(A)$ and $Expr(B)$ their expression values, with $Expr(A) \geq Expr(B)$. In order to evaluate the expression difference between A and B, we compute the *log fold change*, defined as follows:

$$\textsc{LogFold}(A, B) = \log_2 \left(\frac{Expr(A)}{Expr(B)} \right) \tag{4.19}$$

Given N TS-PPI networks and a set of aligned genes $G = \{G_1, G_2, ..., G_N\}$, one for each TS-PPI network, MaxLogFold is the maximum value of LogFold function among all pairs of genes in G:

$$\textsc{MaxLogFold}(G) = \max\{\textsc{LogFold}(G_i, G_j) \ \forall \ 1 \leq i, j \leq n \wedge i < j\} \tag{4.20}$$

We applied the following changes to original GASOLINE algorithm:

- We included the LogFold function in the Gibbs sampling procedure of bootstrap and iterative phases, by multiplying it by the topology and homology scores in the computation of node similarities;
- The number of iterations of Gibbs sampling both in the bootstrap and in the extension phase is governed by a new parameter, called *alpha*, which is a probability threshold related to N, the number of networks, according to the following formula:

$$k = \max \left\{ k' : \left(\frac{N-1}{N} \right)^{k'} > \alpha \right\} \tag{4.21}$$

  where $P = \left(\frac{N-1}{N}\right)^{k'}$ is the probability that a gene is never selected in k' consecutive iterations of Gibbs sampling. The idea is to stop Gibbs sampling when an alignment does not change for k consecutive iterations. The lower is α, the higher is k, so the more precise and slower will be the sampling procedure;
- We introduced a new threshold, MaxLogFoldThreshold, for the value of MaxLogFold function, and we used it to tune the extension process in place of the degree ratio: in particular, we extend the current alignment until the average value of MaxLogFold between the sets of aligned nodes is above such a threshold;
- In the remove phase, the set of aligning nodes with minimum value of MaxLogFold is deleted from the current local alignment;
- Given a local alignment $A = \{A_1, A_2, ..., A_w\}$, where w is the size of the alignment and $A_1, ..., A_w$ are the set of aligned genes, an average value of $\textsc{MaxLogFold}(A_i)$ is computed together with the ISC score to evaluate the quality of the alignment.

4.2.3 Utilities

The architecture of SPECTRA is composed by: (i) a *searching tool* which allows to retrieve TS-PPIs; (ii) a *comparison tool* to look for shared differential expressions patterns between genes of two or more TS-PPI networks. Results can be graphically visualized by using Cytoscape.js (http://js.cytoscape.org) or downloaded as text files.

SPECTRA searching tool: building TS-PPI networks in SPECTRA

SPECTRA builds TS-PPI networks starting from a user-defined set of genes, tissues, expression data and interaction data. Fig. 4.7 depicts the search interface of SPECTRA.

In the "Gene data" section (Fig. 4.7a), the user can look for all genes expressed in a set of tissues or restrict the search to a specific list of genes. Genes can be provided with their official names or Aliases (e.g. Ensembl Gene, Entrez Gene, Affy).

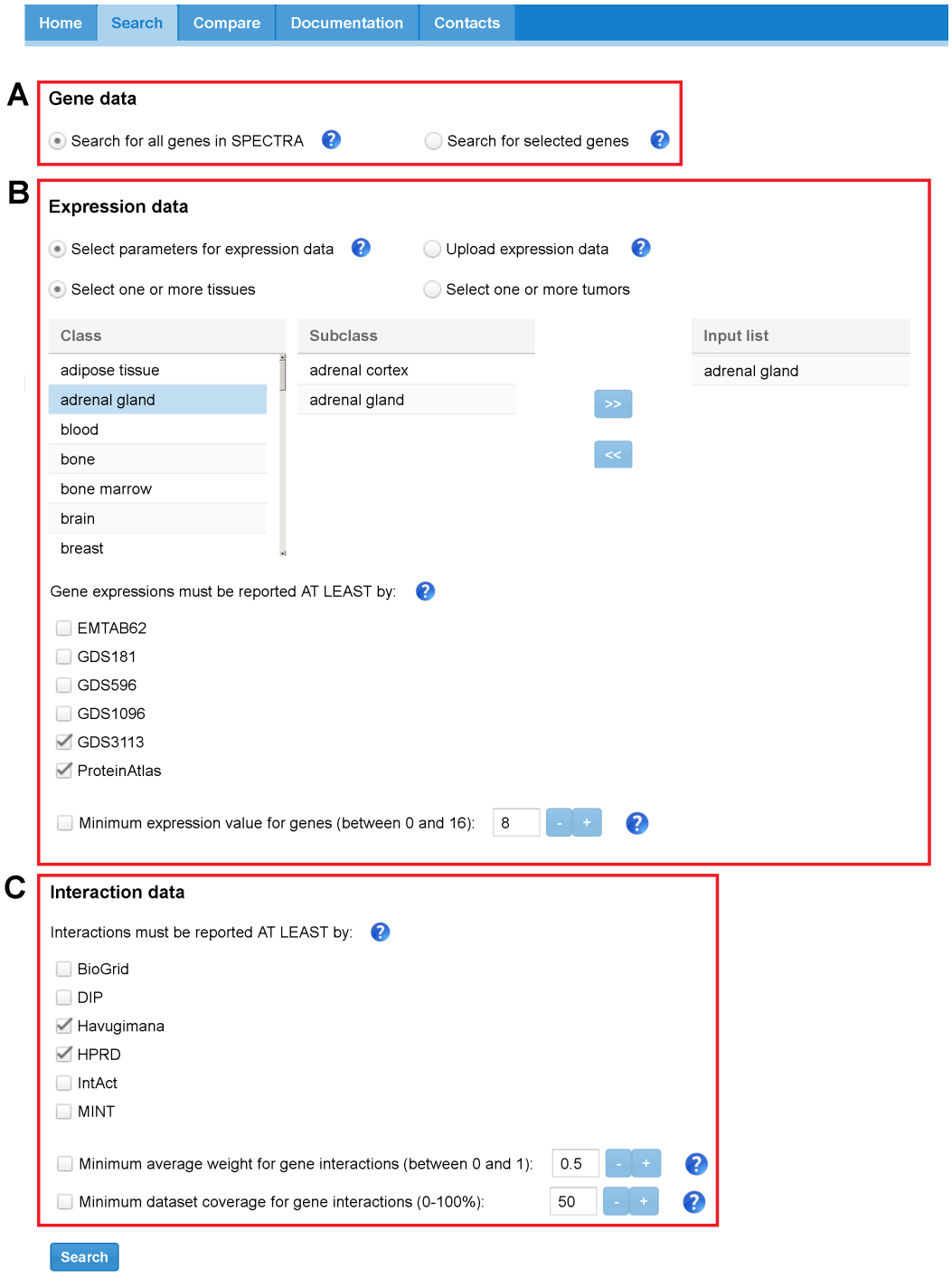


Figure 4.7: SPECTRA search tabbed panel. Red boxes highlight the three sections: a) "Gene data", b) "Expression data" and c) "Interaction data". In this case, the parameters have been set to indicate that we want to retrieve all the interactions that are present at least in Havugimana and HPRD, involving genes that are expressed in "adrenal gland" tissue according at least to GDS3113 and ProteinAtlas. In this example, we neither restrict our search to a predefined set of genes nor provide a threshold for interaction weights, dataset coverage and expression scores.

In the "Expression data" (Fig. 4.7b) section, the user limits the search to a set of tissues/tumors and to a set of expression datasets or uploads a text file with custom expression data. Note that the two options are mutually exclusive, that is, all the settings concerning datasets and tissue/tumors will be ignored if the user provides a custom text file. Available tissues and tumors in SPECTRA are listed in a table and can be easily included in the input query list with a double click in each entry. When no data are provided, all the tissues and tumors in SPECTRA are considered. Tissues and tumors are also mutually exclusive, meaning that a TS-PPI network built in SPECTRA cannot contain interactions defined on both normal and tumor tissues. However, two TS-PPI networks defined upon a specific set of tissues and tumors, respectively, can be always compared for differential analysis with the adapted GASOLINE. The user can also select one or more datasets from which the expression have to be reported. When the expression is in other datasets it will be also given. When no dataset is selected, all expression data in SPECTRA are considered. Finally, a further filter on genes can be applied by indicating a threshold for the minimum normalized value of gene expressions to be considered.

The "Interaction data" section (Fig.4.7c) contains the parameters for filtering interaction data. As above the user can select one or more datasets where protein interactions have to be reported. If no interaction dataset is selected, all PPIs in SPECTRA are considered. A threshold can be provided to select interaction weights above a given value and having a minimum dataset coverage.

When all input parameters have been specified, the user clicks on the "Search" button. At the end of the process, all the TS-PPIs found are listed in a result table (Fig. 4.8).

Figure 4.8: Result table for the query of Fig. 4.7. For each interaction in the table we report the tissue, the average expression scores of interacting genes and the total interaction weight. Expression scores, weights and dataset coverages are represented with a colored progress bar (from cyan to red). By selecting a row (in the example the interaction between NCBP2 and NCBP1), detailed data about the interaction are shown to the right. For each dataset, the corresponding interaction weight (when available) is reported (for example 0.71 for IntAct database).

For each TS-PPI we show the interacting genes, the tissues where they are expressed, the expression values of genes in that tissues, the average interaction weights and dataset coverages of corresponding proteins. Results are ordered by dataset coverage and average interaction weight. Expression values and interaction weights are depicted with colored progress bar, where colors range from cyan (low values) to red (high values).

By selecting a specific TS-PPI in the result table, additional data about the interaction and the interacting genes are shown (Figs. 4.8-4.9). A list of datasets reporting the interaction and the corresponding interaction weight is reported on the right of the result table (Fig. 4.8). Below the result table, two panels with details about the interacting genes are shown (Fig. 4.9). For each gene, description and aliases are provided, together with the lists of tissues and tumors where the gene is expressed, according to the different expression datasets, ordered by expression score.

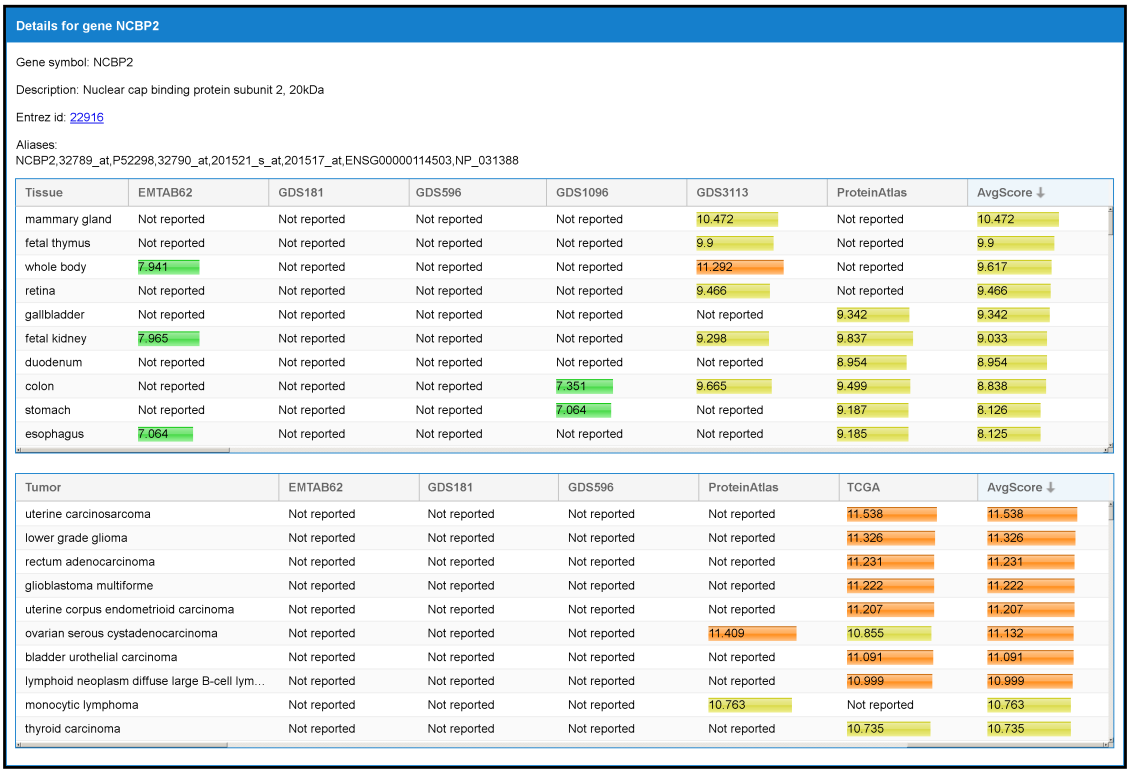


Figure 4.9: The Panel with detailed information of a gene. When an interaction is selected from the result table (Fig. 4.8), two panels with additional data, one for each interacting gene, are shown. This example refers to the detailed panel for gene NCBP2, which appears when the row table of Fig. 4.8 is selected. In the detailed panel the gene symbol, the description, the corresponding ID in Entrez Gene database (when available) and aliases (including references in other databases) are reported. Finally, two tables with the set of tissues and tumors where the gene is expressed are shown. These are shown in decreasing order with respect to the average expression scores.

SPECTRA comparison part: compare TS-PPI sub-networks

TS-PPI networks can be compared in SPECTRA for identifying patterns of differential gene expressions between multiple TS-PPI networks. The goal is to find conserved sub-regions in the TS-PPI networks which maximize the difference of expression values of aligned genes.

Fig. 4.10 shows the "Compare" tabbed panel in SPECTRA. Before running the adapted GASOLINE, the user has to upload at least two TS-PPI networks. For each network, the number of nodes and edges are reported. Networks can also be renamed by double clicking on the corresponding cell. Note that uploaded TS-PPI networks with multi-edges between nodes will be always treated as simple networks, where multi-edges are replaced by a single edge with weight equals to the sum of the weights of multi-edges and label given by the concatenation of the multi-edge labels.

Once the networks have been uploaded, the user can click on "Run GASOLINE" button to set the input parameters for the adapted GASOLINE (Fig. 4.10).

We briefly describe their meaning (default values are reported in brackets):

- "Sigma": the minimum degree of candidate nodes for the initial alignment of seeds (1);
- "Alpha": a value between 0 and 1 which regulates the number of iterations of Gibbs sampling in the bootstrap and extend phases (default 0.05);
- "Overlap threshold": a maximum average overlap threshold between local alignments, which

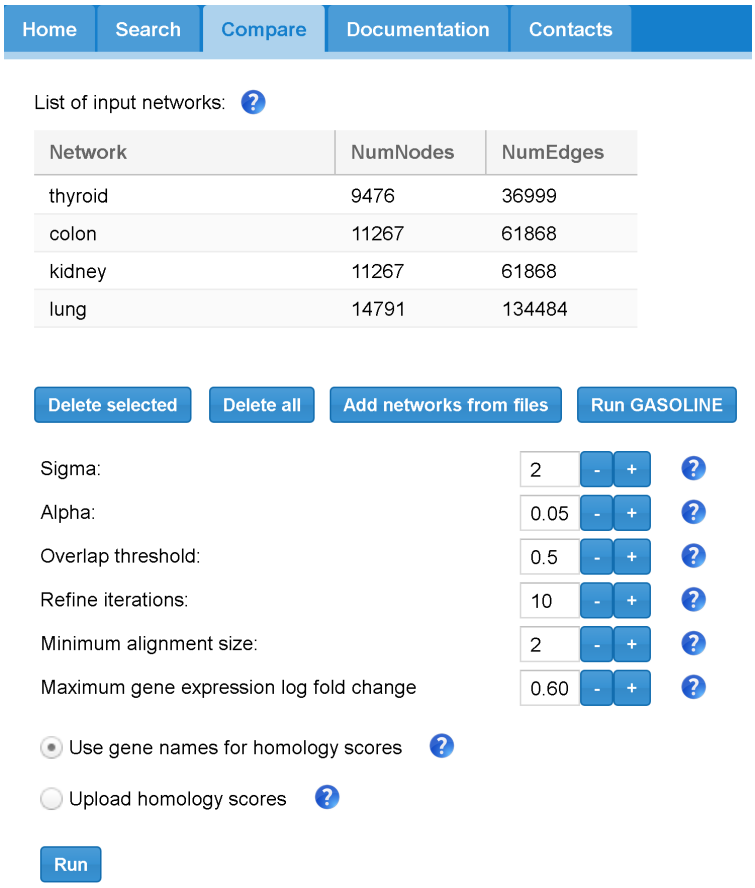


Figure 4.10: The SPECTRA Compare panel. In this example, we first loaded 4 different TS-PPI networks from files using the "Add networks" button. Then by clicking on "Run Gasoline" the form for the selection of the adapted GASOLINE input parameters appears.

is used to remove highly overlapping alignments. It takes values between 0 and 1 (default 0.5, which means 50%);

- "Refine iterations": the number of iterations of the iterative phase, i.e. extend steps followed by a removal step (default 10);
- "Minimum alignment size": the minimum size of a local alignment. Local alignments with size lower than this minimum size are not reported in final list (default 3);
- "Minimum gene expression log fold change threshold": value for MaxLogFoldThreshold, which controls the extension process (default 0.60).

According to the experiments reported in [105] and [106], we assigned to each parameter default values which guarantee a good tradeoff between speed and accuracy of GASOLINE.

"Alpha" and "Refine iterations" parameters are strictly related to the stochastic nature of the algorithm. Lower values for "Alpha" and higher values for "Iter Refine" can be assigned to improve accuracy, however the suggested default values are enough to yield good alignment results. Higher values of "Sigma" can be used to restrict the search to alignments starting from central genes in the input networks and to speedup the algorithm. Lower values of "Overlap threshold" and higher values of "Minimum alignment size" allow to prune the final set of local alignments.

MaxLogFoldThreshold is the most critical parameter for GASOLINE. By increasing this threshold, the number and the size of final local alignments can highly decrease and the algorithm

could become much faster. Notice that there is no constant ideal value for MaxLogFoldThreshold, because it is highly dependent on the properties of input expression data. For log-transformed gene expression data, like the one which are present in SPECTRA database, low values of MaxLogFoldThreshold (0.2-1) are recommended.

Before running the adapted GASOLINE by clicking on "Run GASOLINE" button, the user has to indicate an homology scoring scheme between proteins of different aligning TS-PPI networks (Fig. 4.10). The default naive solution is to use gene names for computing similarities: if two nodes have the same label, then they are considered homologs. Otherwise, user can upload an homology score file.

When the adapted GASOLINE ends, it gives as output a list of local alignments (if any, see Fig. 4.11). For each alignment, the size, the average value of MaxLogFold and the ISC score are reported.

By selecting an alignment, its details are reported on the right (Fig. 4.11). Alignment data include the set of nodes and edges attributes. The final mapping of aligned nodes is represented as a matrix in which columns contain nodes of the same network and rows represent the mapped genes.

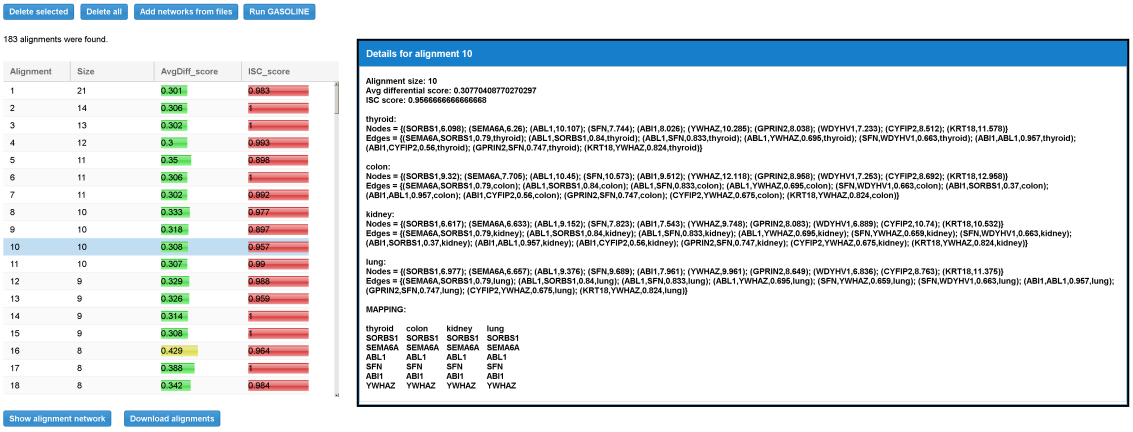


Figure 4.11: Result table for the differential local alignment of the four TS-PPI networks of Fig. 4.10 with the adapted GASOLINE. The table reports, for each alignment, the size (i.e. the number of aligned nodes), the average expression difference between aligned nodes and the ISC (Index of Structural Conservation) score. When the user selects a row in the table, a panel with alignment details is shown to the right. Details include the list of aligned subnetworks (defined by the set of nodes and edges) and the mapping between aligned nodes. Nodes of aligned networks are represented by the corresponding ids, followed by their weights, while edges are represented by the ids of interacting proteins, followed by the interaction weights and the corresponding tissues. Alignment mapping is represented as a matrix where rows contain aligned proteins and columns represent nodes of the same subnetwork.

Alternative Input for SPECTRA

User can upload text files in SPECTRA for building and comparing network. Expression data can be provided as text files in the "Expression data" section (Fig. 4.7b) by selecting the "Upload expression data" option. Expression data files should have a matrix format with a row header representing tissues, a column header representing genes, and matrix elements indicating the gene expression value in a tissue.

There are two ways to provide input TS-PPI networks for comparison. User can either upload a text file or create the TS-PPI network with the SPECTRA searching tool and pass it to the comparison page. In the first case, network files are uploaded by clicking on "Add networks from files" in the "Compare" tabbed panel (Fig. 4.10).

TS-PPI network files for comparison follows the same format of the result table in SPECTRA (Fig. 4.8), except for the dataset coverage, with fields separated by tab characters. In the second case, one or more TS-PPI networks for specific tissues are passed to the comparison tool, by clicking on the "Add to compare list" button. The network is then added as input to the comparison list (Fig. 4.10). By default, networks are added with the name of the corresponding tissue, optionally followed by a progressive number whenever two or more TS-PPI networks for the same tissue are already present in the table. Anyway, networks can be later renamed by the user from the comparison table, before running GASOLINE.

In the homology file, needed to run the adapted GASOLINE algorithm, each row contains a pair of nodes of different TS-PPI networks, followed by a positive score value.

SPECTRA Output

TS-PPI networks (or subnetworks of them) are downloadable from the result panel, by clicking on "Download network" button (Fig. 4.8). The user can filter the set of tissues upon which the TS-PPI network is defined. TS-PPI networks will be saved into different text files, one for each selected tissue or tumor. The file format is the same of the result table (Fig. 4.8), with fields separated by tab characters.

The set of differential alignments returned by the adapted GASOLINE can be saved as .zip archive. The archive will contain a text file for each alignment. Each file contains the same alignment information reported in Fig. 4.11.

Results can also be visualized by using Cytoscape.js (http://js.cytoscape.org), a JavaScript library for the analysis and visualization of networks. In the 2D visualization, TS-PPI networks can be navigated and zoomed. A TS-PPI network can be visualized from the result panel (Fig. 4.8). Fig. 4.12 shows two different examples of visualizations of TS-PPI networks within SPECTRA, with one (Fig. 4.12a) or more (Fig. 4.12b) tissues. Nodes and edges are differently colored according to the tissues of the TS-PPI network. Nodes are represented as pies with multiple colored slices. The diameter of the pie is proportional to the total expression score of the gene (considering all tissues of the TS-PPI network) and the size of each pie slice is proportional to the expression score of the gene in the corresponding tissue. Edge line widths are proportional to the interaction weights.

The alignments can be visualized in 2D (Fig 4.17), by selecting them from the list of local alignments and clicking on the "Show alignment network" button (Fig. 4.11). Aligned nodes are colored according to the network they belong to and their sizes are proportional to the genes expressions. Edges are divided into two categories: intra-edges and inter-edges. Intra-edges connect nodes of the same subnetwork and are represented with solid lines with variable width, depending on the interaction weights. Inter-edges connect aligned nodes of different networks and are drawn with dashed black lines. In both cases, we used the Constraint-Based Layout (COLA) algorithm [40] for network visualization.

4.3 Experimental results and practical case studies

4.3.1 GASOLINE results and discussion

The performance of GASOLINE has been evaluated on synthetic and real biological networks and compared to the state-of-the-art methods for multiple alignment of PPI networks. All tests have been performed on a Intel Core i7-2670 2.2Ghz CPU with a RAM of 8 GB.

Data Description and Experimental Setup

Synthetic biological networks were generated using NAPAbench [135], a large-scale network alignment benchmark for generating families of evolutionary related synthetic PPI networks, evolved from a common ancestor, according to a given phylogenetic tree. It has been recently used as a framework to compare the accuracy and the scalability of different alignment algorithms [135, 136].

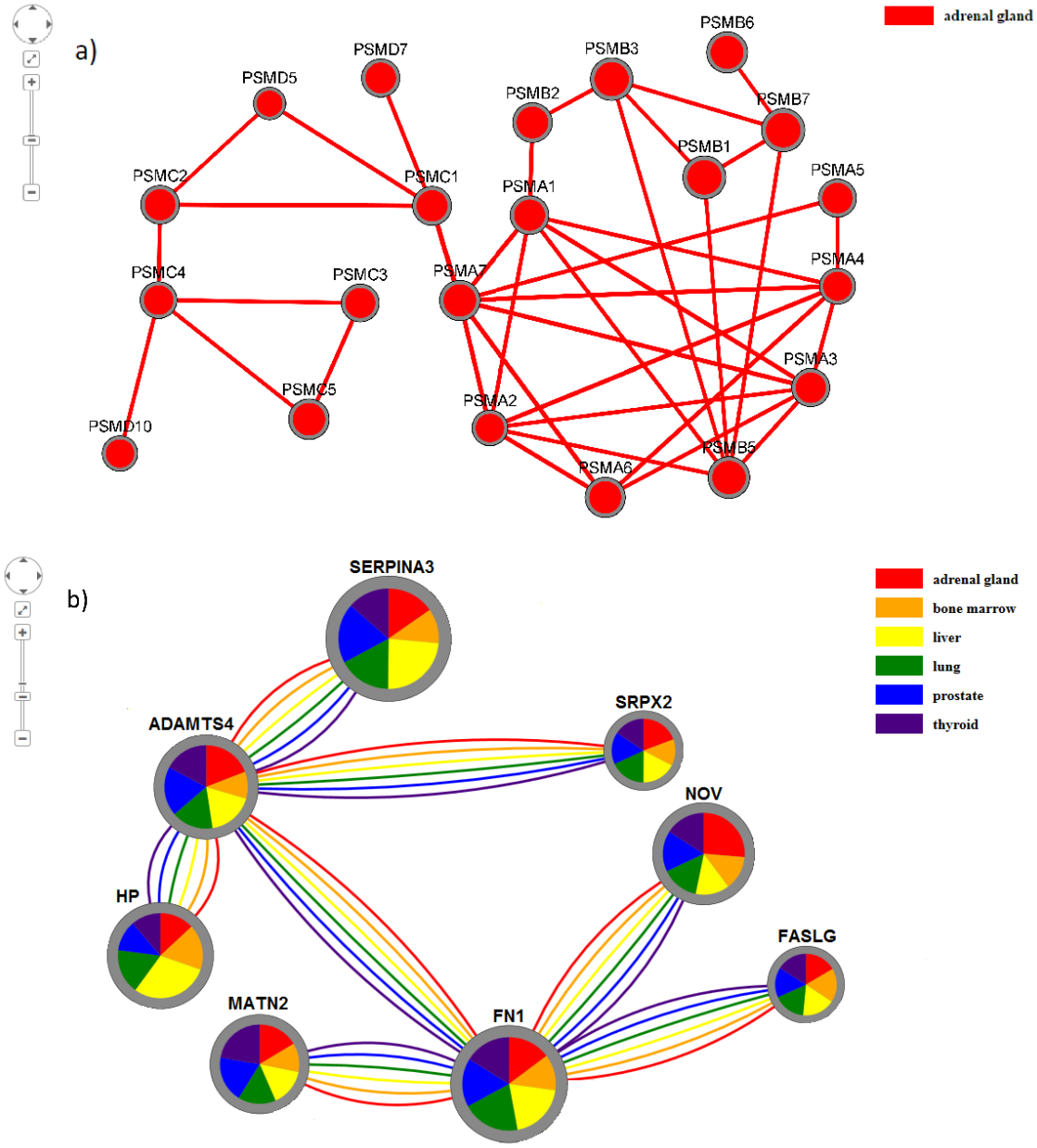


Figure 4.12: The network visualization in SPECTRA. a) A TS-PPI network for a single tissue; b) A TS-PPI network for multiple tissues. In this case, nodes are represented as pies with slice sizes proportional to the expression of corresponding gene in a tissue. Nodes and edges are colored according to the corresponding tissue, and node dimensions are proportional to the total gene expression score.

Real biological networks were taken from STRING (version 9.0) [157], a database of known and predicted PPIs, collected from different high-throughput experiments, coexpression data and publications. For every examined species, we filtered the set of interactions, considering only experimentally supported interactions (i.e. those with positive values on the "Experimental" field). We point out that this kind of protein interactions can also result from experimental knowledge transferred from one species to another.

Three different case studies have been examined:

a) Pairwise and multiple alignments of synthetic networks;

b) 6-way alignment of real PPI eukaryotic networks;

c) 25-way alignment of real PPI vertebrata networks.

In case studies a) and b) we compared our method against three different global and local multiple network alignment algorithms: SMETANA [136], IsoRankN [89] and NetworkBLAST-M [71]. To our knowledge, the first two methods are the best global many-to-many aligners of two or more species, while NetworkBlast-M represents the state-of-the-art for the local alignment problem.

We chose both global and local alignment methods in order to (i) highlight the ability of GASOLINE to correctly map many proteins of different species as a good global aligner does; (ii) find many conserved complexes as a good local aligner does. In our experiments we ran IsoRankN with α=0.7 and $K = 10$ and we used the restricted-order version of NetworkBLAST-M for computational reasons. To compute similarities between proteins, we used Blast bit scores for GASOLINE, SMETANA and IsoRankN, and Blast E-values for NetworkBLAST-M.

We designed several measures to evaluate the specificity, the sensitivity and the functional consistency of the alignment algorithms, both for synthetic and for real biological networks, following the methodology described in [135]. We also tested the robustness of the analyzed methods in case of low sequence similarity between homologous proteins and the scalability with respect to the number and the size of aligned networks.

In case study c) we tested the ability of GASOLINE to find highly conserved complexes across many species in reasonable time. We used a simpler similarity scoring function, since calculating E-values was unfeasible due to the huge number of sequence pairs. Starting from the information about orthologous groups (COG, KOG and NOG) obtained from the STRING database, we computed the Jaccard similarity coefficient [68] between the sets of two proteins' orthology groups. Jaccard coefficient is defined as the number of common groups divided by the cardinality of the union of the two sets.

GASOLINE parameter tuning

The algorithm needs a few parameters to be set out:

- *IterSeed*: number of iterations of Gibbs sampling in the bootstrap phase;
- *IterExtend*: number of iterations of Gibbs sampling in the extension step;
- *IterPhase*: number of iterations of each iterative phase;
- σ: threshold value for the degree of candidate seed nodes;
- *Overlap*: threshold value for overlap percentage;
- *Minimum complex size* (*MCS*): minimum number of proteins of a conserved complex;

Notice that, some parameters are strictly related to the stochastic nature of the algorithm (i.e. *IterSeed*, *IterExtend*, *IterPhase*). Such parameters have been determined in connection to the convergence of the algorithm on the network instances tested. Due to the fast convergence of Gibbs sampling, we experimented that few hundreds of iterations (e.g. 100-300) of Gibbs sampling in both seed and extension phases are enough. We also experienced that few iterations of the iterative phase (e.g. 5-15) are enough to yield good alignment results. Higher values of *IterSeed*, *IterExtend* and *IterPhase* make GASOLINE slower and do not improve significantly its accuracy.

The threshold parameter (σ) for the seed selection represents a tradeoff between speed and accuracy of our method. In order to maximize the accuracy and the coverage of GASOLINE, its value has been set to 1 in all the comparisons experiment we executed. This means that no filtering

on the nodes has been applied for the networks alignment. However, we give the possibility to the users to increase the value of such parameter for large input instance to speed up GASOLINE, as we did in third case study for the 25-way alignment.

The *Overlap* parameter allows to filter the output produced by the algorithm. We chose an intermediate value (0.5) for this parameter. However, the user can vary this parameter to tune the number of subgraphs alignments that GASOLINE gives as output.

The MCS parameter can be used to set the smallest size of subgraphs alignments. In our experiments, MCS takes the minimum value (1), in order to maximize protein coverage, since we are comparing our method with global alignment algorithms too. Unlike the threshold parameter for the seed selection, it does not affect the running time of GASOLINE, since it concerns the postprocessing phase.

Since σ is the most critical parameter for the method, to help user in tuning GASOLINE we give the following general rule which involves only σ. If we increase σ, GASOLINE becomes faster but its accuracy can slightly decreases, because it can miss some small complexes.

Finally, we established experimentally default values for the following parameters:

- $IterSeed = 200$;
- $IterExtend = 200$;
- $Overlap = 0.5$;
- $IterPhase = 10$;

Case study 1: alignment of synthetic networks

We first assessed the performance of GASOLINE on different datasets of synthetic similar PPI networks generated with NAPAbench [135]. We considered three different partitions of datasets. Each partition consists of three families of aligning networks, generated using three different network growth models, i.e. *duplication-mutation-complementation* model (DMC) [163], *duplication with random mutations* model (DMR) [151, 118] and *crystal growth* model (CG) [77]. From now on, we will denote them as DMC, DMR and CG families. We set $q_{con} = 0.1$ and $q_{mod} = 0.48$ for DMC, $q_{new} = 0.2$ and $q_{del} = 0.5$ for DMR and $\delta = 4$ for CG.

The first partition is formed by families of 2 closely related networks, evolved from a common ancestor with $N_a = 5000$ nodes. The families of the second partition consist of 4 evolutionary distant networks, with a common ancestor of $N_a = 4000$ nodes. In the last partition, each family contains 8 networks with different evolutionary distances, generated from a common ancestor of $N_a = 3000$ nodes.

Fig. 4.13 depicts the phylogenetic trees used for the families of each partition. All the branches of the phylogenetic tree have weight 500, meaning that each node of the tree (except the root) is a network obtained from the parent node by adding 500 nodes according to the growth model used.

In this case study, we ran GASOLINE with $\sigma = 1$ (i.e. no filtering based on the node degree of candidate seeds) and $MCS = 1$.

To measure the overall accuracy of the proposed methods, we used functional groups associated by NAPAbench to each protein of the aligning networks. We call equivalence class a set of proteins of different species (one or more for each network), which are mapped together by a given algorithm. An equivalence class is claimed as correct if all the included nodes belong to the same functional group.

For each method we computed three different quality measures:

- *Specificity* (SPE): the relative number of correct equivalence classes;
- *Correct nodes* (CN): the total number of proteins assigned to correct equivalence classes;

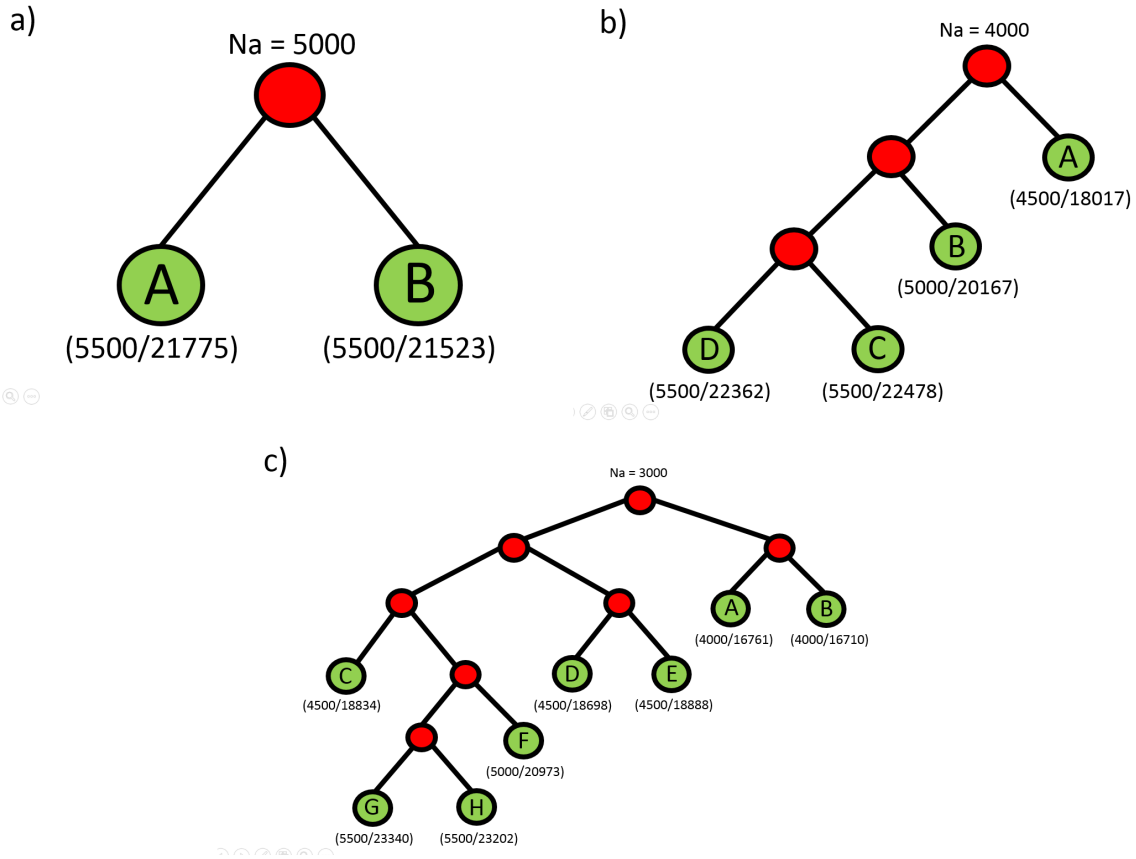


Figure 4.13: Phylogenetic trees for the synthetic networks generated using NAPAbench: (a) 2-way alignment, (b) 4-way alignment, (c) 8-way alignment. Below each leaf node, the number of nodes and the average number of edges across the CG, DMC and DMR families of the corresponding network are shown in parenthesis.

- *Mean normalized entropy* (MNE): the mean normalized entropy of the predicted equivalence classes. Given an equivalence class C, the normalized entropy of C is computed by:

$$H(C) = -\frac{1}{\log d} \sum_{i=1}^{d} p_i \log p_i \tag{4.22}$$

  where p_i is the fraction of proteins in C that belong to the i-th functional group and d is the number of different functional groups.

CN reflects the sensitivity of the method, while MNE measures the consistency of the predicted alignments. For SMETANA and IsoRankN we considered only equivalence classes that contain at least one node from each species.

Tables 4.3, 4.4 and 4.5 summarize the values of SPE, CN and MNE of the proposed methods for all the alignments of 2, 4 and 8 networks, respectively. Each table reports the results obtained for DMC, DMR and CG families. In all cases SMETANA has the highest sensitivity, recovering a high number of CN. However, our method is more precise, especially in the 8-way alignment, resulting in a higher specificity and a lower rate of false positives. The lower sensitivity of GASOLINE is due to the fact that our method is based on 1-to-1 mapping, while SMETANA performs a many-to-many alignment. The other two methods generally exhibit lower specificity, sensitivity and consistency than SMETANA and GASOLINE. Interestingly, the specificity of GASOLINE remains very high (around 90%), even though the number of networks increases, while the accuracy of all the other

algorithms tends to decrease. In particular, the accuracy of NetworkBLAST-M falls down from pairwise to 8-way alignment, going from 88% to 4%.

Table 4.3: Performance of alignment algorithms for pairwise alignments of synthethic PPI networks (CG=crystal growth model, DMC=duplication-mutation-complementation model, DMR=duplication with random mutations model)

	CG			**DMC**			**DMR**		
	SPE	**CN**	**MNE**	**SPE**	**CN**	**MNE**	**SPE**	**CN**	**MNE**
GASOLINE	90.35%	6536	0.096	87.49%	5209	0.125	89.58%	5346	0.104
SMETANA	**96.09%**	**9420**	**0.035**	**94.62%**	**9823**	**0.051**	**95.83%**	**9742**	**0.039**
NetworkBLAST-M	53.92%	7639	0.461	88.1%	5560	0.119	87.85%	5251	0.121
IsoRankN	79%	7048	0.199	83.75%	7818	0.154	85.32%	8042	0.138

Table 4.4: Performance of alignment algorithms for 4-way alignments of synthethic PPI networks (CG=crystal growth model, DMC=duplication-mutation-complementation model, DMR=duplication with random mutations model)

	CG			**DMC**			**DMR**		
	SPE	**CN**	**MNE**	**SPE**	**CN**	**MNE**	**SPE**	**CN**	**MNE**
GASOLINE	**92.75%**	10400	**0.059**	87.62%	10421	0.101	88.74%	9934	0.091
SMETANA	90.41%	**14154**	0.073	**91.06%**	**15495**	**0.07**	**93.15%**	**15255**	**0.055**
NetworkBLAST-M	31.72%	9747	0.639	44.01%	7336	0.514	56.06%	6916	0.395
IsoRankN	62.46%	5793	0.302	74.83%	8856	0.195	74.64%	9077	0.195

Table 4.5: Performance of alignment algorithms for 8-way alignments of synthethic PPI networks (CG=crystal growth model, DMC=duplication-mutation-complementation model, DMR=duplication with random mutations model)

	CG			**DMC**			**DMR**		
	SPE	**CN**	**MNE**	**SPE**	**CN**	**MNE**	**SPE**	**CN**	**MNE**
GASOLINE	**94.52%**	15359	**0.044**	**87.29%**	15735	**0.097**	88.6%	14842	0.092
SMETANA	82.93%	**17489**	0.114	83.89%	**21976**	0.102	**88.7%**	**20315**	**0.081**
NetworkBLAST-M	4.01%	5376	0.851	4.03%	5932	0.836	5.96%	6020	0.818
IsoRankN	32.09%	2433	0.485	51.74%	7112	0.305	50.84%	6677	0.305

Tab. 4.6 compares the running times of the four algorithms for each of the nine network families considered. In the pairwise case, NetworkBLAST-M and SMETANA are the fastest methods, while in the multiple case GASOLINE shows the best performances. Surprisingly, for all DMR families SMETANA performed better than GASOLINE, even in the multiple case. This is probably due to the fact that networks in DMR families are sparser than the others and GASOLINE usually works better with denser networks. This hypotheses seems to be supported by the tests performed on the real biological networks, which are two or three times denser than the synthetic ones (see Tab. 4.7 and Tab. 4.13).

Next, we investigated the effects of sequence similarities on the performances of the algorithms. Following the approach used in [135, 136], we introduced a bias term b on the similarity score distribution of potential orthologs between different networks, in order to increase the differences between the similarity scores of orthologous nodes and those of non-ortologous nodes. We generated 6 different families of aligning networks, by varying b between -150 and 250. Negative values of b penalize sequence similarity scores, while positive values of b enhance them, making the

Table 4.6: Running times (min) of alignment algorithms for the alignments of synthetic networks (CG=crystal growth model, DMC=duplication-mutation-complementation model, DMR=duplication with random mutations model)

	2-way			**4-way**			**8-way**		
	CG	**DMC**	**DMR**	**CG**	**DMC**	**DMR**	**CG**	**DMC**	**DMR**
GASOLINE	2.3	3.1	6.03	**4.15**	**4.15**	10.23	**11.98**	**15.83**	32.43
SMETANA	1.77	1.05	0.97	8.51	5.97	**6.18**	40.13	29.18	**29.68**
NetworkBLAST-M	**1.72**	**0.78**	**0.96**	30.85	50.62	45.35	428.93	717.36	661.22
IsoRankN		103.5	110.7	524.1	641.6	578.4	2081.4	2991.7	2350.4

alignment easier to compute. All families consist of 4 networks generated with CG model, using the phylogenetic tree of Fig. 4.13b).

Fig. 4.14 reports the values of SPE and CN for different values of b. GASOLINE shows the most constant level of accuracy among all the methods, even for negative values of b. This means that our algorithm exploits topological informations well and it can produce many correct alignments even when sequence similarity scores are very noisy (73% of SPE, when $b = -150$). Similarly, SMETANA shows a constant level of accuracy, but its specificity is always below that of GASOLINE for non positive values of b. Surprisingly, for the lowest value of b ($b = -150$), our method recovers more correct nodes than SMETANA. On the other hand, NetworkBLAST-M and IsoRankN take great advantage from the increasing bias with respect to both SPE and CN values, so they seem to strongly rely on sequence similarity scores during the computation of the alignments.

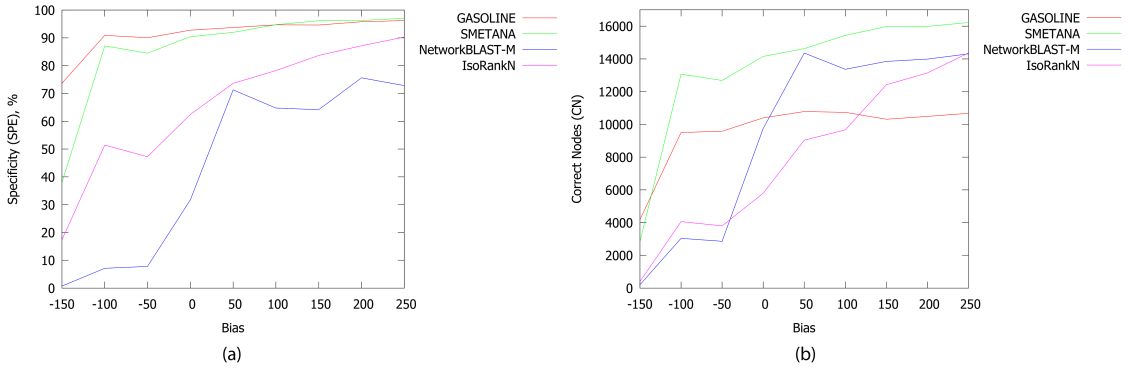


Figure 4.14: a) Specificity (SPE) and b) Number of correct nodes (CN) for various level of bias between the similarity score distribution for orthologs and the similarity score distribution for non-orthologs.

Finally, we tested the scalability of our method, based on the size of aligning networks. We generated 7 different families, by varying the number of nodes of the ancestral network, N_a from 2000 and 5000. Again, all families consist of 4 networks generated with CG model, using the phylogenetic tree of Fig. 4.13b). We performed a comparison between GASOLINE and SMETANA, which are clearly the fastest methods, as shown before. Fig. 4.15 shows the running time for different values of N_a. As can be seen, GASOLINE is always faster than SMETANA and generally shows less variance in running times.

Case study 2: alignment of 6 PPI eukaryotic networks

In the second case study, we compared the four algorithms on real biological networks of 6 species (yeast, worm, fly, human, mouse and rat). Tab. 4.7 describes the features of the networks.

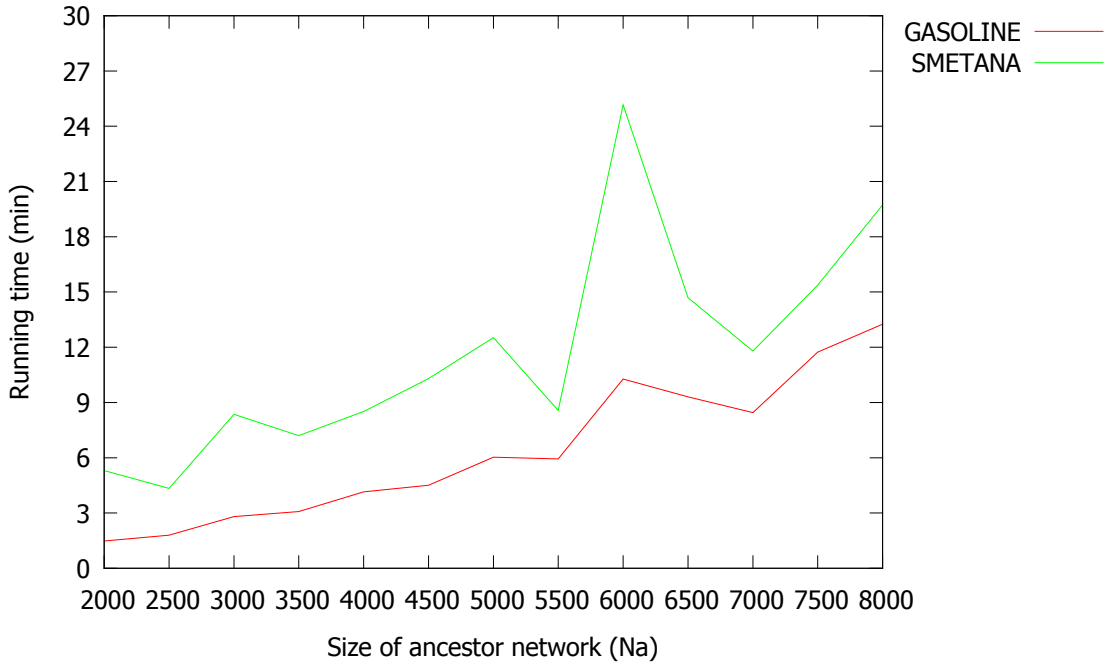


Figure 4.15: Running times of GASOLINE and SMETANA for different number of nodes (N_a) of the ancestor network.

Table 4.7: Features of 6 PPI eukaryotic networks

SPECIES	**# PROTEINS**	**# PPIs**
Caenorhabditis elengans	6173	26184
Drosophila melanogaster	8624	39466
Homo sapiens	12575	86890
Mus musculus	9781	52161
Rattus norvegicus	8763	39932
Saccharomyces cerevisiae	6136	166229

Bit scores and BLAST E-values between all pairs of proteins belonging to different networks were computed. All pairs with E-value greater than 10^{-5} were filtered out. In order to compare the consistency and the accuracy of the algorithms, we used orthologous groups (COG, KOG and NOG), downloaded from STRING [157]. As in the previous case study, we define an equivalence class as a set of proteins of different species which are mapped together by a given algorithm. An equivalence class is claimed as correct if all the included nodes share at least one orthologous group.

To asses the performance of the algorithms, we computed specificity (SPE) and number of correct nodes (CN), and we replaced the mean normalized entropy with a different measure, the mean group consistency (MGC), defined as follows:

$$MGC(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \frac{CommonGr(C)}{Gr(C)} \tag{4.23}$$

where $\mathcal{C}$ is the set of all predicted equivalence classes, $CommonGr(C)$ is the set of groups shared by every protein in C and $Gr(C)$ is the set of groups associated to at least one protein in C.

We decided to change the consistency measure because a protein of a real biological network may be associated to more than one groups, while in the previous case study a protein was always

associated to at most one group, assigned by NAPAbench during the generation of synthetic networks.

Tab. 4.8 reports the quality measures for GASOLINE, SMETANA, NetworkBlast-M and IsoRank-N in the case of pairwise and 3-way alignment. In human-mouse alignment, IsoRank-N unexpectedly failed and did not recover any conserved group. Results show that GASOLINE has much higher SPE and MGC than the compared algorithms, especially in the 3-way alignment case. Moreover, the number of correct nodes found by GASOLINE are now comparable to those of SMETANA, or even higher. Low values of CN in NetworkBlast-M are probably due to the high threshold for the minimum size of complexes (which is 5). Furthermore, NetworkBlast-M exhibit lower values for all considered metrics than GASOLINE in all tested cases.

Table 4.8: Performance of alignment methods for pairwise alignments and 3-way alignments of real PPI networks (W=worm, F=fly, Y=yeast, H=human, M=mouse)

	W-Y			**H-M**			**W-F-Y**		
	SPE	**CN**	**MGC**	**SPE**	**CN**	**MGC**	**SPE**	**CN**	**MGC**
GASOLINE	**98.28%**	**4360**	**0.933**	**98.32%**	17796	**0.973**	**97.52%**	6041	**0.903**
SMETANA	82.89%	4351	0.726	96.1%	**18003**	0.939	77.79%	**6112**	0.662
NetworkBLAST-M	93.77%	2545	0.742	81.2%	6747	0.713	84.13%	3178	0.595
IsoRankN	67.88%	3900	0.601	0%	0	0	56.29%	4526	0.485

Such results are confirmed for the alignment of 4, 5 and 6 species (Tab. 4.9). It is worth noting that the specificity of GASOLINE remains very high (around 95%) and the differences between GASOLINE and the other methods increase (around 20% specificity more than the second best algorithm, SMETANA). In this case, quality measures are not reported for IsoRank-N because of its high running time (more than 2 days of computation).

Table 4.9: Performance of alignment methods for 4-way, 5-way and 6-way alignments of real PPI networks (W=worm, F=fly, Y=yeast, H=human, M=mouse, R=rat)

	W-H-M-Y			**W-F-H-M-Y**			**W-F-H-M-R-Y**		
	SPE	**CN**	**MGC**	**SPE**	**CN**	**MGC**	**SPE**	**CN**	**MGC**
GASOLINE	**95.5%**	**7954**	**0.861**	**94.82%**	9166	**0.847**	**93.8%**	10385	**0.822**
SMETANA	76.95%	7913	0.653	75.19%	**9368**	0.631	73.7%	**10677**	0.612
NetworkBLAST-M	63%	4651	0.303	60.95%	5343	0.268	51.62%	5829	0.228

To sum up, the performance results of GASOLINE in the context of real biological networks are superior to those of synthetic networks, with respect to both specificity and number of correct nodes, which is related to the sensitivity of the algorithm. Moreover, the values of CN are very close to or even higher than SMETANA, though the latter is a global alignment method.

A further comparison between GASOLINE and NetworkBlast-M was made to assess the statistical and biological significance of complexes found by both methods in the alignment of 6 species. We annotated aligned proteins with GO terms (cellular components, processes and functions), taken from BioDBNet [111]. We computed, for every GO category in each complex of the alignments, a p-value based on the hypergeometric distribution. Finally, p-values have been corrected by applying FDR correction for multiple hypotheses testing, with $\alpha = 0.01$.

Tab. 4.10 shows the 10 best complexes identified by GASOLINE, sorted by their size and ISC score. The number of enriched GO categories together with the ranking of the corresponding complexes found by NetworkBlast-M are reported. The table shows that the best results found by GASOLINE are also among the best results identified by NetworkBlast-M.

GASOLINE found more complexes than NetworkBlast-M (46 vs 45). However, most of the results are common to both methods. Nine small complexes (5-7 proteins) have been identified only by GASOLINE and eight small complexes (5-10 proteins) have been recovered only by

Table 4.10: Best 10 complexes found by GASOLINE

RANK	DESCR	SIZE	ISC	GOs	NetBlast RANK
1	Large and small subunit of ribosomes in the cytosol	59	85.6%	16	10, 12 14, 15
2	Spliceosome	40	87.1%	13	5, 9
3	Proteasome	32	95%	17	2, 3
4	Ribosome biogenesis in the nucleolus	25	89.2%	11	4, 16
5	Protein serine/threonine kinase activity	25	75.6%	19	34, 35
6	DNA repair complex	24	92.5%	39	18
7	SSU processome	22	96.4%	4	1
8	DNA directed RNA polymerase	21	94.2%	13	6, 7
9	Vesicle-mediated transport	20	85.5%	20	19
10	Prefoldin complex	19	90.6%	2	37

NetworkBlast-M.

Some of the complexes are correctly split by GASOLINE and wrongly joined in NetworkBlast-M, while other complexes in GASOLINE are actually smaller than the corresponding ones in NetworkBlast-M. This is probably due to the different scoring functions used by the two methods.

All complexes returned by NetworkBlast-M can include non 1-to-1 mapping between proteins of different networks. However, these have a fixed maximum size of 15 proteins. This is a serious limitation in the context of local alignment of biological networks since real biological complexes can be actually bigger [60]. Tab. 4.11 shows that the most significant GO categories found by GASOLINE and NetworkBlast-M for the Proteasome complex have similar significant p-values. Nevertheless, the Proteasome complex found by GASOLINE includes more proteins than the one found by NetworkBlast-M (32 vs 15 proteins).

Table 4.11: GO enriched categories related to the Proteasome complex

GO category	GASOLINE	NetworkBlast-M
GO:0000502	5.551E-17	3.775E-16
GO:0005839	3.701E-17	1.110E-16
GO:0019773	1.199E-15	8.882E-17
GO:0051603	1.480E-16	2.405E-16
GO:0004298	5.551E-17	9.252E-17

In Tab. 4.12 we report the running times of GASOLINE, SMETANA, NetworkBlast-M and IsoRank-N. In the case of pairwise and 3-way alignment, NetworkBlast-M is faster than GASOLINE. However, GASOLINE clearly outperforms NetworkBlast-M and the other algorithms in the multiple case scaling well with the number of networks.

Case study 3: alignment of 25 vertebrata PPI networks

In the last case study, we collected a dataset of 25 vertebrata biological networks. Tab. 4.13 describes the features of these networks.

We ran GASOLINE with higher values of MCS and σ ($MCS = 5$, $\sigma = 7$), for computational

Table 4.12: Running times of GASOLINE, SMETANA, NetworkBlast-M and IsoRank-N

Alignment	**GASOLINE**	**SMETANA**	**NetworkBlast-M**	**IsoRank-N**
W-Y	154 sec	125 sec	**59 sec**	54460 sec
H-M	890 sec	1587 sec	**205 sec**	16620 sec
W-F-Y	**175** sec	351 sec	281 sec	148320 sec
W-H-M-Y	**409 sec**	6310 sec	4854 sec	$>$ 2 days
W-F-H-M-Y	**533 sec**	13380 sec	5999 sec	$>$ 2 days
All networks	**666 sec**	22185 sec	12487 sec	$>$ 2 days

Table 4.13: Features of 25 PPI eukaryotic networks

SPECIES	**# PROTEINS**	**# PPIs**
Anolis carolinensis	6510	31135
Bos taurus	8474	42234
Canis familiaris	8440	42239
Cavia porcellus	8185	42208
Danio rerio	5720	25732
Dasypus novemcinctus	6850	30495
Equus caballus	8144	40703
Felis catus	7200	32547
Gallus gallus	6409	29534
Gasterosteus aculeatus	6018	28276
Homo sapiens	12575	86890
Macaca mulatta	8787	41460
Monodelphis domestica	7800	38002
Mus musculus	9781	52161
Ornithorhynchus anatinus	6035	26467
Oryctolagus cuniculus	8010	39304
Oryzias latipes	5754	26880
Pan troglodytes	8677	44263
Pongo pygmaeus	8551	43984
Rattus norvegicus	8763	39932
Sus scrofa	6752	29852
Taeniopygia guttata	6271	28791
Takifugu rubipres	5872	27077
Tetraodon nigroviridis	5779	25730
Xenopus tropicalis	6153	29769

reasons due to the high number of aligned networks. We found 36 complexes conserved in all species. Tab. 4.14 lists the 10 highest-scored ones, together with the number of significantly enriched GO categories.

Most of the complexes found by GASOLINE in the second case study are also present in this third one. However they are smaller here (i.e. spliceosome), due to (i) the higher number of aligned networks; (ii) incompleteness of PPI networks data in some species. GASOLINE took 2250 seconds ($\sim$38 minutes) to perform the alignment of all 25 vertebrata PPI networks.

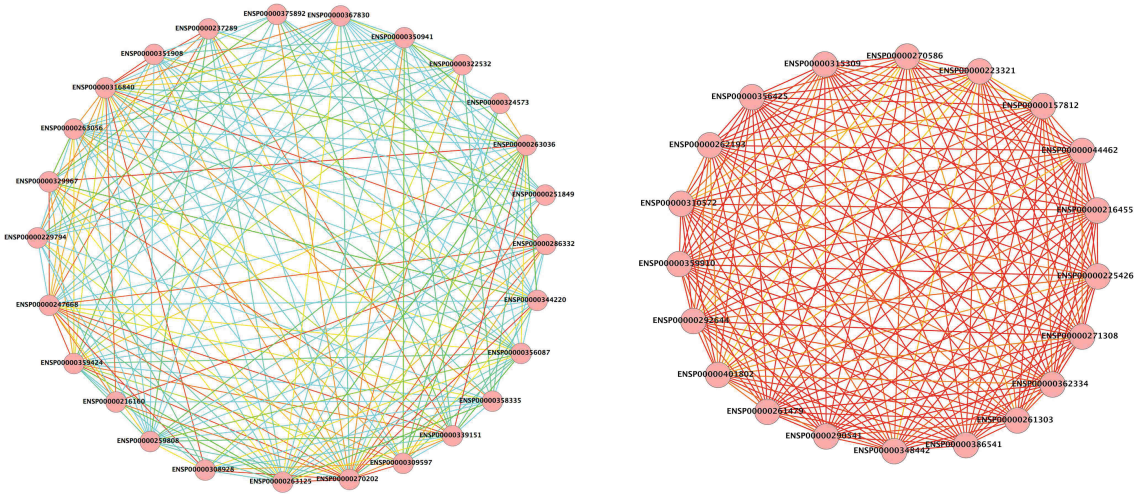
We also analyzed phylogenetic relations among corresponding proteins of distant species in local alignments. Largest and most conserved complexes returned by GASOLINE, the proteasome and the chaperonin, were considered. We represented the conserved cluster of interactions as a single meta-graph (Fig. 4.16), where nodes are classes of aligned proteins (one for each species) and edges are colored according to the conservation extent of the corresponding interaction.

In Fig. 4.16 (a) we depict the meta-graph of Chaperonin complex, whereas in Fig. 4.16 (b) we

Table 4.14: Best 10 conserved complexes found by GASOLINE for the alignment of 25 vertebrata PPI networks

RANK	DESCR	SIZE	ISC	GOs
1	Protein serine/threonine kinase activity complex	26	86.1%	19
2	Proteasome	20	91.3%	14
3	Nuclear receptor DNA complex	16	78.7%	13
4	Histone deacetylase complex	14	85.4%	14
5	Vesicle-mediated transport	13	86.5%	10
6	Cyclin-dependent kinase complex	13	85.9%	8
7	Chaperonin-containing T-complex	13	85.5%	8
8	DNA directed RNA polymerase II	12	94.3%	8
9	Eukaryotic translation initiation factor 3	12	91.8%	5
10	Spliceosome	11	92.6%	5

present the Proteasome complex. In both complexes, we can observe the presence of a big core of highly conserved protein interactions. This may represent a sort of ancestral complex from which all the species-specific complexes have differently evolved.



(a) (b)

Figure 4.16: Meta-graph of complexes found by GASOLINE for the alignment of 25 PPI vertebrata networks: (a) chaperonin complex, (b) proteasome complex. Cyan indicates low conservation, green medium, yellow high and red very high.

4.3.2 A case study for the adapted GASOLINE

In this subsection, we show a practical usage of SPECTRA through a case study. We compared a a set of four TS-PPI networks, built from genes expression data in normal and well differentiated, moderately differentiated and poorly differentiated breast cancer tissues. The aim is to identify subnetworks of differentially-expressed genes across the normal breast and the three different grades of breast tumors.

Data preprocessing

We downloaded four breast cancer expression datasets for which information about the stage of breast tumors were available: GSE2361 [49], GSE2990 [152], GSE4922 [67] and GSE7390 [35]. We normalized data using RMA [97] in R Bioconductor package [51] .

The four expression datasets were then combined using COMBAT [69] into the R InSilicoDb-Merging package. Finally, we grouped samples of the integrated dataset into four categories according to the grade of breast tumor (0 for normal tissue, 1 for well-differentiated tumor cells, 2 for moderately differentiated cells and 3 for poorly differentiated cells). For each category, we computed the average expression value of each gene among samples. Results are stored into four different files (one per category).

Uploading data in SPECTRA and building breast TS-PPI networks

We loaded the expression files in the "Expression data" panel in SPECTRA (Fig. 4.7b) and we selected BioGRID and IntAct as PPI datasets in the "Interaction data" panel (Fig. 4.7c). SPECTRA builds four TS-PPI networks, each of them has 7,472 nodes and 29,765 edges. We added each network to the comparison list of GASOLINE (Fig. 4.10), by clicking on *Add to compare list* from the Result panel (Fig. 4.8).

Results of the adapted GASOLINE on TS-PPI networks

Networks have been aligned by clicking on *Run GASOLINE* with the following parameters:

- Sigma = 1;
- Alpha = 0.05;
- Overlap threshold = 0.5;
- Refine iterations = 10;
- Minimum complex size = 2;
- Maximum gene expression log fold change threshold = 0.3;
- Use gene names for homology score.

GASOLINE took 27 seconds to complete the task and returned 20 local alignments. In Fig. 4.17 the two biggest alignments are shown using the SPECTRA visualization tool.

Both alignments contain genes that are known to be involved in breast cancer at different stages.

More precisely, the major group of aligned nodes in Fig. 4.17a is formed by the chemokine proteins (CXCL10, CXCL9, CXCL11, CCL5) and the chemokine receptors CXCR3 and CCR1, which are all highly overexpressed across the different grades of breast tumor. Chemokines can be responsible for leukocyte migration during processes of tissue development and formation, or can attract immune cells to a site of inflammation. Chemokines and chemokine receptors are known to have an important role on cancer metastasis, by facilitating tumor dissemination [112, 75]. DPP4 gene has a lower expression variation but ensures the communication between CCL5, CCR1 and the other chemokine proteins. This result agrees with the key role of DPP4 in signal transduction and tumor progression [125].

The alignment of Fig. 4.17b is characterized by the Human Leukocyte Antigen (HLA) system (HLA-DRB1, HLA-DMB, HLA-DMA, HLA-DRA). The HLA system is composed by proteins on cells surface that are responsible for regulation of the immune system. HLA genes exhibit very high differential expression between normal and tumor cells and their overexpression in breast cancers is confirmed by several papers [13, 74, 30].

The above case study highlights the capability of SPECTRA in helping researchers in producing novel biologically sound hypothesis and insight in the study of tissue specific diseases.

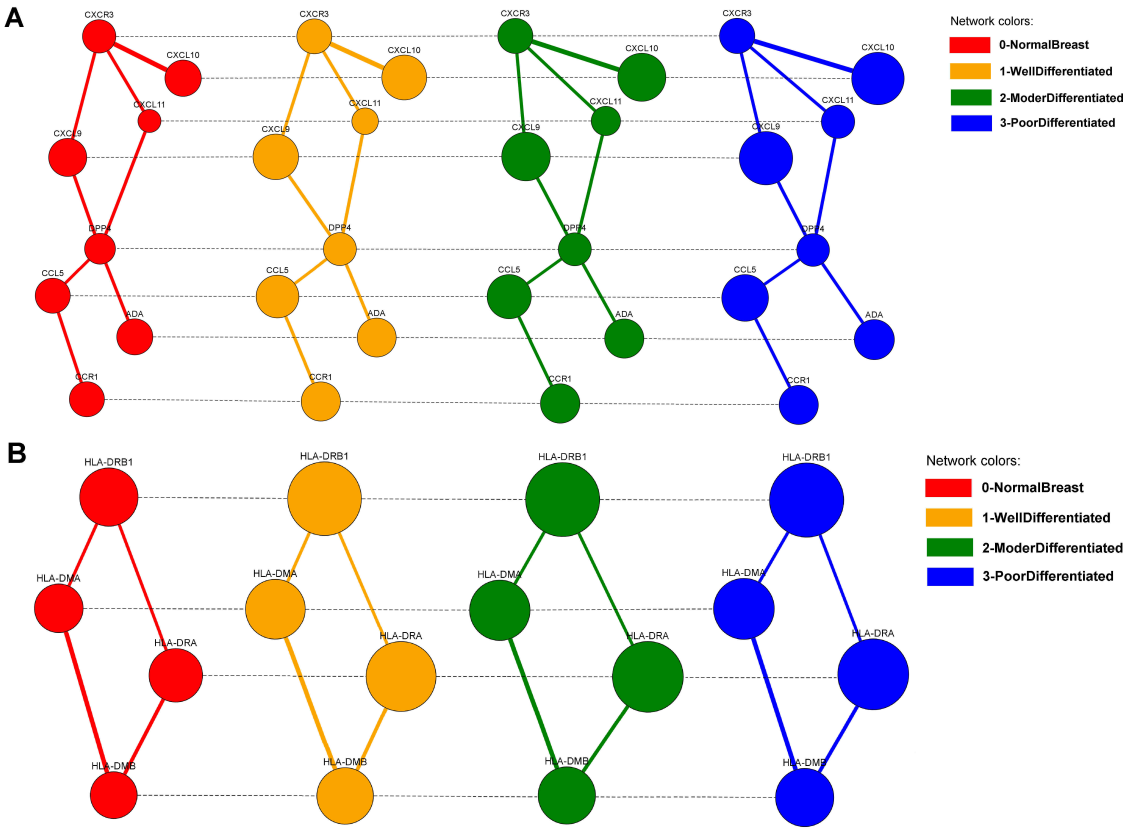


Figure 4.17: The two biggest local differential alignments found by the adapted GASOLINE for the TS-PPI networks of normal breast cells (grade 0), well differentiated cells (grade 1), moderately differentiated cells (grade 2) and poorly differentiated cells (grade 3). A) A complex of chemokine proteins; B) The Human Leukocyte Antigen (HLA) system. Nodes and edges are colored according to the corresponding network. Edge widths are proportional to the strength of interaction. Node dimensions are proportional to the gene expressions. Solid lines (intra-edges) connect the nodes of the same network, while dashed lines (inter-edges) connect the aligned nodes.

Chapter 5

Mining of protein structures

In this chapter, we describe an innovative algorithm for comparing 3D protein structures that also uses Gibbs sampling. Here we aim to identify common structural motifs which could represent conserved binding sites with other macromolecules. An algorithm called PROPOSAL is presented and its results are discussed [106]. Finally, we describe a Java 2D standalone program based on PROPOSAL for local comparison of 3D structures and visualization of corresponding alignments.

5.1 PROPOSAL

5.1.1 Description of the algorithm

PROPOSAL (PROtein comparison through Probabilistic Optimal Structure local Alignment) is a stochastic algorithm for multiple local comparison of 3D structures. The goal is to find potentially conserved binding sites across two or more protein structures.

We formalize the problem as a multiple local structural alignment problem with alignments of fixed size. Let $P = \{P_1, P_2, ..., P_N\}$ be a set of N 3D protein structures w be a positive integer, with $w \geq 3$. We want to find N substructures of w residues, one for each protein, such that structure similarity is locally maximized. We call w the size of the local alignment.

PROPOSAL is able to find approximate solutions to the problem through a greedy and stochastic technique, by using a Gibbs sampling strategy [50] similar to the one described in the previous chapter for GASOLINE.

PROPOSAL is an iterative method. In each iteration it tries to find an optimal local alignment of size w, starting from a predefined triplet of amino acids (e.g. AAC), called fingerprint. Since the fingerprint changes at every iteration and there are 20 amino acids, the maximum number of iterations performed by PROPOSAL has been set to $20^3 = 8000$.

A single iteration consists of three phases. In the first one, called *bootstrap phase*, Gibbs sampling is used to find a local alignment of N substructures (one for each protein), composed by 3 residues each. These substructures, called *seeds* of the alignment, represent small potential conserved motifs shared by the N 3D protein structures.

The quality of the seeds alignment is evaluated according to a proper scoring scheme based on the average Root Mean Square Deviation (RMSD) between the aligned substructures, considering all possible pairs of proteins. The best alignments will have the lowest average RMSD.

Let $C = \{C_1, C_2, ..., C_k\}$ and $D = \{D_1, D_2, ..., D_k\}$ be two sets of residues. The RMSD between C and D is given by the root mean-square deviation of the Cα atomic coordinates of residues, after performing an optimal rigid body superposition. The RMSD is defined as follows:

$$RMSD(C, D) = \sqrt{\frac{1}{w}\sum_{i=1}^{k}((C_{ix} - D_{ix})^2 + (C_{iy} - D_{iy})^2 + (C_{iz} - D_{iz})^2)} \tag{5.1}$$

where C_{ix}, C_{iy}, C_{iz} and D_{ix}, D_{iy}, D_{iz} are the 3D coordinates of residues C_i and D_i, respectively, after the superposition.

We computed RMSDs using QCP [92], a recently proposed algorithm that finds the optimal alignment by using a Newton-Raphson quaternion-based method.

Each seeds alignment having average RMSD $\leq 1\mathring{A}$ is extended by adding one residue at the time, until we reach an alignment of N motifs, each having w residues. The *extension phase* is performed stochastically through Gibbs sampling.

Finally, in the third phase, the alignment is refined, by iteratively removing and adding single nodes to each aligned motif. This *refinement phase* produces the final local alignment. The set of local alignments is then filtered by removing highly overlapping alignments. The outline of PROPOSAL is depicted in Fig. 5.1.

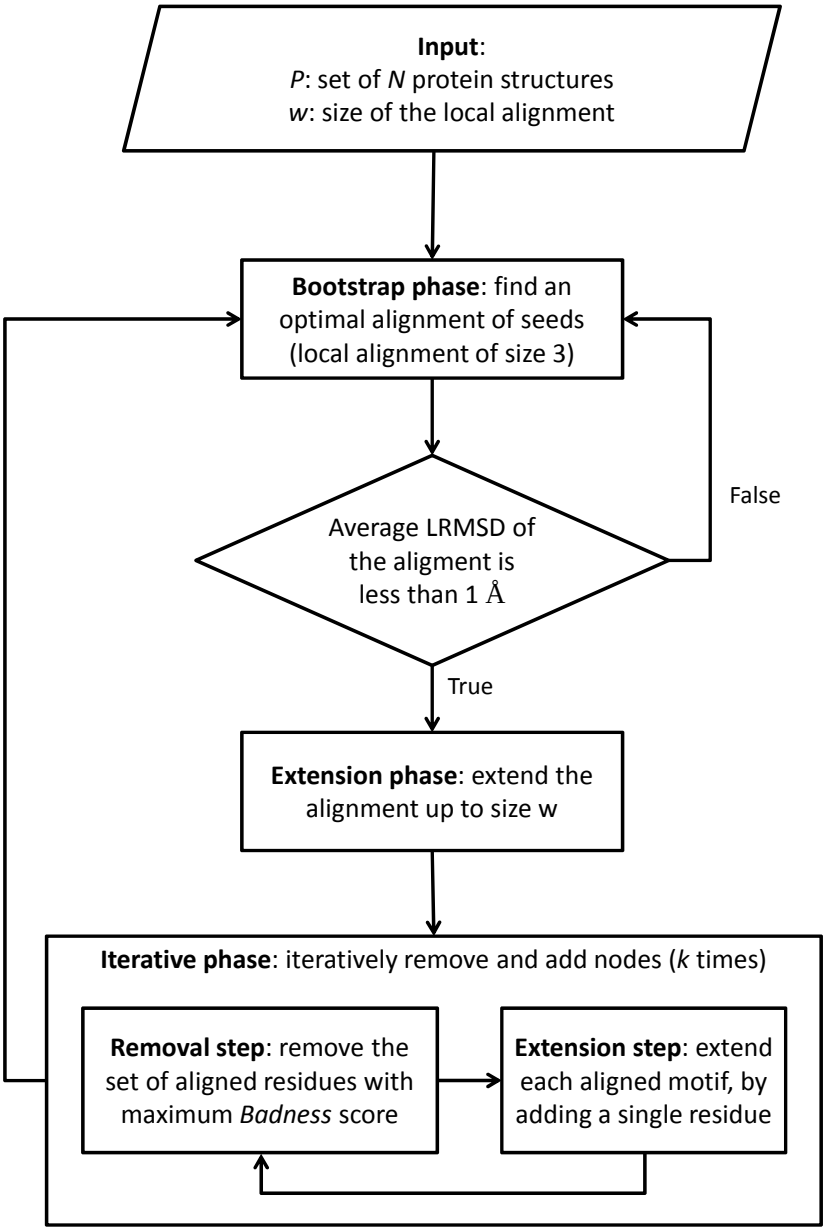


Figure 5.1: Outline of PROPOSAL

The bootstrap phase

The goal of the bootstrap phase is to find an optimal alignment of small substructures of 3 nodes, called seeds. A seed is represented by a triple of residues $A = (A_1, A_2, A_3)$.

The set of possible candidates for the initial alignment consists of all seeds satisfying the following conditions:

a) All residues within the seed are at distance less than 10 $\mathring{A}$;

b) The residue symbols in the triple must match the fingerprint of the corresponding iteration of PROPOSAL.

Feasible candidates are seeds satisfying both a) and b). If one or more proteins contain no feasible candidates, the search stops and a new iteration of PROPOSAL begins.

Once a set of suitable candidates is generated, PROPOSAL tries to construct an optimal initial alignment through Gibbs Sampling on top of a Monte Carlo Markov Chain (MCMC). In the MCMC each state represents an alignment of N seeds, one from each protein structure.

Starting from a random initial state (i.e. a random initial alignment), the sampling method iteratively performs a transition from a state of the chain to another, by replacing a randomly chosen seed of the current alignment with a feasible candidate of the same protein, according to a properly defined transition probability distribution. When Gibbs sampling stops, the last current alignment is returned. If the sampling procedure is iterated a sufficient number of times, it converges to a local optimum solution.

A critical task is to establish when Gibbs sampling can be stopped. The procedure ends when the alignment of seeds does not change. Let $P = \left(\frac{N-1}{N}\right)^i$ be the probability that a protein structure is never selected in i consecutive iterations of Gibbs sampling. The number of iterations of Gibbs sampling is determined by the following parameter k:

$$k = \max\left\{k' : \left(\frac{N-1}{N}\right)^{k'} > \alpha\right\} \tag{5.2}$$

where α is a user-defined probability threshold. If the alignment does not change for k consecutive iterations, the Gibbs sampling is stopped. The lower is α, the more precise and slower will be the sampling procedure. Therefore, α represents a trade-off between accuracy and speed of PROPOSAL.

The transition probability is defined on top of a similarity score, based on the distances between the residues of the seeds. Let $Dist(R_1, R_2)$ be the euclidean distance between the two residues R_1 and R_2 of a 3D structure. Given two seeds $A = (A_1, A_2, A_3)$ and $B = (B_1, B_2, B_3)$, we define the pairwise distance between A and B as:

$$PairDist(A, B) = \prod_{i=1, j=1, i<j}^{3} |Dist(A_i, A_j) - Dist(B_i, B_j)| \tag{5.3}$$

Now, let $S = \{S_1, S_2, ..., S_N\}$ be the alignment of seeds at the i-th iteration of Gibbs sampling and suppose we have to replace S_j by a feasible candidate X of the same protein. The similarity score of X is defined as the inverse of the product of all pair distances between X and the seeds of the current alignment (except S_j):

$$Sim(X) = \frac{1}{\prod_{i=1,\, i\neq j}^{N} PairDist(X, S_i)} \tag{5.4}$$

The transition probability is then computed by normalizing such similarity scores in $[0, 1]$.

The extension of the alignment

In the extension phase, the alignment of residues is extended up to size w by iteratively adding N residues to the current alignment, one from each protein.

Suppose that we start from a substructure alignment of size $w' < w$. The goal is to find an optimal alignment of N residues $R_1, R_2, ..., R_N$, one for each protein, and add such residues to the substructure alignment. R_i must be at distance at most equal to $10\mathring{A}$ from residues in the corresponding current aligned substructure. At the end of this process, the alignment size will be $w' + 1$.

Each extension step is performed through a Gibbs sampling strategy similar to the one used during the bootstrap phase. In the extension phase the similarity score takes into account:

a) The symbol of a candidate residue;

b) The distances between the candidate residue and the aligned residues of the same structure.

Let $SA = \{SA_1, SA_2, ..., SA_N\}$ be the current alignment of size w', where each $SA_i = \{R_{i,1}, R_{i,2}, ..., R_{i,w'}\}$ is a set of residues, and let $A^m = \{A_1^m, A_2^m, ..., A_N^m\}$ be the alignment of candidate residues at the generic m-th iteration of Gibbs sampling.

Suppose we replace A_j^m with a candidate residue X. First, we define a similarity score, $SimSymb(X)$ which evaluates the similarity between the symbol of X and the symbols of residues in A^m (except A_j^m):

$$SimSymb(X) = \prod_{k=1,\, i \neq j}^{N} \textsc{SimMatrix}(X, A_k^m) \tag{5.5}$$

where $\textsc{SimMatrix}(X, A_k^m)$ is a BLOSUM similarity score between X and A_k^m.

Then, we define another similarity function, $SimDist(X)$:

$$SimDist(X) = \frac{1}{\prod_{k=1,\, k \neq j}^{N} PairDist(X, A_k^m)} \tag{5.6}$$

where $PairDist(X)$ is defined as follow:

$$PairDist(X, A_k^m) = \prod_{h=1}^{w'} |Dist(X, R_{j,h}) - Dist(A_k^m, R_{k,h})| \tag{5.7}$$

Finally, the similarity score of X, $Sim(X)$, is the product of $SimSymb(X)$ and $SimDist(X)$. Again, the transition probability of X is the normalization of $Sim(X)$ in [0,1].

The refinement phase

The goal of the refinement phase is to increase the quality of the discovered alignment. An alignment of residues is iteratively removed from the current alignment of substructures and replaced with a new one. The number of iterations is bounded by a user-defined parameter called $IterRefine$. According to our experimental results, a good accuracy can be achieved with relatively small values of such parameter (e.g. 10).

The replaced alignment is chosen according to a *Badness* function defined below.

Let $SA = \{SA_1, SA_2, ..., SA_N\}$ be the final alignment of size w, where each $SA_i = \{R_{i,1}, R_{i,2}, ..., R_{i,w}\}$ is a set of residues. We can view the alignment SA as a matrix $R[N, w]$, where each column represents an alignment of residues and $R[i, j]$ is the j-th aligned residue of the i-th substructure. Our final goal is to compute a *Badness* score for each column of SA and remove the column that maximizes the *Badness* score function from SA.

First, given two aligned residues $R[i, k]$ and $R[j, k]$, we define the function $PairDistAligned$ as follows:

$$PairDistAligned(R[i,k], R[j,k]) = \prod_{h=1\ h \neq k}^{w} |Dist(R[i,k], R[i,h]) - Dist(R[j,k], R[j,h])| \tag{5.8}$$

The *Badness* of a generic column k is:

$$Badness(k) = \sum_{i,j=1,i<j}^{N} PairDistAligned(R[i,k], R[j,k]) \tag{5.9}$$

Once the column with the highest *Badness* score is removed, a new single extension step is performed.

Filtering overlapping alignments

The alignments produced by PROPOSAL are sorted according to the average RMSD across all possible pairs of structures. This sorted list is finally post-processed to filter highly overlapping alignments. Let $SA^i = \{SA^i_1, SA^i_2, ..., SA^i_N\}$ be the local alignment of rank i in the sorted list. We define $Perc(SA^i_k)$ as the percentage of residues in the substructure SA^i_k observed in the previous $i-1$ alignments, and $Perc(SA^i)$ as the average value of $Perc(SA^i_k)$ across all the aligned substructures. If $Perc(SA^i)$ is above a given threshold *Overlap* (between 0 and 1), the alignment is discarded. The choice of the threshold is arbitrary: an average value (e.g. 0.5) gives a good tradeoff between accuracy of and interpretation of final alignments.

5.1.2 Computational complexity

The analysis of the computational complexity of PROPOSAL is similar to the one made for GASOLINE in Chapter 4. Now, the number of executions is a constant (8000) and the size of the alignment, W, is fixed at each execution of PROPOSAL. For simplicity, we suppose to have N input protein 3D structures with the same number of aminoacids, n.

We define the following variables:

- k: the number of iterations of Gibbs sampling in the bootstrap phase and in each extension step of the iterative phase;
- d: the average number of residues at distance at most equal to 10 $\mathring{A}$;
- γ_i: the number of iterations of the iterative phase.

The number of iterations of Gibbs sampling depends on α. In particular, starting from the definition of α (Eq. 5.2), it follows that $\alpha = c\left(\frac{N-1}{N}\right)^k$, that is $\frac{1}{\alpha} = \frac{1}{c}\left(\frac{N}{N-1}\right)^k$, for $0 < c < 1$. Then:

$$k = \log_{\frac{N}{N-1}} \frac{c}{\alpha} = O\left(\log \frac{1}{\alpha}\right) \tag{5.10}$$

As α decreases the number of Gibbs sampling iterations increases.

The time complexity will be expressed as a function of n, W and k. We will assume that the generation of random numbers and the computation of the similarity scores between aminoacids can be done in constant time $O(1)$.

The preprocessing step, i.e. the building of all possible triplets of aminoacids within distance equal to 10 $\mathring{A}$, involves the computation of euclidean distances between all pairs of residues in each structure, therefore it requires $O(Nn^2)$ time. Assuming $N << n$, the preprocessing step costs $O(n^2)$. We can roughly estimate the number of triplets in each structure as $O(nd^2)$.

The bootstrap phase aims at finding an optimal alignment of triplets of aminoacids. The generation of the initial alignment requires $O(N)$ time. The computation of transition probabilities at each iteration of Gibbs sampling involves the computation of similarities between the symbols of aminoacids of different triplets and between their distances. For each pair of triplets such similarities can be evaluated in constant time, so the computation of transition probabilities requires $O(Nnd^2)$ time.

Therefore, the time complexity of the initial phase is:

$$T_{boot}(n) = O(N) + k \times O(Nnd^2) = O(kNnd^2) \tag{5.11}$$

Since, in practice, $N << n$ and $d << n$ we can write:

$$T_{boot}(n) = O(kn) \tag{5.12}$$

As regards the analysis of the extension step, we can follow the same approach described in Subsection 4.1.2. Now, the average number of adjacent nodes is replaced by the average number d of residues at distance at most equal to 10 $\mathring{A}$.

The generation of the initial alignment in the Gibbs sampling costs $O(N)$. The computation of the transition probabilities depends on the size L of the current alignment. Similarities between aminoacids can be evaluated in $O(NL)$ time, so the computation of transition probabilities at each iteration of Gibbs sampling costs $O(dNL)$.

The extension phase simply iterates the extension step $W - 3$ times to produce an alignment of size W from an initial alignment of triplets. The overall cost of the extension phase is:

$$T_{ext}(n) = (W - 3)(O(N) + k \times O(dNL)) \tag{5.13}$$

In the worst case $L = O(n)$. Assuming $N << n$ and $d << n$, we can rewrite the equation as:

$$T_{ext}(n) = O(Wkn) \tag{5.14}$$

The iterative phase consists in a series of single removal steps followed by single extension steps. Each removal step computes the minimum value of a function (*Badness* score) overall L sets of aligned residues. Since the *Badness* score can be evaluated in $O(NL)$ for each set, the removal step requires $O(NL^2)$ time.

The total cost of the iterative phase is:

$$T_{iter}(n) = \gamma_i \times (O(kn) + O(NL^2)) \tag{5.15}$$

Since $N << n$ and $L = O(n)$ in the worst case, we have:

$$T_{iter}(n) = \gamma_i \times (O(kn) + O(n^2)) \tag{5.16}$$

Postprocessing phase consists in filtering highly overlapping complexes and can be done in constant time.

By combining Eqs. 5.12, 5.14 and 5.16 and considering preprocessing operations, the overall cost of PROPOSAL is:

$$T(n) = 8000 \times O(n^2 + kn + Wkn + \gamma_i(kn + n^2)) = O(Wkn + \gamma_i(kn + n^2)) \tag{5.17}$$

From the results of the analysis, it follows that the running time of PROPOSAL is polynomial in n. In fact, in all its applications $\gamma_i << n$ and can be ignored. $\alpha > 10^{-3}$, hence we can approximate k to $O(n)$. Therefore, the final complexity is $O(Wn^2 + n^2)$.

In the worst case, W is very high ($W = O(n)$) and the algorithm requires $O(n^3)$ time. In the average case $W = O(\log n)$ and the running time is $O(n^2 \log n)$.

5.1.3 Results

Three different case studies have been investigated to evaluate the performance of PROPOSAL. In the first one we analysed the accuracy of our method and the effects of input parameters, using the 33 structures of Skolnick's dataset benchmark [86], a set of large protein domains which has been used in several recent studies related to structural comparison of proteins [126, 37].

In the second case study, we compared PROPOSAL to SMAP [174, 175] and ProBis [79, 80], two algorithms for local pairwise structural alignment, on a dataset of known motifs derived from the literature and taken from the Catalytic Site Atlas (CSA) [47].

In the last case study, following the work of [110], we used a subset of these CSA motifs to test PROPOSAL as a local multiple aligner.

PROPOSAL has been implemented in Java 7 and all tests have been performed with an Intel Core i7-2670 2.2Ghz CPU with 8GB of RAM.

PROPOSAL needs a few parameters to be set:

- w: the size of the final alignments;
- α: the probability which determines the number of Gibbs Sampling iterations in the bootstrap and extension phases;
- $IterRefine$: the number of iterations during the refinement phase;
- $AvgOverlap$: a threshold bounding the average overlapping percentage of alignments.

Default values for some parameters have been experimentally established as follows:

- $\alpha = 0.05$;
- $IterRefine = 10$.

Both α and $IterRefine$ parameters have been chosen to guarantee an optimal trade-off between speed and accuracy.

Tests on Skolnick dataset

Skolnick's dataset is divided into four categories, depending on similarity degree and sequence length. Table 5.1 synthesizes the features of each family with respect to the number of proteins, the average sequence length and the average similarity.

Table 5.1: Skolnick's dataset families

FAMILY	PROTEINS	AVG_SEQ_LENGTH	AVG_SIMILARITY
CheY-related	8	124	15-30%
Ferritin	6	170	7-70%
Plastocyanin	8	99	35-90%
TIM Barrel	11	250	30-90%

To evaluate the reliability of PROPOSAL we considered different values of w, depending on proteins sequence similarity. We chose $w = 10$ for the CheY-related proteins, $w = 12$ for the Ferritin family, $w = 15$ for the Plastocyanin proteins, and $w = 20$ for the TIM Barrel family. In all experiments, we set $AvgOverlap = 50\%$ to reduce the final set of alignments and we used default values for α and $IterRefine$. Table 5.2 gives the running time of PROPOSAL and the RMSD of the best alignments.

Table 5.2: Running time and LRMSD of the best alignments on Skolnick's dataset

FAMILY	W	RUNNING_TIME	BEST_RMSD
CheY-related	10	33.95 sec	1.539 $\mathring{A}$
Ferritin	12	46.102 sec	0.428 $\mathring{A}$
Plastocyanin	15	135.936 sec	0.575 $\mathring{A}$
TIM Barrel	20	1542.929 sec	0.428 $\mathring{A}$

The four best structural alignments are represented as 2D contact map alignments with a cut-off of 10 $\mathring{A}$ in Figs. 5.2, 5.3, 5.4, and 5.5. It can be seen that a good structural correspondence between proteins is guaranteed even when the value of w increases. In most cases the absence of few edges or the presence of new links between nodes are due to pairs of residues whose distance is very close to the cut-off.

We analysed label similarity of the four best alignments, by building the sequence logos [28] of mapped residues (Figs. 5.6, 5.7, 5.8, 5.9). Each position contains a graphical representation of the frequencies of residues in that position within the final mapping. Amino acids are represented with different colours, depending on their chemical properties: basic residues (K, R, H) are coloured in

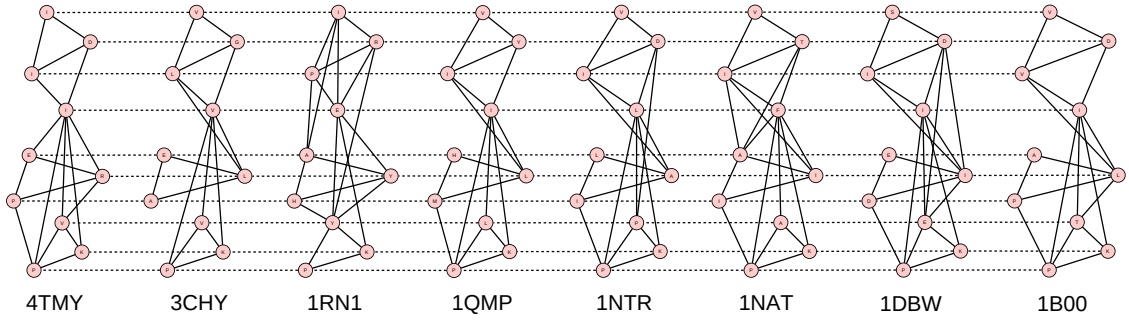


Figure 5.2: Best alignment of the 8 CheY-related protein contact maps with W=10

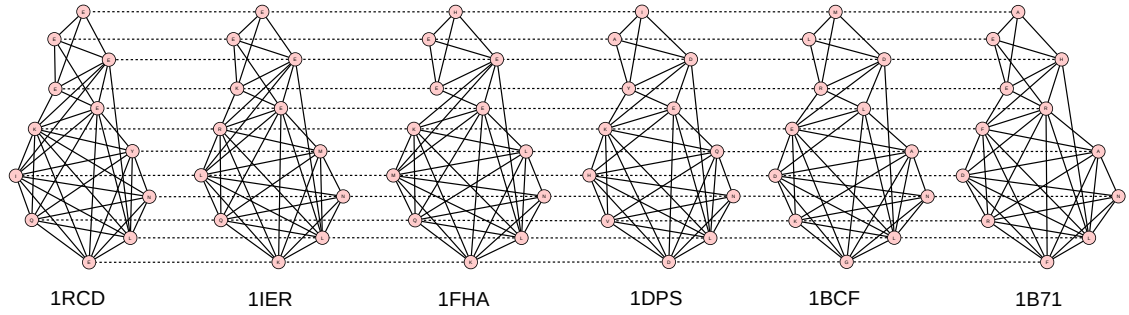


Figure 5.3: Best alignment of the 6 Ferritin protein contact maps with W=12

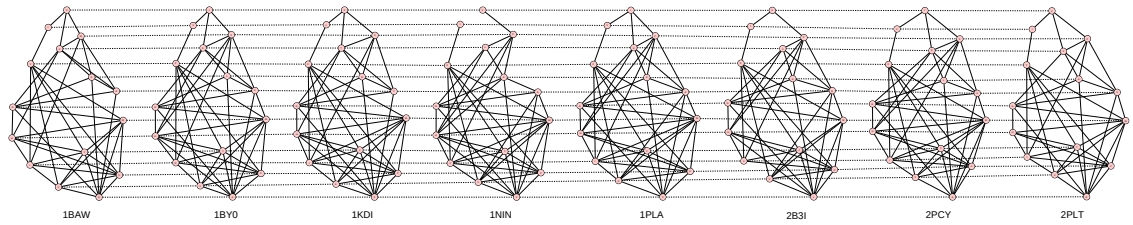


Figure 5.4: Best alignment of the 8 Plastocyanin protein contact maps with W=15

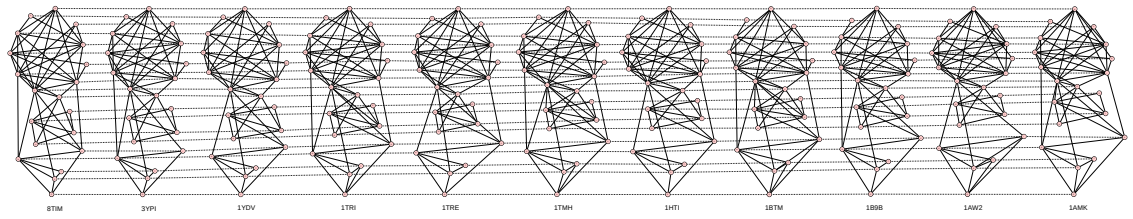


Figure 5.5: Best alignment of the 11 TIM Barrel protein contact maps with W=20

blue, the acidic ones (D, E) in purple, the neutral ones (Q, N, P, S, C) in green, the hydrophobic ones (V, L, I, W, F, M, Y) in orange, and the remaining ones (G, T, A) in red.

Sequence logos reflect the average sequence similarity of proteins within each family: Plastocyanin and TIM Barrel proteins show the best label correspondence. The alignment of Ferritin proteins is quite interesting, since the structural similarity is high, the average LRMSD is very low (0.428 $\AA$, Tab. 5.2), but the corresponding sequence logo shows remarkable dissimilarities between

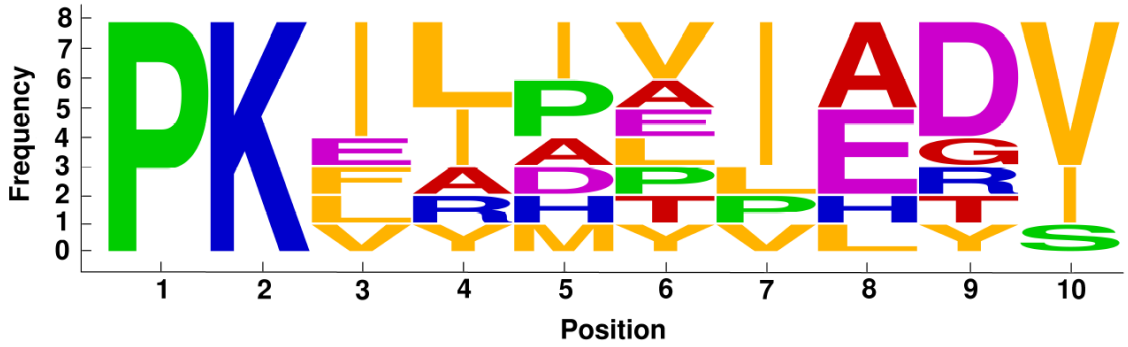


Figure 5.6: Sequence logo of mapped residues in the best alignment of the 8 CheY-related proteins

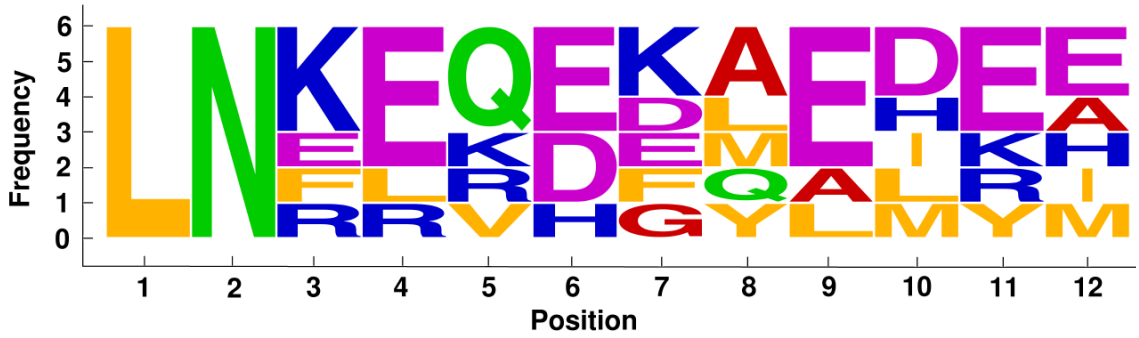


Figure 5.7: Sequence logo of mapped residues in the best alignment of the 6 Ferritin proteins

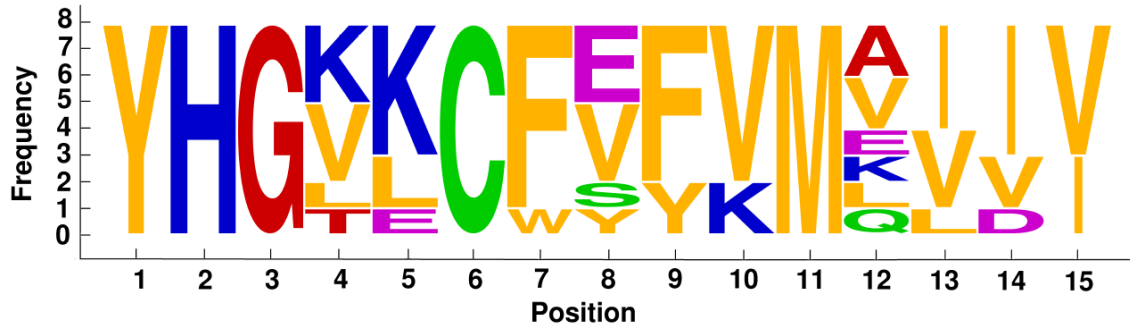


Figure 5.8: Sequence logo of mapped residues in the best alignment of the 8 Plastocyanin proteins

mapped residues. This is an example confirming that protein structural similarity and protein sequence similarity are not always related.

Next, we investigated the effects of varying PROPOSAL parameters. The default values are $N = 6$, $w = 15$, $\alpha = 0.05$, and $IterRefine = 10$.

First, we analysed how parameters influence the running time (Figure 5.10) by varying one parameter and leaving the rest unchanged. Figure 5.10 (a) depicts the running time varying the number N of structures. Figure 5.10 (b) deals with the effect of varying w from 1 to 20. Figure 5.10 (c) reports the PROPOSAL behaviour with α ranging from 0.01 to 0.30. Finally, in Figure 5.10 (d) different values of $IterRefine$ (from 1 to 30) are considered. As expected, when N and w grow and *alpha* decreases, the running time goes up. Such a trend is even more evident in the TIM Barrel family which has the highest average protein sequence length and similarity.

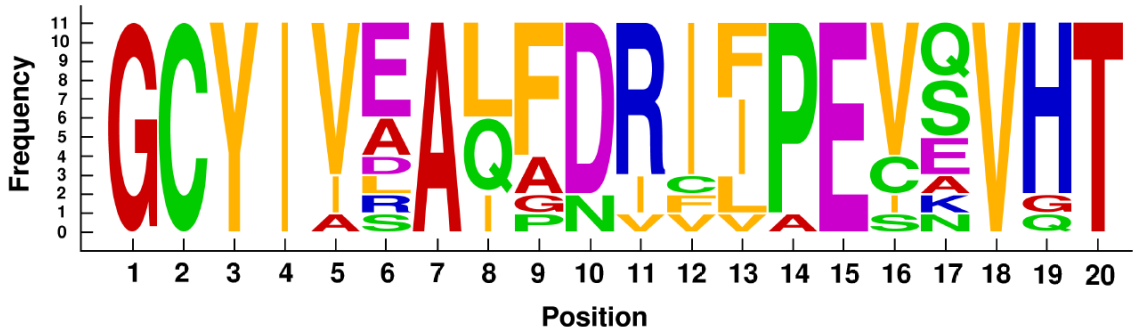


Figure 5.9: Sequence logo of mapped residues in the best alignment of the 11 TIM Barrel proteins

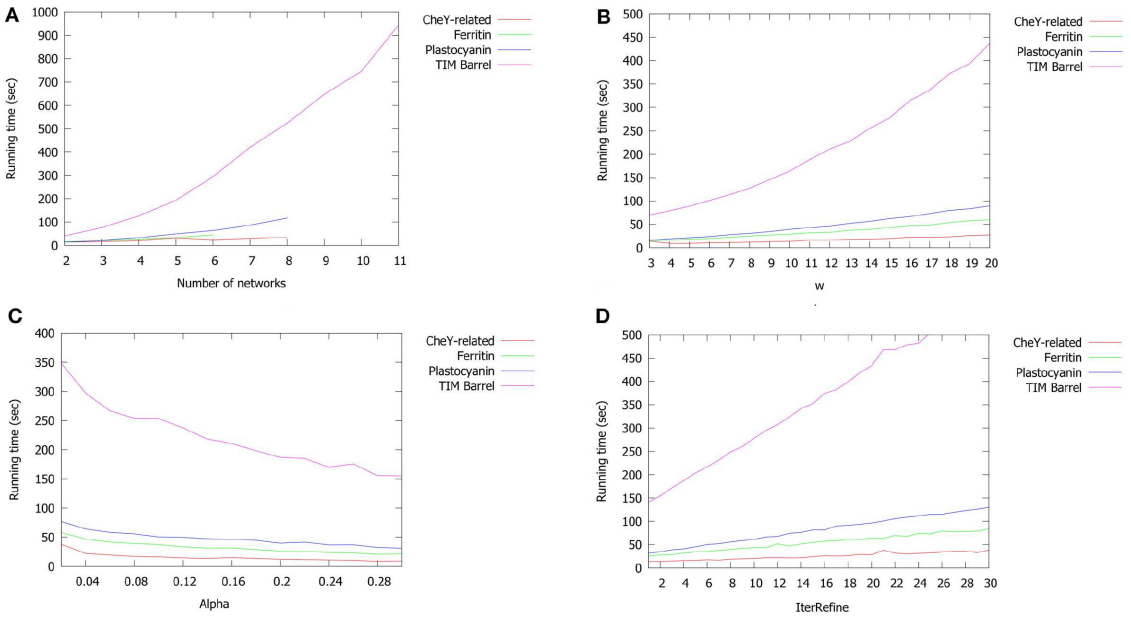


Figure 5.10: Running time of PROPOSAL as a function of a) number of proteins (N); b) w; c) α; d) iterRefine. Default values: $N = 6$, $w = 15$, $\alpha = 0.05$, $IterRefine = 10$

Fig. 5.11 shows the influence of α and $IterRefine$ on the global accuracy of PROPOSAL. We measured the average RMSD over all the computed alignments. In Fig. 5.11 (a) *alpha* varies from 0.01 to 0.30 and $IterRefine$ is set to 10, while in Fig. 5.11 (b) $iterRefine$ varies from 1 to 30 and α is set to 0.05. Default values ($w = 15$ and $N = 6$) were assigned. As expected, the best performance of our method are obtained with low values of α and high values of $IterRefine$. However, if we also consider the influence of such parameters on running time (in particular the $IterRefine$ parameter), the best trade-off between speed and accuracy can be achieved with $0.01 \leq \alpha \leq 0.1$ and $IterRefine = 10$.

Tests on pairwise alignments

As far as we are concerned, PROPOSAL is the first algorithm proposed for multiple local alignments of protein structures. On the other hand a few existing tools can solve the pairwise local structure alignment problem [61, 174, 79]. According to the experiment results reported in [79] and [110], ProBiS and SMAP seem to be the best existing pairwise local structure alignment methods.

In order to compare PROPOSAL with ProBiS and SMAP, we run all the algorithms on a

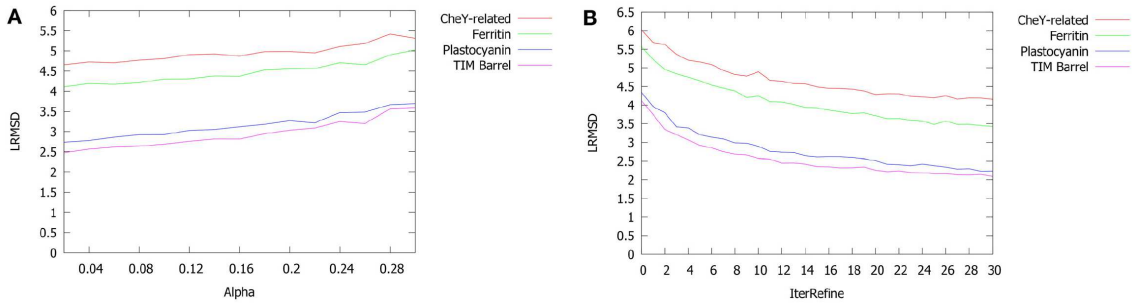


Figure 5.11: Average LRMSD of the alignments returned by PROPOSAL on varying a) α and b) IterRefine. Default values: N=6, $w = 15$, $\alpha = 0.05$, $IterRefine = 10$

properly defined dataset of pairwise alignments.

First of all, we collected a set of 346 non-redundant literature derived small query motifs (having 4-6 residues), taken from CSA (Catalytic Site Atlas) [47]. CSA is a database of hand-annotated entries, containing enzyme active sites (i.e. a set of residues thought to be directly involved in the reaction catalysed by an enzyme).

Then, we used LabelHash, which is the state-of-the-art tool for substructure matching, to search for a match between each query motif and the rest of the dataset. Finally we selected all matches with $RMSD \leq 1.5\,\mathring{A}$. This resulted in a final reference dataset of 6380 pairwise alignments.

The dataset has many highly dissimilar pairs of proteins. In order to analyse the sequence similarity between the 6380 couples of proteins with the lowest RMSD alignments, we run BLAST and considered the percentage of residues with positive matches in the shortest sequence. We call $PPos$ the latter measure. Among the 6380 couples, 3835 ($\simeq 60\%$) have $PPos < 5\%$ and 6173 ($\simeq 97\%$) have $PPos < 15\%$.

For each couple, we run PROPOSAL with no overlapping filter ($AvgOverlap = 100\%$) and w equals to the number of residues of the query motif. We ran SMAP and ProBiS with default parameter values.

We analysed the performance of the three methods on the 6380 pairwise alignments, by taking into account three parameters:

- Query motif coverage (QMC): the highest percentage of residues of the query motif which are present in an alignment returned by each algorithm;
- RMSD of the alignment with highest QMC;
- Running time;

We measured the average values of these parameters by considering different ranges of $PPos$ similarities. All results are plotted in Fig. 5.12.

PROPOSAL exhibits the highest QMC for highly dissimilar proteins, while for medium and high $PPos$ similarities ProBiS is the best method (Fig. 5.12a). However, in all the tested instances PROPOSAL yields the lowest average RMSD with respect to both ProBiS and SMAP. Furthermore, the difference between RMSDs tends to increase as long as $PPos$ decreases (Fig. 5.12 b). We also notice that the average QMC and RMSD of PROPOSAL alignments are approximately constant for all values of $PPos$, while ProBiS and SMAP seem to be quite sensitive to protein similarity.

Finally, ProBiS is by far the fastest algorithm for all possible ranges of $PPos$ similarity values (Fig. 5.12c), while PROPOSAL and SMAP have similar running times (except for $80\% \leq PPos \leq 100\%$, where SMAP is faster). It is worth noting that our method has been designed for solving the multiple alignment problem, while ProBiS and SMAP have been efficiently implemented for comparing pairs of protein structures. Moreover, PROPOSAL and SMAP have been implemented in Java, while ProBiS has been written in C++. Interestingly, our method is faster when $PPos$

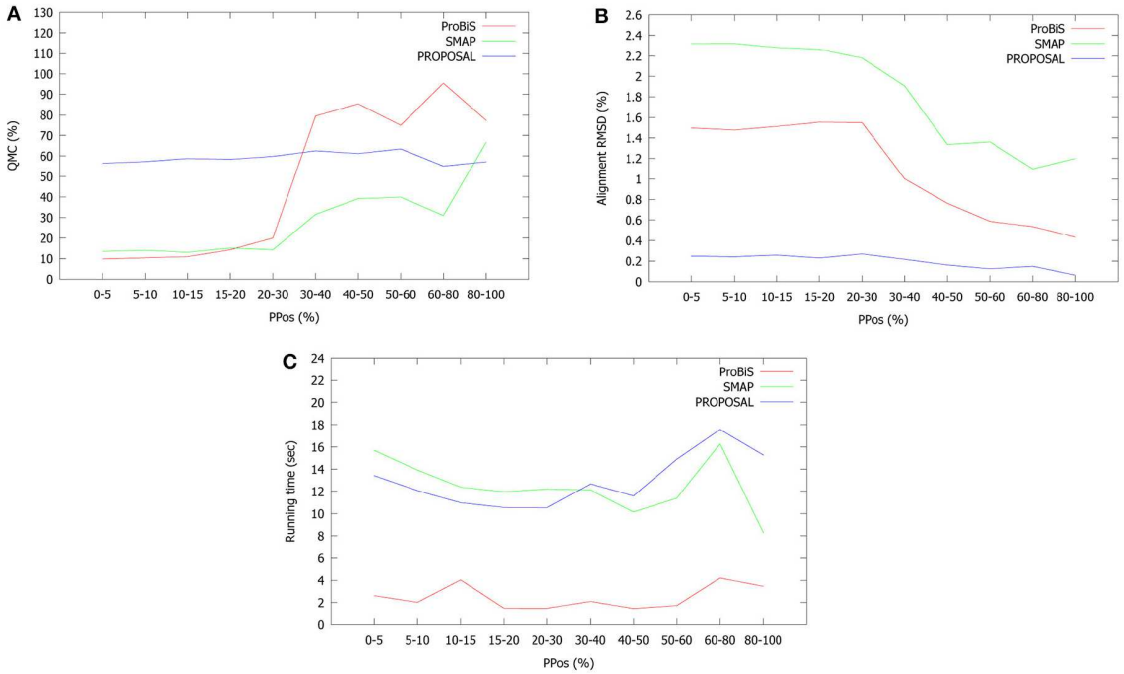


Figure 5.12: Average a) highest QMC, b) corresponding alignment RMSD and c) running time of PROPOSAL, ProBiS and SMAP for different ranges of $PPos$ similarity values

ranges from 10% to 30%. However, when proteins are very dissimilar, the convergence of Gibbs sampling in the bootstrap phase may be slower. On the other hand, when proteins are very similar PROPOSAL performs more extension and refinement phases, producing more feasible alignments. A similar trend holds for ProBiS, where the best performance is obtained when $PPos$ ranges from 15% to 60%.

Tests on multiple alignments

In the last case study, we run PROPOSAL on a different set of 172 motifs taken from CSA to test the capability of our method to detect known conserved binding sites in the multiple case.

The dataset has been built by selecting literature derived motifs of proteins belonging to fully qualified EC classes with at most 25 elements. This resulted in a final set of 172 motifs, spanning 162 distinct EC classes. EC class [41] is a code having the format "EC" followed by four numbers separated by periods. It denotes the type of reaction catalyzed by an enzyme. An EC class is fully qualified if all four numbers are specified (e.g. 1.1.1.149 is fully qualified, while 1.1.1 or 1.1 are not).

For each EC family, we run PROPOSAL on the set of protein structures belonging to that family. We fixed w equals to the number of residues in the corresponding motif and $AvgOverlap = 100\%$ (i.e. no overlapping filter). The remaining parameters were set up to the default values.

We filtered out all alignments with average RMSD above 1 $\AA$, taking for each query motif the local alignment with maximum QMC. In case of ties on QMC, the alignment with minimum average RMSD was chosen. PROPOSAL successfully completed all the alignments in about 29 hours, with an average QMC of 50.08% and average running time of 10 minutes. In Table 5.3 we report motifs with highest QMC and the RMSD of the corresponding alignment. The complete list of results is available as supplementary material of [106]. Results clearly show the ability to identify known motifs from scratch. Out of 172 motifs, 24 have QMC $\geq 75\%$ and 126 have specificity $\geq 50\%$.

In [159, 110], authors observed that the EC-class coverage of a motif has not been considered

Table 5.3: CSA motifs with QMC $\geq$ 75%. Each motif is represented as a list of residue ids of the corresponding reference protein.

PROTEIN	EC_CLASS	MOTIF	QMC	AVG_RMSD
1YBV	1.1.1.252	[138, 182, 164, 178]	100%	0.06814209
1QRR	3.13.1.1	[183, 186, 145, 182]	75%	0.032653827
1MRQ	1.1.1.149	[50, 117, 84, 55]	75%	0.063025678
1GQ8	3.1.1.11	[136, 157, 113, 135]	75%	0.075239285
2JXR	3.4.23.25	[215, 32, 218, 33]	75%	0.088753575
1RK2	2.7.1.15	[252, 253, 255, 254]	75%	0.092735469
2PGD	1.1.1.44	[187, 190, 130, 183]	75%	0.119390475
1VAS	3.1.25.1	[22, 26, 23, 2]	75%	0.126902935
1CZF	3.2.1.15	[180, 201, 202, 223]	75%	0.15027138
1PJB	1.4.1.1	[269, 117, 95, 74]	75%	0.178174017
1RPX	5.1.3.1	[185, 43, 41, 74]	75%	0.222043962
1L1L	1.17.4.2	[119, 408, 419, 410]	75%	0.226235418
1DB3	4.2.1.47	[134, 160, 132, 156]	75%	0.252066199
1IM5	3.5.1.19	[129, 10, 133, 94]	75%	0.294630848
1ODT	3.1.1.41	[181, 269, 182, 298]	75%	0.40228864
1PVD	4.1.1.1	[28, 477, 114, 115]	75%	0.454688806
1U5U	4.2.1.92	[137, 67, 66, 193]	75%	0.613085033
1E94	3.4.25.2	[45, 33, 124, 1]	75%	0.617527201
1Z9H	5.3.99.3	[110, 113, 112, 107]	75%	0.677534589
1B66	4.2.3.12	[88, 42, 133, 89]	75%	0.7033398
2NAC	1.2.1.2	[284, 146, 313, 332]	75%	0.78435381
1QTN	3.4.22.61	[258, 360, 350, 317]	75%	0.798793943
1P4R	2.1.2.3	[431, 267, 592, 266]	75%	0.936798151
1BWZ	5.1.1.7	[217, 73, 208, 159]	75%	0.941959894

for the design of CSA. Consequently, some motifs may be not conserved across all proteins in an EC class. This may be the origin of failures of PROPOSAL on the alignment tasks with QMC $<$ 50%). In some cases CSA motifs could contain one or more residues with few global matches. Moreover, two motifs could match mutually exclusive sets of proteins within the corresponding EC class. These cases may cause a drastic increase of average RMSD for that specific motif. Examples of such CSA motifs are reported in [110]. In order to overcome these problems, methods like Geometric Sieving [24] can be applied to refine a given motif and increase sensitivity while keeping high specificity values.

5.1.4 A Java 2D application

We developed a Java 2D standalone application for multiple local alignment of protein structures, based on PROPOSAL algorithm [106]. The application present a user-friendly interface for the visualization of protein structures and multiple alignments in 3D, using the JMol framework (http://jmol.sourceforge.net). Additionally, users can export the alignments in .sif format as 2D contact maps alignment, for the visualization within Cytoscape [140].

Fig. 5.13 shows the main window of PROPOSAL. It is divided into five panels:

- "Structures list" (Fig. 5.14), containing the list of aligning structures in .pdb format and enables the visualization of one or more structures in 3D using JMol;
- "Log panel", for logging all the operations performed by the tool;
- "Alignment list" (Fig. 5.16, center left panel), containing the list of all local alignments found by PROPOSAL;

- "Alignment details" (Fig. 5.16, center right panel), where the user can obtain more details about a specific alignment, i.e. the aligned residues for each structure and the final mapping;
- "Parameters", for setting the parameters of the algorithm and controlling the progress of the alignment task.

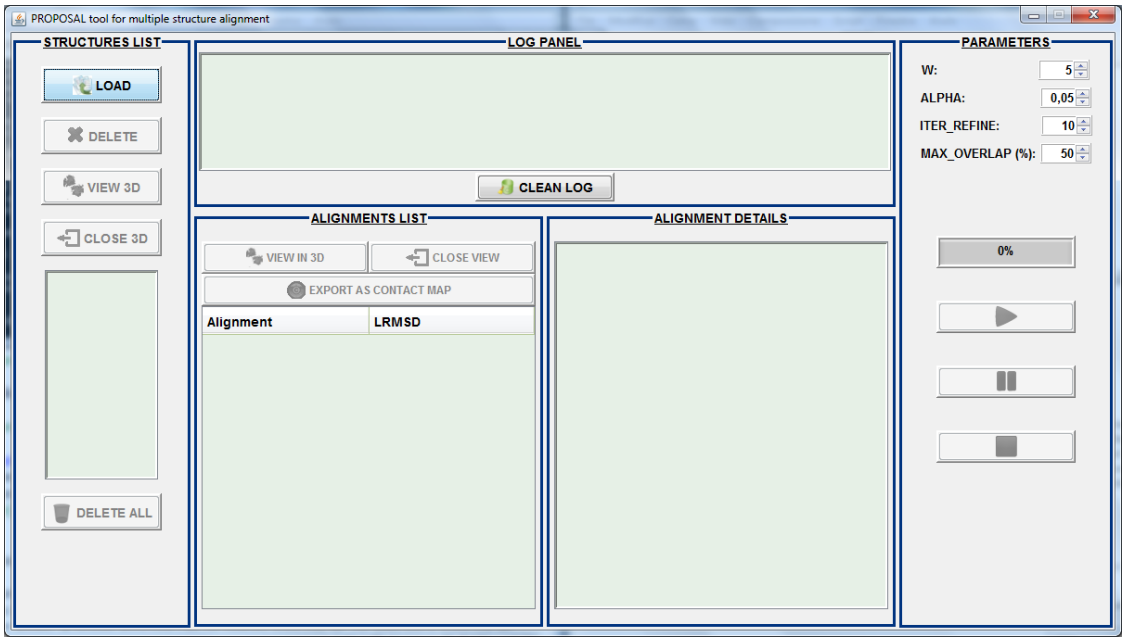


Figure 5.13: General view of PROPOSAL main frame

Loading protein structures

The first mandatory step to run PROPOSAL application is to upload one or more 3D protein structures in Protein Data Bank (PDB) format from local files. The Protein Data Bank (PDB) format [15] is a standard representation of macromolecular structure data derived from X-ray diffraction and NMR studies and it is used by many algorithms related to 3D protein structure analysis. Each protein structure in PDB format is identified by a unique code with 4 letters (e.g. 1PLA, 2B3I).

Structures can be uploaded by clicking on the "Load" button of "Structures List" panel (Fig. 5.14a). A protein structure can be deleted from the list, by selecting it from the list and clicking on "Delete" button. "Delete all" button removes all loaded structures from the input list.

3D visualization of PDB structures

Loaded PDB structures can be visualized in 3D using JMol, by selecting one or more proteins from the structure list and clicking on "View 3D" button. JMol (http://www.jmol.org) is an interactive Java applet for the visualization of chemical structures, which can be rotated and zoomed.

If at least two structures have been selected, the user can choose among two kinds of visualizations:

- "In separate windows" (Fig. 5.14b): selected PDB structures are visualized in different JMol panels;
- "In a single window" (Fig. 5.15): selected PDB structures will be visualized in a single frame, following a grid layout of visualization, with three JMol panels for each row.

Different styles and color schemes can be chosen for the visualization of PDB structures. The "Cartoons" style and the "Amino" color scheme are selected by default. Clicking on "Close 3D", all currently opened 3D windows will be closed.

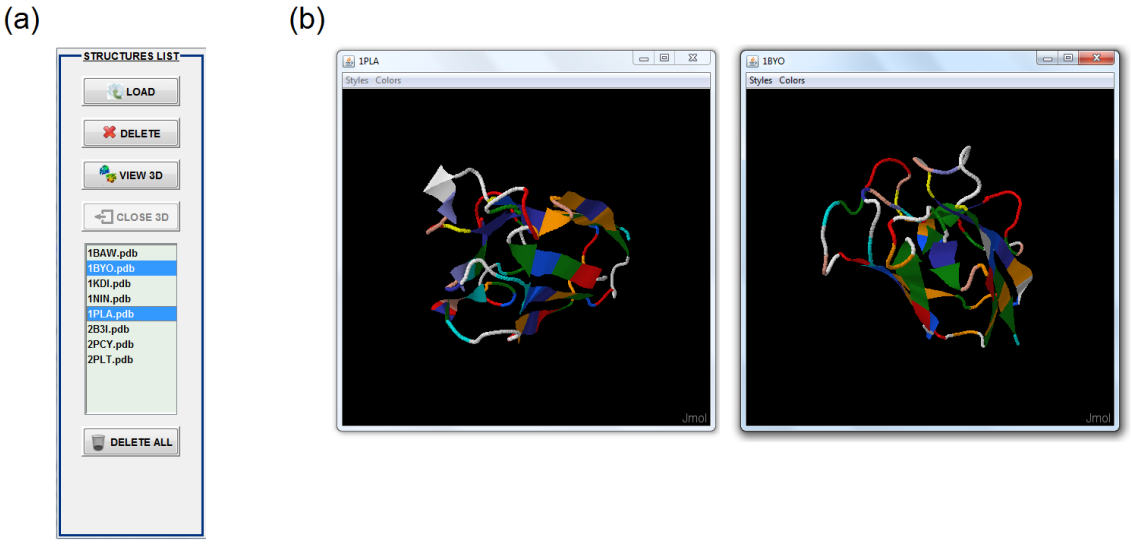


Figure 5.14: "Structures list" panel in PROPOSAL Java application: (a) 8 PDB structures are uploaded from local files; (b) Selected PDB structures in (a) (1BYO and 1PLA) are viewed in 3D in separate windows

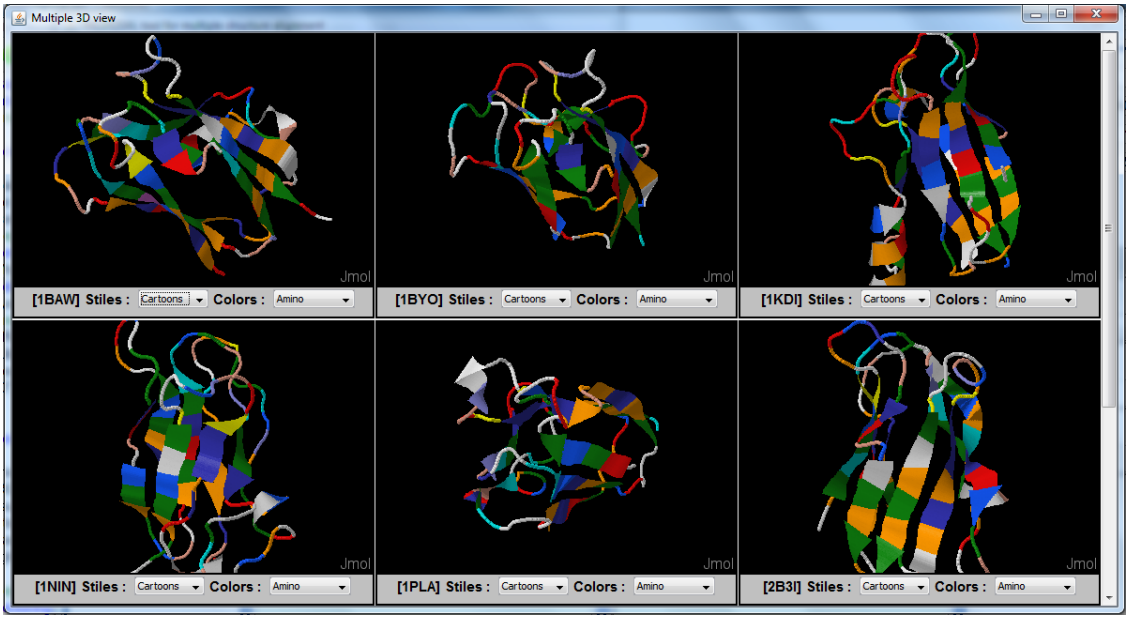


Figure 5.15: Visualization of 6 of the 8 PDB structures uploaded in the "Structures List" panel of Fig. 5.14a) in a single window.

Setting input parameters

The input parameters for PROPOSAL (w, α, $IterRefine$, $AvgOverlap$) are specified in the "Parameters" panel (Fig. 5.13).

Whenever the mouse arrow hovers the name of a parameter, a tool tip string appears, suggesting its meaning. Default values are initially assigned to each parameter, according to the experimental results (Subsection 5.1.3). Lower values of *alpha* and higher values of $IterRefine$ can be set to increase accuracy, but the algorithm could become slower.

Running proposal

After loading PDB structures and setting the input parameters, the user can run PROPOSAL by clicking on the "Play" button, below the progress bar of "Parameters" panel (Fig. 5.13). In any time, the alignment task can be paused, resumed or canceled by clicking on "Pause", "Play" and "Stop" button, respectively. The progress bar shows the advancement of the alignment task. Other information about the state of the process can be visualized in the "Log Panel" (Fig. 5.13).

Alignment visualization

When PROPOSAL ends, the final list of local alignments found by the algorithm will be visualized in the "Alignments list" panel (Fig. 5.16). For each alignment, the corresponding average LRMSD across all pairs of aligned structures is reported. By selecting an alignment from the list, more info about the alignment will be visualized in the "Alignment details" panel (Fig. 5.16). Details include the list of all aligned residues and their 3D coordinates for each structure, and the final mapping of the aligment, represented as a matrix where each column contains all residues of a protein structure and each row contains the aligned residues.

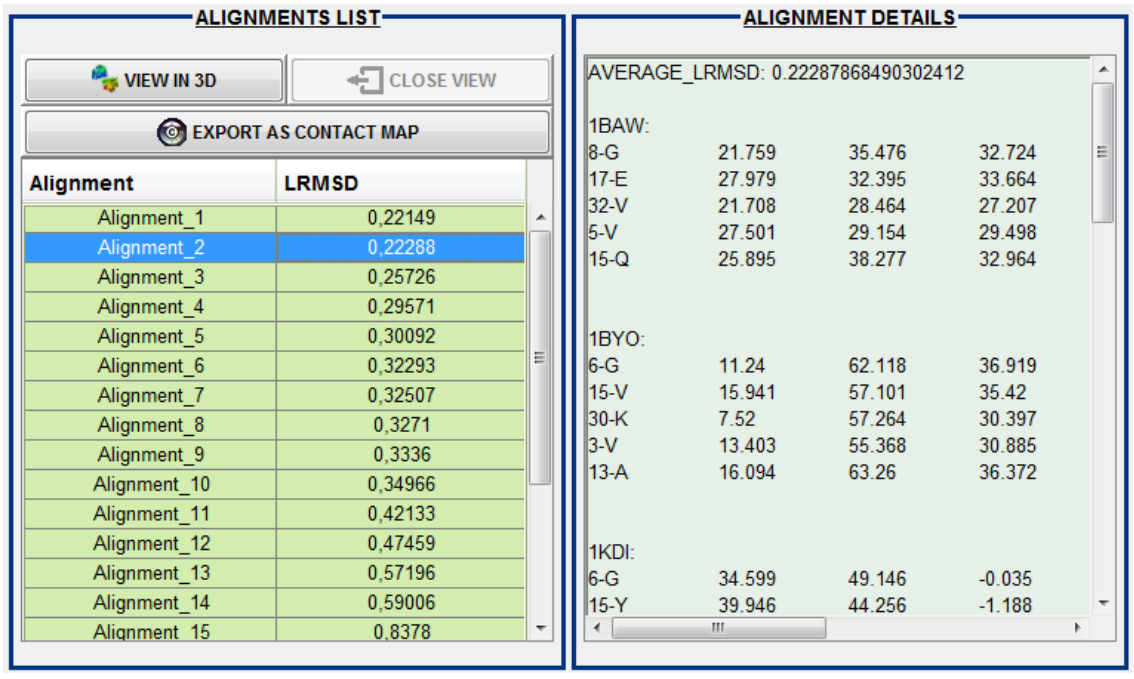


Figure 5.16: List of local alignments for the protein structures listed in Fig. 5.14a). PROPOSAL has been run with default parameter values. If an alignment is selected from the list, details are visualized in the "Alignment details" panel.

Alignments can be further analyzed for visualization. Two kinds of visualization are available:

a) As a 2D contact maps alignment, by clicking on "Export as contact map" (Fig. 5.16);

b) As a 3D structure alignment in JMol, by clicking on "View in 3D" (Fig. 5.16).

In the first case, an alignment graph will be created. The alignment graph is a graph where nodes are all aligned residues and two classes of edges are present:

- Intra-edges, connecting aligned residues of the same protein, that are at distance at most $10\mathring{A}$ one another;
- Inter-edges, connecting aligned residues of different proteins that match one another in the local alignment;

The alignment graph will be then saved in ".sif" format for the visualization on Cytoscape. Additionally, an attribute file for node labels will be created, specifying for each aligned residue the corresponding symbol (in one letter code). The node attribute file will be saved with the name of the corresponding alignment graph, followed by the "_attr" suffix.

In 3D alignment visualization mode, the structures are represented in 3D using JMol (Fig. 5.17). The aligned residues are emphasyzed with different colors and labeled with the corresponding symbol (as a three letter code).

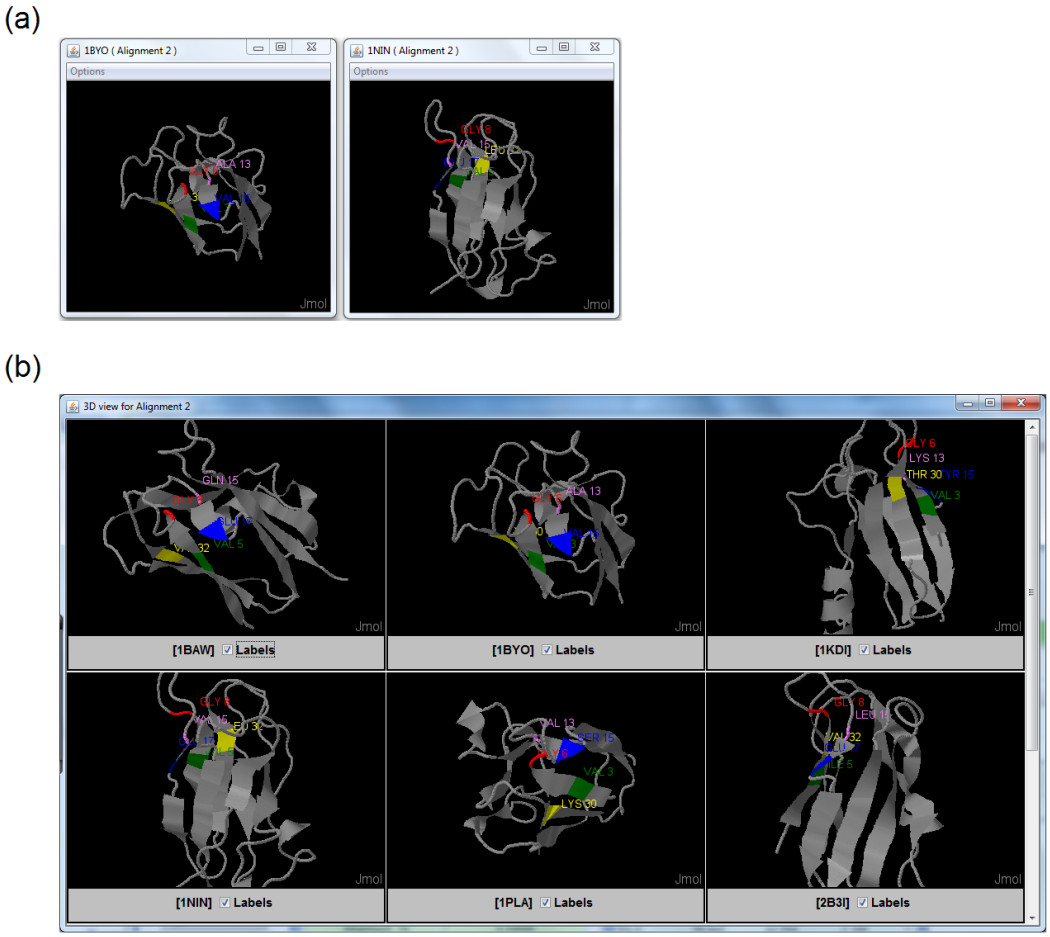


Figure 5.17: Visualization of the structural local alignment selected in Fig. 5.16: a) Visualization of the aligned residues of PDB structures 1BYO and 1NIN in separate windows; b) Visualization of the aligned residues of PDB structures 1BAW, 1BYO, 1KDI, 1NIN, 1PLA and 2B3I in a unique window

Two different kinds of 3D visualization are available:

- "In separate windows" (Fig. 5.17a): PDB structures will be visualized in different JMol panels;
- "In a single window" (Fig. 5.17b): PDB structures will be visualized in a single frame, in a grid layout manner, with three JMol panels for each row.

Structures will be visualized in the "Cartoons" style. Clicking on "Close view" in the "Alignment list" panel Fig. 5.16, all currently opened 3D windows will be closed.

Chapter 6

Conclusion

In this thesis we presented novel algorithms based on Gibbs sampling for network and structure comparison. This is first time that Gibbs sampling has been proposed for comparing structured data.

We applied it for investigating similarities across proteins at two different levels of resolution: the structural one (PROPOSAL) and the more global interaction level (GASOLINE). In both cases, the final goal is to find local patterns, which can give clues to the function of unknown proteins. At the interaction level, comparison helps to identify highly conserved protein complexes in distant species, such as the proteasome and the spliceosome, and understand the evolution of some ancestral processes. Moreover, it can be applied to transfer functional information (such as Gene Ontology terms) from a protein of a well known organism (e.g. yeast) to a protein of a more complex species (e.g. human). GASOLINE outperforms state-of-art systems such as SMETANA, NetworkBlast-M and IsoRank-N and is capable to produce more accurate results than the other methods on both synthetic and real networks. PROPOSAL is the first algorithm for multiple local alignment of 3D protein structures. It can both identify new motifs and refine existing ones with great accuracy.

We also demonstrated that the integration of protein-protein interaction networks with gene expression data for healthy and cancer tissues can help to highlight (i) the role of genes in some diseases and (ii) how a change in the expression of a gene from healthy to disease tissues (e.g. from normal to cancer breast tissues) can influence its interacting partners (SPECTRA). SPECTRA overcomes the current PPI network analysis limitations, representing a unique framework to integrate protein interactions and gene expressions from various datasets. Moreover, it provides for the first time an algorithm to compare tissue and tumor specific PPI networks, in order to identify subnetworks of differentialy-expressed genes.

The non-deterministic algorithms proposed in the thesis are very scalable, with respect to both the size and the number of comparing networks and structures. They are also more accurate than deterministic methods in most cases.

The methodology presented in the thesis is very attractive since it can be applied to identify local similar patterns in any set of complex systems. The strength of the proposed methodology, based on iterative-sampling, relies on the fact that its applicability is domain-independent. Therefore, we have an algorithm which works on *template-data* where entities can be nodes of a graph or, more in general, points in any multidimensional space. Attributes can be labels, coordinates or any kind of features and can be associated to both entities and interactions. To apply the algorithm we only need to define a proper scoring functions capable to evaluate similarities between entities.

6.1 Future developments

The methodology presented in the thesis can further investigated, improved and extended. Our future research will follow three different directions:

1. Extending Gibbs sampling to new applications;
2. Parallelization of Gibbs sampling for analyzing big data;
3. Refining and improving the algorithms presented here.

6.1.1 New potential applications

Global alignment is a natural extension of our Gibbs sampling strategy. It implies the design of new scoring schemes, defined upon global topological features of nodes or entities. Similarity between two entities must be defined from a new perspective, where the behaviour of a single entity is analyzed with respect to the whole system and not just the set of its neighbours. Very recently, new features, such as graphlets and graphlet degree distances, have been proposed to investigate the global behaviour of a node up to a certain distance [83, 107, 84, 98, 27, 96]. Integrating these global features can be very useful, especially when we have no information about the semantic of a node (e.g. label, id, ontologies).

Time-evolving networks have recently received great attention as a powerful model to describe the dynamics of a system just in terms of changes and spread of information through the network [63]. They contain an additional dimension, the time, which is linked to edges and nodes, to indicate when a node or an interaction is active. Processes that can be modeled as temporal networks include, for instance, time-course of gene expressions, disease progression and evolving relationships in social networks. Gibbs sampling could be applied to the analysis of any kind of temporal networks, by looking to each time stamp as a separate graph and finding conserved subgraphs across the different networks, such as biological complexes or social communities which evolve in time. However, in this case, scoring function should be relaxed, in order to consider also small variations, involving label changes, insertions and removals of few nodes and/or edges around a conservative pattern. This is a slight variant of local alignment problem which can be applied to anomaly detection in various domains (biology, physics, security, etc.).

Another possible application of our Gibbs sampling strategy is graph compression. We could use network alignment to find recurrent pattern of interactions within the same network and compress the graph replacing each occurrence of the pattern with a metanode. By iterating this process, we can finally obtain a compact representation of the network. Recurring patterns can be exact or approximated with few nodes and/or edges missing, allowing more flexibility.

6.1.2 Capability to deal with big data

Gibbs sampling is suitable for parallelization at different levels. This can bring a great speedup of our methods on large networks with millions or billions of nodes and edges.

A basic level of parallelization can be performed within a single machine. Indeed, the executions of GASOLINE and PROPOSAL are independent from one another and can be executed in parallel.

Parallelization involving a cluster of machines can be achieved using frameworks for distributed computations, such as Hadoop. For instance, these technologies can be used in connection with Gibbs sampling to count the occurrences of a (possibly approximated) recurrent pattern. One approach would be partitioning the network in smaller subgraphs and running "Map" processes on each partition in different machines in order to count occurrences. "Reduce" machines can combine results to yield the final count.

6.1.3 Refinement of proposed algorithms

Some future developments of the three methodologies presented in the thesis are synthesized below.

Mining protein interaction maps. GASOLINE performs local alignment of PPI networks producing a 1-to-1 mapping between homologous proteins of different species. However, some proteins can have one or more (possibly diverging) copies within the same species. These copies are the results of evolutionary events, such as gene duplications and mutations. This fact motivates the

need to extend GASOLINE to produce many-to-many mappings, or a one-to-one mapping between equivalence classes of proteins, where each class contains homologous proteins of the same organism.

Another improvement concerns the filtering of overlapping alignments. Biological networks tend to have high modularity [9], with modules that cooperate one another to realize more complex biological processes. This implies that two or more complexes can be parts of a bigger complex. In the actual implementation of GASOLINE highly overlapping complexes are filtered in order to favour the biggest complexes, so some small complexes involved in more specific processes could be missed.

GASOLINE Cytoscape app can be extended in its functionalities and integrated with other sources of data. In particular, alignments could be enriched by including cross references, related diseases or pathways for aligned nodes. This part is intended to replace the current manual loading of Gene Ontology data for the analysis of results.

SPECTRA is the first proposed general framework to build and compare TS-PPI networks of different tissues and tumors. It can be extended to include more data as well more functionalities.

We aim to add protein expression data in SPECTRA in order to offer a more complete view of the modifications involving genes and proteins within tissues. In fact, the concentration of a gene and the corresponding protein in a specific tissue can be different due to the action of miRNA and other non coding RNA molecules on the mRNA of the gene, which can repress or stimulate the production of the protein. Moreover, a protein can have one or more splicing variants. Isoforms are currently not considered in SPECTRA but we plan to include them by taking into account protein expressions. As a result, the adapted GASOLINE will be extended to enable the comparison of TS-PPI networks built from protein and gene expression data in the same tissue or in distinct tissues.

Networks could be extended by including non coding RNAs and their interactions with proteins to better elucidate their role in specific tumors.

Finally, we want to refine the confidence score of edges of the integrated PPI network in SPECTRA. The current scoring function takes into account the number of datasets reporting an interaction (dataset coverage) but it is biased by the lack of reliability scores of the interaction for some databases reporting it. We aim to design a new scoring function including information about the experimental methods used to detect the interaction and the number of publications reporting that interaction.

Mining protein structures. The current version of PROPOSAL finds local alignments of a fixed size w, where w is a user-defined parameter. Actually, whenever the user performs local alignment of protein structures, he does not know neither whether exists a potential common binding site nor how big the latter could be. So, we could remove w or make it an optional parameter, using a threshold on maximum LRMSD as an alternative value to control the extension process.

We also aim to design a modified version of PROPOSAL for protein docking in order to predict possible sites of interactions between two proteins. This should be done considering only 3D structural data, with no assumptions about the location of the interaction. The docking algorithm should take into account complementarities between interacting residues and structural correspondences of compared binding sites and integrate such data in the similarity scoring function. Due to the generality of our method, a more ambitious goal could be to extend this approach to any kind of molecular docking (e.g. RNA-proteins, RNA-RNA). Of course, the last scenario requires the design of a more complex scoring scheme.

Bibliography

[1] B. Adamcsek, G. Palla, I.J. Farkas, I. Derenyi, and T. Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

[2] P.K. Agarwal, N.H. Mustafa, and Y. Wang. Fast molecular shape matching using contact maps. *Journal of Computational Biology*, 14(2):131–143, 2007.

[3] S. Alaimo, A. Pulvirenti, R. Giugno, and A. Ferro. Drugtarget interaction prediction through domain-tuned network-based inference. *Bioinformatics*, 29(16):2004–2008, 2013.

[4] F. Alkan and C. Erten. BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 30(4):531–539, 2013.

[5] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[6] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. *Annual IEEE Symposium on Foundations of Computer Science (FOCS '06)*, pages 475–486, 2006.

[7] R. Andonov, N. Yanev, and N. Malod-Dognin. An efficient lagrangian relaxation for the contact map overlap problem. *Proc. WABI 08*, pages 162–173, 2008.

[8] S. Angaran, M. E. Bock, C. Garutti, and C. Guerra. Molloc: a web tool for the local structural alignment of molecular surfaces. *Nucleic Acids Research*, 37(2):W565–W570, 2009.

[9] A. Barabasi and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2004.

[10] T. Barrett, S.E. Wilhite, P. Ledoux, and C. et al. Evangelista. Ncbi geo: archive for functional genomics data setsupdate. *Nucleic Acids Research*, 41(D1):D991–D995, 2013.

[11] R. Barshir, O. Basha, A. Eluk, I.Y. Smoly, A. Lan, and E. Yeger-Lotem. The tissuenet database of human tissue proteinprotein interactions. *Nucleic Acids Research*, 41(D1):D841–D844, 2013.

[12] R. Barshir, O. Shwartz, I.Y. Smoly, and E. Yeger-Lotem. Comparative analysis of human tissue interactomes reveals factors leading to tissue-specific manifestation of hereditary diseases. *PLoS Computational Biology*, 10(6), 2014.

[13] J. Bartek, M. Petrek, B. Vojtesek, J. Bartkova, J. Kovarik, and A. Rejthar. Hla-dr antigens on differentiating human mammary gland epithelium and breast tumours. *British Journal of Cancer*, 56(6):727–733, 1987.

[14] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[15] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, and T.N. et al. Bhat. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

[16] B.M. Bolstad, R.A. Irizarry, M. Astrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[17] A. Bossi and B. Lehner. Tissue specificity and the human protein interaction network. *Molecular Systems Biology*, 5(260), 2009.

[18] S. Brohèe and J.V. Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7(488), 2006.

[19] L. Cabusora, E. Sutton, A. Fulmer, and C.V. Forst. Differential network expression during drug and stress response. *Bioinformatics*, 21(12):2898–2905, 2005.

[20] MD) Center for Medical Genetics, Johns Hopkins University (Baltimore and MD) National Center for Biotechnology Information, National Library of Medicine (Bethesda. Online mendelian inheritance in man, omim (tm). 1996.

[21] E. Cerami, J. Gao, U. Dogrusoz, B.E. Gross, and S.O. et al. Sumer. The cbio cancer genomics portal: An open platform for exploring multidimensional cancer genomics data. *Cancer Discovery*, 2(401), 2012.

[22] V. Cerny. Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.

[23] B. Chen, J. Wang, M. Li, and F. Wu. Identifying disease genes by integrating multiple data sources. *BMC Medical Genomics*, 7(S2), 2014.

[24] B. Y. Chen, V. Y. Fofanov, D. H. Bryant, B. D. Dodson, D. M. Kristensen, A. M. Lisewski, M. Kimmel, O. Lichtarge, and L. E. Kavraki. The mash pipeline for protein function prediction and an algorithm for the geometric refinement of 3d motifs. *Journal of Computational Biology*, 14(6):791–816, 2007.

[25] G. Chen and J. Wang. Identifying functional modules in tissue specific protein interaction network. *Bioinformatics and Biomedicine Workshops (BIBMW), 2012*, pages 581–586, 2012.

[26] G. Ciriello, M. Mina, P.H. Guzzi, M. Cannataro, and C. Guerra. Alignnemo: A local network alignment method to integrate homology and topology. *PLoS ONE*, 7(6), 2012.

[27] J. Crawford and T. Milenkovic. GREAT: Graphlet edge-based network alignment. 2014.

[28] G. E. Crooks, G. Hon, J. Chandonia, and S. E. Brenner. Weblogo: a sequence logo generator. *Genome Research*, 14:1188–1190, 2004.

[29] P. Csermerly, T. Korcsmaros, H.J. Kiss, G. London, and R. Nussinov. Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review. *Pharmacology and Therapeutics*, 138(3):333–408, 2013.

[30] G.B. Da Silva, T.G. Silva, R.A. Duarte, N.L. Neto, and H.H. et al. Carrara. Expression of the classical and nonclassical hla molecules in breast cancer. *International Journal of Breast Cancer*, 2013.

[31] P. Dao, K. Wang, C. Collins, M. Ester, A. Lapuk, and S.C. Sahinalp. Optimally discriminative subnetwork markers predict response to chemotherapy. *Bioinformatics*, 27(13):i205–i213, 2011.

[32] D. De Martino, M. Figluzzi, A. De Martino, and E. Marinari. A scalable algorithm to explore the gibbs energy landscape of genome-scale metabolic networks. *PLoS Computational Biology*, 8(6), 2012.

[33] M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. *Journal of Computational Biology*, 11(2):463–475, 2004.

[34] M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology*, 10(6):947–960, 2003.

[35] C. Desmedt, F. Piette, S. Loi, Y. Wang, and F. et al. Lallemand. Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series. *Clinical Cancer Research*, 13(11):3207–3214, 2007.

[36] Z. Dezso, Y. Nikolski, E. Sviridov, W. Shi, and T. et al. Serebriyskaya. A comprehensive functional analysis of tissue specificity of human gene expression. *BMC Biology*, 6(49), 2008.

[37] P. Di Lena, P. Fariselli, L. Margara, M. Vassura, and R. Casadio. Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, 26(18):2250–2258, 2010.

[38] M.T. Dittrich, G.W. Klau, A. Rosenwald, T. Dandekar, and T. Muller. Identifying functional modules in proteinprotein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.

[39] S.V. Dongen. Graph clustering by flow simulation. *PhD thesis, University of Utrecht*, 2000.

[40] T. Dwyer, K. Marriott, F. Schreiber, P. Stuckey, M. Woodward, and M. Wybrow. Exploration of networks using overview+detail with constraint-based cooperative layout. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1293–1300, 2008.

[41] Webb E.C. Enzyme nomenclature 1992: recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes. *Enzyme Nomenclature*, 1992.

[42] I. Eidhammer, I. Jonassen, and W. R. Taylor. Algorithmic aspects of protein structure similarity. *40th Annual Symposium on Foundations of Computer Science*, pages 512–521, 1999.

[43] I. Eidhammer, I. Jonassen, and W. R. Taylor. Structure comparison and structure patterns. *Journal of Computational Biology*, 7(5):685–716, 2004.

[44] D. Emig and M. Albrecht. Tissue-specific proteins and functional implications. *Journal of Proteome Research*, 10(4):1893–1903, 2011.

[45] J. Flannick, A. Novak, B.S. Srinivasan, H.H. McAdams, and S. Batzoglou. Grmlin: General and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, 2006.

[46] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, and A. et al. Roth. String v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):D808–D815, 2013.

[47] N. Furnham, G. L. Holliday, T. A. P. De Beer, J. O. B. Jacobsen, W. R. Pearson, and J. M. Thornton. The catalytic site atlas 2.0: cataloging catalytic sites and residues identified in enzymes. *Nucleic Acids Research*, 42(D1):D485–W489, 2013.

[48] J. Gao, B.A. Aksoy, U. Dogrusoz, G. Dresdner, and B. et al. Gross. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science Signaling*, 6(269), 2013.

[49] X. Ge, S. Yamamoto, S. Tsutsumi, Y. Midorikawa, S. Ihara, S. M. Wang, and H. Aburatani. Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. *Genomics*, 86(2):127–141, 2005.

[50] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6(6):721–741, 1984.

[51] R.C. Gentleman, V.J. Carey, D.M. Bates, B. Bolstad, and M. et al. Dettling. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10), 2004.

[52] J. Gibrat, T. Madej, and S.H. Bryant. Surprising similarities in structure comparison. *Current Opinion in Structural Biology*, 6(3):377–385, 1996.

[53] V. Gligorijevic, N. Malod-Dognin, and N. Przulj. FUSE: Multiple network alignment via data fusion. *arXiv:1410.7585 [q-bio.MN]*, 2014.

[54] A. Godzik and J. Skolnick. Flexible algorithm for direct multiple alignment of protein structures and sequences. *Computer applications in the biosciences*, 10(6):587–596, 1994.

[55] J. Goecks, A. Nekrutenko, J. Taylor, and Galaxy Team. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11(8), 2010.

[56] P.J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[57] Y. Guo, Q. Sheng, J. Li, F. Ye, and D.C. et al. Samuels. Large scale comparison of gene expression levels by microarrays and rnaseq using tcga data. *PLoS ONE*, 8(8), 2013.

[58] Z. Guo, Y. Li, X. Gong, C. Yao, and W. et al. Ma. Edge-based scoring and searching method for identifying condition-responsive proteinprotein interaction sub-network. *Bioinformatics*, 23(16):2121–2128, 2007.

[59] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[60] P.C. Havugimana, G.T. Hart, T. Nepusz, H. Yang, A.L. Turinsky, and A.L. et al. Zhihua. A census of human soluble protein complexes. *Cell*, 150(5):1068–1081, 2012.

[61] L. Holm and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, 16(6):566–567, 2000.

[62] L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *Journal of Molecular Biology*, 233(1):123–138, 1993.

[63] P. Holme and J. Saramaki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[64] H. Huang, X. Wu, R. Pandey, J. Li, G. Zhao, S. Ibrahim, and J.Y. Chen. C2maps: a network pharmacology database with comprehensive disease-gene-drug connectivity relationships. *BMC Genomics*, 13(6):S17, 2012.

[65] T. Ideker, O. Ozier, B. Schwikowski, and A.F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(1):S233–S240, 2002.

[66] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8):4569–4574, 2001.

[67] A.V. Ivshina, G. Joshy, O. Senko, and B. et al. Mow. Genetic reclassification of histologic grade delineates new clinical subtypes of breast cancer. *Cancer Research*, 66(21):10292–10301, 2006.

[68] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.

[69] W.E. Johnson and C. Li. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

[70] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, 1976.

[71] M. Kalaev, V. Bafna, and R. Sharan. Fast and accurate alignment of multiple protein networks. *Journal of Computational Biology*, 16(8):989–999, 2009.

[72] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.

[73] A. Kamburov, U. Stelzl, H. Lehrach, and R. Herwig. The consensuspathdb interaction database: 2013 update. *Nucleic Acids Research*, 41(D1):D793–D800, 2013.

[74] K. Kaneko, S. Ishigami, Y. Kijma, Y. Funasako, and M. et al. Hirata. Clinical implication of hla class i expression in breast cancer. *BMC Cancer*, 11(454), 2011.

[75] A.E. Karnoub and R.A. Weinberg. Chemokine networks and breast cancer metastasis. *Breast Disease*, 26(1):75–85, 2007.

[76] B.P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B.R. Stockwell, and T. Ideker. Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Research*, 32(S2):W83–W88, 2004.

[77] W.K. Kim and E.M. Marcotte. Age-dependent evolution of the yeast protein interaction network suggests a limited role of gene duplication and divergence. *PLoS Computational Biology*, 4(11), 2008.

[78] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[79] J. Konc and D. Janezic. Probis algorithm for detection of structurally similar protein binding sites by local structural alignment. *Bioinformatics*, 26(9):1160–1168, 2010.

[80] J. Konc and D. Janezic. Probis-2012: web server and web services for detection of structurally similar binding sites in proteins. *Nucleic Acids Research*, 40(W1):W214–W221, 2012.

[81] 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.

[82] 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(1):2256–2268, 2004.

[83] O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes, and N. Przulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7(50):1341–1354, 2010.

[84] O. Kuchaiev and N. Przulj. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, 27(10):1390–1396, 2011.

[85] K. Lage, N.T. Hansen, E.O. Karlberg, A.C. Eklund, and F.S. et al. Roque. A large-scale analysis of tissue-specific pathology and gene expression of human disease genes and complexes. *PNAS*, 105(52):20870–20875, 2008.

[86] G. Lancia, R. Carr, B. Walenz, and S. Istrail. 101 optimal pdb structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. *Proceedings of the fifth annual international conference on Computational biology (RECOMB '01)*, pages 193–202, 2001.

[87] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[88] S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19(S1):i197–i204, 2003.

[89] C. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.

[90] L. Licata, L. Briganti, D. Peluso, L. Perfetto, and M. et al. Iannuccelli. Mint, the molecular interaction database: 2012 update. *Nucleic Acids Research*, 40(D1):D857–D861, 2012.

[91] J.S. Liu, F. Liang, and W.H. Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[92] P. Liu, D.K. Agrafiotis, and D.L. Theobald. Fast determination of the optimal rotational matrix for macromolecular superpositions. *Journal of Computational Chemistry*, 31(7):1561–1563, 2010.

[93] T.J.S. Lopes, M. Schaefer, J. Shoemaker, and Y. et al. Matsuoka. Tissue-specific subnetworks and characteristics of publicly available human protein interaction databases. *Bioinformatics*, 27(17):2414–2421, 2011.

[94] M. Lukk, M. Kapushesky, J. Nikkila, H. Parkinson, A. Goncalves, W. Huber, E. Ukkonen, and A. Brazma. A global map of human gene expression. *Nature Biotechnology*, 28(4):322–324, 2010.

[95] O. Magger, Y.Y. Waldman, E. Ruppin, and R. Sharan. Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLoS Computational Biology*, 8(9), 2012.

[96] N. Malod-Dognin and N. Przulj. GR-Align: fast and flexible alignment of protein 3d structures using graphlet degree similarity. *Bioinformatics*, 30(9):1259–1265, 2014.

[97] M.N. McCall, B.M. Bolstad, and R.A. Irizarry. Frozen robust multiarray analysis (frma). *Biostatistics*, 11(2):242–253, 2010.

[98] V. Memisevic and N. Przulj. C-GRAAL: Common-neighbors-based global graph alignment of biological networks. *Integrative Biology*, 4(7):734–743, 2012.

[99] X. Meng, P. Lu, H. Bai, P. Xiao, and Q. Fan. Transcriptional regulatory networks in human lung adenocarcinoma. *Molecular Medicine Reports*, 6(5):961–966, 2012.

[100] M. Menke, B. Berger, and L. Cowen. Matt: Local flexibility aids protein multiple structure alignment. *PLoS Computational Biology*, 4(1), 2008.

[101] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[102] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[103] G. Micale, A. Continella, A. Ferro, R. Giugno, and A. Pulvirenti. Gasoline: a cytoscape app for multiple local alignment of ppi networks [v2; ref status: indexed, http://f1000r.es/4f7]. *F1000Research*, 3(140), 2014.

[104] G. Micale, A. Ferro, A. Pulvirenti, and R. Giugno. Spectra: an integrated knowledge base for comparing tissue and tumor specific ppi networks in human. *Frontiers in Bioengineering and Biotechnology*, 3(58), 2015.

[105] G. Micale, A. Pulvirenti, R. Giugno, and A. Ferro. GASOLINE: a greedy and stochastic algorithm for optimal local multiple alignment of interaction networks. *PLoS ONE*, 9(6), 2014.

[106] G. Micale, A. Pulvirenti, R. Giugno, and A. Ferro. Proteins comparison through probabilistic optimal structure local alignment. *Frontiers in Genetics*, 5(302), 2014.

[107] T. Milenkovic, W. L. Ng, W. Hayes, and N. Przulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9(30):121–137, 2010.

[108] M. Mina and P.H. Guzzi. AlignMCL: Comparative analysis of protein interaction networks through markov clustering. *International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*, pages 174–181, 2012.

[109] M. Mina and P.H. Guzzi. Improving the robustness of local network alignment: Design and extensive assessmentof a markov clustering-based approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 11(3):561–572, 2014.

[110] M. Moll, D. H. Bryant, and L. E. Kavraki. The labelhash algorithm for substructure matching. *BMC Bioinformatics*, 11(555), 2010.

[111] U. Mudunuri, A. Che, M. Yi, and R.M. Stephens. Biodbnet: the biological database network. *Bioinformatics*, 25(4):555–556, 2009.

[112] A. Muller, B. Homey, H. Soto, N. Ge, and D. et al. Catron. Involvement of chemokine receptors in breast cancer metastasis. *Nature*, 410:50–56, 2000.

[113] L. Nersisyan, R. Samsonyan, and A. Arakelyan. Cykeggparser: tailoring kegg pathways to fit into systems biology analysis workflows. *F1000Research*, 3(145), 2014.

[114] S. Orchard, M. Ammari, B. Aranda, L. Brueza, and L. et al. Briganti. The mintact project-intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Research*, 2013.

[115] C.A. Orengo, S. Michie, A.D. Jones, D.T. Jones, M.B. Swindells, and J.M. Thornton. CATH: A hierarchical classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[116] C.A. Orengo and W.R. Taylor. SSAP: Sequential structure alignment program for protein structure comparison. *Computer Methods for Macromolecular Sequence Analysis*, 266:617–635, 1996.

[117] R.A. Pache and P. Aloy. A novel framework for the comparative analysis of biological networks. *PLoS ONE*, 7(2), 2012.

[118] R. Pastor-Satorras, E. Smith, and R.V. Sole. Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, 222(2):199–210, 2003.

[119] A. Patil, K. Nakai, and H. Nakamura. Hitpredict: a database of quality assessed proteinprotein interactions in nine species. *Nucleic Acids Research*, 39(1):D744–D749, 2011.

[120] A. S. Peleg, M. Shatsky, R. Nussinov, and H. J. Wolfson. Spatial chemical conservation of hot spot interactions in protein-protein complexes. *BMC Biology*, 5(43), 2007.

[121] A. S. Peleg, M. Shatsky, R. Nussinov, and H. J. Wolfson. Multibind and mappis: webservers for multiple alignment of protein 3d-binding sites and their interactions. *Nucleic Acids Research*, 36(2):W260–W264, 2008.

[122] D.A. Pelta, J.R. Gonzalez, and M.M. Vega. A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, 9(161), 2008.

[123] S. Peri, J.D. Navarro, T.Z. Kristiansen, and R. et al. Amanchy. Human protein reference database as a discovery resource for proteomics. *Nucleic acids research*, 32(1):D497–D501, 2004.

[124] G. Plata, T. Fuhrer, T. Hsiao, U. Sauer, and D. Vitkup. Global probabilistic annotation of metabolic networks enables enzyme discovery. *Nature Chemical Biology*, 8:848–854, 2012.

[125] B. Pro and N.H. Dang. Cd26/dipeptidyl peptidase iv and its role in cancer. *Histology and Histopathology*, 19(4):1345–1351, 2004.

[126] V. Pulim, B. Berger, and J. Bienkowska. Optimal contact map alignment of protein-protein interfaces. *Bioinformatics*, 24(20):2324–2328, 2008.

[127] X. Qian, S.M.E. Sahraeian, and B. Yoon. Enhancing the accuracy of hmm-based conserved pathway prediction using global correspondence scores. *BMC Bioinformatics*, 12(S6), 2011.

[128] X. Qian, S. Sze, and B. Yoon. Querying pathways in protein interaction networks based on hidden markov models. *Journal of Computational Biology*, 16(2):145–157, 2009.

[129] X. Qian and B. Yoon. Effective identification of conserved pathways in biological networks using hidden markov models. *PLoS ONE*, 4(12), 2009.

[130] D. Rajagopalan and P. Agarwal. Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(6):788–793, 2005.

[131] S. Razick, G. Magklaras, and I.M. Donaldson. irefindex: A consolidated protein interaction database with provenance. *BMC Bioinformatics*, 9(405), 2008.

[132] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Seraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnology*, 17(10):1030–1032, 1999.

[133] G. Rustici, N. Kolesnikov, M. Brandizi, and T. et al. Burdett. Arrayexpress updatetrends in database growth and links to data analysis tools. *Nucleic Acids Research*, 41(D1):D987–D990, 2013.

[134] S.M.E. Sahraeian and B. Yoon. Fast network querying algorithm for searching large-scale biological networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6008–6011, 2011.

[135] S.M.E. Sahraeian and B. Yoon. A network synthesis model for generating protein interaction network families. *PLoS ONE*, 7(8), 2012.

[136] S.M.E. Sahraeian and B. Yoon. SMETANA: Accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS ONE*, 8(7), 2013.

[137] G. Sanguinetti, J. Noirel, and P.C. Wright. Mmg: a probabilistic tool to identify submodules of metabolic pathways. *Bioinformatics*, 24(8):1078–1084, 2008.

[138] V. Saraph and T. Milenkovic. MAGNA: Maximizing accuracy in global network alignment. *Bioinformatics*, 30(20):2931–2940, 2014.

[139] V. Satuluri, S. Parthasarathy, and D. Ucar. Markov clustering of protein interaction networks with improved balance and scalability. *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 247–256, 2010.

[140] P. Shannon, A. Markiel, O. Ozier, N.S. Baliga, J.T. Wang, and D. et al. Ramage. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[141] R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24:427–433, 2006.

[142] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuine, and P. et al. Uetz. Conserved patterns of protein interaction in multiple species. *PNAS*, 106(6):1974–1979, 2005.

[143] M. Shatsky, R. Nussinov, and H.J. Wolfson. A method for simultaneous alignment of multiple protein structures. *Proteins: Structure, Function and Bioinformatics*, 56(1):143–156, 2004.

[144] M. Shatsky, A. S. Peleg, R. Nussinov, and H. J. Wolfson. The multiple common point set problem and its application to molecule binding pattern detection. *Journal of Computational Biology*, 13(2):407–428, 2006.

[145] Y. Shih and S. Parthasarathy. Identifying functional modules in interaction networks through overlapping markov clustering. *Bioinformatics*, 28(18):i473–i479, 2012.

[146] I.N. Shindyalov and P.E. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering Design And Selection*, 11(9):739–747, 1998.

[147] B.A. Shoemaker and A.R. Panchenko. Deciphering proteinprotein interactions. part i. experimental techniques and databases. *PLoS Computational Biology*, 3(3), 2007.

[148] B.A. Shoemaker and A.R. Panchenko. Deciphering proteinprotein interactions. part ii. computational methods to predict protein and domain interaction partners. *PLoS Computational Biology*, 3(4), 2007.

[149] R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *PNAS*, 105(35):12763–12768, 2008.

[150] F. Sohler, D. Hanisch, and R. Zimmer. New methods for joint analysis of biological networks and expression data. *Bioinformatics*, 20(10):1517–1521, 2004.

[151] R.V. Sole, R. Pastor-Satorras, E. Smith, and T.B. Kepler. A model of large-scale proteome evolution. *Advances in Complex Systems*, 5(1):43–54, 2002.

[152] C. Sotiriou, P. Wirapati, S. Loi, and A. et al. Harris. Gene expression profiling in breast cancer: Understanding the molecular basis of histologic grade to improve prognosis. *Journal of the National Cancer Institute*, 98(4):262–272, 2006.

[153] O. Souiai, E. Becker, C. Prieto, A. Benkahla, and J. et al. De Las Rivas. Functional integrative levels in the human interactome recapitulate organ organization. *PLoS ONE*, 6(7), 2011.

[154] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic acids research*, 34(1):D535–D539, 2006.

[155] A.I. Su, M.P. Cooke, K.A. Ching, Y. Hakak, and J.R. et al. Walker. Large-scale analysis of the human and mouse transcriptomes. *PNAS*, 99(7):4465–4470, 2002.

[156] A.I. Su, T. Wiltshire, S. Batalov, H. Lapp, and K.A. et al. Ching. A gene atlas of the mouse and human protein-encoding transcriptomes. *PNAS*, 101(16):6062–6067, 2004.

[157] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, and P. et al. Minguez. The string database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(S1):D561–D568, 2011.

[158] A.H.Y. Tong, M. Evangelista, A.B. Parsons, and H. et al. Xu. Systematic genetic analysis with ordered arrays of yeast deletion mutants. *Science*, 294(5550):2364–2368, 2001.

[159] J. W. Torrance, G. J. Bartlett, C. T. Porter, and J. M. Thornton. Using a library of structural templates to recognise catalytic sites and explore their evolution in homologous families. *Journal of Molecular Biology*, 347(3):565–581, 2005.

[160] P. Uetz, L. Giot, G. Cagney, T.A. Mansfield, and R.S. et al. Judson. A comprehensive analysis of proteinprotein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[161] M. Uhlen, P. Oksvold, L. Fagerberg, and E. et al. Lundberg. Towards a knowledge-based human protein atlas. *Nature Biotechnology*, 28:1248–1250, 2010.

[162] A. Vazifedoost, M. Rahgozar, B. Moshiri, M. Sadeghi, H.N. Chua, S.K. Ng, and L. Wong. Using data fusion for scoring reliability of protein-protein interactions. *Journal of Bioinformatics and Computational Biology*, 12(4):1–24, 2014.

[163] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1:38–44, 2003.

[164] J. Vlasblom and S.J. Wodak. Markov clustering versus affinity propagation for the partitioning of protein interaction graphs. *BMC Bioinformatics*, 10(99), 2009.

[165] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[166] S. Wang, J. Ma, J. Peng, and J. Xu. Protein structure alignment beyond spatial proximity. *Scientific Reports*, 3(1448), 2013.

[167] Y. Wang and X. Qian. Joint clustering of protein interaction networks through markov random walk. *BMC Systems Biology*, 8(S9), 2014.

[168] Z. Wei and H. Li. A markov random field model for network-based analysis of genomic data. *Bioinformatics*, 23(12):1537–1544, 2007.

[169] Z. Wei and H. Li. A hidden spatial-temporal markov random field model for network-based analysis of time course gene expression data. *Annals of Applied Statistics*, 2(1):408–429, 2008.

[170] Q. Wu, Y. Ye, M.K. Ng, S. Ho, and R. Shi. Collective prediction of protein functions from protein-protein interaction networks. *BMC Bioinformatics*, 15(S9), 2014.

[171] I. Xenarios, L. Salwinski, X.J. Duan, P. Higney, S.M. Kim, and D. Eisenberg. Dip: the database of interacting proteins. *Nucleic acids research*, 28(1):289–291, 2000.

[172] X. Xiao, A. Moreno-Moral, M. Rotival, L. Bottolo, and E. Petretto. Multi-tissue analysis of co-expression networks by higher-order generalized singular value decomposition identifies functionally coherent transcriptional modules. *PLoS Genetics*, 10(1), 2014.

[173] J. Xie, C. Xiang, Z. Zhou, D. Dai, and H. Zhang. NetCompare: A visualization tool for network alignment on galaxy. *International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*, 2:881–884, 2014.

[174] L. Xie and P. E. Bourne. Detecting evolutionary relationships across existing fold space, using sequence order-independent profile-profile alignments. *PNAS*, 105(14):5441–5446, 2008.

[175] L. Xie, Xie L., and P. E. Bourne. A unified statistical model to support local sequence order independent similarity searching for ligand-binding sites and its application to genome-based drug discovery. *Bioinformatics*, 25(12):i305–i312, 2009.

[176] J. Xu, F. Jiao, and B. Berger. A parameterized algorithm for protein structure alignment. *Journal of Computational Biology*, 14(5):564–577, 2007.

[177] Y. Zhang and J. Skolnick. TM-align: a protein structure alignment algorithm based on the tm-score. *Nucleic Acids Research*, 33(7):2302–2309, 2005.

[178] J. Zhao, S.H. Lee, M. Huss, and P. Holme. The network organization of cancer-associated protein complexes in human tissues. *Scientific Reports*, 3(1583), 2012.