CALI: A NOVEL MODEL FOR VISUAL MINING OF BIOLOGICAL RELEVANT PATTERNS IN PROTEIN-LIGAND GRAPHS

SUSANA MEDINA GORDILLO

CALI: A NOVEL MODEL FOR VISUAL MINING OF BIOLOGICAL RELEVANT PATTERNS IN PROTEIN-LIGAND GRAPHS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Raquel C. de Melo-Minardi
Coorientadora: Sabrina A. Silveira

Belo Horizonte
Outubro de 2016

SUSANA MEDINA GORDILLO

CALI: A NOVEL MODEL FOR VISUAL MINING OF BIOLOGICAL RELEVANT PATTERNS IN PROTEIN-LIGAND GRAPHS

M.SC Thesis presented to the Graduate Program in Ciência da Computação of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Ciência da Computação.

Advisor: Raquel C. de Melo-Minardi
Co-Advisor: Sabrina A. Silveira

Belo Horizonte

October 2016

© 2016, Susana Medina Gordillo.
Todos os direitos reservados.

Medina Gordillo, Susana

G661c CALI: A novel model for visual mining of biological relevant patterns in protein-ligand graphs / Susana Medina Gordillo. — Belo Horizonte, 2016

xxvii, 104 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais- Departamento de Ciência da Computação.

Orientadora: Raquel C. de Melo-Minardi
Coorientadora: Sabrina A. Silveira

1. Computação - Teses. 2. Bioinformática. 3. Interação Proteína-ligante. 4. Teoria de redes complexas. 5. Mineração de grafos. 6. Visualização de dados. I. Orientadora. II. Coorientadora. III. Título.

CDU 519.6*93(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

CALI: a novel model for visual mining of biological relevant patterns in protein-ligand graphs

SUSANA MEDINA GORDILLO

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROFA. RAQUEL CARDOSO DE MELO MINARDI - Orientadora
Departamento de Ciência da Computação - UFMG

PROFA. SABRINA DE AZEVEDO SILVEIRA - Coorientadora
Departamento de Informática - UFV

PROF. ARISTÓTELES GÓES -Neto
Instituto de Ciências Biológicas - UFMG

PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 27 de outubro de 2016.

To my parents, Cecilia and Carlos, you are always my inspiration and my strength.

Acknowledgments

There are so many people to thank for helping me during the last years. First of all, are my parents, Cecilia and Carlos, whose always inspired me to become the person that I am today. They teach me to do everything with love, passion and to persevere, because success is the result of a hard work. Also, they gave me lots of advice and supported me, no matter the situations I was in.

Thank you to professors Simone, Angel, Irene and Martha. You gave support and encourage me to began a Master. I am glad to have special friends that, even when they are far, still keep in touch with me, giving me support. Thanks to Marcela, Claudia, David E., Ricardo and Vanessa.

Brazil gave me a huge opportunity to do this master Thesis. In this country I met a lot of wonderful people. I am very thankful to the help from David J., Juan and Luis before and after I arrived.

I am thankful for the guidance of my advisor Raquel Minardi during this work. Also, I thank to everyone from the LBS, specially Alexandre and Sabrina. You gave me a huge support from the beginning of my graduate studies until the end. I need to thank other LBS' members for the great, nice and lovely laboratory that you had built up. There are Pedro, Marcos, Larissa, Carmelina, Diego and Guilherme S.. The LBS became like my second home and it is because of them.

To my DCC colleagues from other laboratories as Danilo, Jhielson, Natalia and Guilherme V.. To the persons that helped me or gave advice or support to correct this document as Alberto and Rensso. To the first friends I made in Brazil: Kelly and Naiana, my favourite neighbors forever. Also, to my dearest friends Samira and França, my partners in crime.

I have so many people to thank for, which would make this too long. Met them and learn about their lives was an honor and a great pleasure. I wish them a path long enough in life to find happiness.

Abstract

Protein-ligand interaction (PLI) networks show how proteins interact with small non-protein ligands and can be used to study molecular recognition, which plays an important role in biological systems. The binding and interaction of molecules depend on a combination of conformational and physicochemical complementarity. There are several methods to predict protein-ligand interactions, but a few are designed to identify and describe implications of intelligible factors in protein-ligand recognition.

We propose CALI (Complex network-based Analysis of protein-Ligand Interactions), a strategy based on complex network modeling of protein-ligand interactions, revealing frequent and relevant patterns among them. We compared patterns obtained with CALI to those computed using Frequent Subgraph Mining (FSM) paradigm. FSM needs to run several times for a variety of support values and it also needs a mapping step, in which computed patterns are mapped to the graph input dataset through a subgraph isomorphism algorithm. On the other hand, CALI is executed once and without applying the mapping step to the input dataset. Additionally, patterns obtained with CALI were compared to experimentally determined protein-ligand interactions from previous studies involving two datasets: one composed by the well studied CDK2 enzymes and, the other, by the Ricin toxin. For CDK2 dataset, CALI found 90% of such residues and, for Ricin dataset, CALI found all residues that interact with ligands.

CALI was able to predict residues experimentally determined as relevant in protein-ligand interactions for two diverse datasets. This new model requires neither running FSM nor analyzing its wide number of output patterns to find the most common protein-ligand interactions. Instead, we propose using network topological properties coupled with a powerful visual and interactive representation of data to analyze interactions. Furthermore, our strategy provides a general view of the input interaction dataset, showing the most common PLIs from a global perspective.

Keywords: protein-ligand interaction, complex networks, frequent pattern mining, visualization, graph-mining.

List of Figures

List of Tables

Contents

List of Abbreviations

Abbreviation	*Description*
2D	Two-dimensional
3D	Three-dimensional
NMR	Nuclear magnetic resonance
PLI	Protein-Ligand Interaction
PPI	Protein-Protein Interaction
ATP	Adenosine triphosphate
DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
CDK	Cyclin-Dependent Kinase
RTA	Ricin A-chain
SAP	Saporin L1
SVM	Support Vector Machines
FSM	Frequent Subgraph Mining. Includes FGM and FTM.
FTM	Frequent Subtree Mining
FGM	Frequent Subgraph Mining
PTE	Predictive Toxicology Evaluation Challenge
HTS	High-throughput
CALI	Complex Network-Based Analysis of Protein-Ligand Interaction
Å	Angströms

Chapter 1

Introduction

1.1 Motivation

Biological data and its relationships are commonly represented as graphs or networks. Entities such as DNA, genes, proteins, just to name a few, are usually interacting with each other and these interactions can be studied in a straightforward manner using classical graph theory and complex network theory as well. For instance, when proteins and/or genes interact in a complex network, there may be a group of them that contribute to a disease process and at the same time a subgroup may become active at different stages of the disease. If the sequence of biological events at the different stages of the disease can be identified, it may be possible to propose pharmaceutical interventions, accomplishing more effective treatment of the disease [Wang et al., 2005].

Some of the most well known biological networks are metabolic, gene regulation, protein-protein interaction (PPI) and protein-ligand interaction (PLI) networks. Proteins are essential macro-molecules composed by units called amino acids. Protein interactions are the basis for the function of a larger protein complex [Junker and Schreiber, 2008], and lately have been studied through PPI and PLI networks perspectives. PPI networks show how proteins interact with others for different roles, as to assembly cell structural components, metabolic channeling and transmission of regulatory signals. On the other hand, PLI networks show how proteins interact with small non-protein ligands.

An important phenomenon that can be studied under the light of PPI and PLI networks is molecular recognition. It plays an important role in biological systems and refers to interactions between two or more molecules through noncovalent bonding, such as aromatic stacking, hydrogen bonding, hydrophobic forces and salt bridges. The conditions responsible for an interaction between two or more molecules are a

combination of conformational and physicochemical complementarity [Kahraman et al., 2007]. Understanding and predicting protein-ligand interactions are essential steps towards ligand prediction, target identification, lead discovery and drug design [Pires et al., 2013]. Despite the existence of several methods to predict protein ligands, few methodologies are devised to identify and describe implications of intelligible factors in protein ligand recognition.

According to Kitano [2002], computer techniques and bioinformatics approaches had become the most promising options for effective drug discovery based on structures and on how a receptor recognizes its ligands. These computational models and algorithms have showed their potential in early stage of drug-target identification and target validation [Chen et al., 2015]. Advances on the understanding of PPI and PLIs have led the development of algorithms to predict interactions [Medina-Franco et al., 2014] for proteins [Szilágyi et al., 2005], to find missing links [Clauset et al., 2008] and detecting overlapping protein complexes (proteins which belong to two families or more) in PPIs [Nepusz et al., 2012].

1.2 Biological Background

1.2.1 Protein

Proteins are macro-molecules with crucial functions for all biological processes in organisms. Their work include to transport and store other molecules, such as oxygen, bring mechanical support and immunity protection, generate movement, nervous impulse transmission, growth and differentiation control.

Proteins are made of linear sequence of units called *amino acids*. An amino acid (Figure 1.1) is a simple molecule composed by an organic acid. Additionally, proteins have a wide range of chemical functional groups, which permit some proteins, known as *enzymes*, to catalyze specific reactions in biological systems. The interaction between proteins and other biological macro-molecules is very important because it allows to form complex structures. Flexibility allows proteins to form diverse structural elements, each one with a peculiar function [Stryer et al., 2013].

1.2.1.1 Protein Structural Organization

The protein synthesis produces thousands of macro-molecules in the cell, which is controlled by instructions coded in the DNA. Every type of protein has an unique amino

acid sequence with a particular shape, size and function [Restrepo O. and Zuluaga, 2011].

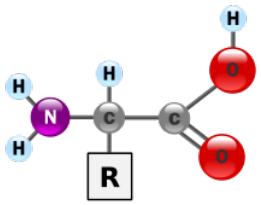

Figure 1.1: The structure of an amino acid molecule. Source: Wikipedia, "Amino acid", June 23 2016.

The basic protein components, the amino acids, contain three groups: an amine group (-NH2), a carboxylic group (-COOH) and a radical group (-R), also known as side-chain (see Figure 1.1). There are 20 amino acids that compound all proteins from all species, differenced only by their side chain [Stryer et al., 2013]. The names of these 20 types of amino acids are usually abbreviated with one-letter or three-letters as shown in Table A.1. The amino acids inside proteins are called *residues*. Residues establish bonds with one another, specifically known as *peptide* bonds, forming *peptide chains*. The protein structure has a hierarchy containing four levels (Figure 1.2). The *primary* structure is the linear sequence of amino acids or a peptide chain, which can have a length between 50 and 2000 amino acids. The *secondary* structure describes the folds done by the amino acid sequence. There are three basic folds: alpha helices, beta sheets and turns or loops. The secondary structure is mainly formed by helices and sheets in an approximated proportion of 70%-20% respectively and the loops are considered random shapes connecting them [Wang et al., 2005]. The *tertiary* structure describes another folds done by the secondary structure. Resulting in connections among amino acids side-chains that are widely separated in the peptide chain. The *quaternary* structure occurs only in proteins formed by two or more polypeptide chains, these chains are also known as *subunits*. The specific manner that subunits fit together in the native protein conformation is the quaternary structure. The function is determinated by all this specific structural organization adopted by the protein [Restrepo O. and Zuluaga, 2011]. For instance, the hemoglobin is an oxygen-carrying protein in red blood cells, its quaternary structure is formed by four subunits in roughly a tetrahedral arrangement.

1.2.1.2 Protein Complex

Protein complex is a form of quaternary structure. The polypeptide chains in a complex are linked by non-covalent protein-protein interactions, causing proteins to have

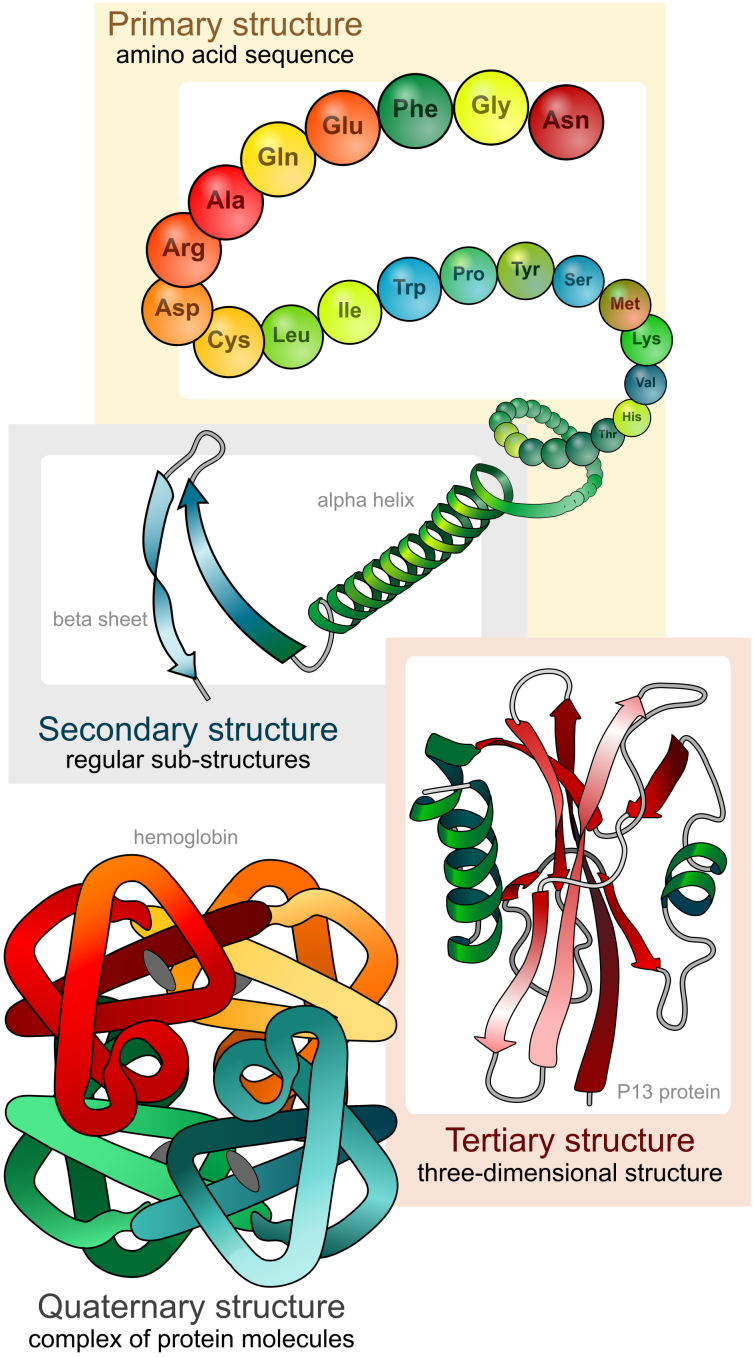

Figure 1.2: Protein Hierarchy. Source: Wikipedia. "Protein Structure", 5 June 2016.

different folded conformations over time. Many biological processes are supported by the functions of protein complexes as *enzymes* that catalyze chemical reactions. *Enzymes* are proteins acting alone or forming complexes usually with a non-proteic chain. Enzymes speed up (*catalyze*) chemical reactions decreasing the activation energy of those reactions.

Additionally, enzymes have a high specificity confered by a special region called *active site*, which according to Kahraman and Thornton [2008] is composed by regions on their surface specially modeled to interact with other molecules. The active site has usually two parts: the *binding site* and the *catalytic site*. The first one checks if the incoming molecule has the right characteristics to bond the enzyme. The second one is the group composed commonly by two to six amino acids interacting with the molecule that perform the catalytic reaction. The molecule bound to the enzyme active site, commonly known as *substrate* is transformed creating the *products* of the catalytic reaction [Restrepo O. and Zuluaga, 2011]. Proteins composed only by polypeptide chains are *protein-protein complexes*. When a protein is binding to a ligand forming a complex is called *protein-ligand complex*, as is shown in Figure 1.3.

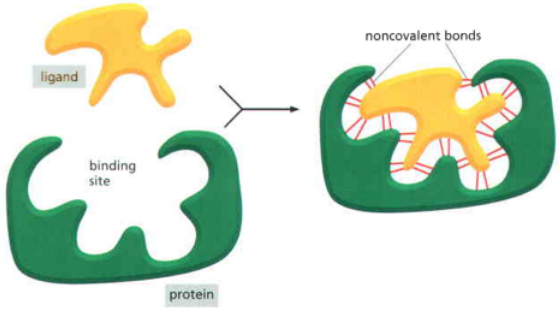

Figure 1.3: Protein-ligand complex. The protein (green) adopts a specific shape to interact with a ligand (yellow) through non-covalent bonds in particular sites of the protein structure. Source: Alberts, et al. 2008. Molecular Biology of the Cell (5th ed). Garland Science: New York.

1.2.1.3 Ligand

Most enzymes are proteins, forming protein-ligand complexes which create specific bonds to the ligand, through the binding site. Some proteins are not enzymes but they also form protein-ligand complexes, as the *metalloproteins*, a well-known example of them is the *hemoglobin*, an oxygen-transport protein present in the blood of almost all vertebrates. Some ligands produce a signal causing inhibition of the protein function. These ligands are called *inhibitors*, which decrease the velocity of reactions catalyzed by enzymes. Inhibitors are able to regulate enzyme activity (see Figure 1.4). If an inhibitor binds to the active site it can block the chemical reaction of the enzyme. There are two principal types of inhibitors: reversible and irreversible. *Reversible inhibitors* form non-covalent bonds with the enzyme, which are weak and can be easily removed by dilution or dialysis. *Irreversible inhibitors* form covalent bonds with the enzyme, causing chemical changes to the enzyme active site. Some inhibitors are known as

drugs or *medicines*, because are used in disease treatments [Restrepo O. and Zuluaga, 2011]. A very famous drug that have its mechanism of action by inhibition is *penicillin*. Other ligands are known as *poisons*, because they inactivate irreversibly an enzyme, even causing death. *Arsenate* is a potent poison that leads to death, because it disrupts the production of ATP, which is the molecule that transports chemical energy where it is necessary.

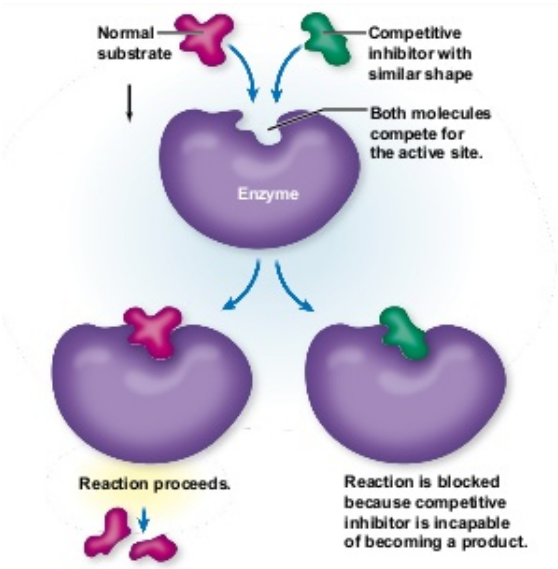

Figure 1.4: **Enzyme inhibition**. In a competitive inhibition, the substrate and the inhibitor compete to bind the active site of the enzyme. If the substrate wins to bind the enzyme, the reaction occurs normally. When the inhibitor wins the reaction is blocked incapable of generating a product. Source: "Microbiology Fundamentals: A Clinical Approach". McGraw-Hill Education. Cowan and Bunn [2015].

1.2.1.4 Interactions

Molecular interactions are essential for biological processes. Chemical *bonds* make possible molecular interactions and can be classified as strong or weak. The strong bonds are known as *covalent* bonds and keep together atoms by sharing a pair of electrons between them. Single (i.e. σ and π), double and triple bonds are types of covalent bonds. However, *non-covalent* bonds are the essential of life, because they are reversible. These interactions are able to maintain the protein structure and to form protein complexes. The three main types of non-covalent bonds, which create reversible interactions in biomolecules, are electrostatic interactions (e.g. coulombic charge-charge), hydrogen bonds and Van der Waals interactions (e.g. dipole-ion) [Stryer et al., 2013].

1.2.1.5 Protein-Protein Interactions

Protein-protein interactions (PPI) have a vast area of application principally for drug discovery, which include subareas as molecular docking, structure-based design, clustering of complexes, virtual screening of molecular fragments, small molecules and other types of compounds [Medina-Franco et al., 2014].

Interactions driving the affinity of a pair of proteins are not distributed evenly across their surfaces. Instead they are located in regions known as *hot spots* [Scott et al., 2016]. These regions are composed by residues with high binding free energy, essential for protein-protein binding. Frequent residues found in hot spots are tryptophan, arginine and tyrosine. The "anchor residues" are tyrosine, phenylalanine, tryptophan and leucine which are known as characteristic of small-molecule binding pockets in the interface of protein-protein interactions [Medina-Franco et al., 2014].

There are experimental methods for detecting PPIs (as the described in Section C) but bioinformatic approaches have been also used for the same purpose. The study of PPIs became supported by the creation of several databases as SPRING [Szklarczyk et al., 2011], TIMBAL [Higueruelo et al., 2009] and PICCOLO [Bickerton et al., 2011].

PPIs can be classified as obligate (strong and long lived) or non-obligate (weaker and transient). Additionally, PPIs can be divided in four classes according their structure: pairs of globular proteins (Figure 1.5 a), interactions between a pair of globular proteins in which one or both proteins undergo a substantial conformational change on binding (Figure 1.5 b), PPIs involving a globular protein interacting with a single peptide chain (Figure 1.5 c-d) and PPIs between two peptide chains (Figure 1.5 e).

1.2.1.6 Protein-Ligand Interactions

Almost all processes in living organisms involve protein-ligand interactions (PLI). These interactions describe the relation between a particular protein and diverse ligands. This relation is chemical and comprise biological recognition at the molecular level (Figure 1.6). Proteins have specific sites designed to bind ligands depending on the cell needs. The common weak bonding forces that create binding interactions are hydrophobic forces or charge dipole interactions. Studying structures of protein-ligand complexes at atomic resolution make possible to conceive drugs for many disease treatments [Dunn, 2001]. Most of the biochemical and pharmaceutical researches focus on the equilibrium binding affinities in PLIs. The similarities between protein-ligand binding and protein folding, allow to apply methods or principles that worked out in one to another [Held et al., 2011].

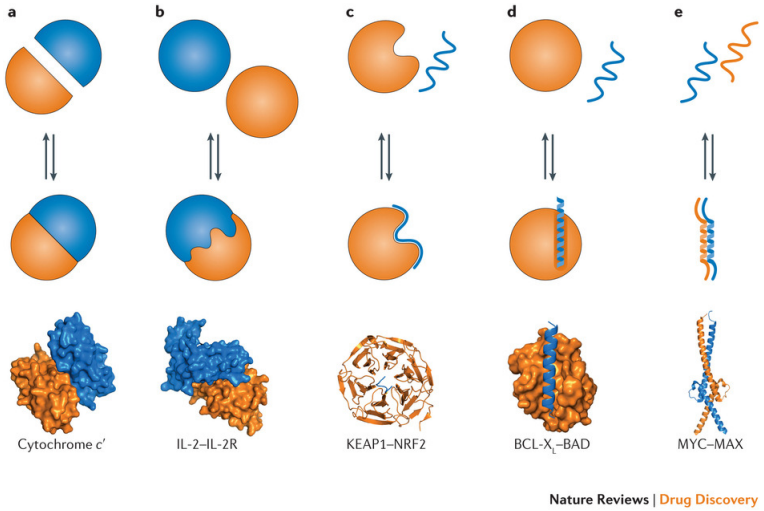

Figure 1.5: **Structural classification of protein-protein interactions**. In the upper part of the figure, simplified illustrations are used to represent protein and/or peptide partners, and in the lower part of the figure, example crystal structures for each interaction type are shown. a | An interaction between two globular proteins with preformed surfaces (PDB ID: 2CCY). b | An interaction between two globular proteins with an induced binding surface (PDB ID: 1z92). c | An interaction of a rigid globular protein with a peptide (PDB ID: 2DYH). d | An interaction of a flexible globular protein with a peptide (PDB ID: 2XA0). e | An interaction of two peptides (PDB ID: 1NKP). Source: "Small molecules, big targets: drug discovery faces the protein-protein interaction challenge". [Scott et al., 2016].

Bonds between proteins and ligands are discovered by biochemical and biophysical assays[1] which are labor-intensive. Recently, new PLIs have been discovered by developing novel approaches as the affinity-based separation by affinity chromatography [Liu et al., 2011]. However, it is becoming evident that the dynamical and kinetic properties are crucial for the effectiveness of PLIs. The dynamical properties of binding are inherently linked to structural aspects (such as the size, concentration, and spatial distribution) of the binding partners, as well as their detailed atomic structures and changes that occur therein [Held et al., 2011].

1.2.1.7 Sequence Alignment

Macro-molecule structure analysis involves several diverse topics as prediction of secondary structure of RNA and proteins, comparison of protein structures, protein structure classification and visualization of protein structures. Sequence alignment permits comparison between biological sequences (DNA, RNA, protein). Analysis of the amino acid sequences allow to identify changes in proteins, through time. Multiple sequence

[1] An assay is a biological experiment.

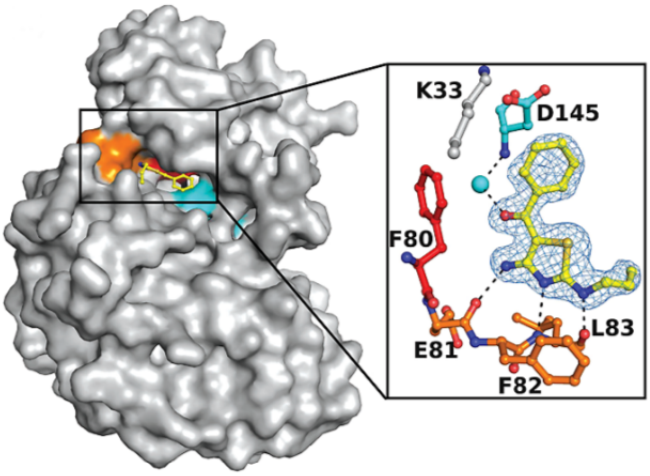

Figure 1.6: **Protein-ligand interaction of a compound (inhibitor) with the enzyme CDK2**. The enzyme is shown in gray and the inhibitor in yellow. The exploded view show in red and orange residues from the enzyme and in cyan is a motif. The water molecules are also shown as cyan spheres. PDB ID:3QQK. Source: Schonbrunn et al. [2013].

alignment process takes three or more input sequences forcing them to have the same length by inserting a gap symbol. Algorithms implementing this process are able to identify similar regions in all sequences, defining a conserved consensus pattern. DALI is a software tool for structural alignment and Clustal (with its different versions, as V, X and omega) is a software tool for sequence alignment. Progressive alignment [Hogeweg and Hesper, 1984] and iterative methods from PRRN/PRRP software package [Gotoh, 1996] are some of the most popular strategies for sequence alignment. An example of sequence alignment is provided in Figure1.7.

1.2.1.8 Three-Dimensional Structure

Biological macro-molecules structure are obtained preeminently by X-ray crystallography, NMR spectroscopy, or cryo-electron microscopy. The Protein Data Bank (PDB) [Rose et al., 2015] is the main online public database [2] that storages three-dimensional structural data, including proteins and nucleic acids, as DNA. There are other databases that use structures deposited in the PDB, with specific purposes as the classification by class, architecture, topology, and homology (CATH) database, the structural classification of proteins (SCOP) database, Molecular Modeling Database (MMDB), and Swiss-Model resource [Wang et al., 2005].

[2] http://www.rcsb.org/pdb

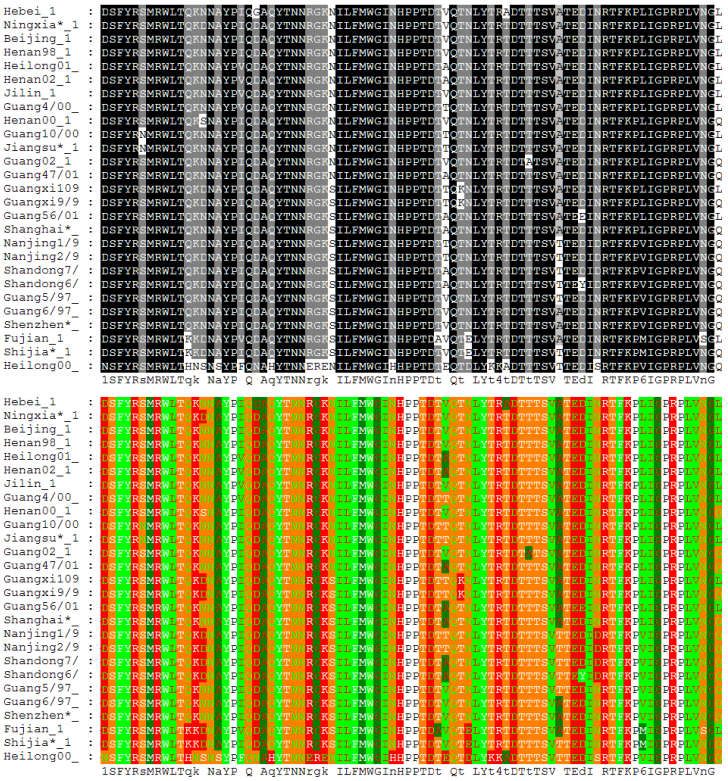

Figure 1.7: Alignment of 27 avian influenza hemagglutinin protein sequences colored by residue conservation (top) and residue properties (bottom). Alignment produced with ClustalW software. Source: Wikipedia, "Sequence alignment", 5 June 2016.

Visualization of the 3D protein structures takes crystal structures, usually from the PDB, showing details for every organization level of the protein (as the one in Figure 1.8). This is crucial to scientists, because it allows them to understand how protein interactions are formed in space. Cn3d, Rasmol and PyMOL are well-known visualization tools for 3D protein structures [Wang et al., 2005].

1.2.1.9 Contacts

The files stored in the PDB contain the 3D conformation of proteins, which means they include the atomic coordinates. However, PDB files do not include contacts formed by the atoms. Contacts have diverse definitions depending on their applications. According to Sobolev et al. [1999b], two elements (e.g. amino acids, ligands) were postulated as forming a contact if the closest distance between their atoms is below a certain threshold. There are various approaches to calculate which atoms establish *contact*, according to the distance between them [3], creating physicochemical interactions. Some methods are based on contact surface area, as the one proposed by Sobolev et al., 1999b,

[3]Distance between protein atoms is measured in Angstroms Å.

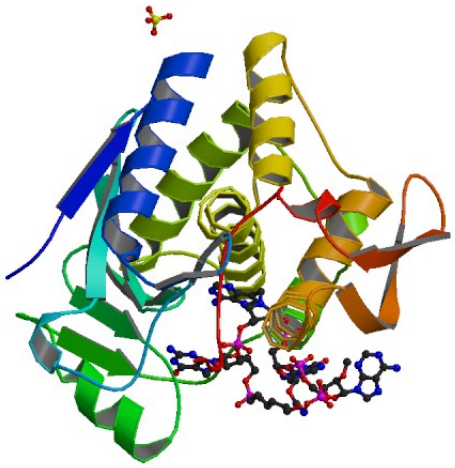

Figure 1.8: Crystal structure of Ricin A-chain in complex with the cyclic tetranucleotide inhibitor, a transition state analogue. Source: PDB web site (http://www.rcsb.org), PDB ID: 3HIO. 6 September 2016.

others propose algorithms as the based on Voronoi diagrams (polyhedra draws around atoms and residues) [Poupon, 2004], Delaunay triangulation (a unique partition of 3D space with non-overlapping tetrahedron) [Zhou and Yan, 2014] or both (see Figure 1.9). Types of contacts are hydrogen bonds, hydrophobic-hydrophobic, aromatic-aromatic, aromatic-polar, etc.

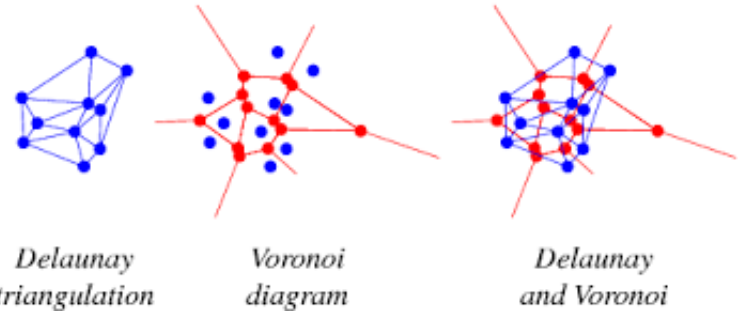

Figure 1.9: Delaunay triangulation, Voronoi diagram and the combination of both. Source: Weisstein, Eric W. "Voronoi Diagram." From MathWorld–A Wolfram Web Resource.

1.2.2 Mining Chemical Compounds

Mining frequent structural patterns is an important strategy to analyze and comprehend protein complexes or chemical compound structures (i.e. inhibitors). *Chemical compounds* are groups of different atoms bound together adopting a well-defined geo-

metric configuration, as hexagons, pentagons, etc. (see Figure 1.10), allowing a graphical representation which chemists are used to work with. There are many representations for chemical compounds. The most known and simplest is the molecular formula [Wang et al., 2005]. For example, the molecular formula of caffeine is `C8H10N4O2`.

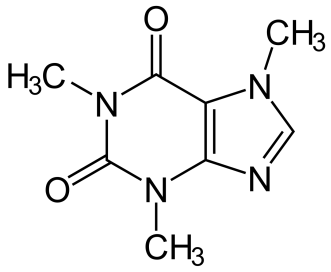

Figure 1.10: Graphical representation of the caffeine molecule. Source: Wikipedia. "Caffeine". June 27 2016.

The chemical 3D structure determinates the compound activity and can be naturally represented as a *graph*. In the atomic level, a compound can be viewed as a graph where vertices are the atoms and the edges are the bonds between the atoms. Also, this representation is refered as *topological graph*.

Drug discovery is an area related to bioinformatics, which involves searching ligands that act as inhibitors and then modifying or designing them in order to bind protein targets. The search that leads to drug discovery requires study of ligand-target interaction, which is a traditional domain of medicinal chemistry and computational chemists [MacCuish and MacCuish, 2010]. Protein kinases are the second most important group of drug targets, because their abnormal activity causes nearly 400 human diseases, such as cancer, diabetes and asthma [Chen et al., 2015]. The cyclin-dependent kinases (CDK) are protein complexes regulators of the cell cycle progression, composed by two subunits: the regulatory (cyclin) and the catalytic (kinase). CDK2 is an enzyme (belongs to the CDKs family) that controls the transition of some cell mitosis phases (from G1 to S) [Shapiro, 2006]. It has been demonstrated by Akli et al., 2011, that the cyclin E binding to the CDK2 is directly related with the formation of breast cancer.

Computational techniques have been used to identify chemical compounds. However, the classical machine learning approaches can not handle the natural structure of chemical compounds, creating an accurate or representative model of them. Thus, new computational techniques were developed with this objective. One of the first techniques developed builds a classification model based on physicochemical properties usually derivated from the compounds structure, known as *quantitative-activity relantionships* (QSAR). The second and actually most successful technique attempts to identify a small number of substructures in the chemical compound representing them

as graphs, applying frequent subgraph mining (FSM). In this last approach there are two manners to identify these substructures [Wang et al., 2005]. The first one is through apriori candidate generation as the frequent subgraph discovery algorithm proposed by Kuramochi and Karypis, 2004. The second one is by growing frequent substructure patterns and pruning those that are non-frequent based on a threshold (support) value, as the gSpan algorithm [Yan and Han, 2002]. Computational techniques avoid manual comparisons among all compounds, which is a time-consuming task and requires a huge specific knowledge. Finding frequent patterns applying FSM algorithms can be easier in data which naturally are able to be represented as a graph or graphs, as it is the case of chemical compounds. In the next section, detailed descriptions about FSM algorithms are presented.

1.3 Related Work

Protein-ligand Interactions (PLI) have been recently studied using diverse approaches, here are mentioned four of them. *Biochemistry* is the first approach, which studies PLIs doing laboratory experiments (assays). Computer science offers a huge variety of techniques and methods, which sometimes reduces the search space and/or time applying algorithms or simulations (i.e. Monte Carlo). In other cases, computer science contributes to improve a characterization or model, using the knowledge of emerging areas. This work focus barely on two contrasting approaches of the computer science. *Network Science* [Barabási, 2016], the second approach, also known as *Complex Network Theory*, emerged in 2000, allowing to see the data as a system and deduce non-trivial behaviors. *Graph Mining* [Zaki and Wagner Meira, 2014] the third approach, arises for the need to extract relevant information (e.g. FSM algorithms) in data, which can be modeled using *graphs*, such as PPI and PLI. *Data Visualization*, the fourth and last approach is fundamental for biological sciences, such as in biochemistry where it is mainly used for 2D or 3D molecular structure representations. In this Thesis, we propose an interdisciplinary model based on these four approaches to represent PLIs and find biological patterns. Next, we present the most prominent works of biochemistry, network science, graph mining and data visualization related to our research topic.

1.3.1 Biochemistry

There are a huge amount of *biochemical* studies that aim to generate inhibitors or try to understand the inhibition process for a determinate protein. We relate two main papers as basis for our biological analysis, for the two datasets (Section 3.2) used in

this work: CDK2 and Ricin. These two proteins dissimilar between them (first one is from humans, second one is from plants), were useful as case studies for this work and both represent actually interesting research targets. CDK2 is related to the formation of breast cancer [Akli et al., 2011] and a promising target for the development of nonhormonal male contraceptives [Schonbrunn et al., 2013]. On the other hand, Ricin is a toxic protein, classified as a category B bioterrorism agent, which has been used in clinical trials to kill leukemia and lymphoma cells, but it is limited by its side effects [Ho et al., 2009].

Schonbrunn et al. [2013] developed highly potent diaminothiazole inhibitors of the protein family Cyclin-Dependent Kinases (CDKs), with preference for the CDK2 and CDK5. They describe all the chemical processes for the compound development, the experiments, the chemical composition and, most important of all, their mechanism of action.

The main topic in the work of Ho et al. [2009] is the transition state analogue inhibitors in two homologous structures: Ricin A-chain(RTA) and Saporin-L1 (SAP). They studied structural similarities (30% identity) and differences (in the N-terminal region) between these two proteins, with the goal to establish the catalytic site features. They found that the inhibitor-binding modes, in both proteins, are reversed and these proteins have an active site composed by two tyrosines and one arginine, to form a polarized quadruple π-stack. Additionally, both papers include 2D maps and 3D visualizations of the structure of these proteins and their inhibitors.

1.3.2 Network Science

Several biochemistry studies begin to use *Network Science*. Taylor [2013] did an interesting survey, with synopsis of many articles that study protein structures applying network strategies. Several types of networks are described modeling molecular structure data. Also, the author brings attention to the network descriptors and the fact that the network approach can be incorporated into computational methods.

Kuhn et al. [2011] capture local cooperativity in protein-ligand interactions using interaction networks. They modeled the network constructing a set of noncovalent contacts, three training sets with the same quality criteria.A fourth training set[4] with compounds in which a small change in the ligand causes a drastic change in binding affinity. They also define network descriptors and develop ScorpionScore, a scoring function to predict binding affinity in these four datasets. Their results were promising,

[4]This dataset also includes no public information from the company Roche: 93 protein complexes. The reason related by the authors is a well-known problem in molecular interactions: There was no data with samples of unfavorable interactions.

even when they highlight some problems as absence of a good training set and inability of the model to capture long-range cooperativity effects. Besides, they emphasize interest to apply the network approach in other protein families, as kinases.

The work of Liu and Hu [2011] focuses on Heme proteins, which are a group of proteins characterized for containing an iron-porphyrin complex and are essential for the survival of organisms. Liu et al. model the network with residues as nodes and the contacts between them as edges. They aim to identify the binding residues through their topological characteristics. The predictor of key residues was done using support vector machines (SVM). They concluded that residues with high centrality values are more likely to be involved in the binding.

Cheng et al. [2012] construct a network of drug-target interactions (DTI), as a bipartite graph, where one node set represents drugs and the second node set represents targets (i.e. proteins, enzymes, hormones, ligands, etc.). They take data from several databases, as the Drugbank [Wishart et al., 2006], and develop inference methods based on the interaction network, which allows them to predict new DTI. They did experimental validation and found that five old drugs, indeed, are able to inhibit several targets.

Network science is continually evolving and it is important to highlight three articles that collaborate to it. Latapy et al. [2008] propose new network metrics for bipartite networks, demonstrating the inconveniences of analyzing these networks by their projections. They develop a new definition of clustering coefficient and propose a new coefficient, called redundancy, which focuses on neighborhood overlapping in nodes. They call attention to maintain the nature of the network, as in bipartite networks, which allows to obtain more precise information.

Estrada has several works related to networks. Here, we briefly comment on two of them. The first one [Estrada and Rodríguez-Velázquez, 2005] analyses several complex networks, as yeast protein-protein interactions (PPI). The second one [Estrada and Hatano, 2008] develop a new manner to calculate communicability (Section 2.1.2.6), using the graph spectral theory. It permits calculations based on eigenvalues and eigenvectors from the adjacency matrix of the network. This method is the implementation done for communicability in the *networkx* library [Hagberg et al., 2008].

These network studies show the applicability of networks in biochemistry and guided us to model PLIs as an interaction network, which is the basis of the proposal model.

1.3.3 Graph Mining

Graph mining is an important field within data mining. The goal of graph mining is to identify frequent subgraphs in graph datasets. The survey done by Jiang et al. [2013] presents main definitions for frequent subgraph mining (FSM) and its most popular and used algorithms. They separately discuss frequent subgraph (FGM) and frequent subtree mining (FTM). The FTM algorithms applied to chemical compound analysis are FreeTreeMiner [Chi et al., 2003] , FTMiner [Rückert and Kramer, 2004], F3TM [Zhao and Yu, 2008], CFFTree [Zhao and Yu, 2007] and HybridTreeMiner [Chi et al., 2004]. A huge number of FGM algorithms use a expansion technique, such as VSIGRAM and HISGRAM [Kuramochi and Karypis, 2005], gSpan [Yan and Han, 2002] (restrictive extension), CloseGraph [Yan and Han, 2003] and CloseCut-Splat [Yan et al., 2005]. There are some FGM algorithms as SPIN, developed by Huan et al. [2004], which only mines maximal frequent subgraphs.

A noteworthy algorithm for frequent subgraph mining is gSpan, the most cited FSM algorithm according to Jiang et al. [2013], considered as the state-of-art as it has successfully applied in many different graph data, using an efficient candidate generation based on a lexicographic order. gSpan experiments performed by Yan and Han [2002] used a chemical compound dataset (DTP Antiviral Screen), however they do not conclude anything about the semantic of the patterns found. They focus on the general efficiency of the algorithm as the runtime data and the low memory consumption. As gSpan became reference for FSM algorithms, many articles (as the mentioned above) used this same chemical dataset to compare proposed algorithms to gSpan, also considering complexity, runtime and memory consumption, without discussing the semantic of computed patterns.

There are approximately 2^n frequent subgraphs in a *n-edge* labeled graph G. Mining complete subpatterns in G, which represents a molecule structure, generates subgraphs predominantly with redundant information when they share the same support. This means that exploring all the possibles subpatterns in G have exponential space and time complexity. gSpan reduces both complexities by using a lexicographic order avoiding the generation of all candidates and pruning false positives. However, it still generates recurring patterns. Using the mining frequent itemsets approach offer advantages in large graph datasets, reducing the number of mined subgraphs and consequently reducing the execution time. A frequent pattern I is closed if there exists no proper super-pattern of I with the same support. CloseGraph mines only *closed frequent graph patterns* and it was proposed by the same authors of gSpan. A frequent pattern I is maximal if none of its supersets are frequent. SPIN is a mining algorithm

which finds *maximal frequent subgraphs*.

Graph mining algorithms designed specifically for biological or chemical data are few, because of the complexity to do a generalization in this context. Hu et al. [2005] propose an scalable algorithm called CODENSE, for pattern mining in biological networks (e.g. protein-protein interaction, genetic regulatory and co-expression), which is able to mine coherent dense subgraphs and identifies modules, useful for biological applications (to discover unrelated genes). An optimization for gSpan is proposed by Jahn and Kramer [2005] for molecular datasets, reducing the number of subgraphs isomorphisms, taking advantage of the symmetries inherent in molecules. They did tests in the DTP Antiviral Screen (October 99 release) dataset and in the Predictive Toxicology Evaluation Challenge (PTE) dataset (also used by Kuramochi 2001). The symmetries enhance the performance just for the first one dataset.

Vanetik [2010] propose mining graphs with constraints on symmetry and diameter based on tree decomposition. The evaluation was done using two types of datasets. The first two sets with molecular data from the PTE and the DTP Antiviral Screen. The other two sets were synthetic graphs generated using the VFLib library. The results showed that the algorithm using constrains saves approximately 2 orders of magnitude for a support value less than 10%.

Silveira et al. [2015] proposed a model to represent PLI networks that used gSpan to find frequent subgraphs in an attempt to find patterns in PLIs. This proposal was tested in two datasets, CDK2 and Ricin. gSpan generated for both datasets a large amount of very small frequent subgraphs (with one or two edges), making it difficult to discover relevant patterns that can help in ligand binding comprehension.

The above mentioned works showed several and diverse limitations when they model biological data. When graph mining algorithms are used it is necessary to apply an isomorphism algorithm, which is computationally expensive, to be able to analyze the patterns/subgraphs found. Additionally, these patterns do not include biological features, because almost all graph mining algorithms do not allow it. Mining closed or maximal subgraphs could be a solution but remains the problem of pattern reconstruction to be able to analyze its biological meaning. However, as gSpan is the most common used FSM algorithm it was chosen to compare the patterns found by the proposal model (Chapter 4).

1.3.4 Data Visualization

Data Visualization is an interdisciplinary emerging area, which comprises the creation and study of visual representation of data. It aims at finding the most effective way to

communicate discoveries found in a dataset. Also, it allows to identify hidden trends, behaviors and anomalies. There are two main classes of visualizations. The first one is *data visualization and techniques* that offers guidance to create 2D and 3D easily interpretative graphics to obtain knowledge and insights into the dataset analyzed. The second one is *visual data mining tools and techniques* that gives the foundations to create visualizations of data mining models and the patterns identified by the data mining algorithms [Soukup and Davidson, 2002].

Several factors drive the need for visual data in the biological domain. First, the huge complexity, diversity and size of biological databases. A clear example is the exponential growth of the PDB since 1990, which actually has 120.057 molecular structures and 21.224 ligands deposited[5]. Second, the data produced by biotechnology as DNA *microarrays* which are 2D arrays on a solid substrate (lab-on-a-chip) that permits experiments with a large amount of biological material using HTS. Third, a colossal increase in the demand for bioinformatic services, principally to process huge data, as genomes. Finally, the integration of multiple biological data resources requires powerful visualization tools (Figure 1.11) allowing to extract knowledge as patterns and tendencies in complex biological systems [Wang et al., 2005]. Biological networks (i.e. metabolic and genetic) are another example of data that needs to be represented with powerful graphics that allow a better understanding and analysis of them.

There are several visualization tools and software developed specifically for protein-ligand interactions (PLI). One of the most used tools is LigPlot+ [Laskowski and Swindells, 2011], which generates interactive 2D protein-ligand interaction diagrams allowing overlaid them when they have a certain degree of similarity, which is done by the sequence alignment algorithm of Needleman and Wunsch [1970]. Also, it highlights conserved interactions involving protein residues that are in equivalent 3D positions when the two structural models are superposed and show the 3D representation of such diagram in molecular viewers.

PoseView [Stierand and Rarey, 2010] is a software[6] that automatically generates structure diagrams of molecular complexes (containing the amino acids and the interacting ligand). It calculates a collision-free layout that provides information about the interacting pattern modeling amino acids as rigid structures and ligands can be modified in order to allow an intersection-free arrangement of interaction lines. This tool receives as input data from the PDB and docking results generated by FlexX [Rarey et al., 1999], which calculates the position of absent hydrogen atoms. After that it generates the pose diagram, showing only the amino acids that interact with the ligand

[5]Number of macromolecular structures at June 30 2016. Site web: http://www.rcsb.org/pdb/
[6]Available online at: http://poseview.zbh.uni-hamburg.de/

and have an energy contribution of at least 33% (of the maximal energy possible for the interaction type).

LigDig [Fuller et al., 2015] is a web server [7] that links diverse data sources [8] allowing basic manipulations and analyses of protein-ligand complexes. Seven tools compose LigDig, each one can be used separately: (i) text compound search, (ii) query interaction network based on a search for suitable ligands (inhibitors) of a protein, (iii) search for the likely function of a ligand, (iv) batch search for compound identifiers, (v) find structures of protein ligand complexes, (vi) comparison of 3D structures of ligand binding sites and (vii) process coordinate files of protein-ligand complexes for further calculations. Despite LigDig offers diverse services, it has the disadvantage that reusing software do not allow a complete control of the output. Specifically, visualizations of huge interaction networks are drawn with their nodes, edges and its respective labels superposed, creating images where it is not possible to distinguish them clearly.

PLIP [Salentin et al., 2015] is a web service[9] for fully automated detection and visualization of non-covalent protein-ligand contacts in 3D structure. The input can be 3D structure files or PDB files. The output of PLIP are 2D and 3D interaction diagrams and a table with the interaction binding site details.

Initially it was not intended to create visualizations of the proposal model or the patterns found. However, it emerged as an imperative need in the analysis stage of this work. Despite that 3D plots of molecular structures are highly used, 2D plots of PLIs are more useful when the atomic contacts are analyzed. The model proposal is graph-based which allow to create 2D representations of the PLIs. Though these 2D visualizations are different from those generated by all the software tools described in this section and consequently they are not comparable. The proposal model transform the input graphs into other comprised graphs with the aim to facilitate find PLI patterns. This whole process is described in Chapter 3.

1.4 Justification

The four areas mentioned previously offer different approaches to find patterns in PLI networks, as tools and / or algorithms with some limitations. The FSM algorithms, the 2D or 3D visualizations tools for PLI and the network models for PLIs are examples of these approaches.

[7] Available online at: http://mcm.h-its.org/ligdig

[8] Including the databases: ChEMBL, PubChem, and the software programs: cytoscape.js, PDB2PQR, ProBis and Fconv.

[9] https://projects.biotec.tu-dresden.de/plip-web/

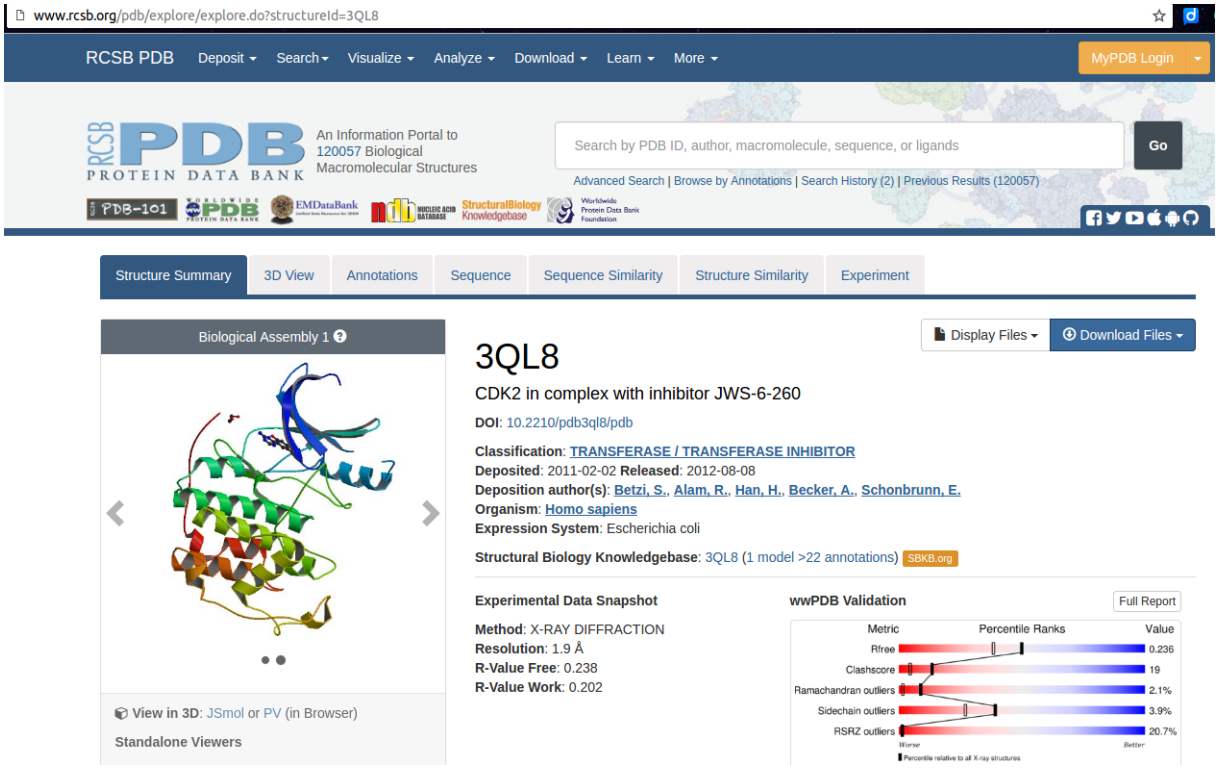

Figure 1.11: Graphical user interface of the PDB web site. Its showing a CDK2 with the identifier 3QL8, which corresponds to a protein complex of the CDK family. On the left side is a 3D visualization of the protein structure. Source: PDB web site. June 30 2016.

The FSM algorithms have several limitations when are applied to biological datasets. The original input data need to be transformed in order to fit the specific model of the FSM algorithm and be able to run it. This implies that the output of the FSM algorithm, which are the patterns found also need to be transformed, returning to the original data domain. The transformation can be defined as a function, which maps from the original domain X to a domain Y, preserving as much as possible the nature of the data. However, the output needs a second transformation to be able to analyze the patterns. This second transformation corresponds to an isomorphism algorithm to identify which graph contents the pattern in the original input. Unfortunately, isomorphism graph detection is computationally expensive, non solvable in polynomial time. On the other hand, subgraph isomorphism is NP-complete [Garey and Johnson, 1979]. The first transformation function is restricted by the graph definition of the FSM algorithm. For example, gSpan allows as input undirected graphs with vertices and edges represented by sequential numbers, admitting one attribute per vertex and edge. The molecule data (e.g. PLI and PPI) does not depend barely on its topology, it is complex and involves other factors as the atom types (defined by experimental

observations), physicochemical properties of the interactions, the Van der Waals effect (attraction between long-range interactions), etc.. Execution time for FSM algorithms applied in biological datasets, usually takes a huge amount of time, because the majority of them are large. Moreover, it is necessary to run the FSM algorithm several times with different support values in order to find interesting patterns according to the biological context.

The network approach allows to model PLI as interaction networks providing global and local metrics to analyze them. Nevertheless, it is not the focus of this area to identify patterns, it aims to find the tendencies followed by a network which are grouped in three main models (Random, Small-World and Free-scale). The majority of biological network studies need to cross references with another databases to obtain interesting inferences and many of these studies are for specific datasets. However, it seems very promising because molecular data can be easily modeled as graphs.

Specific software is continually developed according to biochemists or molecular biologists needs. Many of these software tools require deep biochemistry knowledge to manage them and analyze its results. Visualizations are crucial to analyze and understand the behavior of macromolecules, almost all biochemistry studies includes at least one 2D or 3D visualization as its first approach [O'Donoghue et al., 2003]. Visualization tools, as those mentioned in Section 1.3.1 (i.e. LigPlot+), create useful and powerful graphics but for a limited number of PLIs. This restriction is due to the complexity of the calculations for 3D structural superposition, which is time-consuming [Gallina et al., 2013].

Thus, we decide to combine all these mentioned approaches, eliminating the majority of their limitations, proposing an interdisciplinary strategy to model and find patterns in PLI: CALI (Complex network-based Analysis of protein-Ligand Interactions). This strategy allows us to generate a simple graph-based model represented by a 2D visualization easy to interpret, with the graph components comprising the interaction patterns, which do not require an extensive biochemistry knowledge.

The CALI model is based on compression of the PLI data, with two graphs G' and G". The first one allow us to see how the patterns (frequent and non-frequent) are distributed. The second one is created after applying a merge algorithm, presenting patterns in a more compact view, by grouping edges and nodes according to biological features. We choose two biological datasets for test: CDK2 (Section 3.2.1) and Ricins (Section 3.2.2). This new model does not require running data mining algorithms to find the most common PLIs, because it uses network topological properties (particularly centralities) coupled with a powerful visual and interactive data representation.

1.5 Objectives

1.5.1 General Objective

The objective of this Master Thesis is to identify relevant biological patterns in interaction networks of protein-ligand complexes using a visual and graph-based strategy.

1.5.2 Specific Objectives

The following objectives have been set to fulfill this main goal:

- To propose a graph-based model for protein-ligand interactions (PLI) that allows to identify relevant biological patterns.
- To characterize and analyze the PLI network with different ligands using local and global descriptors from network science (degree, shortest path, node centralities, edge centralities, and so on).
- To propose and implement a strategy to group common protein-ligand interactions based on their physicochemical properties and their atom types, allowing to identify relevant biological patterns in a PLI network.
- To develop a visual strategy to identify relevant biological patterns in a PLI network.
- To evaluate and compare the proposed methodology (the graph model, the algorithm to group common interactions and the visual strategy) results with the state of the art graph mining algorithm.
- To analyze the results and identify its biological meaning.

1.6 Contributions

The contributions of this Master Thesis are:

- A graph-based visual model for representing protein-ligand interactions (PLI).
- A strategy for visual-pattern discovery in protein-ligand interaction networks.
- A strategy that groups common interactions based on their physicochemical properties and their atom types, from the PLI dataset.

- A prototype tool for 2D visualization of protein-ligand interaction.
- Information about relevant biological patterns and its localization in the protein-ligand interaction dataset.

1.7 Organization of the Document

This Master's Thesis has 5 chapters. Chapter 1 is the introduction, including a biological background and related works for biochemistry, network science, graph mining and data visualization. Chapter 2 describes and explains the main concepts of network science, as global properties and metrics. The last section of the Chapter 2 is dedicated to describe well-known biological networks. Chapter 3 explains the proposed model, which we named CALI. Also, we formalize the PLI graph model and the pattern definition. Furthermore, the datasets used in this study are described. Chapter 4 shows the results of the Thesis, divided in three sections. In the first section is done a network analysis according to the metrics and properties calculated. This analysis is complemented by visualizations of the PLI graphs. The second section makes a comparison of the patterns found by CALI with the patterns found by the frequent subgraph mining paradigm. In the third section are discussed the patterns found by CALI with those found in experimental papers, which used the same datasets. Finally, Chapter 5 presents the conclusions of this work.

Chapter 2

Complex Networks

A network, also called a *graph* in the mathematical literature, is a collection of vertices joined by edges. According to Diestel [2000], a graph is a pair $G = (V, E)$ of sets such that $E \subseteq [V]^2$. Thus the elements of E are 2-element subsets of V. The elements of V are the vertices of the graph G and the elements of E are its edges. To avoid notational ambiguities, we assume that $V \cap E = \emptyset$. In diverse areas such as physics, sociology, chemistry and biology, many interactions can be understood as networks [Newman, 2010]. Complex networks is a term relatively new, used to describe real dynamic complex networks with a particular topology and with non-trivial properties [Boccaletti et al., 2006].

In this chapter, we present and define network properties (Section 2.1), including global properties (Section 2.1.1) and some of the principal metrics (Section 2.1.2) which are the basis to characterize networks through models. In addition, we describe bipartite graphs (Section 2.2) and biological networks (Section 2.3), which are concepts important to understand the proposed model for protein-ligand interactions (Section 1.2.1.6).

2.1 Network Properties

Complex networks usually represent dynamical systems through graphs. A huge number of the network elements interact nonlinearly, bringing emergent properties. The characterization and analysis of graphs representing networks is vital to understand the system behavior. Network properties allow to characterize and, in some cases, to predict the system behavior. Understanding the collective behavior of different levels of biological organization (cells, tissues and organisms) in terms of their component properties is one of the most important challenges in biological sciences [Junker and

Schreiber, 2008]. This section describes diverse measures and network topology characteristics, frequently used in complex network analysis.

2.1.1 Global Properties

The global properties of a network allow us to understand the topology and the internal structure, which makes possible to fit the network into a well-known model (Random, Small-World and Scale-free).

2.1.1.1 Distance

The distance d_{ij} between two vertices in the network is the minimal number of edges that need to be traversed from the initial vertex (v_i) to the final vertex (v_j). This definition is most common known as the *shortest path*. For example, in Figure 2.1, the distance between vertices A and F is 5, because there are 5 edges connecting them. However, A is directly connected with B, which means a distance 1. All distances for every pair of vertices are in Table 2.1.

Table 2.1: Matrix of distances for graph in Figure 2.1

A	0						
B	1	0					
C	2	1	0				
D	3	2	1	0			
E	4	3	2	1	0		
F	5	4	3	2	1	0	
G	5	4	3	2	1	2	0
	A	B	C	D	E	F	G

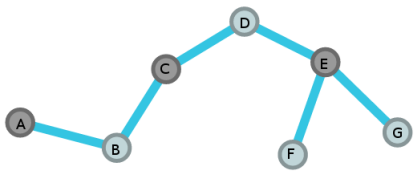

Figure 2.1: Example of a graph. This graph corresponds to one of the components of the CDK2 PLI (protein-ligand interaction) network. These nodes represent protein atoms: A is GLN131:O, C is ASN132:OD1 and E is ASP145:OD1. The nodes B, D, F and G represent atoms from different ligands.

2.1.1.2 Diameter

The diameter is defined as the maximal distance of any pair of vertices of a graph. Another definition for diameter is the maximum eccentricity (this metric is explained in Section 2.1.2.5).

$$d_m = max(d_{ij}) \tag{2.1}$$

For the graph in Figure 2.1 the diameter is 5, because it is the longest distance between the vertices (A-F and A-G).

2.1.1.3 Average Path Length

The average path length of a network is defined as the average distance of all pairs of vertices in G. It is also called *average shortest path length* and *characteristic path length*. The well-known algorithms to calculate *shortest paths* are the Dijkstra and Floyd-Warshall [Bang-Jensen and Gutin, 2007]. For the graph in Figure 2.1 $L = 1.19$, according to Equation 2.2. This value is the sum of all distances for every pair of vertices in the graph (see Table 2.1), multiplied by the fraction of $N_v - 1$ times the number of vertices (N_v).

$$L = \frac{1}{N_v(N_v - 1)} \sum_{i,j \in \mathbb{N}, i \neq j} d_{ij} \tag{2.2}$$

2.1.1.4 The Degree Distribution

This is one of the most important properties because defines the network structure, which means the frequency distribution of vertex degrees. The degree of a vertex v_i is the number edges adjacent to it. The degree distribution represented by p_k can be defined as the fraction of vertices in a network that have degree k [Newman, 2010]. The degree distribution can also be seen as the probability p_k that the *degree* of a randomly chosen vertex in the network equals k. The distribution degree is commonly analyzed making a plot (see Figure 2.2) of the values as a function of k, specially when the network is large. Many empirical networks, such as social networks, have a peculiarity related with their distribution p_k: follow a *power law*. These networks have the majority of their vertices with low degree, meanwhile a small number of vertices are highly connected, conforming *hubs*(Figure 2.3).

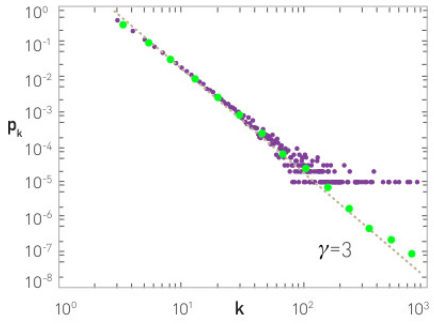

Figure 2.2: Degree distribution (p_k) for a network with 100.000 vertices generated by the scale-free model. k represents the degree values of the network vertices. This plot shows the p_k linearly-binned (purple) and log-binned (green). The log bins (green) allow to identify easily the scale-free behavior, because the bins tend to form a straight line. Source: Network Science. Barabasi. May 2016.

Figure 2.3: Component of the CDK2 PLI (protein-ligand interaction) network. The graph representing this component is also known as *star graph* or *hub*. The central node is representing an atom of the CDK2 protein complex, which is an oxygen of the glutamic acid (GLU) in the position 81, summarized as GLU81:O.

2.1.1.5 Power Laws

There are degree distributions that follow the form described in Equations 2.3 and 2.4. The variation occurs as power of k, hence these distributions receive the name of *power laws*. The power-law degree distributions happen in non-homogeneous networks and follow a straight line when they are plotted in a logarithmic scale. The logarithm of the degree distribution p_k is a linear function of degree k [Newman, 2010].

$$ln\, p_k = -\alpha\, ln\, k + c \tag{2.3}$$

Taking the exponential of both sizes:

$$p_k = Ck^{-\alpha} \tag{2.4}$$

The majority of real networks, as social networks, have their exponents varying in the range $2 < \alpha < 3$. Values outside this range are possible but observed occasionally.

A true power-law distribution is monotonically decreasing over its entire range (Figure 2.4). However, for small values of k, the degree distribution presents fluctuations from the expected behavior of a power-law. Additionally, it is very common that the power-law is only followed in the tail of the distribution, where values of k are higher.

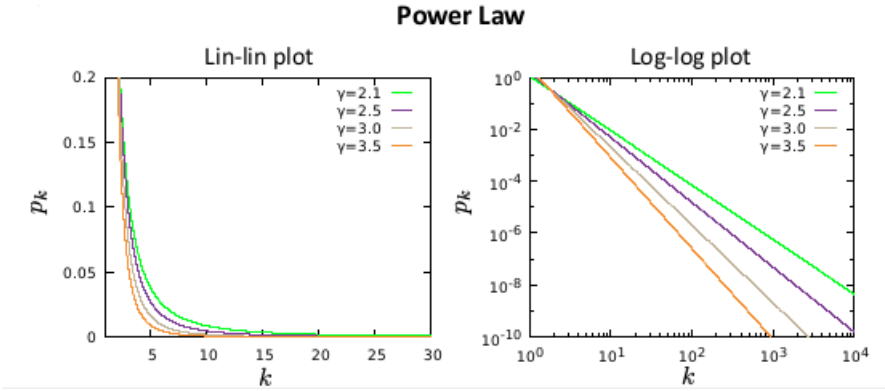

Figure 2.4: Power law distributions with different values for the exponent (γ). The plot at right is in linear scale and the plot at left is in logarithmic scale. Source: Network Science. Barabasi. May 2016.

2.1.1.6 Clustering Coefficient

The clustering coefficient measures the probability that two vertices with a common neighbor have an edge between them. This property is related to the local cohesiveness and transitivity of a network, helping to understand the internal structure [Junker and Schreiber, 2008]. A simple definition of the clustering coefficient of a vertex i is given by:

$$C_i = \frac{E_i}{E_{max}} \tag{2.5}$$

where E_i is the number of edges between v_i neighbors and E_{max} is the number of all possible edges between v_i neighbors. E_{max} can be defined in terms of the number of neighbors (k_i) of v_i:

$$C_i = \frac{E_i}{\frac{1}{2}(k_i(k_i - 1))} \tag{2.6}$$

C_i is known as the *local clustering coefficient* for a single vertex (Figure 2.5).

Another way to calculate local clustering coefficient is by counting the number of triangles, but it does not make sense for *bipartite networks* (it is impossible to have

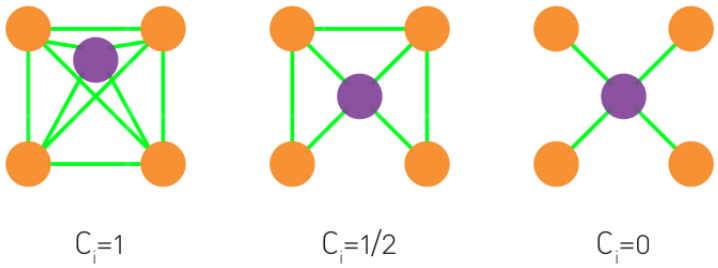

Figure 2.5: Clustering coefficient for the central node of three graphs. Source: Network Science. Barabasi. May 2016.

triangles on them) and it is not described here, because we represented protein-ligand interactions as a bipartite network.

The *global clustering coefficient* C of the whole network is the average cluster coefficient of all vertices. A network with a high value for C indicates a local cohesiveness and a disposition to form groups.

2.1.2 Centrality Metrics

Centrality metrics are measures that aim to characterize networks by characterizing edges and vertices with respect to their position inside the network. The centrality of a graph G, is a function that assigns a numerical value to each vertex of G. These numerical values are useful to create vertex rankings depending on the computing concepts used to calculate the centralities. Also, it permits pairwise vertex comparison by comparing their values. A vertex is more important or central than another if the centrality value of the first one is greater than the second one [Junker and Schreiber, 2008]. When network analysis is done by using centralities, it is important to consider that comparison among centrality values is only allowed inside the same network and some measures can only be calculated in connected networks. The majority of the centralities explained here are based on the concept of *shortest path* (Section 2.1.1.3). The shortest path of a graph is the minimum distance between a pair of nodes i and j. Besides, it is very common to represent the edges between nodes with a matrix. The adjacency matrix of a graph G has size n x n, where n is the number of nodes, and the position a_{ij} is filled with 1 if the vertices i and j are linked, otherwise it is filled with 0. In the next subsections we present the centralities most used in literature and applied to this work.

2.1.2.1 Degree

The simplest centrality metric is the degree of a node, which is the number of edges connected to it. [Newman, 2010]. This metric is equivalent to the number of neighbors of the node. For example, in the Figure 2.6 the degree of the central node is 14, which means that this node is central and important (connecting all other nodes), because without it the network would not exist.

Figure 2.6: Hub corresponding to one of the components of the CDK2 PLI (protein-ligand interaction) network. The central node of the hub (dark gray) has degree 14, meaning that it is linked to 14 nodes that represent ligand atoms (light gray). This node represents a nitrogen (NZ) atom, part of a lysine (LYS) residue, which is in the sequence position 33. Briefly, this is resumed in LYS33:NZ.

2.1.2.2 Node Betweenness

The node betweenness centrality measures the extent to which a node lies on paths between other nodes. This metric allows to identify important nodes with low degree, which also work as a bridge, joining two or more groups. The betweenness centrality of a node i is the number of all shortest paths in the graph that pass through i:

$$x_i = \sum_{st} n_{st}^i \tag{2.7}$$

s and t are nodes of the graph and if there is no path between them n_{st}^i is zero. In the Figure 2.6 the node betweenness is 1 for the dark grey node, because all paths pass through it, making it the most central node. A high betwenenness as 1 (in a normalized scale) is considered high and identifies the most central nodes of the network. On the other hand, a low value for betweenness, near to zero, implies that the vertex do not concentrate a high number of paths, making it less important principally for communication goals.

2.1.2.3 Edge Betweenness

This metric shares the betweenness concept with Section 2.1.2.2, it measures the extent to which an edge lies on paths between nodes. It can be defined as the number of shortest paths between pairs of nodes that pass through the measured edge [Bang-Jensen and Gutin, 2007]. Edge betweenness allow to identify the most important pairs of vertices communicating essential parts of the network. In the Figure 2.6, all edges have the same value of betweenness, because every edge is in just one shortest path. Besides, in Figure 2.1 the edges *C*-*D* and *D*-*E* have the highest betweenness (0.55) because several paths pass through them.

2.1.2.4 Closeness

Measures the mean distance from one node to other nodes. Suppose $d_{i,j}$ is the length of a shortest path between nodes i and j, then the mean shortest path distance from i to j, averaged over all vertices j in the network is, according to Newman [2010]:

$$l_i = \frac{1}{n} \sum_j d_{ij} \tag{2.8}$$

The mean distance l_i is not considered a measure of centrality as the measures discussed above because it gives low values for more central nodes and high values for less central ones. To circumvent this problem, researchers from social networks community calculate the inverse of l_i, which is called *closeness centrality Ci*. In Figure 2.6, the closeness for the central node is 1 (normalized scale), because it is the only node connected to the rest of nodes and also, it is the nearest to all of them. Additionally, these nodes have all the same value for closeness (0.5).

2.1.2.5 Eccentricity

Eccentricity is the maximum distance between a node s and any reachable node t (the opposite definition for closeness) of the graph [Junker and Schreiber, 2008]. The distance between two nodes that are not connected is defined as infinite. In this case, eccentricity can be calculated for the graph connected components. This metric allow us to identify the most far vertices of a network. The eccentricity centrality for s is defined by the formula:

$$ecc(s) = \frac{1}{max\{dist(s,t) : t \in V\}} \tag{2.9}$$

If an eccentricity value is high, it implies that the vertex needs a long path to reach the most distant vertex from it. For example in Figure 2.1 the eccentricity for nodes A, F and G is 5. Meanwhile, for nodes B and E is 4 and for nodes C and D is 3.

2.1.2.6 Communicability

The network communicability is the sum of closed walks of all lengths starting at a node s and ending at a node t. Using the graph spectrum notation in the generalization proposed by [Estrada and Hatano, 2008], where $\phi_j(s)$ is the s th element of the j th orthonormal eigenvector of the adjacency matrix associated with the eigenvalue λ_j , the communicability between the nodes s and t is given by:

$$c(s,t) = \sum_{j=1}^{n} \phi_j(s)\phi_j(t)e^{\lambda_j} \tag{2.10}$$

The communicability function express how the propagation from one node to another occurs. This extension of the communicability definition done by Estrada and Hatano [2008] permits to obtain global and local information about the network structures. This metric also allows to identify communities as sets of nodes displaying large internal communicability. A community characterize a group of nodes that communicate better among them than with the rest of the network nodes. Metrics only based in shortest paths are not very perceptive with structural bottlenecks in the network.

Largest values for communicability usually take place among nodes with high degree and low communicability occurs between nodes with low degree. For example, in Figure 2.1 the node E has the highest communicability centrality (2.95), this node also has the highest degree (3). The lowest value for communicability is 1.6 for node A, which has the lowest degree (1).

2.2 Bipartite Networks

A graph $G = (V, E)$, where V is the vertex set and E is the edge set is known as a *bipartite graph* or *bigraph* if there is a partition in $V = S \cup T$, such that every edge in E has one end-vertex in S and one end-vertex in T [Junker and Schreiber, 2008]. This kind of graphs are commonly used to represent entities and the groups they belong, which are connected by edges representing interactions between them [Newman, 2010]. In network theory literature, *bipartite networks*, *affiliation networks* and *two-mode networks* [Latapy et al., 2008] are all referencing the same type of network, representing

it with a bipartite graph and they are usually found in the context of membership of groups.

There are two types of vertices (S and T) in bipartite networks, which constitute two disjoint vertex sets that do not establish edges between the same set (Figure 2.7). The adjacency matrix for the graphs representing this kind of networks is called *incidence matrix*. Being n the number of vertices in one group and g the number of vertices in the other group, the *incidence matrix* B has a size $g \times n$:

$$B_{ij} = \begin{cases} 1 & \text{if vertex } j \text{ belongs to group } i \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

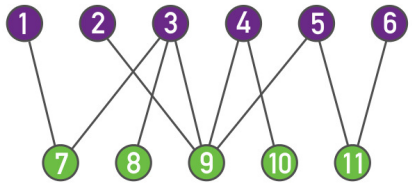

Figure 2.7: Graph representing an affiliation or bipartite network. The network has 11 nodes: 6 purple and 5 green. However, there are no edges between nodes with the same color (membership). Source: Network Science. Barabasi. May 2016.

A notable example of a bipartite network in Medicine is the *Human Disease Network* (the *diseasome*), Figure 2.8, which shows how diseases and genes are connected [Barabási, 2016].

2.3 Biological Networks

Networks are used in biology to represent interactions between biological elements and are known as **biological networks**. These can be classified according to the scale they exhibit: *macroscopic* or *microscopic*, as proposed by Junker and Schreiber, 2008.

Macroscopic networks involve organisms and interactions between them. In this case, nodes represent organisms and edges represent interactions between these organisms. Two famous examples of this kind of networks are Ecological (food webs) and Phylogenetic (i. e. genealogical or family trees).

Microscopic networks are in at the molecular level, as the biochemical networks, which have been attracted attention during the last years. These networks represent molecular patterns of interaction and mechanisms of control in the biological cell. They model biochemical processes through graphs, where nodes are molecules and

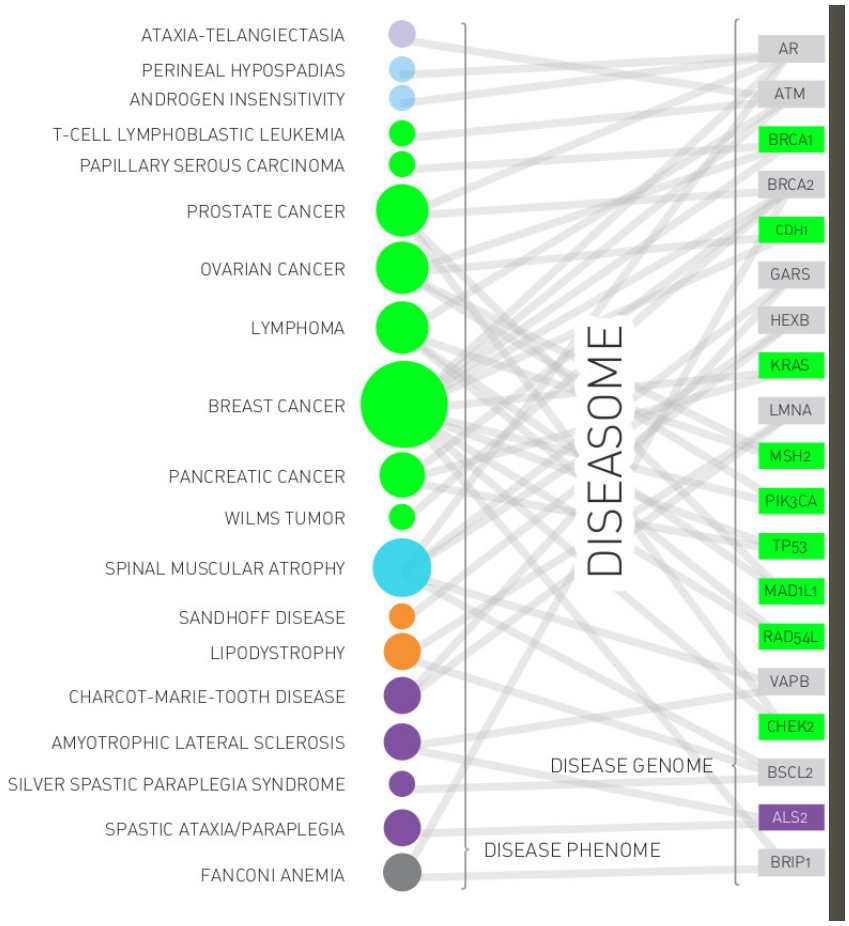

Figure 2.8: The Human Disease Network. Nodes are diseases (circles) and genes(rectangles). An edge appear between them if a mutation in a gene affect a particular disease. Source: Network Science. Barabasi. May 2016.

edges are interactions between them. Representative examples of this kind of networks are metabolic networks, genetic regulatory networks and protein-protein interaction networks.

The metabolic networks represent all the chemical reactions necessary for the cell to break down food and build biological molecules to complete its tasks. This type of network is properly represented by directed graphs with nodes as the chemical products consumed and produced (known as metabolites) and the edges represent the direction of the reactions. The genetic regulatory networks (GRN) allow to understand interactions between genes in a cell. The vertices in this network are proteins or the genes that code them and a directed edge indicates that one gene controls another. The protein-protein interaction (PPI) network is the set of all chemical interactions between proteins. The vertices represent proteins and the edges represent interactions between such proteins [Newman, 2010]. In a similar manner, protein-ligand interaction

(PLI) networks model interactions between proteins and their ligands. In the next section we explain PPI and PLI networks in detail, as they involve concepts closely related to our work.

2.3.1 Protein-Protein Interaction Networks

Proteins interact among them and with other biomolecules. The interactions are usually chemical (exchanging small subgroups), but some are physical (without exchanging subunits), specially between a protein and another. Protein interactions occur due to physico-chemical forces such as electrostatic interactions, hydrogen bonds, van der Waals attraction and hydrophobic effects. These particular interactions create protein complexes by folding its shapes to interlock. A graph representing a protein-protein interaction (PPI) network has nodes as proteins and each edge between a pair of nodes means that these two proteins interact. A famous PPI network is the baker's yeast (Figure 2.9). In this network a vertex with high degree corresponds to an important protein for the organism survival [Junker and Schreiber, 2008]

There are several experimental techniques available to probe interactions between proteins as immunoprecipitation, co-immunoprecipitation, the high-throughput (such as the *two-hybrid screen*) and the affinity purification methods (such as the tandem affinity purification) [1]. These methods, even the most accurate and efficient of them, require time and laboratory experiments to be done. In some cases, when the function of a protein is unknown, it can be discovered by applying the information inside a PPI network and using computational techniques from machine learning, such as clustering [Zaki et al., 2013].

Global analysis of the networks formed by PPI is essential to understand biological mechanisms and disease processes. PPI study contributes to understand the alterations caused by certain diseases and can elicit a potential therapeutic strategy. The information inside a PPI network allows to construct protein-linkage maps on a genome wide scale, which is known as *interactome*. The *interactome* is the whole set of PPI inside a cell. Actually, the interactome is still under construction, as its complexity is greater than the genome, but it is crucial for a complete understanding of biology [Bonetta, 2010].

[1] Detailed description of these methods and techniques is available in Section C.

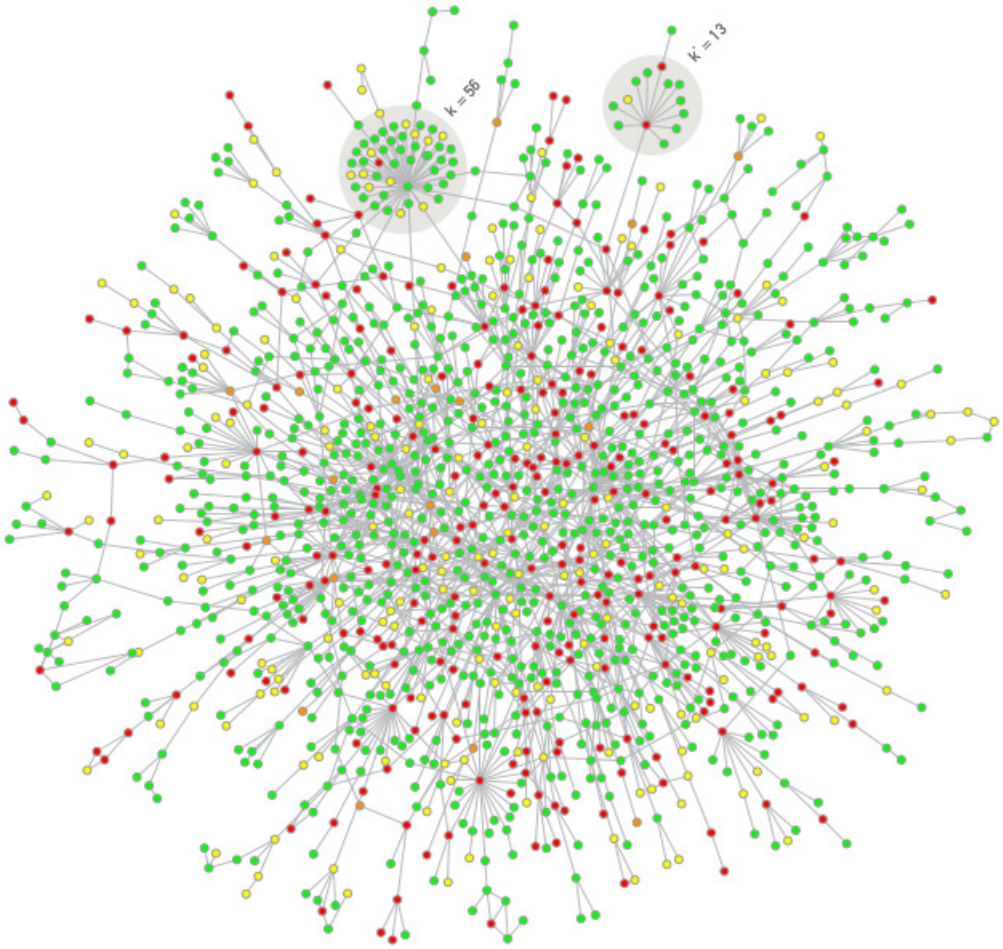

Figure 2.9: The protein interaction map of yeast (*Saccharomyces cerevisiae*). It is shown only the largest component of the network, which is composed by 1870 nodes and 2277 edges (interactions). The red nodes represent proteins that are vital for the organism and the green are not. Source: Network Science. Barabasi. May 2016.

2.3.2 Protein-Ligand Interaction Networks

The understanding of protein-ligand interactions is key for molecular recognition. The study of protein-ligand complexes allows scientists, as chemists, to discover new drugs. In the areas of network science, biochemistry, bioinformatics and chemoinformatics literature, to the best of our knowledge, the PLI networks are mentioned in PPI network studies or referred as special cases of PPI networks. The acronym PLI (protein-ligand interaction), appear in the work of Gallina et al., 2013, who create a web-based tool for comparison of protein-ligand interactions, and in the work of Medina-Franco et al., 2014, who did a compilation of PPI and PLI studies.

The network approach applied to PLI networks surge as a plus to collaborate in the analysis of these interactions. The PLI network can be easily build by using a

graph, modeling interactions as edges and the protein and the ligand atoms as nodes. Another possibility to model a PLI network as a graph is to define residues as nodes and the interactions between them as edges. Actually, there is no consensus for PLI models, it relies on the author's goal.

The visualizations of protein-ligand complexes are used as a first impression to obtain information about their structure [O'Donoghue et al., 2003]. These visualizations represent the molecules forming the protein-ligand complex. The majority of PLI studies begin with a 3D visualization of the protein complex. However, 2D visualizations are usually generated in an attempt to simplify the structure analysis and increase the understanding for non-experts in molecular modeling. Both types of visualization are done with specialized software tools as LigPlot+ [Laskowski and Swindells, 2011], PoseView [Stierand and Rarey, 2010], MOE [Clark and Labute, 2007] and Maestro [Laimer et al., 2016].

Chapter 3

Methodology

This Chapter explains **CALI: Complex Network-Based Analysis of Protein-Ligand Interaction**. The datasets (Section 3.2) used to analyze this proposed model are CDK2 and Ricin. A scheme of the model is in Figure 3.1. The web prototype application is available at: www.lbs.dcc.ufmg.br/projetos/cali.

3.1 The Strategy: a General View

A protein-ligand interaction (PLI) complex is usually formed by two units: a protein and a ligand. Extracting data from the PDB for a particular protein complex enables to reconstruct their contacts at an atomic level (Section 3.3). Protein-ligand contacts can be seen in a simple manner as a *link* between a protein atom and a ligand atom. Graphs allow to model this type of data and observe global and local characteristics. When PLI are modeled as graphs, they tend to form tiny subgraphs, commonly with two or three edges. The majority of interactions in these complexes are protein-protein type, which means contacts between two protein atoms. There are less contacts between a protein and a ligand atom, because a ligand is a smaller molecule (around ten atoms) than a protein (in average a protein can have 500 amino acids [1] and each one have 19 atoms on average [2] [Lefranc et al., 2009]). Hence, PLI graphs for a specific protein complex are usually highly disconnected. Also, when the ligand is an inhibitor there is a small amount of them developed or discovered for a specific protein. The PLI graphs are commonly analyzed by protein families or functional groups, which are composed by tiny subgraphs with repeated protein vertices, when their sequences have

[1] yeast proteins are on average 466 amino acids.

[2] The smallest (Alanine) amino acid have 13 atoms and the biggest (Tryptophan) have 27 amino acids, the complete table with the 20 amino acids can be found in: www.imgt.org/IMGTeducation/Aide-memoire/

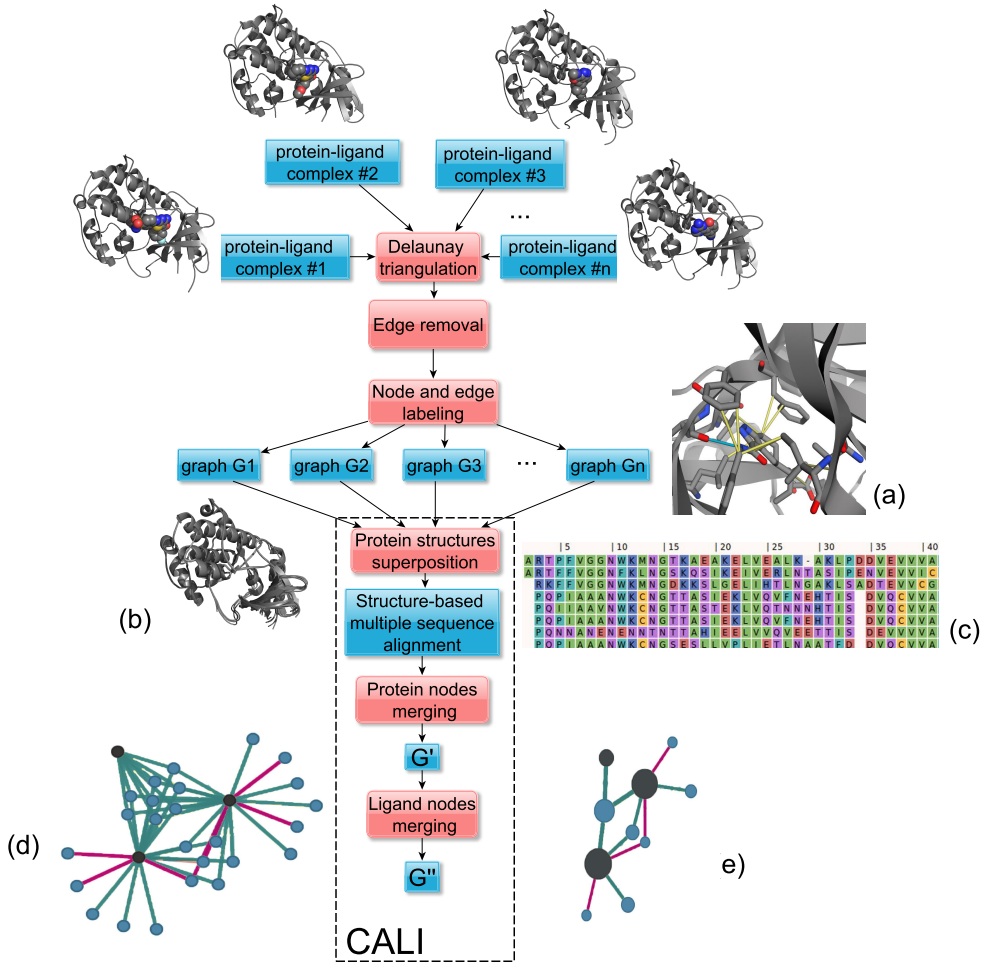

Figure 3.1: CALI workflow. This diagram includes the major steps to create the proposal models G' and G'', basis of CALI. (a) Collect the data from the PDB, pre-process the protein-ligand complexes to calculate their atomic contacts using the Delaunay triangulation, remove the protein-protein interactions and construct graphs only with protein-ligand interactions. (b) Superposition of protein complexes with similar 3D structure. (c) Run the multiple sequence alignment algorithm for protein complexes with different 3D structure. (d) Graph G' with merged protein nodes. (e) Graph G'' based on G' with merged ligand nodes and edges grouped by interaction types.

high similarity (over 80%). Next, we will briefly comment on some strategies used to investigate PLIs.

The first approach, before the conception of CALI, was to apply graph theory and data mining algorithms to investigate if there are patterns in PLIs. Classical graph algorithms in PLI graphs to find communities (relevant groups) as clique percolation clustering [Derenyi et al., 2005] or the Girvan-Newman algorithm [Newman and Girvan, 2004] based on betweeness, can not be always used, because these algorithms first calculate *paths* or a *clique* of a specific size. However, many paths between pairs of

vertices do not exist neither cliques with the required size. A similar problem happened with spectral clustering, because its graph adjacency matrix has infinite distances for pairs of vertices without a path. Other cluster algorithms, such as DBSCAN, are seriously affected by the imbalance of the PLI data, where the majority of interactions (in our datasets) are hydrogen bond and hydrophobic, creating huge clusters which can not classify accurately another type of interactions, such as aromatic stackings.

The second approach was based on the *egonet* concept, proposed by Akoglu et al. [2010], which compares neighbors of networks centered in one node (ego) that aim to discover abnormal networks, which are those highly different from others. The comparison is done with the four features or new power laws discovered by the same authors, which were calculated for a PLI dataset. However, the interpretation of the results was hard (one feature produced complex numbers with an imaginary part). Also, to create definitions of normal and anormal for PLIs is hard.

However, in many cases it is necessary to analyze a set of protein complexes and, to avoid the problems described above, we propose CALI: Complex Network- Based Analysis of Protein-Ligand Interaction. The Section 3.2 describes the PLI datasets. The calculation of atomic contacts of the PLI datasets are presented in Section 3.3. The graph models G' and G", which are the basis of CALI, are explained in Section 3.4.

3.2 Datasets

In this section we discuss the two datasets of PLI modeled using CALI. Here we detail how CDK2 and Ricin datasets were obtained (from Protein Data Bank [Rose et al., 2015] in August 2014). The list of PDB identifiers for each dataset is provided in Table A.3 for CDK2 and Table A.2 for Ricin.

3.2.1 CDK2

This dataset is based on the work of Schonbrunn et al., 2013, and it comprises a specific protein for which several inhibitors are known. This same protein was crystallized with a variety of ligands and the 73 experimental structures are available in PDB. In the mentioned work, authors experimentally determined important residues for PLI, so this dataset is also used to check if our proposed model CALI is able to find relevant interaction patterns by comparing CALI patterns with those experimentally obtained.

Schonbrunn and colleagues in their work depict how they discovered by high-throughput screening the compound 2-(allylamino)-4-aminothiazol-5-yl-(phenyl) methanone as a potent inhibitor of the human CDK2. Through the co-crystal structure

(PDB id 3QQK), they show the importance of hydrogen bonds in the binding of this compound with the ATP site. The hydrogen bonds occur between the thiazolamine moiety and the hinge region (GLU81-LEU83).

Departing from previously cited compound, the authors developed other 95 analogues by replacing systematically the flanking allyl and the phenyl moieties whereas the aminothiazole core was maintained unchanged to preserve its functionality. Thenceforth, they evaluated analogues concerning their inhibitory potential, but only 35 from these analogues had their crystal structure determined.

Besides these 35 structures, we have found another 38 related structures (totaling 73 PDB files) by searching on the PDB[3] website. These files are supposed to be discussed in another work from the same authors which is not published yet to the best of our knowledge.

3.2.2 Ricin

This dataset is composed by 29 experimental structures from the PDB which have at least one ligand and 50% or more identity with Ricin (2AAI, chain A in PDB). We consider this dataset a more realistic (or frequent) one, as the sequences are not exactly the same, which is common, for instance, in a protein family. Thus this dataset was chosen to show the applicability of the proposed model CALI for a diverse, real-world protein set by comparing CALI patterns with those experimentally determined by Ho et al., 2009.

We searched the PDB for key words *ricin*, *ricin-like* and *ribosome inactivating protein* and obtained 136, 126 and 163 results respectively. As there was overlap among results, the total number of different PDB entries was 266.

Sequences from all 266 PDB entries were split by chain using PDBest tool [Gonçalves et al., 2015] and were aligned against PDB id 2AAI [Rutenber et al., 1991] chain A, which we call 2AAI.A, using an in-house implementation of Needleman-Wunsch algorithm [Needleman and Wunsch, 1970]. PDB entry 2AAI.A is the catalytic subunit of ricin toxin without any ligands. Those 47 structures which have 50% or more identity were taken as our initial Ricin dataset.

The final step to obtain the Ricin dataset was to select entries which have at least one ligand, as we were interested in patterns of interactions between a protein and its ligands. So we computed probable PLI at atomic level using a geometric approach (which is detailed in Section 3.3) to determine ligands that were interacting with protein residues. Only ligands with seven or more atoms were considered, in a

[3]http://www.rcsb.org

similar manner to the work of Pires et al. [2013]. This process resulted in 29 PDB chains.

3.3 Pre-processing Datasets

The graph for each protein chain and its ligands was computed through a cutoff-independent and geometric-based approach which consists of building a Voronoi diagram and its dual graph, the Delaunay triangulation [Poupon, 2004; Okabe et al., 2009]. In a Delaunay graph, edges connecting two nodes represent atoms that are likely in contact and not occluded by other atoms (Figure 3.1-a). For this computation, CGAL library [4] was used. The Delaunay graph was post processed by keeping only edges connecting protein and ligand atoms, which are the interest of this research, generating a bigraph.

Thereafter, by using the generated contacts, edges are labeled according to the distance between atoms and their physicochemical properties. Atoms were classified as acceptor, aromatic, donor, hydrophobic, negative (anion) or positive (cation) in accordance with [Sobolev et al., 1999a]. The chemical properties of each protein atom were obtained from previous works [de Melo et al., 2006] while the ligand atom properties were computed by using Pmapper software from Chemaxon (Pmapper 5.3.8, 2010, Chemaxon [5]) at pH 7.0.

Finally, interactions were labeled by considering each physicochemical property and a distance criterion [Mancini et al., 2004; Silveira et al., 2015]. The distance criterion for each type of interaction is provided in Table 3.1. This pre-processing data was done by the Bioinformatics PhD student Alexandre V. Fassio from UFMG, as part of his master Thesis in Bioinformatics [Fassio, 2015].

Table 3.1: Criteria used to compute the interactions.

Type of interaction	Atom types	Min. distance	Max. distance
Aromatic stacking	two aromatic atoms	1.5	3.5
Hydrogen bond	an acceptor and a donor atom	2.0	3.0
Hydrophobic	two hydrophobic atoms	2.0	3.8
Repulsive	two atoms with the same charge	2.0	6.0
Salt bridges	two atoms with opposite charge	2.0	6.0

[4] http://www.cgal.org
[5] http://www.chemaxon.com

3.4 PLI Models: G' and G''

3.4.1 Graph Model for One Protein-Ligand Complex

Denoting $I = \{(p_1, l_1), (p_2, l_2), (p_3, l_3), ..., (p_n, l_m)\}$ as the interaction set composed by two subsets, $V_P = \{p_1, p_2, p_3..., p_n\}$ as protein atoms, and $V_L = \{l_1, l_2, l_3..., l_m\}$ as ligand atoms, an edge e_{ij} of the PLI graph G is defined as a pair of a protein atom and a ligand atom $e_{ij} = (p_i, l_j)$, which means that these two atoms are linked configuring a contact. $G(V_P, V_L, E)$, by its definition, is a bipartite graph where the set of edges, $E = \{e_{ij} : p_i \in V_P, l_j \in V_L\}$, are only possible between a ligand atom l_i and a protein atom p_j that are able to form a chemical bond. Thus, $I = E$. The edges from set E have as attributes the names of the two atoms in contact, the interaction physicochemical type, the contact distance (in Angstroms) and the edge betweenness centrality (the only network metric calculated for edges). The nodes from sets V_P and V_L have as attributes the atom type and the atom name. The calculated network metrics (Section 2.1.2) are used also as attributes for both vertex sets. The graph $G(V_P, V_L, E)$ (Figure 3.2) represents interactions for one protein-ligand complex.

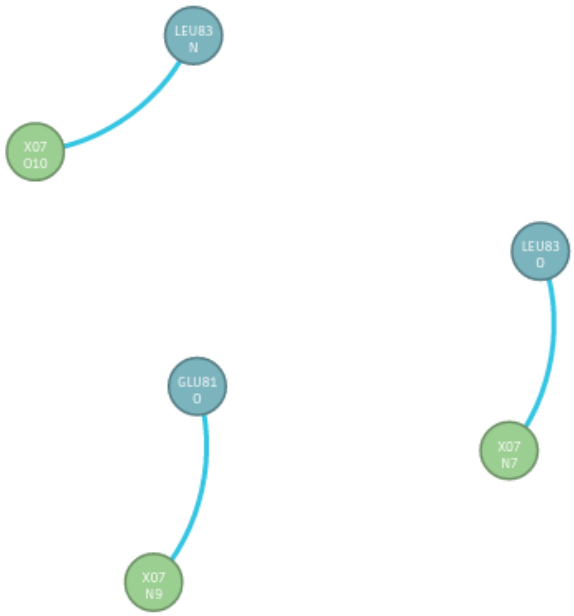

Figure 3.2: Set I of protein-ligand interactions of the protein complex CDK2, corresponding to the structure 3QQF from the Protein Data Bank (PDB). Source: http://www.napoli.dcc.ufmg.br/, a bioinformatic tool for protein-ligand visualization.

3.4.2 Graph Model Formalization

In this Section, we will formally state CALI model. Let γ be a set of graphs, $\gamma = G_1, G_2, G_3, ..., G_n$ where n is the number of graphs in the input dataset (each graph G_i represents a protein-ligand complex as in Section 3.4.1), which can be seen in Figure 3.1-a. Each graph i, $G_i = (V_{Pi}, V_{Li}, E_i)$ is composed by a set of nodes, V_{Pi}, representing protein atoms, and V_{Li}, representing ligand atoms. Also, each graph has a set of edges, E_i, that are interactions between nodes in V_{Pi} and in V_{Li}, so we have a set of bipartite graphs. The problem is then finding frequent subgraphs in the set γ .

Our model consists of building a new graph, $G' = (V'_P, V_L, E)$ where V'_P is a new set of nodes comprised by compositions of nodes in V_{Pi} $\forall i \in n$. This is described by a transformation function $T_{SA}(V_P) = V'_P$. We define the function T_{SA} as a multiple sequence alignment (Figure 3.1-c) on the set of protein vertices $V_P = \{V_{P1}, V_{P2}, V_{P3}, ..., V_{Pn}\}$, where every subset V_{Pi} corresponds to the nodes of a protein complex. The idea behind this structural superposition (Figure 3.1-b) is to join together protein residues to form a connected graph of PLI, where V'_P is a new vertex set, composed by protein atoms grouped by alignment residue positions. An example of this process is provided in Figure 3.3. Using this new model or graph G' (Figure 3.1-d), we see at a first glance all the interacting protein and ligand atoms, as well as conserved patterns as big components. Residues establishing a high number of interactions and its atoms simply emerge as hubs in the network. We define this whole process of generating G' as *graph superposition*.

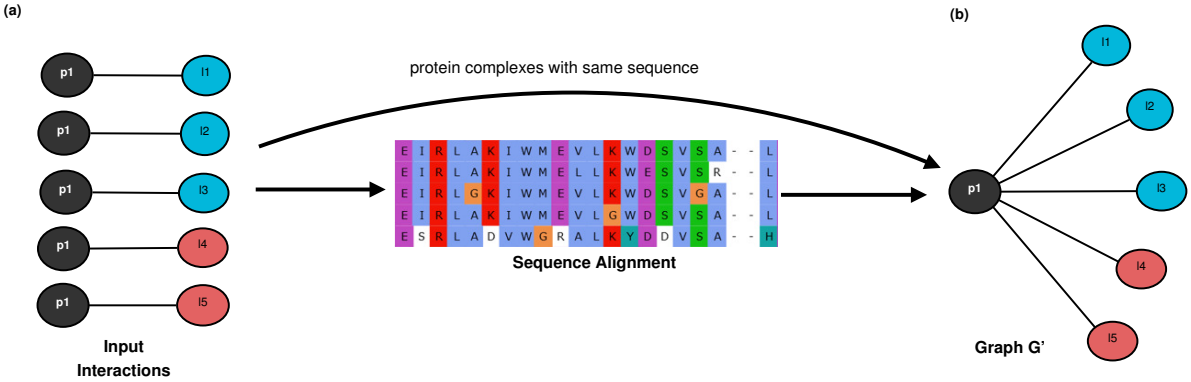

Figure 3.3: Transformation of protein-ligand interactions. **(a)** An interaction I is composed by one protein atom node p_i linked to one ligand atom node l_i from an a specific protein-ligand complex. **(b)** The bipartite graph G' conformed by one protein atom $p1$ which have interactions with 5 different ligands l. The multiple sequence alignment (T_{SA}) is needed if the sequences of the protein-ligand complexes are different. Source of the sequence alignment: `2013.igem.org/wiki/images/4/4b/Heidelberg_IndPD_Fig4.png`.

We also propose a second graph, $G'' = (V'_P, V'_L, E')$ (Figure 3.1-e) where V'_L is a new set of nodes, that consists of node compositions in V_{Li} $\forall i \in n$. Each composition is obtained by merging or aggregating nodes sharing exactly the same physicochemical properties (interaction type and atomic ligand type), interacting with exactly the same protein atoms and establishing the same type of interactions (established in the Table 3.1). An example of this process is provided in Figure 3.4. G'' is obtained by applying simultaneously a two-part merge T_m. One is $T_{m1}(V_L) = V'_L$, applied on vertex ligand atom set $V_L = \{l_1, l_2, l_3, ..., l_m\}$, generating V'_L, a new vertex set representing ligand atoms grouped by atom type. The other is $T_{m2}(E) = E'$, which creates a new edge set E' as result of grouping interactions by type. T_{m1} is done for a set of k ligand vertices $V_k = \{v_1, v_2, v_3, ..., v_c\}$ of a component $\kappa \in K$ (the set of connected components of G') with the same atom type if and only if for T_{m2}, $e_{ij} : p \in V'_P, l \in V_k$ and all candidates for merge e_{ij} have the same interaction type. Ligand atom vertices that can not be grouped by atom type or with edges that are not candidates for merge, remain in the new graph, implying that the whole transformation T_m of G' requires a *strict consensus*. Then the new graph $G''(V"_P, V'_L, E')$ is obtained as a result of this merge transformation function T_m. We define the process of shrinking nodes and edges of G' to generate G'' as *merge*.

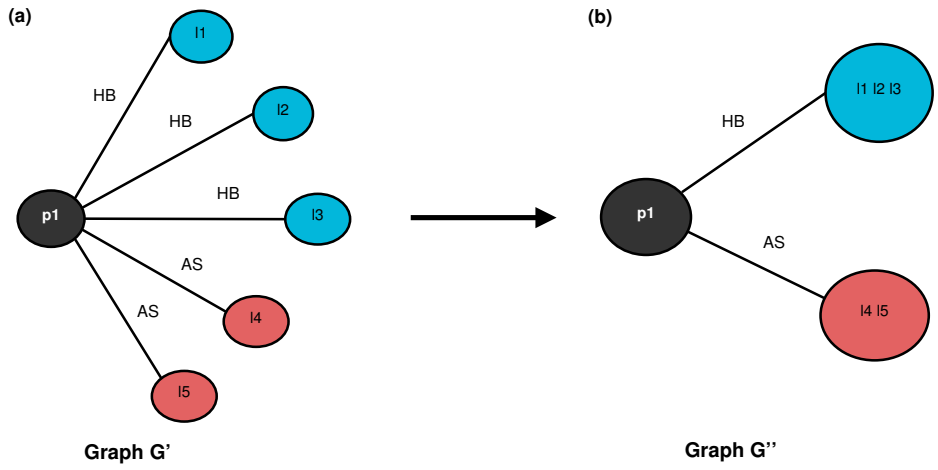

Figure 3.4: Transformation of G' in G''. **(a)** The graph G' is composed by two types of ligand atoms and two types of interactions. The ligand atoms $l1$, $l2$ and $l3$ are connected with $p1$ by hydrogen bonds (HB). On the other hand, the ligand atoms $l4$ and $l5$ are connected to $p1$ by aromatic stackings (AS). **(b)** The graph G'' is a comprised representation of G' grouping edges by interaction type and ligand nodes by atom type.

3.4.3 Model Generator Algorithm

The whole formalization of PLI graphs described in Section 3.4.2 is done by a script in python using the *networkx* library to represent the γ set of PLI graphs. The particular choice of this library is because it allows to add attributes to nodes and edges. We take advantage of this feature saving all the biological information available of the dataset as edge and node attributes.

The attributes for a protein node of G' are: atom label (unique string), atom type (acceptor, aromatic, donor, hydrophobic, negative or positive), color (according to node type= aligned, normal, heteratom), bipartite (0 or 1) and aligned (boolean value). The interaction types defined for PLI are: aromatic stacking, hydrogen bond, hydrophobic, repulsive and salt bridge. The attributes to construct edges of G' are: protein atom label, ligand atom label, distance (measured in Angstroms - Å), protein complex id (from PDB), interaction type and color (according to interaction type).

For the best of our knowledge, there is no graph mining algorithm that allows a representation of PLI as the one made in CALI, because it considers atom types and physicochemical types of interactions. Many graph mining algorithms do not model attributes or do not make them part of the mining process. These mining algorithms generalize their graph representation, allowing the graphs to contain different types of data.

The calculated components of G', which we denote by K, have attributes which make possible to shrink it into the G'' model. We call the resulting components *frequent PLI patterns* and denote them by K'. These attributes correspond to biological data inherent to the protein-ligand atom contacts.

After pre-processing the PLI input (Section 3.3) we applied the transformation function T_{SA} only for datasets that have proteins with different amino acid sequences. However, some datasets, as CDK2 (Section 3.2.1), do not need application of T_{SA} because they are special cases where all the proteins from the dataset have the same amino acid sequence, which means that all graphs in γ have the same set of nodes representing protein atoms V_P.

The steps to generate G' are:

1. Pre-process interaction data applying T_{SA}, which for PLI graphs is a sequence alignment function.

2. Construct edges and nodes of G' from input interaction data.

3. Calculate connected components, using the *networkx* function which traverses all the vertices of the graph and uses the Dijkstra algorithm to calculate all the

shortest path lengths.

4. Calculate global descriptors, using implementations of *networkx* library.

5. Calculate network metrics for every component in the K set, using implementations of *networkx* library.

The process to generate G' takes $\mathcal{O}(|V||E|)$ space and its time complexity depends mainly on the connected components calculation, which is $\mathcal{O}(|E| + |V|log|V|)$. The network metrics calculated for each component could be time-consuming if the biggest components of G' are dense and have large size. However, the G' components (K) are bipartite, implying that time complexity for centralities as betweenness, will be always lower than for an undirected weighted complete graph, which is $\mathcal{O}\big(|V||E| + n^2 log\, n\big)$ [Brandes, 2001].

The steps to generate G'', described in Algorithm 1 are:

1. Merge interactions for all components K of G', using the function T_m.

2. Construct edges and nodes of G"

Algorithm 1: Merge PLI

Data: I

Result: G'', K'

```
1  initialization;
2  M_c := matrix for counting all interactions ;
3  L_c := list of protein nodes;
4  for k_i ∈ K do
5      for p_i ∈ V'_P ∈ k_i do
6          % Identify candidates for merge ;
7          V'_c := set of candidates for merge;
8          if p_i.degree() > 1 then
9              V'_c.add(p_i);
10             for ne ∈ p_i.neighbors() do
11                 if ne.degree() == 1 then
12                     c_1 := count_types_interaction(M_c);
13                     la_1:=join_labels(M_c);
14                 end
15                 if ne.degree() > 1 then
16                     c_2 := count_types_interaction(M_c);
17                     la_2 := join_labels(M_c);
18                 end
19             end
20             L_c.add(p_i, c_1, la_1, c_2, la_2) ;
21         else
22             p_i is not candidate;
23         end
24     end
25     κ' =construct_component(V'_c, L_c);
26     K'.add(k')
27 end
28 G'' = construct_G_composed(K');
```

Algorithm 1 generates G'' creating its components κ_i by counting all interactions. The matrix M_c is created for counting all interaction types. First, all nodes representing protein atoms, p_i, are evaluated to select candidates to merge. All p_i with degree greater than one are selected and stored. Second, the p_i neighbors, which are nodes representing ligand atoms, are evaluated and separated into two groups: c_1 for nodes with degree

equal to one, and c_2 for nodes with degree greater than one. The labels of the respective nodes are stored in la_1 and la_2. The list L_c save the information necessary for every p_i candidate: c_1, c_2, la_1 and la_2. Third, components κ' are calculated with the function *construct_component* described in algorithm 2 and G'' is constructed based on the set of components K'.

Algorithm 2: construct component κ' of G''

```
   Data: V'_c, L_c
   Result: κ'
 1 for each p_i ∈ L_c and p_i ∈ V'_c do
 2     for each interaction type ∈ M_c do
 3         forall e_ij ∈ L_c with same interaction type do
 4             if e_ij ⊂ l_j.degree() == 1 then
 5                 merge edges → E_c;
 6             end
 7             if found ∃ e_ij ⊂ l_j.degree() == 1 then
 8                 add edges with degree == 1 → E_c;
 9                 add the rest of edges → E_c without merge;
10             end
11             if e_ij ⊂ l_j.degree() > 1 then
12                 % This edges have interaction with two or more proteins;
13                 merge edges just for p_i → E_c ;
14             end
15         end
16     end
17 end
18 return κ'(E_c) ;
```

The set of connected components K' is constructed using Algorithm 2. The protein nodes p_i in V'_c are processed according to the counting data saved in M_c. Edges e_{ij}, with the same interaction type in M_c, are merged if all their ligand nodes l_j have degree equal to one. On the other hand, only edges e_{ij}, whose ligand nodes l_j have degree equal to one, are added without merge. The rest of edges are processed separately, to avoid merging edges whose ligands have interactions with two or more protein nodes p_i.

The whole process of constructing G'' requires traverse the graph twice, first for counting the interactions and second to do the merge. Thus, G'' calculation can be approximately done in time $\mathcal{O}(|V||E|)$.

3.4.4 Relevant Biological Patterns

A component can be defined as an independent piece of the graph, a subgraph where each node is connected to every other and the subgraph is not a part of any larger subgraph [Kleinberg and Easley, 2010]. Analyzing components is essential when a network is highly sparse [Newman, 2010] to calculate metrics and network properties, which occurs on most PLI graphs. G is the disconnected graph that represents protein-ligand interactions for a set γ of protein complexes. Doing some transformations on γ, we generate new graph models G' and G'', whose components represent interaction patterns.

The set $K = \{\kappa_1, \kappa_2, \kappa_3, ..., \kappa_c\}$ of connected components of G' shows conserved and big components in the PLI graph. Many of these components present the shape of a hub or a star, which means there is only one node in the center with a lot of connections to other nodes that do not have edges among them.

The graph G'' was conceived as another representation of PLIs, grouping ligand interactions with the aim to improve the recognition of patterns by identifying the more common types of ligand interactions in every component. G'' components, which belong to the set $K' = \{\kappa'_1, \kappa'_2, \kappa'_3, ..., \kappa'_c\}$, are shrink in comparison with those in the set K of G' components.

The component sets K and K' from the graph models G' and G'' can be defined as *biological patterns* of PLIs, where the degree centrality is the measure for frequency. Nodes with a high degree in G' correspond to protein atoms belonging to residues that have biological importance for the protein complexes. These nodes represent atoms of residues with high binding affinity, which usually are in hubs. A high number of hubs in G' could mean that there are few residues well-conserved and important to the ligand binding. Also, this could mean that the ligands in the dataset are derived from the same inhibitor or a set of inhibitors. A huge component in K indicates that there is a extensive region (composed by several residues) of the protein complex involved in the ligand binding.

Chapter 4

Results

This chapter discusses the results obtained using CALI, our visual model for frequent pattern mining in protein-ligand interactions. The PLI visualization developed based on the CALI graph models is explained in Section 4.1 including the application web (Section 4.1.1) and the visual strategy used to identify relevant biological patterns (Section 4.1.2). A brief network analysis for the PLI graphs is presented in section 4.2. Complementing this analysis, a network model characterization is provided in Appendix D. Finally, two comparisons are done. The first one, in Section 4.3, we compare CALI with the state-of-the-art algorithm for graph mining, *gSpan*. The second one, in Section 4.4, we compare CALI with results experimentally generated for CDK and Ricin.

4.1 PLI Visualization

The graphs G' and G'' represent a PLI network. However, a visualization focusing on the components (K and K') allow to identify easily the biggest ones, specially hubs, which biologically are more interesting to analyze. Due to this fact, we developed a prototype tool (Section 4.1.1) allowing flexible visualizations of the PLI networks with the aim to explore the centrality metrics of these networks join with biological relevant data (e.g. residue name, atom name, residue number, ligand name, distance between atoms, etc.) displayed in the same network. The strategy used to discover biological patterns with this visualization is described in Section 4.1.2.

4.1.1 Visualization Tool for PLI Graphs

The visualization web application for PLI graphs modeled using CALI was developed with the collaboration of the PhD student in Bioinformatics, Alexandre V. Fassio from the ICB and DCC, UFMG (Belo Horizonte, Brazil), as part of a teamwork of the Laboratory of Bioinformatics and Systems (LBS) from the same University.

The PLI network visualization is coupled with several filters to support exploration, investigation and analysis of the model and its emerging patterns. We use a force directed layout to draw the bipartite PLI graph, grouped by components, where nodes depict atoms and edges are interactions between them. Different colors distinguish between protein and ligand atoms and we have five colors for edges, to represent the five different types of interactions, as shown in Figure 4.1. The interactive visualization was implemented in D3[1], a JavaScript library to build a wide spectrum of visual representations.

We also implemented several filtering and highlighting possibilities. The user can filter out the network by the types of atoms (Figure B.1) and by the types of interactions (Figure B.2). When the user checks or unchecks an option, the corresponding atoms (nodes) / interactions (edges) lose contrast with the background and all the others are highlighted. Users can search for a particular residue and / or atom and highlight it (picking any color they prefer) which makes easy to find a residue and / or atom in the network. In Figure 4.2, we present an example where we select the TYR80 (all atoms from this residue) which it is highlighted in a brilliant shade of blue. Finally, there are also a total of eight measures, that can be used to filter out the network elements through sliders. In Figure B.3, we show an example where we filter out nodes whose degrees are bellow 10% of the maximum value. For every element of the graph, details can be obtained on demand by passing the mouse over it (Figure B.4).

4.1.2 The Visual Strategy

The graphs G' and G'' are plotted in a way that allow to identify directly by visual inspection their connected components. The biggest components in G' (Figures 4.1 and 4.4) are the result of a *graph superposition* and can be visually identified by their size, which is characterized by two main types. The first type corresponds to hubs: a central protein atom node with many edges to different atom ligand nodes. The second type are subgraphs forming odd cycles including several protein atom nodes interacting with many atom ligand nodes. Using the number of edges filter of the developed tool, the

[1] http://d3js.org/

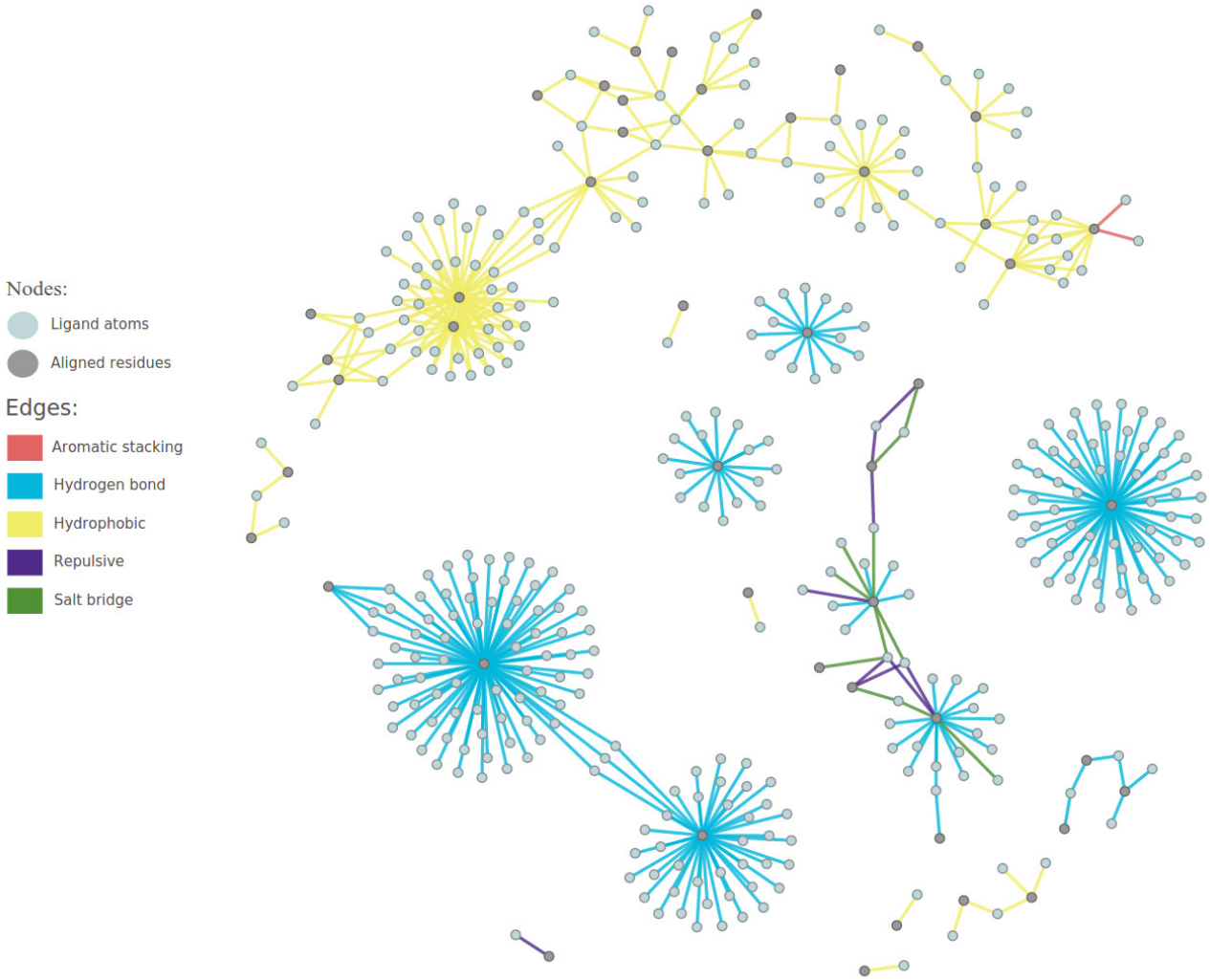

Figure 4.1: CALI model bipartite graph drawn using a force directed layout. Nodes depict atoms and edges are interactions between them. Different colors distinguish between protein and ligand atoms and also, we have five colors for edges, to represent the five different types of interactions. This graph G' represents the CDK2 dataset, with its components K.

biggest components remain while the others loss emphasize. Also, the filter to search specific residues, ligands or atoms, locate them in the network, allowing to identify at once if the element searched is or not in a big component. The described features in Section 4.1.1 allow to explore the PLI networks in many ways. Preserving the biological information of the PLI in the network visualizations allow to complement its analysis with other molecular visualization tools as PyMOL.

The graph G'' (Figures 4.3 and 4.5) is the result of the *merge* described in Section 3.4.2, grouping edges (interactions) and ligand atom nodes by their types respec-

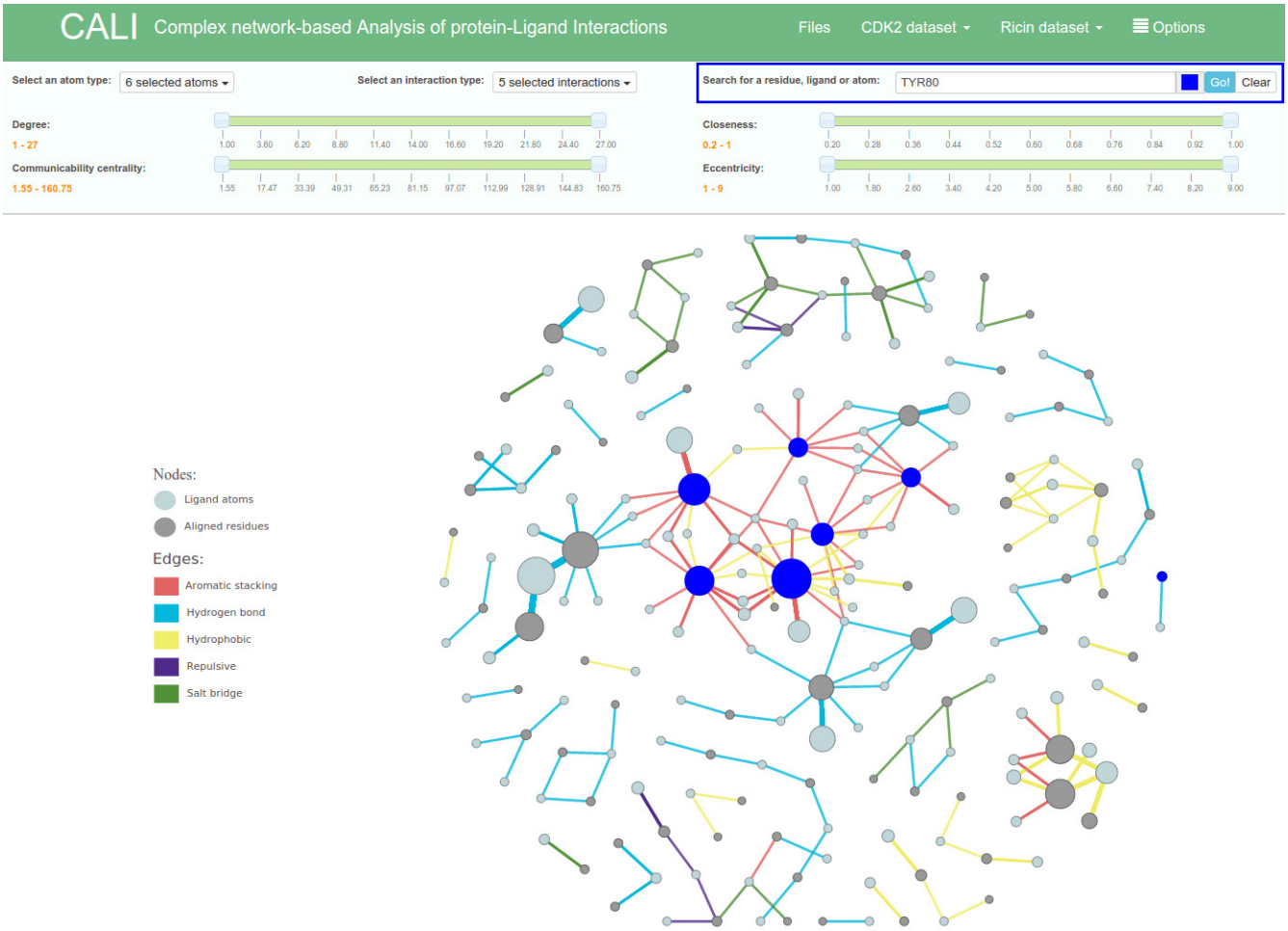

Figure 4.2: CALI search example. Users can search for a particular residue and / or atom and it is highlighted (users can pick any color they prefer), which makes easy to find a residue and / or atom in the network. In this figure, we searched for TYR80 and it was highlighted in a brilliant shade of blue. The graph is from Ricin dataset.

tively. The visualization of G'' maintain the focus on components (K') but as a result of the *merge* are only shown the principal types of interactions and several ligand atom nodes are represented by one node. Edges merged are represented by thicker lines. However, the information of the ligand nodes and the edges merged remain in the visualization and can be obtained by passing the mouse over them. The protein nodes with high degree are drawn bigger than those with low degree, highlighted that these nodes represent atoms from residues with high binding affinity from the PLI dataset. This big picture of the whole PLI network, combined with the filters of the tool (Figure 4.2), allow to identify the more frequent atoms involved in the ligand binding. Also, it is possible to discover atoms from residues with high degree, which are not reported as important in experimental results. The biggest components from both graphs (G' and

G'') also represent the main regions of interaction between the protein complex and the ligands from the dataset.

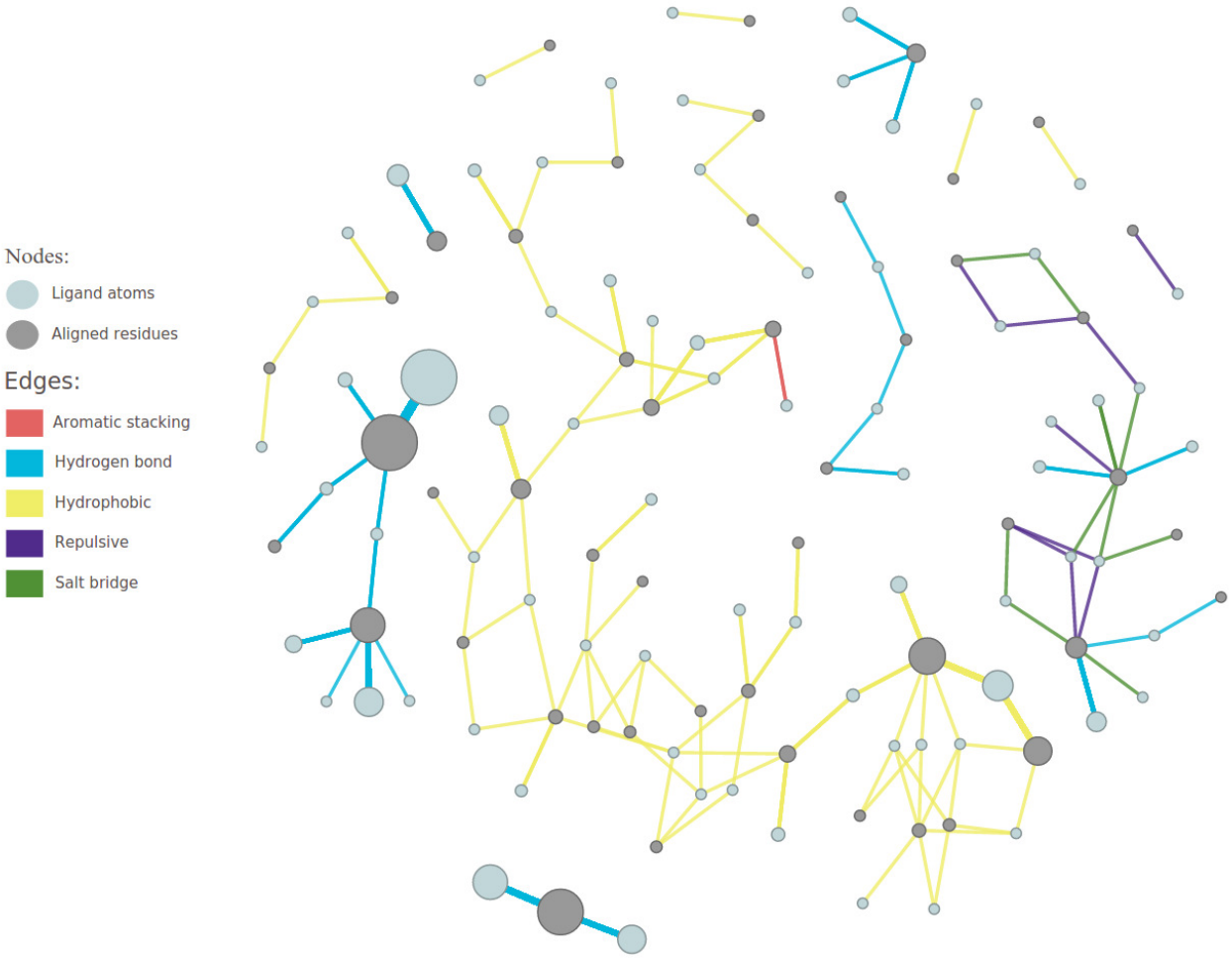

Figure 4.3: CALI model bipartite graph G'' for the CDK2 dataset, with its components K'.

4.2 PLI Network Analysis

Global network descriptors were calculated using *networkx*[2] library [Hagberg et al., 2008]. Some useful global measures are the *total number of edges and nodes*, the *number of components* and the *density* (this measure describes the portion of the potential connections in a network that are actual connections). This last one is remarkably low and it is something potentially interesting because being so sparse (*global clustering coefficient* is zero), the network can be more effectively visualized and relevant patterns

[2]https://networkx.github.io/documentation

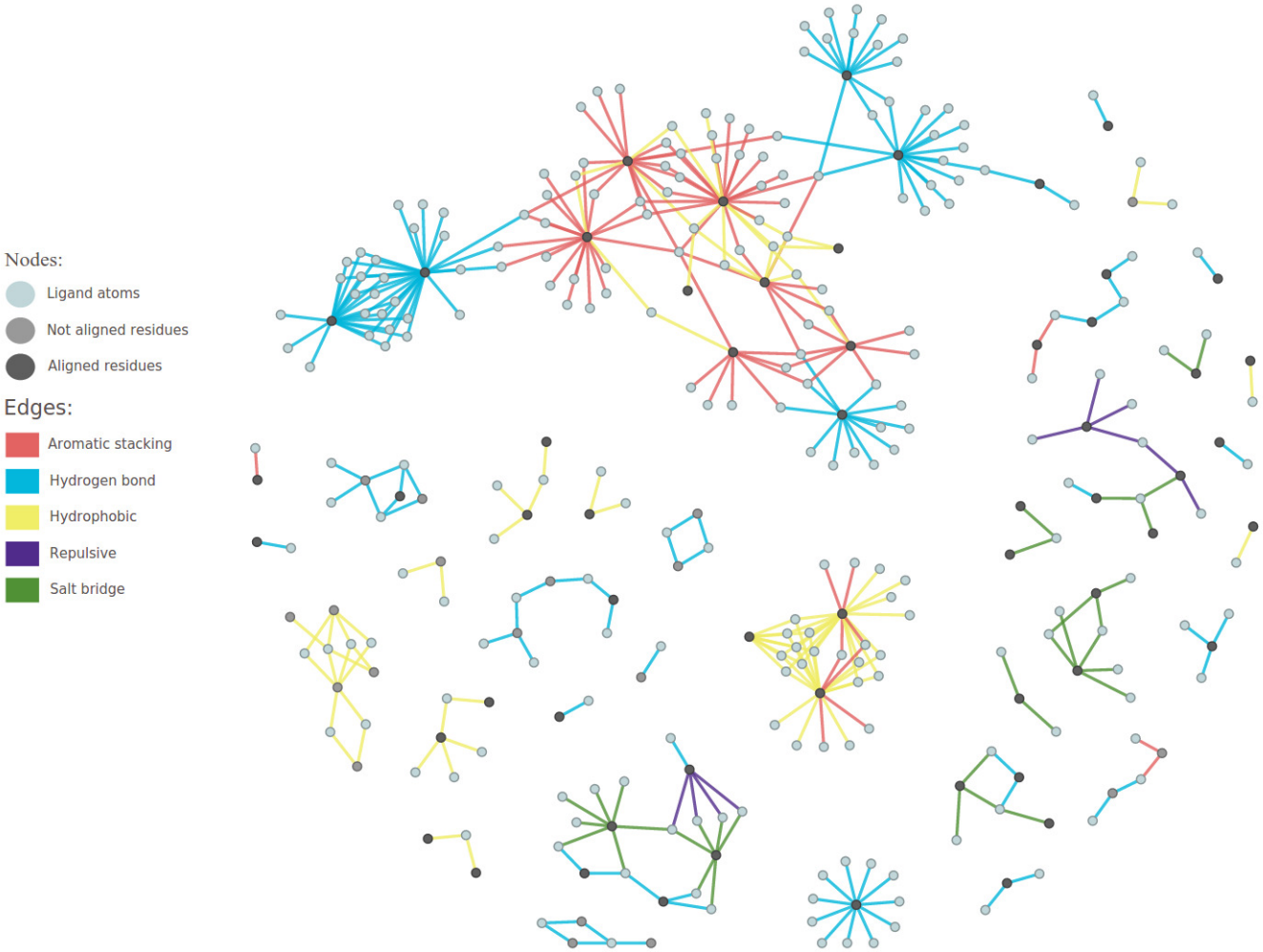

Figure 4.4: CALI model bipartite graph G' representing the Ricin dataset, with its components K. Nodes depict atoms and edges are interactions between them. Different colors distinguish between protein and ligand atoms and also, we have five colors for edges, to represent the five different types of interactions.

can emerge in a straightforward manner. To characterize the model, we computed the centrality metrics: *degree*, *closeness*, *communicability*, *eccentricity*, *node betweenness* and *edge betweenness*. Additionally, another two metrics are used as filters: the *distance* between pairs of (node) atoms measured in Angstroms and the *number of edges* for each component, which allow us to filter them by its size.

Global network descriptors for the G' graphs of CDK2 and Ricin are presented in Table 4.1. Note that both are sparse as the number of edges and nodes are the same order of magnitude (about 400 for CDK2 and 300 for Ricin). Also, notice that both are disconnected presenting several components (12 for CDK2 and 34 for Ricin). This is interesting and could indicate that ligands present far (disconnected) points of contacts

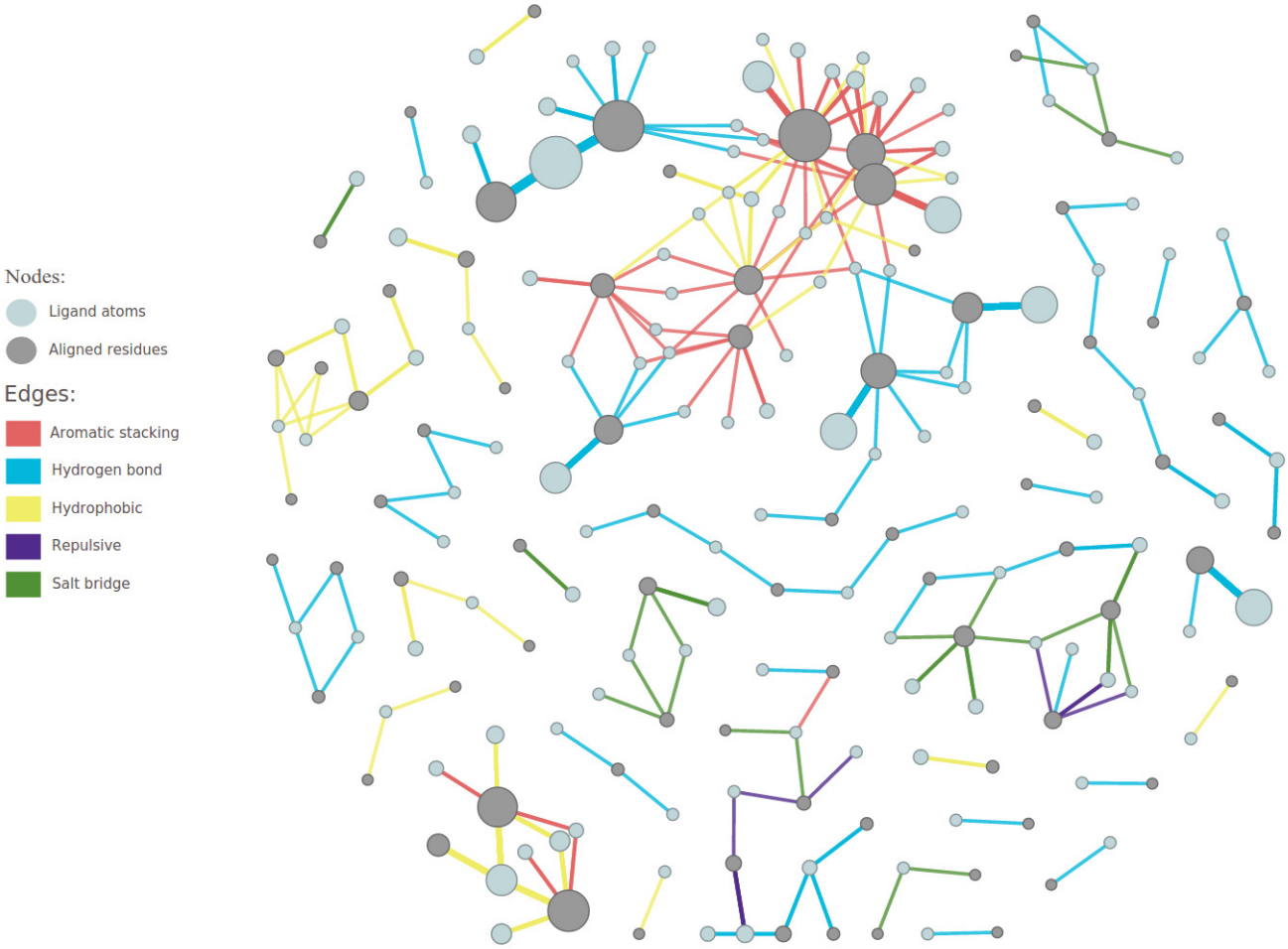

Figure 4.5: CALI model bipartite graph G'' representing the Ricin dataset with its components K'.

Table 4.1: Global network descriptors

Descriptors	CDK2	Ricin
Number of nodes	407	321
Number of edges	459	377
Global clustering coefficient	0.0	0.0
Number of connected components	12	34
Order of largest component	128	134
Order of second largest component	120	28
Order of third largest component	62	18
Density	0.026	0.020
Average bipartite clustering coefficient	0.748	0.547
Average bipartite clustering coefficient for ligands	0.823	0.623
Average bipartite clustering coefficient for proteins	0.188	0.306

with proteins. Additionally, there is a big component for the Ricin (134 nodes), which may suggest that the protein-ligand contacts are in great part close, and there are two components for CDK2 (128 and 120 nodes), which may indicate two main regions of contact with ligands for CDK2.

The CDK2 G' network (Figure 4.1) is more transitive, it has a higher bipartite clustering coefficient [3] than Ricins. This happens because there are a considerable number of hydrophobic interactions in the largest component of CDK2, forming odd cycles larger than triangles. This behavior is similar to the metabolic network of yeast in the work of Estrada and Rodríguez-Velázquez, 2005. It is also noteworthy the fact that the average bipartite clustering coefficient is considerably higher for ligand atoms than for proteins. It happens because groups of ligand atoms tend to establish interactions with groups of protein atoms, those that can be considered more conserved in the protein structure. If the CDK2 G' network did not have its biggest component(hydrophobic), the bipartite clustering coefficient would be very low. The other components are mainly hubs (Figure 2.3) or planar graphs (Figure 2.1).

PLI in CDK2 involve an identical protein with several ligands. Consequently, protein atoms are more conserved and then more connected in the network. In the Ricin graph G' (Figure 4.4), protein nodes were generated using a structural superposition to define a similarity function that allows us to group nodes. In this case, we have an heterogeneous residues sequence set, where not every position is necessarily well conserved and thus nodes tend to be less connected and less transitive as well. Moreover, both datasets are dissimilar in biological context (CDK2 forms a human protein complex and Ricin forms a toxic protein from the seeds of a plant).

Despite the Ricin PLI network is smaller than the CDK2 network, it has a huge connected component with 134 nodes composed mainly by hydrogen bonds and aromatic stackings and a few hydrophobic interactions. The majority of the other components have less than 16 edges and there are no hubs, excepting two components. The first one is a hub formed by a node representing a Nitrogen from the TYR123 with degree 11 (interactions). The second one is composed by three atom nodes (with degrees 19, 20 and 8) from the TYR123.

Also, we addressed the question of which are the most important nodes (central ones according to our modeling) of a PLI network produced by CALI. It is important to point out that we have a premise that the most important patterns should comprise important nodes of the network, as the largest component of hydrogen bonds in CDK2

[3] Latapy et al., 2008 propose this definition for bipartite graphs because in these graphs it is not possible to have triangles. This definition is based on the possibility that 4 nodes are connected with 4 links, forming a square.

(see the light blue subgraph on the left side in Figure 4.3), which have two biggest protein nodes (in dark gray) with high degree, corresponding to an oxygen and a nitrogen of Leucine in the position 83 of the residue sequence. Among the several centrality measures we computed, we concluded degree is the most important for frequent pattern detection as expected. In fact, in our PLI model, conserved elements tend to have a high degree. Despite in some cases, considering other metrics, can help to filter out irrelevant elements in the PLI graphs.

4.3 CALI Comparison with gSpan

This section explains the patterns found in the CDK2 and Ricin datasets using CALI. Moreover, we compare CALI patterns with those computed by gSpan, the most cited graph mining algorithm according to Jiang et al., 2013, which is commonly used to find frequent patterns in biological data.

4.3.1 gSpan Pattern Generation

gSpan is a depth-first search-based mining algorithm for efficient mining of frequent subgraphs in large graph databases. The two main advantages of gSpan are better memory utilization and effective subgraph testing [Huan et al., 2004]. This algorithm introduces a novel lexicographic ordering by mapping each graph to an unique minimum DFS code and then performs a search tree based on this lexicographic order [Yan and Han, 2002]. We selected gSpan to perform a comparison with CALI because gSpan is the most cited frequent subgraph mining algorithm [Jiang et al., 2013].

In order to be able to compare the patterns found by CALI with the patterns found by gSpan, we have to follow several steps (Figure 4.6) to be able to run this algorithm properly for the PLI datasets. These steps are pre-process the protein-ligand complexes, calculate their atomic contacts, choose a model and a mapping for the PLI graphs and create the input file required to run gSpan with different support values.

The gSpan algorithm requires a specific input file, which was generated processing each dataset, resulting in one file with 341 graphs representing CDK2 PLIs and another file with 168 graphs representing Ricin PLIs. gSpan was run for both datasets with support values between 90% and 10%, decreasing 10% each time. After 10% it was decreased in 1% each time, until 1% was reached. To test the gSpan limits and the maximal number of patterns it is able to find, gSpan was executed with support 0,10%. Patterns began to appear for CDK2 with 50% support and for Ricin with 30% support.

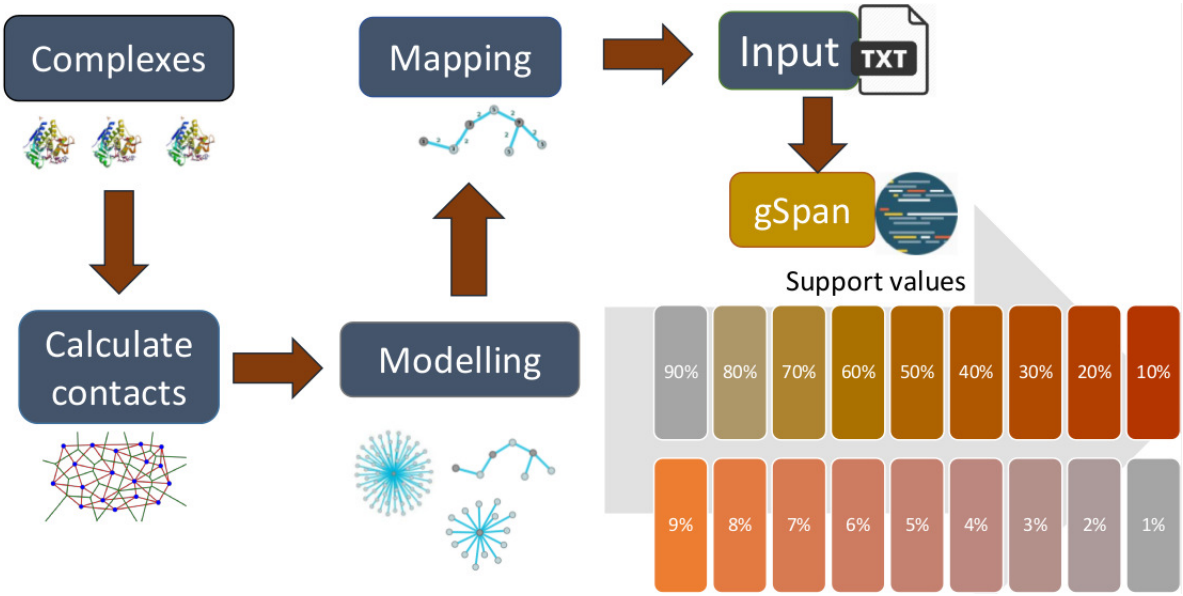

Figure 4.6: Process to apply gSpan to find frequent patterns in PLI datasets. There are four steps requiered to generate the input file for gSpan. First, obtain the protein-ligand complexes from the PDB. Second, calculate the atomic contacts from the PDB files. Third, model the biological information to construct graphs which represent the PLIs. Fourth, choose the most relevant characteristics and map them to construct graphs according to the gSpan input format. Finally, gSpan is tested with several support values to find frequent patterns.

Figures 4.7 and 4.8 show the number of patterns found for each dataset, with support values between 1% and 90%. The highest number of patterns found, as expected, was when gSpan run with 1%, resulting in 37 graph-patterns for CDK2 and 224 graph-patterns for Ricin.

The gSpan implementation [4] used for this comparison found patterns (graphs) composed by one node, which were not included in our analysis because a graph with one node does not present any interactions.

Even gSpan, which is the most used, save-memory FSM algorithm, did not offer useful results searching patterns in PLI graphs. We found the following limitations for gSpan in the comparison performed:

- It is not able to treat input graphs with several node and edge attributes.
- Loss of biological data for PLI graphs (as contact distance), when these graphs are converted to gSpan input format with just one label for node/edge.

[4]The implementation used was the binary code (2009) for linux. Available at the web site of professor Xifeng Yan https://www.cs.ucsb.edu/ xyan/software/gSpan.htm

- Running gSpan is not time-consuming (takes less than 0.01 seconds for each support value), because our datasets are small. However, for larger datasets this algorithm takes considerable time to process graphs, because it does not mine maximal frequent subgraphs and it only finds frequent patterns with very low support values (around 10% or lower) in PLI datasets.

The frequent pattern-graphs obtained by gSpan have two main problems. First, among the patterns, many can be structurally repetitive, as a frequent subgraph can have other frequent subgraphs within it [Yan and Han, 2003]. Second, gSpan output does not have labels indicating which nodes from input graph G_i match to a pattern g_i. Thus we need to apply an algorithm of subgraph isomorphism (as the VF2 algorithm [Cordella et al., 2004]) to find which nodes from input graph G_i match to a subgraph g_i, as occurs in the work of Silveira et al., 2015. The problem of graph isomorphism has not been yet determined as solvable in polynomial time or NP-complete, while subgraph isomorphism is well-known to be NP-complete [Garey and Johnson, 1979].

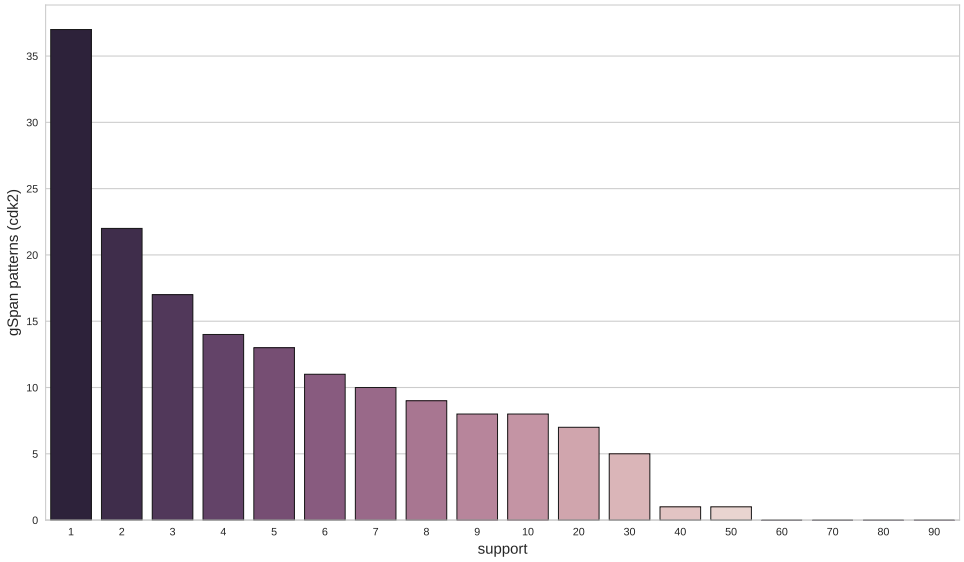

Figure 4.7: Number of patterns found by gSpan for different support values in the CDK2 dataset.

4.3.2 Mapping gSpan Graph-Patterns

To be able to analyze the frequent gSpan patterns at the protein complexes, it was necessary to choose an adequate mapping function to maintain the biological information

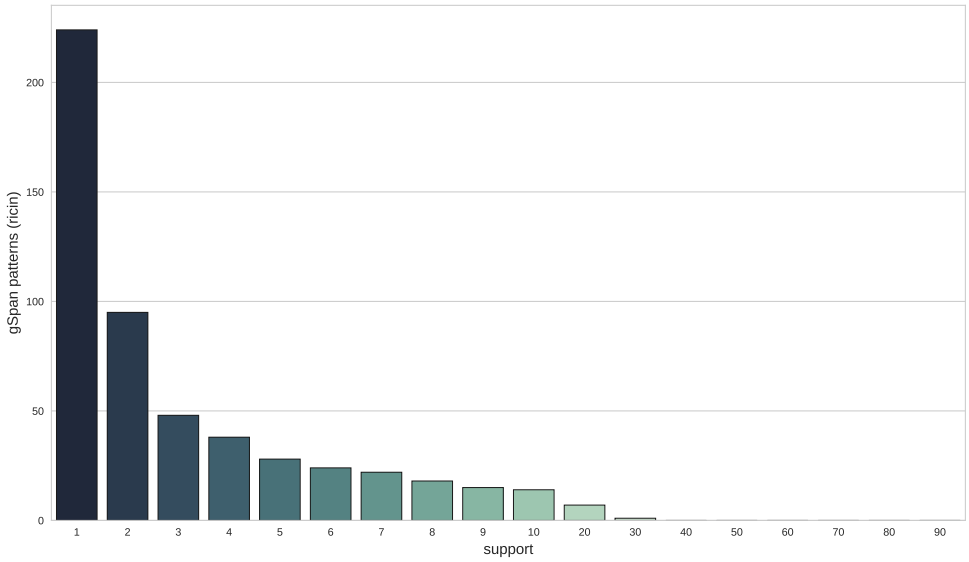

Figure 4.8: Number of patterns found by gSpan for different support values in the Ricin dataset.

in the PLI graph. In the work done in gSpan [Yan and Han, 2002] and in CloseGraph [Yan and Han, 2003], biological node and edge labels were mapped [5] as numbers. Both used synthetic data and also a real one, which is a chemical compound dataset (AIDS antiviral screen). This dataset is sparse: on average 27 nodes and 28 edges per graph. The authors goal was to find frequent substructures in graph datasets and decrease the memory consumption by creating a new graph mining algorithm.

Our interest here is to find patterns (similar to Yan and Han) in the PLI datasets. To accomplish this goal, we used an analogous mapping choosing to model only two physicochemical properties as the most representatives for the input graphs of gSpan. These properties are type of interactions (Table 3.1) as edge labels and type of atoms as node labels of the input graphs. The type of atoms were defined (as described in Section 3.3) with their possible combinations (i.e. an atom can be positive, donor and aromatic). There are 14 types of atoms and 5 types of interactions, corresponding to the types present in the two datasets (CDK2 and Ricin) used. The input graphs map the interaction and atom types as integer numbers, as the example in Figure 4.9.

To find where and which are the gSpan patterns in the PLI graphs it was necessary to define a process composed by two main steps (Figure 4.10). The first one was to search for the original subgraph in the gSpan input where the pattern belongs to.

[5]The exact mapping used was not available in neitheir of their publications. Professor Xifeng Yan answered us by e-mail that he did not find the mapping file.

This search is done running a subgraph isomorphism algorithm. Second, as every gSpan pattern was encoded with numbers (to allow gSpan execution), it needs to be decoded (to our original data): a reverse mapping, from numbers to labels (categorical data). Finally, the patterns with their location (on the input graphs) and the original categorical data (atom and interaction types) were obtained.

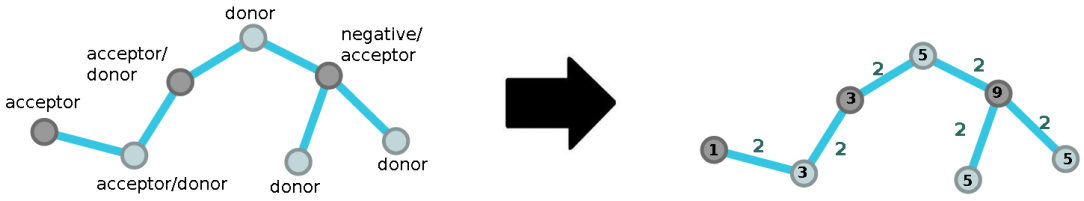

Figure 4.9: Mapping atom and interaction types as numbers. This graph (a component from CDK2) has only hydrogen bond interactions, linking 3 protein atoms and 4 ligand atoms.

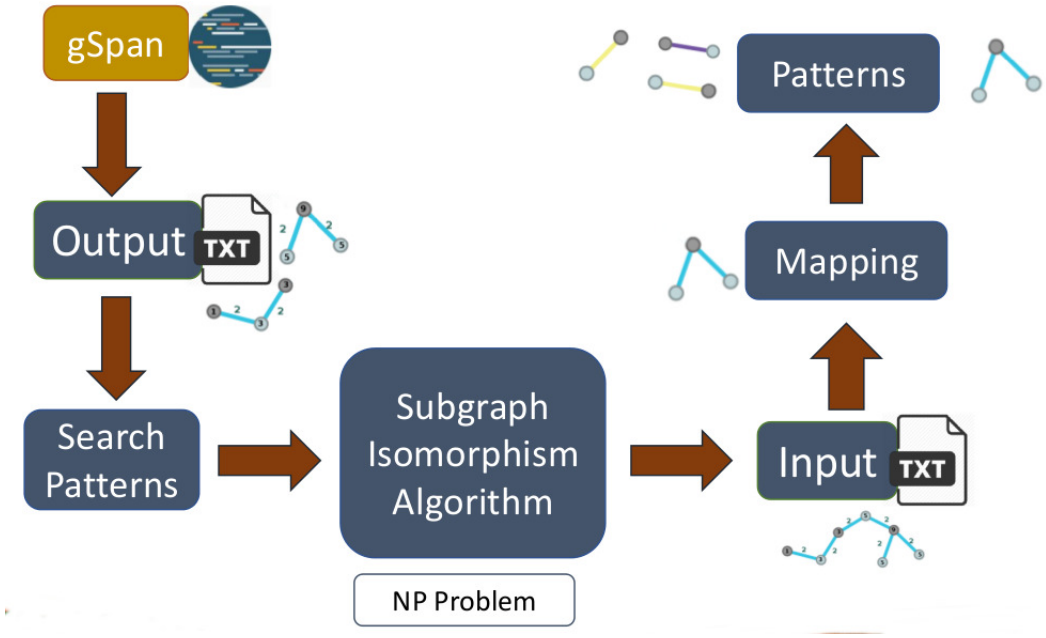

Figure 4.10: Process required to analyze the patterns in gSpan output.

4.3.3 CALI and gSpan Pattern Comparison

The comparison between the patterns found by CALI and by gSpan was done using graph and subgraph isomorphism algorithms (Figure 4.11). The gSpan patterns decoded in the original (PLI) data are searched running a graph isomorphism algorithm in the PLI graph generated by CALI. This search was done through the PLI graph components, because all of them are disconnected. Moreover, we applied subgraph isomorphism for cases where the gSpan pattern is smaller than the component, and graph isomorphism for cases where the gSpan pattern has the same size as the component (just a few of these cases found). Both isomorphism algorithms were implemented to allow the comparison between the CALI model and gSpan. This whole process takes around 2 seconds, beginning with the reading of the gSpan output file and finishing with the identification of every pattern in the graph components generated by CALI.

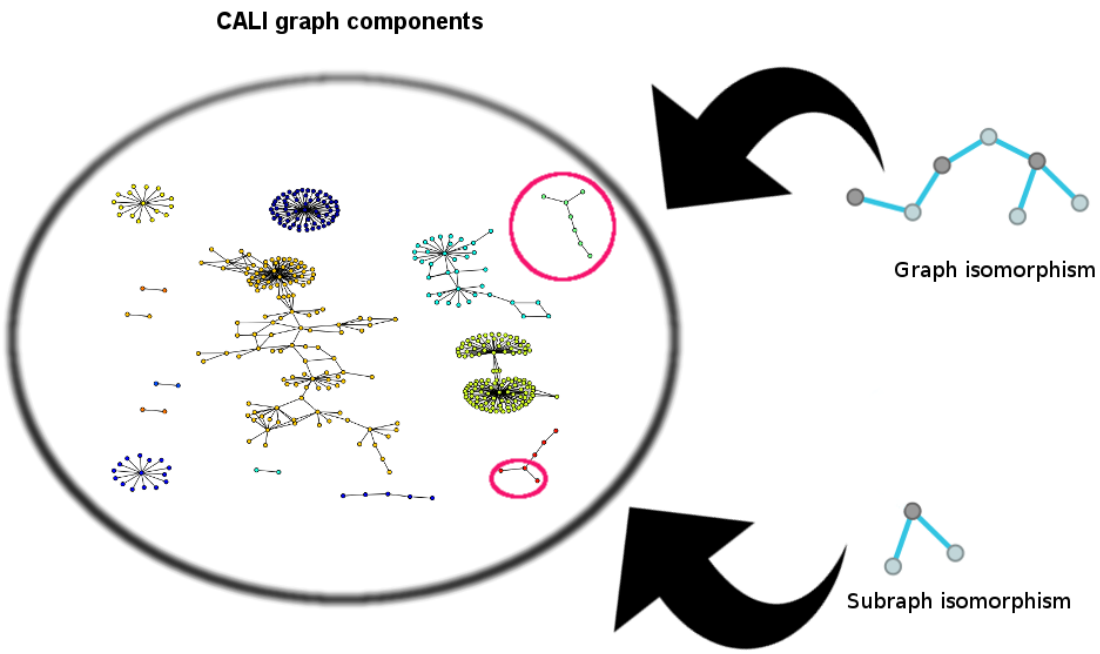

Figure 4.11: Search for gSpan patterns. There is necessary to apply two isomorphism algorithms for search gSpan patterns. One is for graph isomorphism, when the pattern is exactly the same as a CALI graph component. The another is for graph isomorphism, when the pattern is composing a CALI graph component.

All gSpan patterns were found in the components of CALI graphs for both datasets. The difference between the two representations is that with CALI is possible to observe all patterns (frequent and infrequent) in a same diagram and filter them according to the user's demand. CALI is simpler, more direct and faster, needing

one execution to create the graph with all PLI patterns. Nevertheless, gSpan requires testing with several support values (particularly lower or equal than 10%) until a reasonable number of patterns is reached.

The majority of the gSpan patterns have one or two edges (interactions) in the CDK2 dataset as seen in Figure 4.12. The majority of patterns for Ricin dataset have one edge for support values between 5% and 10%, but lower than 5% the distribution changes, specially for support values 1% and 2%, where appear numerous patterns with three to eight edges (Figure 4.15). These particular small patterns, which are much like tree, are due to the bipartite model used to build the PLI graphs, but it is also found on chemical compound datasets as reported by Yan and Han, 2002 and Huan et al., 2004.

Larger components in both datasets have the majority of gSpan patterns. This search was done implementing a subgraph isomorphism algorithm, checking either the topology and the atom and interaction types of each gSpan pattern inside the PLI components. The CDK2 dataset contains most of the gSpan patterns in the three largest components of the PLI graph, as shown in Figure 4.13. The largest component has principally hydrophobic interactions and two aromatic stackings, the second and the third largest components are composed only by hydrogen bonds. In the Ricin dataset, the gSpan patterns are in the two largest components of the PLI graph, as shown in Figure 4.16. The Ricin PLI graph has one largest component with 128 interactions, and most of them are hydrogen bonds and aromatic stackings, and a few of them are hydrophobic. The second largest component has only 28 interactions, where most of them are hydrophobic and eight of them are aromatic stackings. We conclude that CALI contains all gSpan patterns, which are located mainly in the largest components. Moreover, PLI graphs are able to show non-frequent patterns in their medium or smaller components, which can also have biological importance.

The search done applying a graph isomorphism algorithm implemented according to the PLI data found only small gSpan patterns, as those with two nodes and one edge. This search was faster than the first done (described above) because many CALI components can avoid comparison by checking their size first. Figure 4.14 shows four gSpan patterns isomorphic with CALI components in the CDK2. All of them correspond to hydrophobic interactions with different atoms [6] from a Lysine (residue). Figure 4.17 shows three gSpan patterns isomorphic with CALI components in the Ricin.

[6] The PDB file format nomenclature is used to abbreviate atom names: C (Carbon), N(Nitrogen), O(Oxygen) = main chain. CA = Carbon, Alpha (1st). CB = Carbon, Beta (2nd). CG, OG = Carbon/Oxygen, Gamma (3rd). CD = Carbon, Delta (4th). CE, NE, OE = Carbon/Nitrogen/Oxygen, Epsilon (5th). NZ = Nitrogen, Zeta (6th). NH = Nitrogen, Eta (7th)

The biggest one has three protein atoms and four ligand atoms. The others have one (hydrogen bond) and two (hydrophobic) interactions.

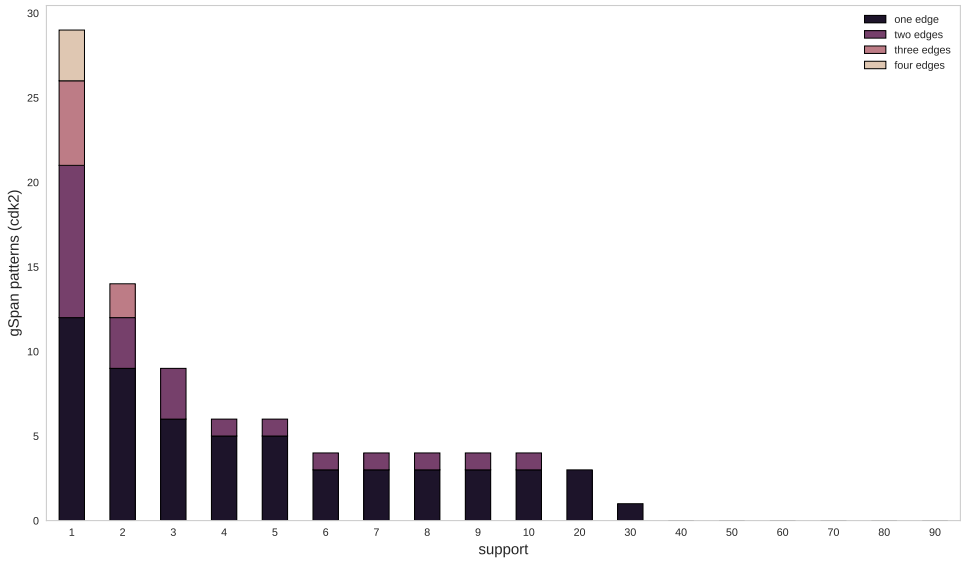

Figure 4.12: Number of patterns found by gSpan and their size in number of edges, for different support values in the CDK2 dataset.

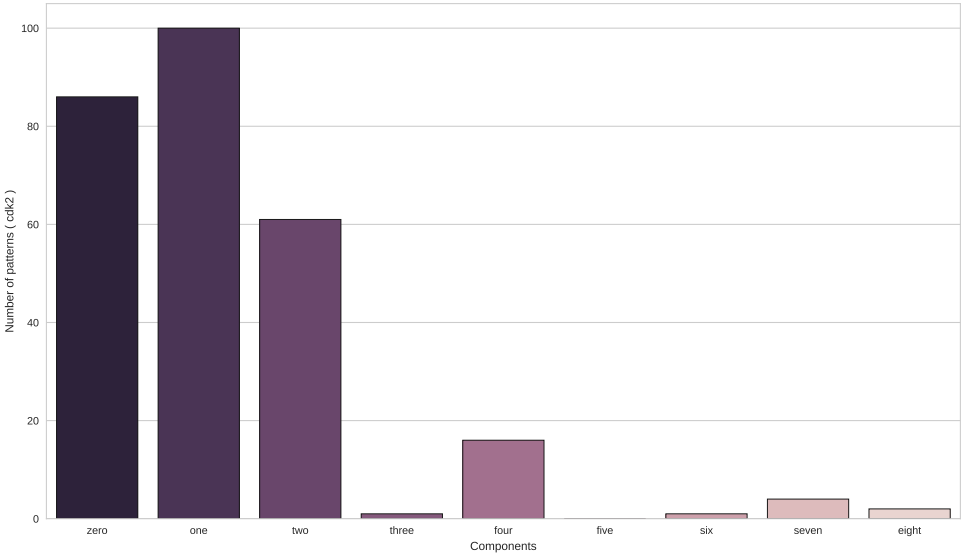

Figure 4.13: Distribution of gSpan patterns in CALI components of the CDK2 graph. These pattern-graphs were searched in CALI components applying an exact matching subgraph algorithm.

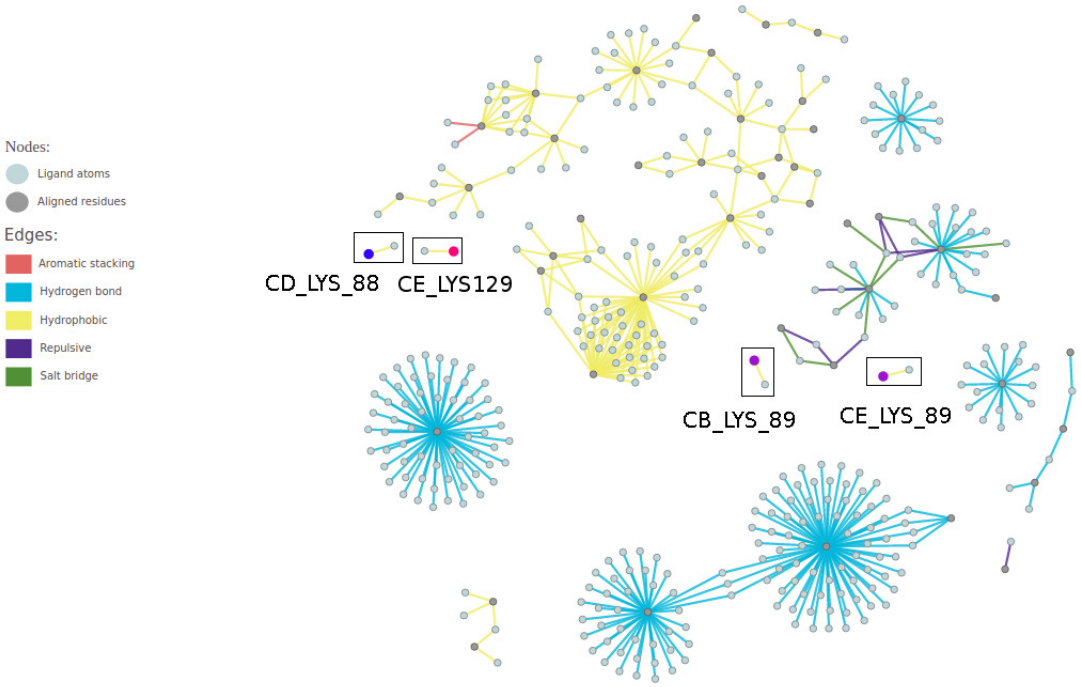

Figure 4.14: gSpan patterns found in CALI components of the CDK2 graph applying exact matching graph isomorphism. There are four gSpan patterns, highlighting in blue(CD LYS 88), magenta (CE LYS 129) and two in purple (CB LYS 89 and CE LYS 89).

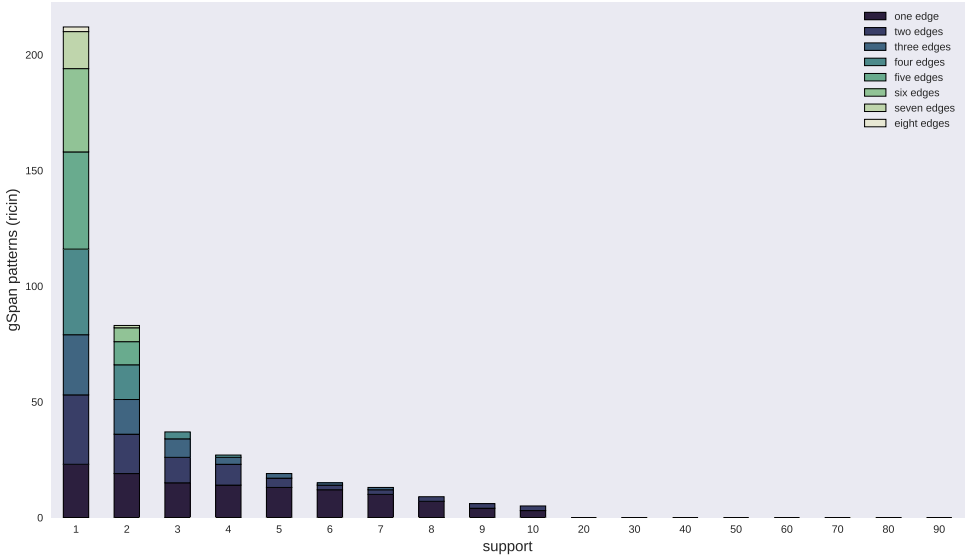

Figure 4.15: Number of patterns found by gSpan and their size in number of edges, for different support values in the Ricin dataset.

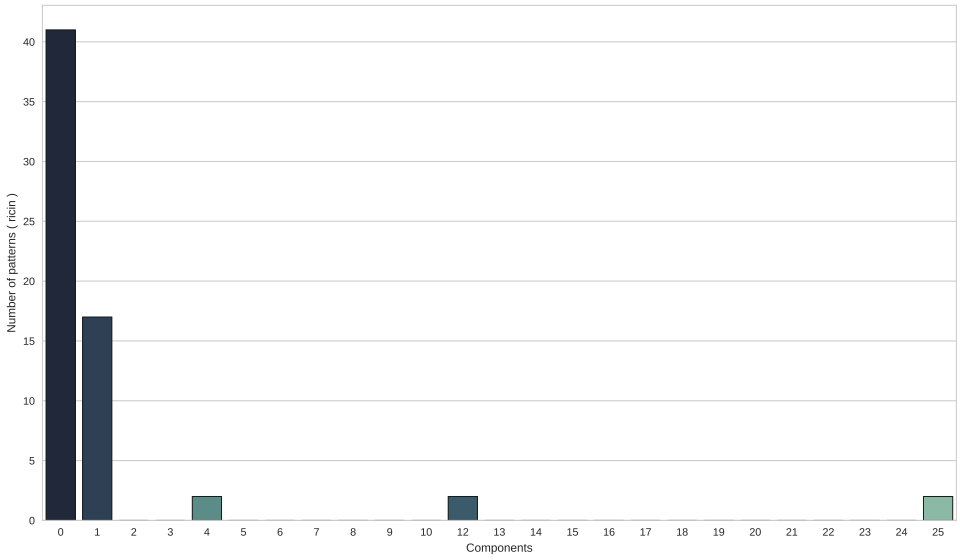

Figure 4.16: Distribution of gSpan patterns in CALI components of the Ricin graph. These pattern-graphs were searched in CALI components applying an exact matching subgraph algorithm

4.4 CALI Comparison with Experimental Results

Analysis of patterns and crucial interactions in PPI and PLI are usually done by checking manually the results of laboratory experiments (assays), with the support of visualization software tools. This requires a huge and deep biochemical knowledge about the protein complexes being analyzed and the tools used. However, as we are proposing a PLI graph model, which does require neither laboratory experiments nor deep biochemical knowledge for its generation, we compare CALI results for the CDK2 dataset with the work of Schonbrunn et al., 2013 and CALI results for the Ricin dataset with the work of Ho et al., 2009.

4.4.1 CDK2

Schonbrunn et al. [2013] present figures highlighting interacting residues and we present these residues in Table 4.2. We notice that residues from the hinge region (GLU81, PHE82 and LEU83) are consistently interacting according to CALI and appear in Figure 4.18. Moreover, our model was able to spot some residues that frequently interact with ligands through hydrophobic interactions: ILE10, LYS33, ALA31 and LEU134.

Research done by Kuhn et al. [2011] of a 3-aminoindazole compound with CDK2

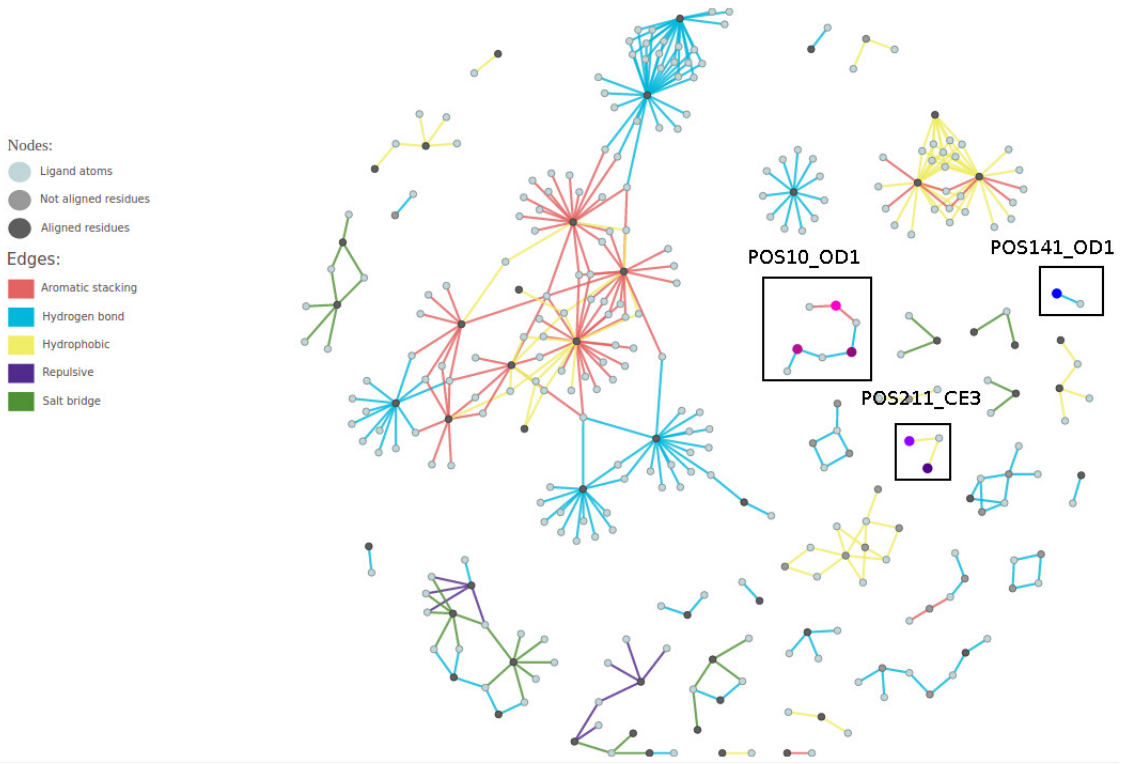

Figure 4.17: gSpan patterns found in CALI components of the ricin graph applying exact matching graph isomorphism. There are three gSpan patterns, with the protein atoms highlighting in blue (one hydrogen bond interaction), magenta (a planar graph with 4 hydrogen bond interactions and 2 aromatic stacking interactions) and purple (two hydrophobic interactions).

(PDB id 2R64), which is not in our CDK2 dataset, shows high scores for correlated hydrogen bond interactions. They identify three nitrogen hydrogen bond donors and acceptors that interact with the axis backbone (GLU81 - LEU83). Using CALI, these interactions are easily detected, without using score functions, just watching the two components formed in our graph by GLU81 e LEU83, see Figure 4.18. The residue GLN85 from the front specificity pocket does not appear in the CALI graphs because it interacts with a water molecule (that are not considered in the calculation of contacts), which binds to inhibitor. However, the residue ASP145 from the binding packet appears in the CALI graphs because it interacts with a water molecule and the inhibitor.

4.4.2 Ricin

The work of Ho et al. [2009] call our attention to 2 conserved Tyrosines (TYR80 and TYR123) establishing π-stacking (aromatic interactions); ARG180 at one end of the π stacking providing cationic polarization and GLU177 serving to activate H_2O

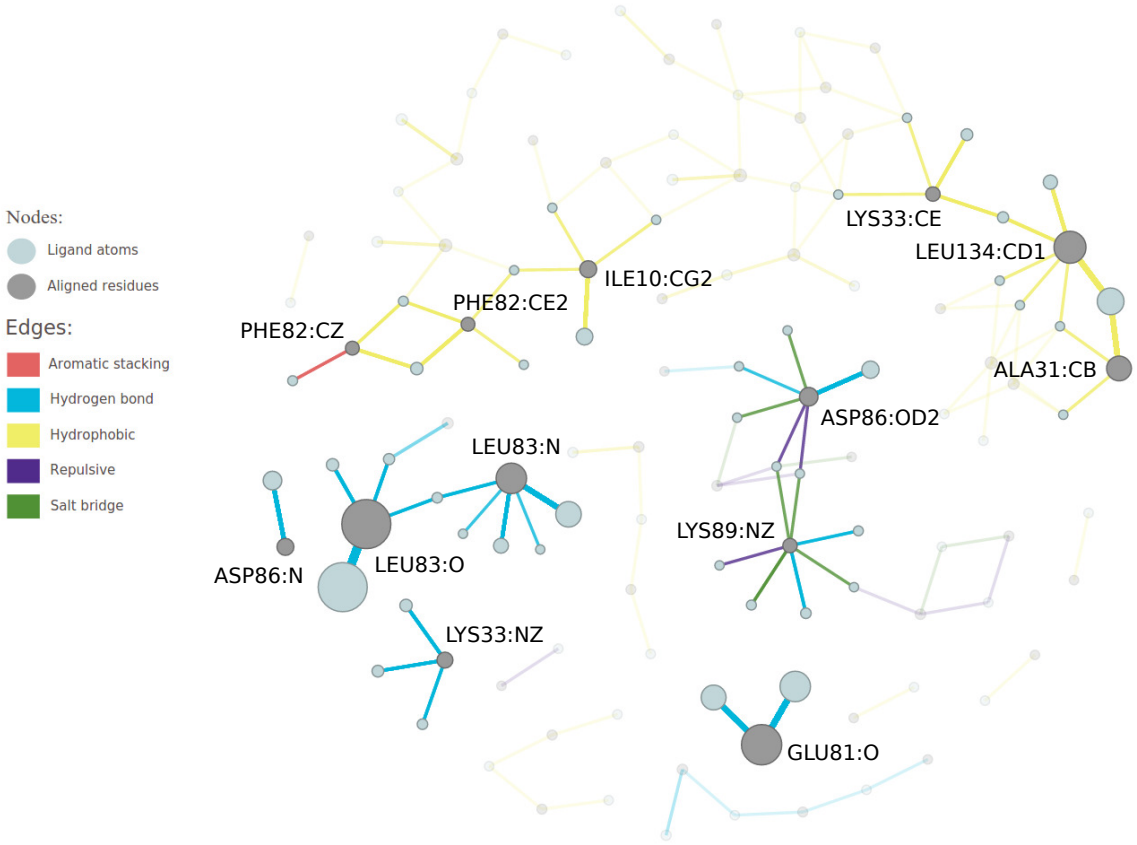

Figure 4.18: Residues from the hinge region of CDK2. In this figure, we highlight two important results: *(i)* CALI was able to spot residues from the hinge region (GLU81, PHE82 and LEU83) according to [Schonbrunn et al., 2013]. Moreover, our model was able to spot some residues that frequently interact with ligands through hydrophobic interactions: ILE10, LYS33, ALA31 and LEU134. *(ii)* In a research done by [Kuhn et al., 2011] of a 3-aminoindazole compound with CDK2 (PDB id 2R64), which is not in our CDK2 dataset, they identified three nitrogen hydrogen bond donors and acceptors that interact with the axis backbone (GLU81 - LEU83). Using CALI, these interactions are easily detected just watching the two components formed in our graph by GLU81 e LEU83. These patterns were obtained by CALI G" model.

nucleophiles. These residues are shown in Figure 4.19.

Authors present a two-dimensional map of the cyclic transition state analogue inhibitor bound to the active site of RTA. They highlight hydrogen bonds between the protein and the ligand, π-stackings and draw water molecules as well. The residues involved in such interactions are presented in Table 4.3, with each of their interacting atoms and some of their properties according to our model (degree and closeness).

CALI information on Table 4.3 was obtained by an empirical choice of filters in CALI, using the sliders of the visual interface that, by eliminating the nodes with few

Table 4.2: Binding site residues of CDK2 interacting with the 2 most potent sulfonamide analogue inhibitors.

Residue	Atom	Degree	Closeness	CALI	FSM
ASP145	CB	2	0.15	•	×
	CG	6	0.20	•	×
	OD1	3	0.50	•	×
LYS33	CB	1	0.15	•	×
	CD	3	0.20	•	×
	CE	11	0.25	✓	×
	CG	3	0.20	•	×
	NZ	14	1.00	✓	×
ASP86	N	16	1.00	✓	•
	CB	2	0.10	•	×
	OD1	3	0.35	•	×
	OD2	19	0.50	✓	×
LYS89	CB	1	1	•	×
	CE	1	1	•	×
	NZ	11	0.45	✓	×
GLN85	-	-	-	×	×
HIS84	O	4	0.3	•	×
LEU83	N	42	0.45	✓	✓
	O	78	0.6	✓	✓
PHE82	CE2	10	0.15	✓	×
	CZ	10	0.15	✓	×
GLU81	O	61	1	✓	✓
PHE80	CB	6	0.15	•	×
	CG	4	0.15	•	×
	CD2	3	0.15	•	×
	CE2	3	0.20	•	×
	CZ	2	0.15	•	×

Residues in bold are part of the hinge region. The column **CALI** is filled as follows: ✓ refers to nodes considered by us as frequent because their degrees are above the lower 10% (above 8.70); • is used when the residue is present in the model but is not a hub (degree is bellow the 10%) and × for residues not found in our model. The column **FSM** shows which residues were recovered in frequent patterns from FSM strategy as follows: ✓ is used for residues found interacting in patterns; • is used for residues found as an isolated pattern (one node graph) and × for residues not found among patterns.

connections (under 10% of the highest degree), keep the most connected and interesting nodes or, in other words, we clean up the network of the non-conserved and weakly connected atoms.

Table 4.3 shows that from the 12 mentioned residues, 2 (GLU208 and ARG134) did not directly interact with the ligand according to Ho et al. [2009]. From the 10 interacting residues, 6 were considered frequent and highly connected by our approach, which configures a conserved pattern of interaction. Note that TYR80 and TYR123 have highly connected atoms with considerable degrees and ARG180 as well is remarkably connected. On the other hand, GLU177 show up only interacting with two inhibitors, which are also interacting with ARG180, forming just two edges in the CALI graphs. Despite GLU177 is a catalytic site residue, it is not directly connected with the ligand (it is connected with a water molecule), which we did not consider in our computation.

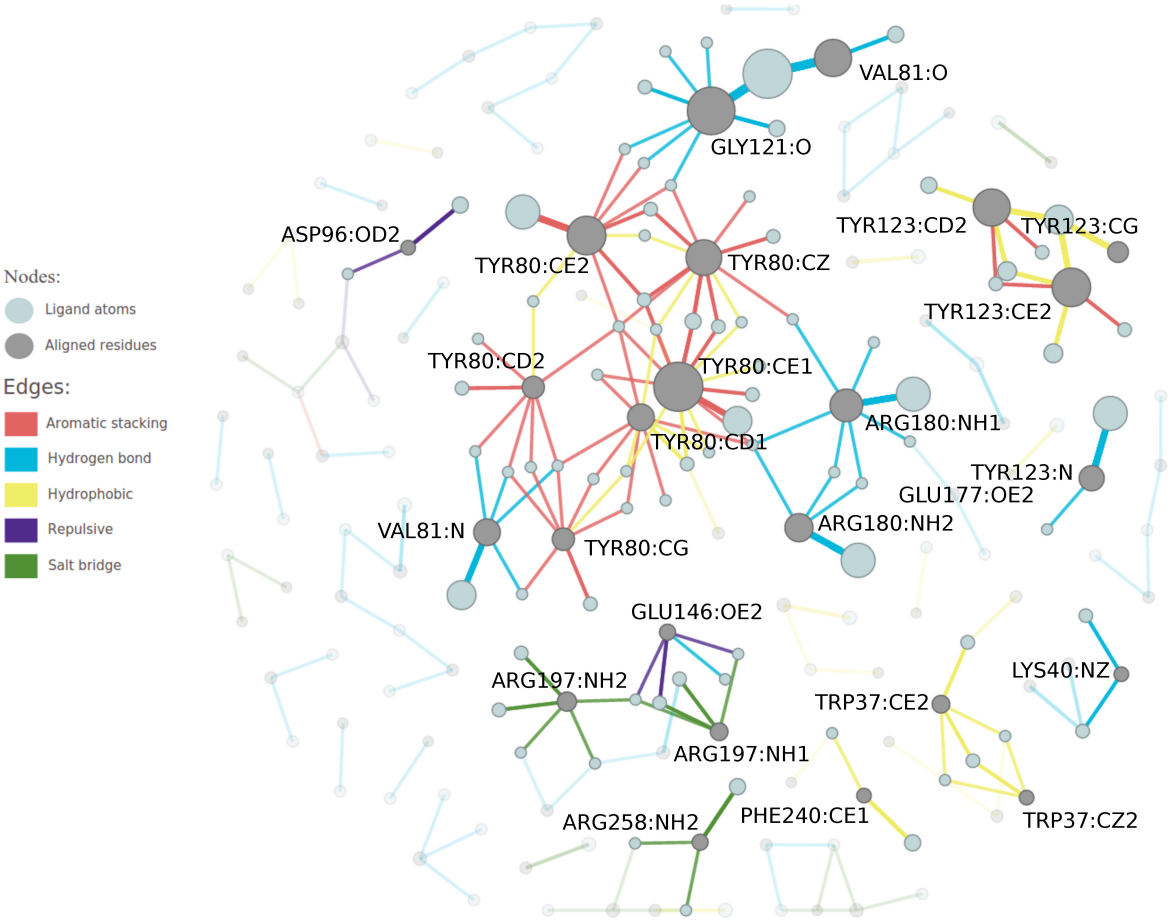

Figure 4.19: Important residues in the interaction between Ricin A chain and 28S rRNA. In [Ho et al., 2009], authors co-crystalize RTA with a transition state analogue inhibitor that mimics sarcin-ricin recognition loop of the 28S rRNA. They call our attention to 2 conserved TYR residues (TYR80 and TYR123) establishing π-stacking (aromatic interactions); ARG180 at one end of the π stacking providing cationic polarization and GLU177 serving to activate H_2O nucleophiles. CALI was able to spot the mentioned residues. These patterns were obtained by CALI G" model.

Table 4.3: Active site residues of Ricin RTA interacting with a cyclic transition state analogue inhibitor.

Residue	Atom	Degree	Closeness	CALI	FSM
GLY121	O	26	0.25	✓	✓
ARG180	NH1	16	0.25	✓	•
	NH2	13	0.25	✓	•
VAL81	N	12	0.20	✓	•
	O	19	0.20	✓	•
ASP96	OD1	3	0.50		✓
	OD2	4	0.40	✓	•
ASP100	OD2	1	0.30	•	•
ASP75	OD2	2	0.30	•	•
ASN78	ND2	3	1.00	•	✓
TYR80	CD1	12	0.30	✓	•
	CD2	9	0.25	✓	✓
	CE1	27	0.30	✓	•
	CE2	20	0.30	✓	✓
	CG	9	0.20	✓	•
	CZ	18	0.30	✓	•
TYR123	N	11	1.00	✓	•
	CD2	19	0.60	✓	•
	CE2	20	0.70	✓	✓
	CG	8	0.45	✓	•
GLU208*	-	-	-	×	-
GLU177	OE2	2	0.20	•	×
ARG134*	-	-	-	×	-

According to Ho et al., 2009, residues marked with an * are not directly interacting with the inhibitor. The column **CALI** is filled as follows: ✓ refers to nodes considered by us as frequent because their degrees are above the lower 10% (above 3.17); • is used when the residue is present in the model but is not a hub (degree is bellow the 10%) and × for residues not found in our model. The column **FSM** shows which residues were recovered in frequent patterns from FSM strategy as follows: ✓ is used for residues found interacting in patterns; • is used for residues found as an isolated pattern (one node graph) and × for residues not found among patterns.

Chapter 5

Conclusions and Future Works

In this work we proposed CALI, a complex network based strategy to model protein-ligand interactions and reveal frequent relevant biological patterns among them. Our proposal intended to increase the comprehension of key factors in molecular recognition, which is an essential step towards the prediction of protein-ligand complexes.

The PLI graphs modeled using CALI represent protein and ligand atoms as nodes labeled according to their physicochemical properties and the interactions between protein and ligand atoms are represented as edges. Through a sequence alignment is derived a structure superposition and protein nodes are merged. Each new merged node keeps all the interactions established with several ligands by original protein atoms. Ligand atoms can also be merged according to their atom type, protein atoms with whom they interact and the type of established interactions. This novel modeling, associated to complex network metrics and interactive visualization techniques, allows to highlight those residues that frequently interact with ligands in the whole dataset, revealing patterns of interactions. Larger components in the graphs (G' and G'') generated by CALI are equivalent of key regions of the binding site.

We compared patterns obtained with CALI to some proven relevant protein-ligand interactions from experimental studies for Ricin and CDK2 dataset and our strategy was able to highlight the 90% of the CDK2 binding site residues and the 100% [1] of the Ricin RTA active site residues. We compared CALI with a previous FSM based strategy and CALI found more patterns experimentally determined in an easier and intuitive manner. We also made a comparison with gSpan and we find that CALI discover the same patterns but it is able to show their localization in the protein complex, conserving their biochemical context and without fragmenting them in many

[1] Here are not included two residues (GLU208 and ARG134) which do not interact with the inhibitor.

small repetitive patterns. On the other hand, gSpan needs several tests with different support values and is only able to find a considerable amount of patterns when the support value is bellow than 10%.

It is important to point out that CALI is not computationally expensive, its execution time depends only of the PLI graph size, avoiding the expensive computation of FSM and the subgraph isomorphism algorithm to map a frequent subgraph into the input graph. Moreover, our strategy provides a general view of the input interaction dataset (several protein complexes), showing the most common PLI from a global perspective, while previous strategies provide a result segmented in protein complexes or groups of them, which makes it difficult to understand what is globally relevant. The prototype web application is available at: www.lbs.dcc.ufmg.br/projetos/cali/, with the datasets of CDK2 and Ricin.

There are several future works for CALI. The first one is to do more experiments for larger PLI datasets. One possibility could be the *PDBbind* database, which has 1300 diverse protein-ligand complexes and was used in the paper of Ballester et al. [2014]. Also, it would be interesting to collect experimental data to create protein family datasets for a more accurate analysis as for the CDKs. The database KLIFS [Kooistra et al., 2016] of structural kinase-ligand interactions could improve the process of finding patterns for the kinase protein family.

The second one, includes using others biological databases together with CALI, which could allow us to be more conclusive about the network centrality metrics (i.e. betweenness). Some promising databases for this purpose are the catalytic site atlas (CSA) and Platinum [Pires et al., 2015], a structural database of the effects of mutations in protein-ligand complexes. We also want to improve the web application allowing users to select their own dataset from the PDB.

The third one, is improve the CALI model including water molecules, which can interact with the protein complex and play a crucial role in the ligand binding. This will need to be done when the atomic contacts are calculated. However, this fact need to be studied carefully, because water molecules are not always interacting with protein-ligand complexes and include many of them can be computationally expensive. In the two datasets used, CALI was not able to detect interactions involving water molecules because there were not considered in the atomic contact calculation.

Bibliography

Akli, S., Van Pelt, C. S., Bui, T., Meijer, L., and Keyomarsi, K. (2011). Cdk2 is Required for Breast Cancer Mediated by the Low-Molecular-Weight Isoform of Cyclin E. *Cancer Research*, 71(9):3377--3386.

Akoglu, L., McGlohon, M., and Faloutsos, C. (2010). OddBall: Spotting Anomalies in Weighted Graphs. In *Proceedings of the 14th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Volume Part II*, PAKDD'10, pages 410--421, Berlin, Heidelberg. Springer-Verlag.

Ballester, P. J., Schreyer, A., and Blundell, T. L. (2014). Does a More Precise Chemical Description of Protein - Ligand Complexes Lead to More Accurate Prediction of Binding Affinity ? *Journal of Chemical Information and Modeling*, 54(3):944--955. ISSN 1549-9596.

Bang-Jensen, J. and Gutin, G. (2007). Theory, algorithms and applications. 101.

Barabási, A.-L. (2016). *Network Science*. Cambridge University Press, Place of publication not identified. ISBN 9781107076266.

Bickerton, G. R., Higueruelo, A. P., and Blundell, T. L. (2011). Comprehensive, atomic-level characterization of structurally characterized protein-protein interactions: the PICCOLO database. *BMC Bioinformatics*, 12:313. ISSN 1471-2105.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U. (2006). Complex networks: Structure and dynamics. 424(4):175--308. ISSN 0370-1573.

Bonetta, L. (2010). Protein-protein interactions: Interactome under construction. *Nature*, 468(7325):851--854. ISSN 0028-0836.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163--177.

Chen, Q., Luo, H., Zhang, C., and Chen, Y.-P. P. (2015). Bioinformatics in protein kinases regulatory network and drug discovery. *Mathematical Biosciences*, 262:147--156.

Cheng, F., Liu, C., Jiang, J., Lu, W., Li, W., Liu, G., Zhou, W., Huang, J., and Tang, Y. (2012). Prediction of Drug-Target Interactions and Drug Repositioning via Network-Based Inference. *PLOS Comput Biol*, 8(5):e1002503. ISSN 1553-7358.

Chi, Y., Yang, Y., and Muntz, R. R. (2003). Indexing and mining free trees. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 509--512. IEEE.

Chi, Y., Yang, Y., and Muntz, R. R. (2004). Hybridtreeminer: An efficient algorithm for mining frequent rooted trees and free trees using canonical forms. In *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, pages 11--20. IEEE.

Clark, A. M. and Labute, a. P. (2007). 2d Depiction of Protein-Ligand Complexes. *Journal of Chemical Information and Modeling*, 47(5):1933--1944.

Clauset, A., Moore, C., and Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98--101. ISSN 0028-0836.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub) graph isomorphism algorithm for matching large graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1367--1372.

Cowan, M. K. and Bunn, J. (2015). *Microbiology Fundamentals: A Clinical Approach*. McGraw-Hill Education, New York, NY, 2 edition edition. ISBN 9780078021046.

de Melo, R. C., Lopes, C., Fernandes Jr, F. A., da Silveira, C. H., Santoro, M. M., Carceroni, R. L., Meira Jr, W., and Araújo Ade, A. (2006). A contact map matching approach to protein structure similarity analysis. *Genet. Mol. Res*, 5(2):284–308.

Derenyi, I., Palla, G., and Vicsek, T. (2005). Clique Percolation in Random Networks. *Physical Review Letters*, 94(16). ISSN 0031-9007, 1079-7114.

Diestel, R. (2000). Graph theory, volume 173 of. *Graduate texts in mathematics*, pages 24--26.

Dunn, M. F. (2001). *Protein-Ligand Interactions: General Description*. John Wiley & Sons, Ltd.

Estrada, E. and Hatano, N. (2008). Communicability in complex networks. *Physical Review E*, 77(3):036111.

Estrada, E. and Rodríguez-Velázquez, J. A. (2005). Spectral measures of bipartivity in complex networks. *Physical Review E*, 72(4):046105.

Fassio, A. V. (2015). nAPOLI: uma ferramenta para análise de interações proteína-ligante.

Fuller, J. C., Martinez, M., Henrich, S., Stank, A., Richter, S., and Wade, R. C. (2015). LigDig: a web server for querying ligan–protein interactions. *Bioinformatics*, 31(7):1147--1149. ISSN 1367-4803, 1460-2059.

Gallina, A. M., Bisignano, P., Bergamino, M., and Bordo, D. (2013). PLI: a web-based tool for the comparison of protein-ligand interactions observed on PDB structures. *Bioinformatics*, 29(3):395--397. ISSN 1367-4803, 1460-2059.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA. ISBN 9780716710448.

Gonçalves, W. R., Gonçalves-Almeida, V. M., Arruda, A. L., Meira, W., da Silveira, C. H., Pires, D. E., and de Melo-Minardi, R. C. (2015). Pdbest: a user–friendly platform for manipulating and enhancing protein structures. *Bioinformatics*, page btv223.

Gotoh, O. (1996). Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *Journal of molecular biology*, 264(4):823--838.

Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11--15, Pasadena, CA USA.

Held, M., Metzner, P., Prinz, J.-H., and Noe, F. (2011). Mechanisms of Protein-Ligand Association and Its Modulation by Protein Mutations. *Biophysical Journal*, 100(3):701--710. ISSN 0006-3495.

Higueruelo, A. P., Schreyer, A., Bickerton, G. R. J., Pitt, W. R., Groom, C. R., and Blundell, T. L. (2009). Atomic Interactions and Profile of Small Molecules Disrupting Protein-Protein Interfaces: the TIMBAL Database. *Chemical Biology & Drug Design*, 74(5):457--467. ISSN 1747-0285.

Ho, M.-C., Sturm, M. B., Almo, S. C., and Schramm, V. L. (2009). Transition state analogues in structures of ricin and saporin ribosome-inactivating proteins. *Proceedings of the National Academy of Sciences*, 106(48):20276--20281.

Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of molecular evolution*, 20(2):175--186.

Hu, H., Yan, X., Huang, Y., Han, J., and Zhou, X. J. (2005). Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl 1):i213--i221. ISSN 1367-4803, 1460-2059.

Huan, J., Wang, W., Prins, J., and Yang, J. (2004). SPIN: Mining Maximal Frequent Subgraphs from Graph Databases. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 581--586, New York, NY, USA. ACM.

Jahn, K. and Kramer, S. (2005). Optimizing gSpan for molecular datasets. In *Proceedings of the third international workshop on mining graphs, trees and sequences (MGTS-2005)*, pages 77--89.

Jiang, C., Coenen, F., and Zito, M. (2013). A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(01):75--105.

Junker, B. H. and Schreiber, F. (2008). *Analysis of Biological Networks*. Wiley. ISBN 9780470041444.

Kahraman, A., Morris, R. J., Laskowski, R. A., and Thornton, J. M. (2007). Shape variation in protein binding pockets and their ligands. *J. Mol. Biol.*, 368(1):283--301.

Kahraman, A. and Thornton, J. M. (2008). Methods to characterize the structure of enzyme binding sites. In Schwede, T. and Peitsch, M. C., editors, *Computational Structural Biology: Methods & Applications Vol. 1*, pages 189--221. World Scientific Pub. Co., Singapore.

Kitano, H. (2002). Computational systems biology. *Nature*, 420(6912):206--210.

Kleinberg, J. and Easley, D. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge university press edition. ISBN 9780521195331.

Kooistra, A. J., Kanev, G. K., van Linden, O. P. J., Leurs, R., de Esch, I. J. P., and de Graaf, C. (2016). KLIFS: a structural kinase-ligand interaction database. *Nucleic Acids Research*, 44(D1):D365--D371. ISSN 0305-1048, 1362-4962.

Kuhn, B., Fuchs, J. E., Reutlinger, M., Stahl, M., and Taylor, N. R. (2011). Rationalizing Tight Ligand Binding through Cooperative Interaction Networks. *Journal of Chemical Information and Modeling*, 51(12):3180--3198. ISSN 1549-9596, 1549-960X.

Kuramochi, M. and Karypis, G. (2004). Grew-a scalable frequent subgraph discovery algorithm. In *Data Mining, 2004. ICDM 04. Fourth IEEE International Conference on*, pages 439--442. IEEE.

Kuramochi, M. and Karypis, G. (2005). Finding Frequent Patterns in a Large Sparse Graph*. *Data Mining and Knowledge Discovery*, 11(3):243--271. ISSN 1384-5810, 1573-756X.

Laimer, J., Hiebl-Flach, J., Lengauer, D., and Lackner, P. (2016). MAESTROweb: a web server for structure-based protein stability prediction. *Bioinformatics*, 32(9):1414--1416. ISSN 1367-4803, 1460-2059.

Laskowski, R. A. and Swindells, M. B. (2011). LigPlot+: Multiple Ligand-Protein Interaction Diagrams for Drug Discovery. *Journal of Chemical Information and Modeling*, 51(10):2778--2786. ISSN 1549-9596.

Latapy, M., Magnien, C., and Vecchio, N. D. (2008). Basic notions for the analysis of large two-mode networks. *Social Networks*, 30(1):31--48. ISSN 03788733.

Lefranc, M.-P., Giudicelli, V., Ginestoux, C., Jabado-Michaloud, J., Folch, G., Bellahcene, F., Wu, Y., Gemrot, E., Brochet, X., Lane, J., Regnier, L., Ehrenmann, F., Lefranc, G., and Duroux, P. (2009). IMGT(R), the international ImMunoGeneTics information system(R). *Nucleic Acids Research*, 37(Database):D1006--D1012. ISSN 0305-1048, 1362-4962.

Liu, P.-F., Kihara, D., and Park, C. (2011). Energetics-Based Discovery of Protein-Ligand Interactions on a Proteomic Scale. *Journal of Molecular Biology*, 408(1):147--162. ISSN 00222836.

Liu, R. and Hu, J. (2011). Computational prediction of heme-binding residues by exploiting residue interaction network. *PLoS One*, 6(10):e25560--e25560.

MacCuish, J. and MacCuish, N. (2010). *Clustering in Bioinformatics and Drug Discovery*. Chapman & Hall/CRC Mathematical and Computational Biology. CRC Press. ISBN 9781439816790.

Mancini, A. L., Higa, R. H., Oliveira, A., Dominiquini, F., Kuser, P. R., Yamagishi, M. E. B., Togawa, R. C., and Neshich, G. (2004). Sting contacts: a web-based application for identification and analysis of amino acid contacts within protein structure and across protein interfaces. *Bioinformatics*, 20(13):2145–2147.

Medina-Franco, J. L., Mendez-Lucio, O., and Martinez-Mayorga, K. (2014). The Interplay Between Molecular Modeling and Chemoinformatics to Characterize Protein Ligand and Protein Protein Interactions Landscapes for Drug Discovery. In *Advances in Protein Chemistry and Structural Biology*, volume 96, pages 1--37. Elsevier.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443--453.

Nepusz, T., Yu, H., and Paccanaro, A. (2012). Detecting overlapping protein complexes in protein-protein interaction networks. *Nat Meth*, 9(5):471--472. ISSN 1548-7091.

Newman, M. (2010). *Networks: An Introduction*. OUP Oxford. ISBN 9780199206650.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113.

O'Donoghue, S. I., Goodsell, D. S., Frangakis, A. S., Jossinet, F., Laskowski, R. A., Nilges, M., Saibil, H. R., Schafferhans, A., Wade, R. C., Westhof, E., and Olson, A. J. (2003). Visualization of macromolecular structures. *Nature Methods*, 7:S42--S55. ISSN 1548-7091.

Okabe, A. et al. (2009). *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. Wiley.

Pires, D. E., de Melo-Minardi, R. C., da Silveira, C. H., Campos, F. F., and Meira, W. (2013). acsm: noise-free graph-based signatures to large-scale receptor-based ligand prediction. *Bioinformatics*, 29(7):855--861.

Pires, D. E. V., Blundell, T. L., and Ascher, D. B. (2015). Platinum: a database of experimentally measured effects of mutations on structurally defined protein-ligand complexes. *Nucleic Acids Research*, 43(D1):D387--D391. ISSN 0305-1048, 1362-4962.

Poupon, A. (2004). Voronoi and voronoi-related tessellations in studies of protein structure and interaction. *Current Opinion in Structural Biology*, 14(2):233–241.

Rarey, M., Kramer, B., and Lengauer, T. (1999). Docking of hydrophobic ligands with interaction-based matching algorithms. *Bioinformatics*, 15(3):243--250. ISSN 1367-4803, 1460-2059.

Restrepo O., J. and Zuluaga, H. F. (2011). *Fundamentos Estructurales de Bioquimica*. Programa Editorial Universidad del Valle, Cali, Colombia, 1 edition.

Rose, P. W., Prlić, A., Bi, C., et al. (2015). The rcsb protein data bank: views of structural biology for basic and applied research and education. *Nucleic acids research*, 43(D1):D345--D356.

Rückert, U. and Kramer, S. (2004). Frequent free tree discovery in graph data. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 564--570. ACM.

Rutenber, E. et al. (1991). Crystallographic refinement of ricin to 2.5 å. *Proteins: Structure, Function, and Bioinformatics*, 10(3):240--250.

Salentin, S., Schreiber, S., Haupt, V. J., Adasme, M. F., and Schroeder, M. (2015). PLIP: fully automated protein-ligand interaction profiler. *Nucleic Acids Research*, 43(W1):W443--W447. ISSN 0305-1048, 1362-4962.

Schonbrunn, E., Betzi, S., Alam, R., Martin, M. P., Becker, A., Han, H., Francis, R., Chakrasali, R., Jakkaraj, S., Kazi, A., Sebti, S. M., Cubitt, C. L., Gebhard, A. W., Hazlehurst, L. A., Tash, J. S., and Georg, G. I. (2013). Development of Highly Potent and Selective Diaminothiazole Inhibitors of Cyclin-Dependent Kinases. *Journal of Medicinal Chemistry*, 56(10):3768--3782. ISSN 0022-2623, 1520-4804.

Scott, D. E., Bayly, A. R., Abell, C., and Skidmore, J. (2016). Small molecules, big targets: drug discovery faces the protein-protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8):533--550. ISSN 1474-1776.

Shapiro, G. I. (2006). Cyclin-Dependent Kinase Pathways As Targets for Cancer Treatment. *Journal of Clinical Oncology*, 24(11):1770--1783. ISSN 0732-183X, 1527-7755.

Silveira, S., Fassio, A., Silveira, C., and Melo-Minardi, R. (2015). Revealing protein-ligand interaction patterns through frequent subgraph mining. In *BIOCOMP 2015*.

Sobolev, V. et al. (1999a). Automated analysis of interatomic contacts in proteins. *Bioinformatics*, 15(4):327--332.

Sobolev, V., Sorokine, A., Prilusky, J., Abola, E. E., and Edelman, M. (1999b). Automated analysis of interatomic contacts in proteins. *Bioinformatics*, 15(4):327--332. ISSN 1367-4803, 1460-2059.

Soukup, T. and Davidson, I. (2002). *Visual Data Mining: Techniques and Tools for Data Visualization and Mining*.

Stierand, K. and Rarey, M. (2010). Drawing the PDB: Protein-Ligand Complexes in Two Dimensions. *ACS Medicinal Chemistry Letters*, 1(9):540--545.

Stryer, L., Berg, J., and Tymoczko, J. (2013). *Bioquímica. Con aplicaciones clínicas*. Reverte, Barcelona, edición: 7 edition. ISBN 9788429176025.

Szilágyi, A., Grimm, V., Arakaki, A. K., and Skolnick, J. (2005). Prediction of physical protein-protein interactions. *Physical biology*, 2(2):S1.

Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L. J., and Mering, C. v. (2011). The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(suppl 1):D561--D568. ISSN 0305-1048, 1362-4962.

Taylor, N. R. (2013). Small-world network strategies for studying protein structures and bindings. *Computational and Structural Biotechnology Journal*, 5(6):1--7. ISSN 20010370.

Vanetik, N. (2010). Mining Graphs with Constraints on Symmetry and Diameter. In Shen, H. T., Pei, J., Özsu, M. T., Zou, L., Lu, J., Ling, T.-W., Yu, G., Zhuang, Y., and Shao, J., editors, *Web-Age Information Management*, number 6185 in Lecture Notes in Computer Science, pages 1--12. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-16720-1_1.

Wang, J. T., Zaki, M. J., Toivonen, H. T., and Shasha, D. (2005). *Data Mining in Bioinformatics*. Springer-Verlag, London. ISBN 1852336714.

Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., and Woolsey, J. (2006). Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl 1):D668--D672.

Yan, X. and Han, J. (2002). gspan: Graph-based substructure pattern mining. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 721--724. IEEE.

Yan, X. and Han, J. (2003). Closegraph: mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 286--295. ACM.

Yan, X., Zhou, X. J., and Han, J. (2005). Mining closed relational graphs with connectivity constraints. In *21st International Conference on Data Engineering (ICDE'05)*, pages 357--358.

Zaki, M. J. and Wagner Meira, J. (2014). *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press. ISBN 9780521766333.

Zaki, N. M., Dmitry, D., and Berengueres, J. (2013). Protein complex detection using interaction reliability assessment and weighted clustering coefficient. *BMC Bioinformatics*, 14:163.

Zhao, P. and Yu, J. X. (2007). Mining closed frequent free trees in graph databases. In *International Conference on Database Systems for Advanced Applications*, pages 91--102. Springer.

Zhao, P. and Yu, J. X. (2008). Fast frequent free tree mining in graph databases. *World Wide Web*, 11(1):71--92.

Zhou, W. and Yan, H. (2014). Alpha shape and Delaunay triangulation in studies of protein-related interactions. *Briefings in Bioinformatics*, 15(1):54--64. ISSN 1467-5463, 1477-4054.

Appendix A

Additional Tables

Table A.1: Abbreviation of amino acids

Amino Acid	Three letter code	One letter code
Alanine	ALA	A
Arginine	ARG	R
Asparagine	ASN	N
Aspartic Acid	ASP	D
Asparagine or aspartic acid	ASX	B
Cysteine	CYS	C
Glutamic Acid	GLU	E
Glycine	GLY	G
Histidine	HIS	H
Isoleucine	ILE	I
Leucine	LEU	L
Lysine	LYS	K
Methionine	MET	M
Phenylalanine	PHE	F
Proline	PRO	P
Serine	SER	S
Thereonine	THR	T
Tryptophan	TRP	W
Tyrosine	TYR	Y
Valine	VAL	V

Table A.2: Ricin dataset.

PDB id and Chain	Ligand name	PDB id and Chain	Ligand name
1BR5.A	NEO	1BR6.A	PT1
1IFS.A	ADE	1IFU.A	FMC
1IL3.A	7DG	1IL4.A	9DG
1IL5.A	DDP	1IL9.A	MOG
1J1M.A	TRE	1OBT.A	AMP
1RZO.A	GAL	1RZO.B	GAL
1RZO.C	GAL	1RZO.D	GAL
2P8N.A	ADE	3EJ5.X	EJ5
3HIO.A	C2X	3PX8.X	JP2
3PX9.X	JP3	3RTI.A	FMP
3RTI.B	GAL	3RTJ.B	GAL
4ESI.A	0RB	4HUO.X	RS8
4HUP.X	19M	4HV3.A	19L
4HV7.X	19J	4MX1.A	1MX
4MX5.X	5MX		

Table A.3: CDK2 dataset.

PDB id and Chain	Ligand name	PDB id and Chain	Ligand name
3QL8.A	X01	3QQF.A	X07
3QQG.A	X06	3QQH.A	X0A
3QQJ.A	X11	3QQK.A	X02
3QQL.A	X03	3QRT.A	X14
3QRU.A	X19	3QTQ.A	X35
3QTR.A	X36	3QTS.A	X46
3QTU.A	X44	3QTW.A	X3A
3QTX.A	X43	3QTZ.A	X42
3QU0.A	X40	3QWJ.A	X6A
3QWK.A	X62	3QX2.A	X63
3QX4.A	X4B	3QXO.A	X65
3QXP.A	X64	3QZF.A	X66
3QZG.A	X67	3QZH.A	X69
3QZI.A	X72	3R1Q.A	X75
3R1S.A	X73	3R1Y.A	X76
3R28.A	XA0	3R6X.A	X84
3R71.A	X86	3R73.A	X87
3R7E.A	X88	3R7I.A	X9I
3R7U.A	X96	3R7V.A	Z02
3R7Y.A	Z04	3R83.A	Z14
3R8L.A	Z30	3R8M.A	Z19
3R8P.A	Z46	3R8U.A	Z31
3R8V.A	Z62	3R8Z.A	Z63
3R9D.A	X6B	3R9H.A	Z67
3R9N.A	Z68	3R9O.A	Z71
3RAH.A	O1Z	3RAI.A	X85
3RAK.A	03Z	3RAL.A	04Z
3RJC.A	06Z	3RK5.A	07Z
3RK7.A	08Z	3RK9.A	09Z
3RKB.A	12Z	3RM6.A	18Z
3RM7.A	19Z	3RMF.A	20Z
3RNI.A	21Z	3ROY.A	22Z
3RPO.A	24Z	3RPR.A	25Z
3RPV.A	26Z	3RPY.A	27Z
3RZB.A	02Z	3S00.A	Z60
3S0O.A	50Z	3S1H.A	56Z
3SQQ.A	99Z		

Appendix B

Additional Images

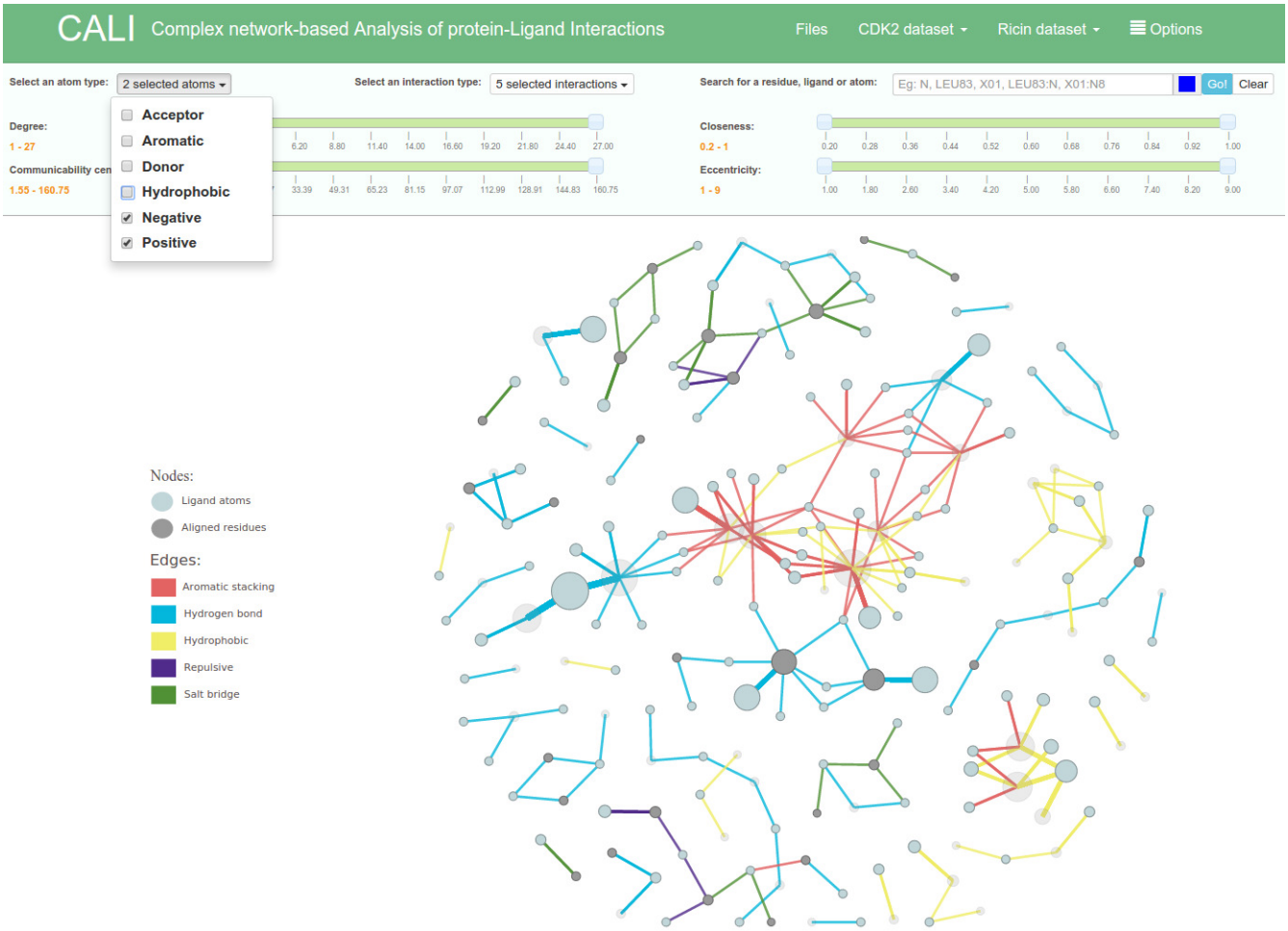

Figure B.1: Example of an atom type filtering possibility. The user can filter out the network by the types of atoms. When he/she checks or unchecks an option, the corresponding atoms (nodes) lose contrast with the background and the others are highlighted. The graph in this example is from Ricin dataset.

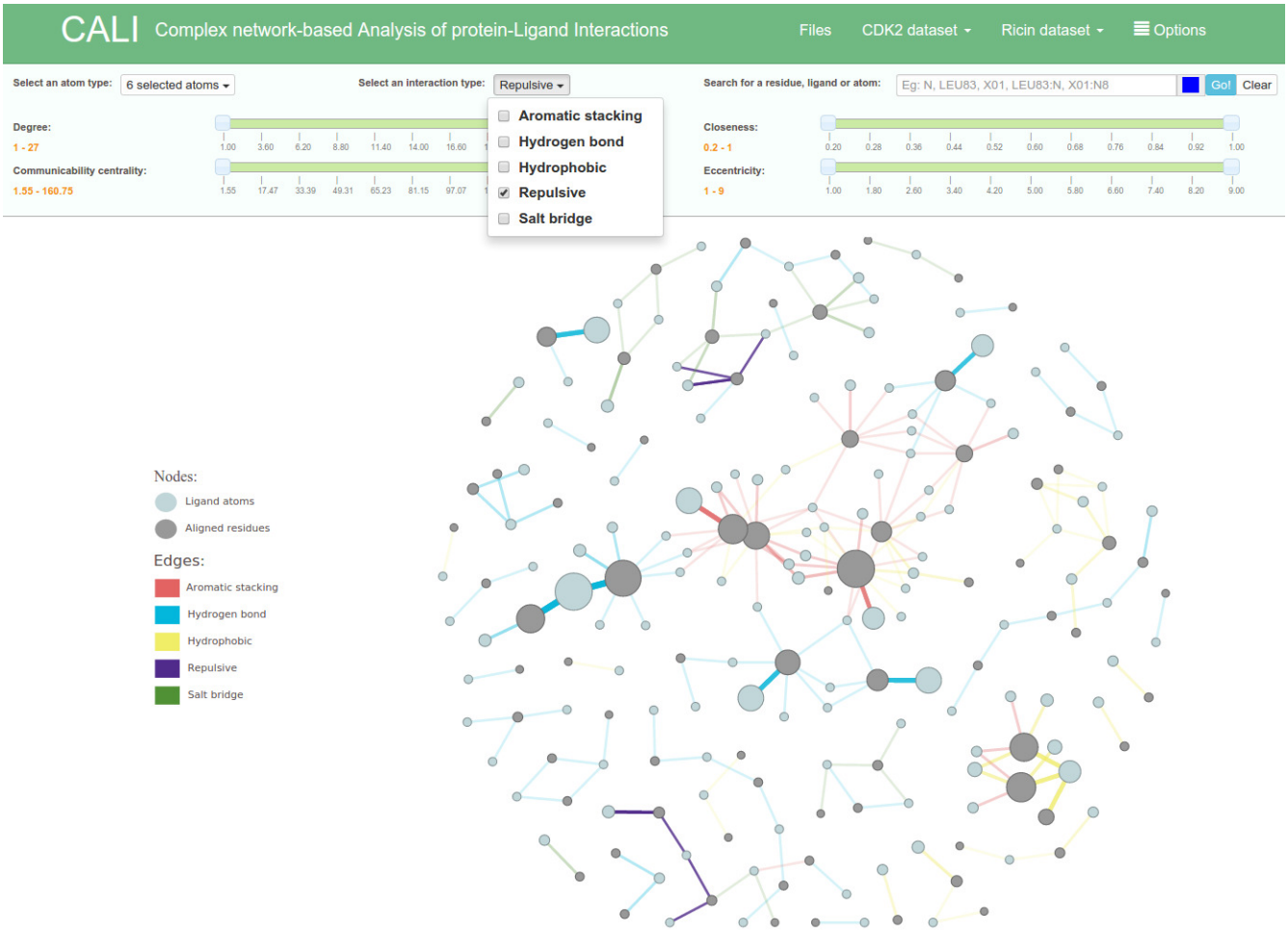

Figure B.2: Example of an interaction type filtering possibility. The user can filter out the network by the types of interactions. When he/she checks or unchecks an option, the corresponding interactions (edges) lose contrast with the background and the others are highlighted. The graph in this example is from Ricin dataset.

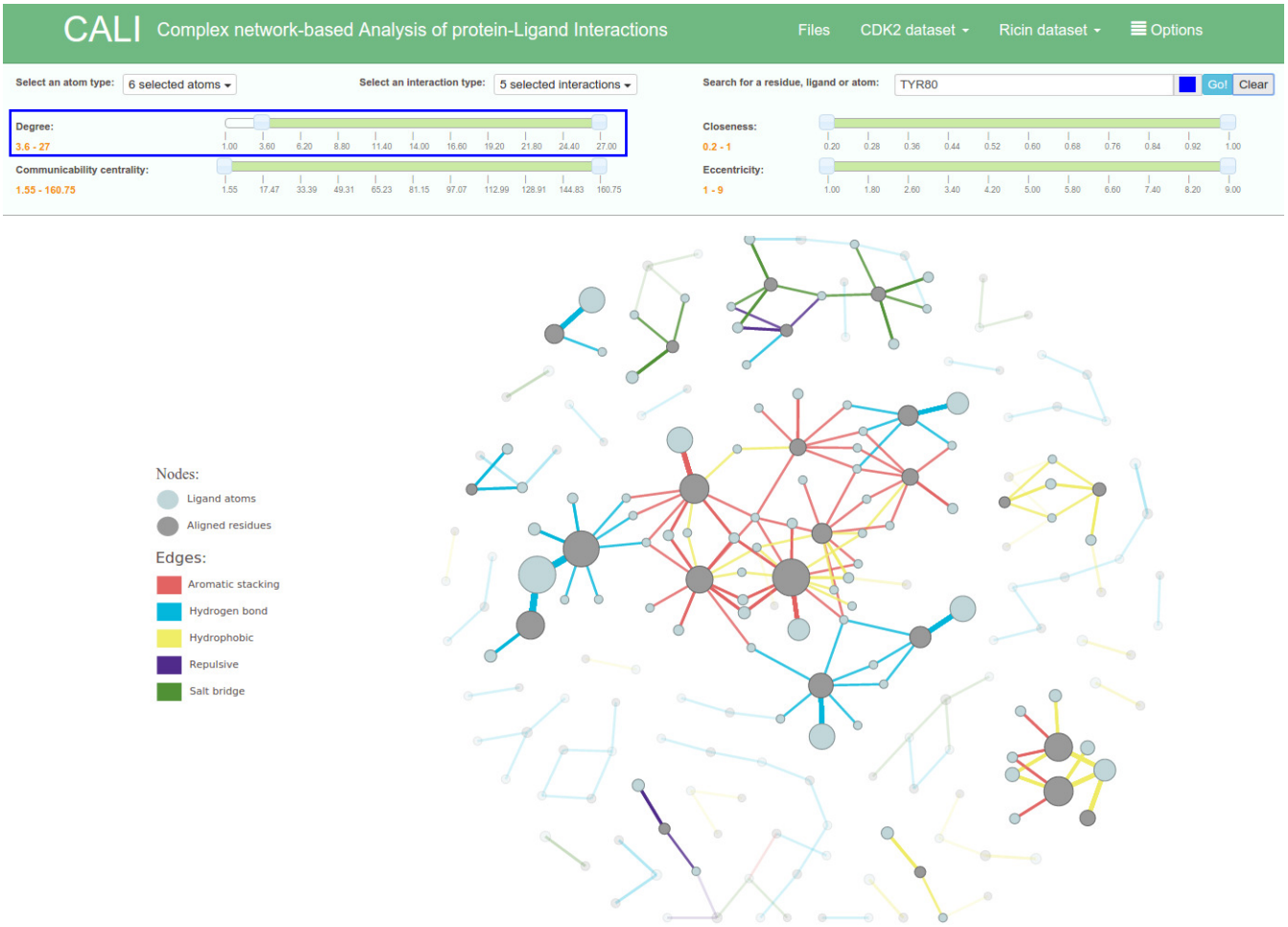

Figure B.3: Centrality measures filters. There are a total of eight different complex network centrality measures that can be used to filter out the network elements through sliders. In this figure, we filter out nodes whose degrees are bellow 10% of the maximum value. The graph is from Ricin dataset.

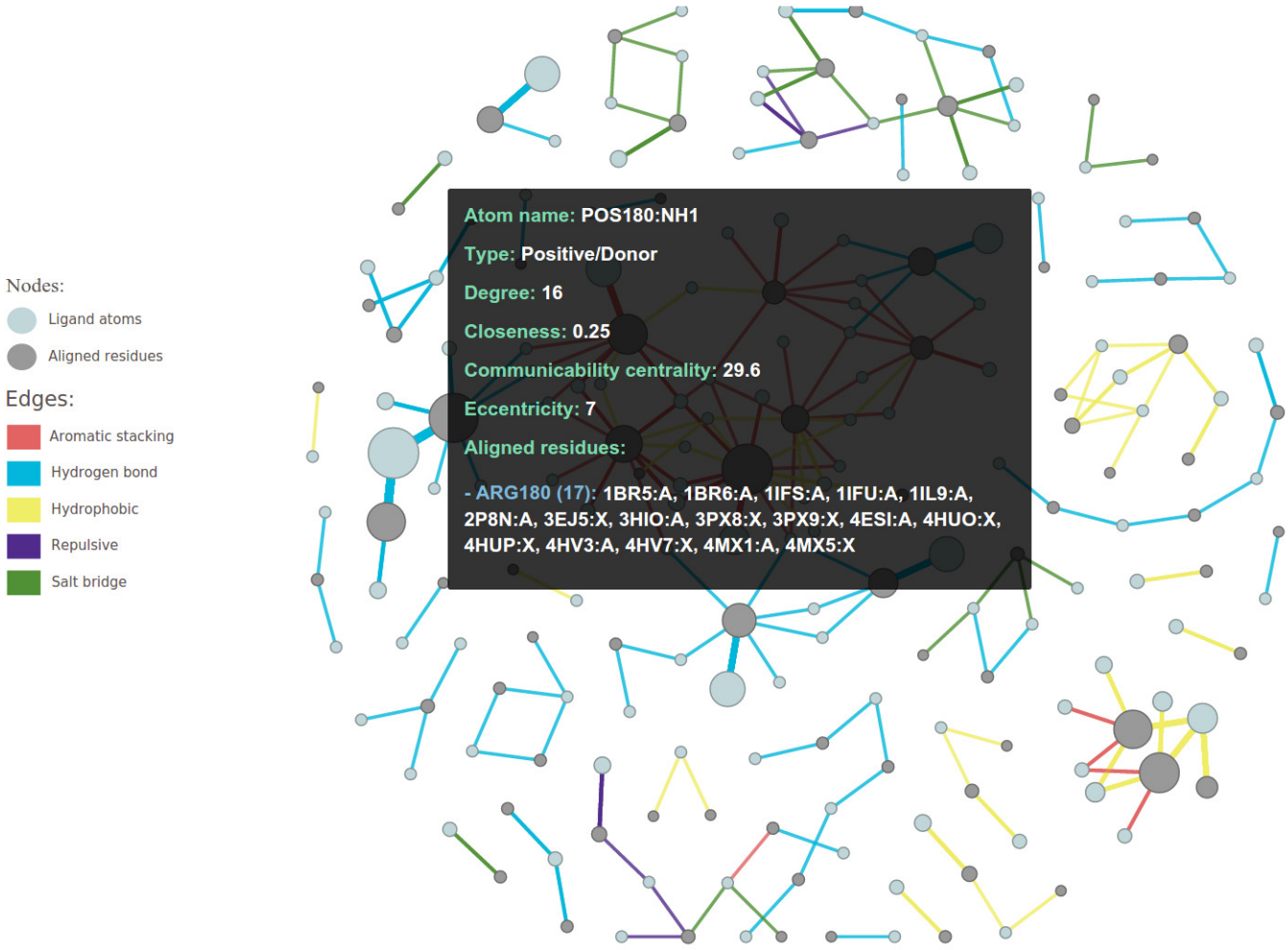

Figure B.4: Details on demand. For every element of the graph, details can be obtained on demand by passing the mouse over it. In this example, we obtain node details by positioning the mouse over such node. The graph is from Ricin dataset.

Appendix C

Detecting Protein Interactions

One of the experimental techniques available to probe interactions between proteins is *co-immunoprecipitation*. Immunoprecipitation is a technique that permits to extract a single protein from a sample. This technique replicates in laboratory what the immune system does. The immune system uses antibodies to neutralize harmful structures (protein, complexes, etc.) to the body. In the immunoprecipitation an antibody is attached to a solid surface (glass bead), then a solution containing the target protein is passed over the surface. The protein target is attached to the surface by the action of the antibody that binds together with it. Co-immunoprecipitation uses the same method but the protein can attach to others, forming a protein complex. This technique allows to recover the complex and identify every protein on it.

Unfortunately, co-immunoprecipitation is not practical for reconstruction of a whole interaction network. This approach requires individual experiments (which takes days) for every interaction to be identified. Appropriate antibody creation for the experiments could take even longer (weeks or months) and demands a considerable amount of money. In the 1990s and early 2000s the *high-throughput* methods for quickly interaction identification were adopted. The best established of these methods is the *two-hybrid screen*. In this method, a cell is persuaded to produce two proteins of interest, each attached to one of the domains of the transcription factors (binding and activation). Therefore, if the two proteins interact to form a complex, the two domains of the transcription factor will be brought together. Two-hybrid experiments takes a known protein (called bait) to probe if it interacts with a larger number of other proteins (called prey), which ones are produced using *plasmids*. The advantages of this method is that several proteins can be tested for interactions in a single experiment. Also, it is cheaper and faster than co-immunoprecipitation.

The presence of the two domains of the transcription factor, in a two-hybrid screen

experiment, can get in the way of the bait interacting with the prey protein and block the formation of a protein complex. This means that some real protein interactions are not going to be detected because of the experiment conditions. Additionally, the results of the two-hybrid screen are inaccurate, because it generates false positives (interactions that do not exist) and false negatives (not detect interactions that exist) in high proportion. The *affinity purification* methods surged as an alternative, which are able to offer accurate results. These methods use *high-throughput* detection but with antibodies as in co-immunoprecipitation, producing protein complexes that can be analyzed to identify the proteins interacting on them. The method *tandem affinity purification* generates high-quality results which is the reason to be considered the most reliable source to create protein-protein interaction (PPI) networks [Newman, 2010].

Appendix D

Model Characterization

When a network is being analyzed is common to characterize it in one of the three well-known models: Random, Small-World or Scale-free. The global properties in Table 4.1 allow us to confirm that our PLI graph model is not small-world, because it has isolated components causing that the diameter can not be calculated. This also implies that PLI graphs have not local structure inside and their connectivity is limitated. Additionally, the PLI graph do not present a random behavior, which require a low diameter and vertices with similar degree values (following the Poisson distribution). Instead, the PLI graphs have a few vertices with high degree and a huge number of vertices (protein central nodes of hubs) with very low degree (specially ligand nodes). Analyzing the network degree distribution (Section 2.1.1.4) is possible to observe if a network fits or not in a model. When the histogram of a network degree distribution is plotted, is possible to identify if it follows a power law, which allows us to say that it tends to be or in fact, that it is a scale-free network. Figures D.1 and D.2 show the degree distribution for the PLI networks. Both Figures have three lines plotted: the red one is in order 1, the green one in order 2 and the cyan one in order 3. In Figure D.1 the cyan line fits better for the CDK2 dataset, which means that it tends to be as a scale-free network, with a value of 3 for the exponent (α). In the scatter plot (Figure D.2) the degree distribution shows a curve, near the point (10,10), which barely allow us to say that it follows a power law. One more time the cyan line is the nearest to the scatter data for ricins. When logarithmic plots do not allow us to complete elucidate if the network has a power law, it is suggested (by Newman, 2010) to calculate the cumulative distribution function of the network. Figures D.3 and D.4, show this function for both datasets, but neither of them follow what is expected (a straight line) for a scale-free network. We can conclude that both PLI networks have a partially behavior of a scale-free network, but they do not follow a power law completely, because of their small

size (number of nodes and edges) and their sparsely. As PLI networks are evolving and growing (discovering new inhibitors ligands and new protein structures), is possible to say that in the future they will follow these tendency and they could follow a power law.

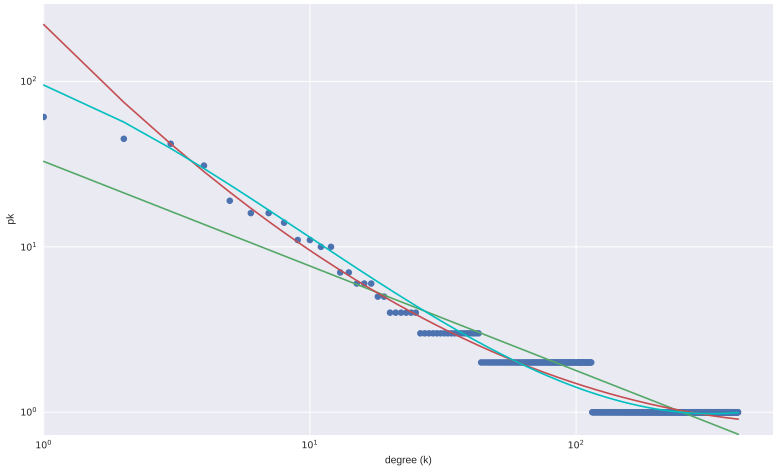

Figure D.1: Degree distribution (in logarithmic scale) for the CDK2 PLI network. In blue are represented the node degrees for this graph. Three lines are plotted represented three different functions. The green one is a straight line, $\alpha = 1$. The red one is quadratic, $\alpha = 2$. The cyan one is cubic, $\alpha = 3$. This last one line is the nearest to the degree nodes.

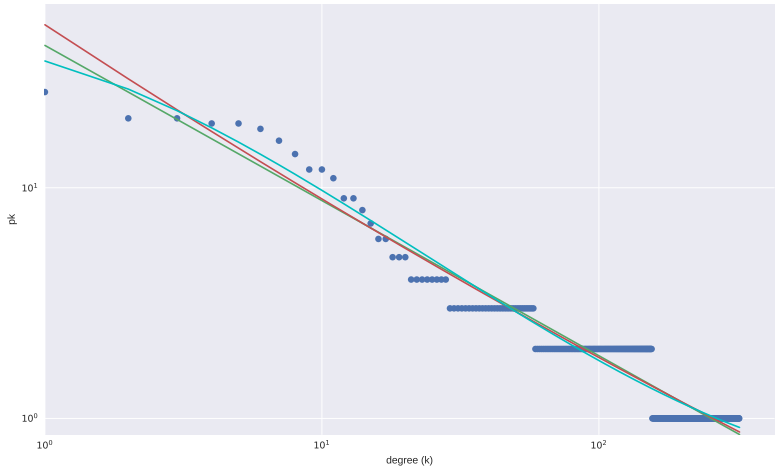

Figure D.2: Degree distribution (in logarithmic scale) for the Ricin PLI network. In blue are represented the node degrees for this graph. Three lines are plotted represented three different functions. The green one is a straight line, $\alpha = 1$. The red one is quadratic, $\alpha = 2$. The cyan one is cubic, $\alpha = 3$. This last one line is the nearest to the degree nodes, nevertheless the top degree distribution is far from all lines.

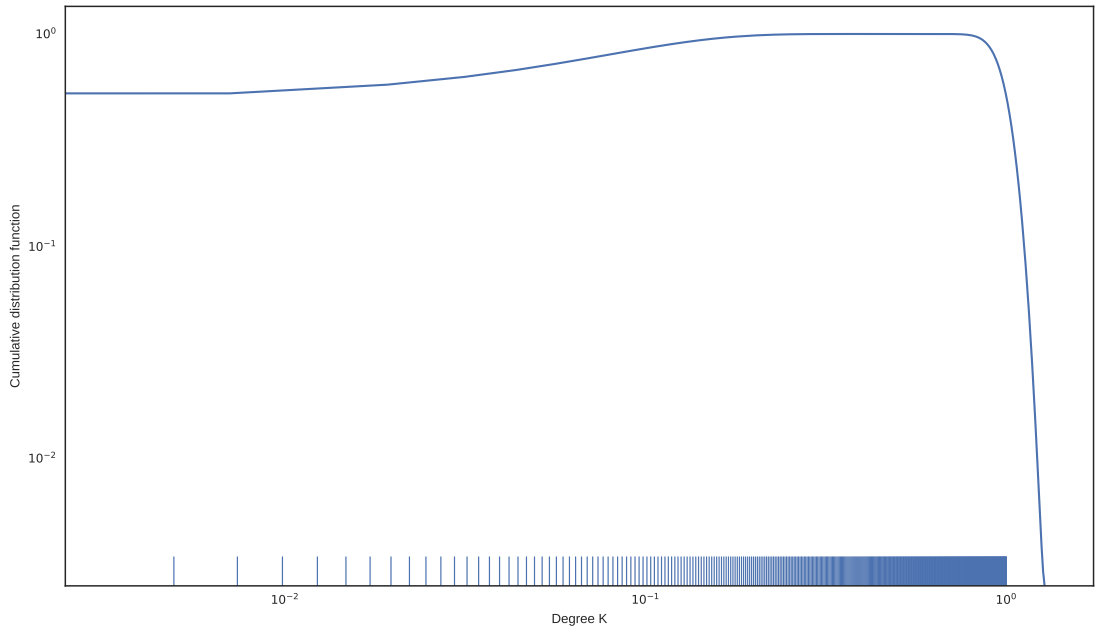

Figure D.3: Cumulative degree distribution function (in logarithmic scale) for the PLI network from the CDK2 dataset.

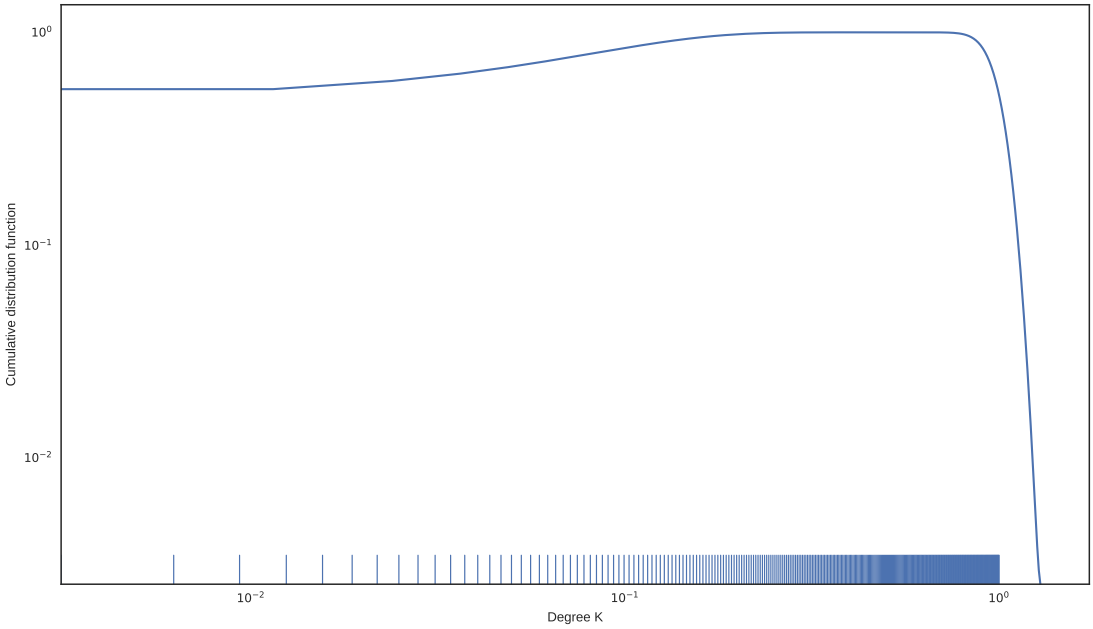

Figure D.4: Cumulative degree distribution function (in logarithmic scale) for the PLI network from the Ricin dataset.