Ca'Foscari University of Venice

Master Degree in Computer Science

Department of Environmental Sciences, Informatics and Statistics

Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks Using Dominant Sets

Jona Boscolo Cappon

Student Number 815717

Supervisor

Prof. Marcello Pelillo

Ca'Foscari University of Venice

Co-supervisor

Dr. Alberto Paccanaro

Royal Holloway University of London

Academic Year 2012/2013

"The saddest aspect of life right now is that science gathers knowledge faster than society gathers wisdom"

Isaac Asimov

Abstract

Proteins facilitate most biological processes in a cell, including cell growth, proliferation, catalyzing metabolic reactions, replicating DNA, motility, and intercellular communication. Nevertheless, proteins seldom act alone. Many times they team up into "molecular machines", called protein complexes, to undertake biological functions at cellular and systems levels.

Given the assumption that proteins perform their tasks by interacting with each other, determining the interactions is a task of great importance. This realisation results in Protein-Protein Interaction (PPI) networks. These networks can be represented as undirected graphs, in which nodes represent proteins and edges represent interactions between pairs of proteins. These networks allow us to tackle the problem of complex prediction with the aid of clustering techniques.

Unfortunately most of the standard clustering algorithms present in literature suffer from several limitations and are not ideal for PPI networks. Some of them operate only on unweighted graphs while others partition the set of input objects forcing each element to belong to no more than one cluster. However it is well known that proteins may belong to more than one complex, and therefore the corresponding nodes may belong to more than one cluster.

In this work we applied the Dominant Sets framework, a recent method whose purpose is to provide a general formulation of the clustering problem and an elegant way to address it. It addresses the clustering problem from a novel point of view, which is provided by game theory. The advantages of this game-theoretic perspective are many, thus making Dominant Sets a very general framework, which can be applied in a wide rage of scenarios, including protein complex prediction. It allows handling weighted and unweighted data and detecting overlapping clusters.

Our results show that the quality of protein complexes predicted by Dominant Sets algorithm is better than that obtained by other standard clustering techniques, sometimes even more accurate, than that achieved by ClusterOne -the state of the art for detecting protein complexes.

Moreover the Dominant Sets algorithm shows a high tendency to detect dimers, namely protein complexes of size two. To find such small complexes is a hard task and many authors agree that standard clustering algorithm for PPI underestimate their number. To the best of our knowledge, this is the first attempt at defining a method with the express aim of finding dimers, which raises the need to define an evaluation methodology to validate the putative dimers produced by Dominant Sets.

Acknowledgements

Acknowledgements often concern people who have allowed to achieve a specific target in some way. Unfortunatelly, to find the whole set of people who, directly or indirectly, helped you is not trivial. In most cases, these people do not even realise how important they have been.

First of all I would like to thank my supervisor Marcello Pelillo, who supported me giving me the possibility of working on an exciting topic in an equally exciting environment.

I would like to express my gratitude to my co-supervisor Alberto Paccanaro who allowed me to join his research team. His enthusiastic guidance and always sincere advice have been crucial during this experience. With his strong passion for research he transmitted to me the most genuine support and motivation, helping me find my way when I got lost.

Special thanks go also to the other members of the group, who in countless occasions were always available to spare some time and share their helpful suggestions. They worked together with me to solve numerous issues, helping me discover what research actually is and teaching me how to approach it with the best of attitudes. I would especially like to thank Horacio, for reviewing this thesis providing invaluable suggestions. Thanks to Alfonso for his also invaluable feedback and constructive criticism in discussing the problems I was confronted with.

A special mention goes to Michele Bugliesi, Salvatore Orlando, Riccardo Focardi and Alessandra Raffaetà for introducing me in the university with an extraordinary passion, and for all the great advice given during these years.

I would also like to acknowledge Simone, Max, Mario, Francesco, Zorzi, Lorenzo, Pretotto, Amadio and il Filosofo, for sharing with me during these years both the grey moments and most pleasant discussions. In particular, thanks to Emily and to her corals, for having handled better than me my paperwork; all those PhD applications have not been useless.

My gratitude and all my love go to Giulia, who shared this adventure in the UK with me. Her patience, trust and support have been crucial during the hard times of my research. Can we go back to Conche now?!

Finally, the most important thank you goes to my family. They give me their unconditional love and support during all my studies, teaching me the most important lessons of life.

Contents

List of Figures

List of Tables

Chapter 1

Protein Complexes

1.1 Introduction

In the past few decades a huge amount of research regarding proteins of all organism has been produced. However, proteins rarely act alone and, in most cases, in order to perform their task in a biological system they work in groups.

These particular groups of proteins are called protein complexes. Proteins complexes perform many crucial tasks within living beings, including catalyzing metabolic reactions, replicating DNA, and transporting molecules from one place to another. Being workhorses that assist a vast amount of biological processes, their detection is needed for expanding our knowledge about cellular organization and function. Their importance is reflected in the volume of recent research conducted in the field of protein associations.

Only recently, Protein-Protein Interaction datasets have been proposed due to the development of experimental procedures such as AP-MS and affinity purification. These experiments produce weighted graphs, where each node represents a protein, each link represent a possible interaction between proteins, and weights on edges are related to their interaction confidence values.

1.2 Proteins and Proteins Complexes

Nowadays, it is possible to list the genes and the relative proteins for an increasing number of organisms. The knowledge gathered in the past decades about cell biology, molecular biology, biochemistry, structural biology and biophysics allowed us to expand our comprehension of the function and molecular properties of individual proteins. This knowledge is maintained into vast protein databases like UniProt[6, 31].

However, proteins rarely act alone. In most of the cases they aggregate in "molecular machines", called protein complexes, that perform biological functions at both cellular and system levels.

Proteins complexes assist most of the biological processes in a cell, including gene expression, cell growth, proliferation, nutrient uptake, morphology, motility, intercellular communication and apoptosis[61]. Moreover, they mediate biochemical phenomena such as enzyme cooperativity and signal transduction. Therefore, in order to fully understand the dynamics of biological processes within an organism it is not sufficient to just list its proteins. Indeed, it is necessary to determine all the physical interactions between them.

Hence, it is not surprising that, even if until the late 1990's the research concerning the analysis of protein function was focused on single proteins, nowadays most efforts are focused towards understanding how proteins interact with each other. It is clear that, also considering a single protein, to fully comprehend its real function in an organism, it is necessary to study it in the context of its interacting partners.

The field studying proteins is often called proteomics, while with the term interactome we usually refer to the complete map of protein interactions that can occur in a living organism at any time. As pointed out before, interactome mapping is one of the main topics of current biological research, in the same way as the "genome" projects were 20 years ago[17].

However, there are remarkable differences between proteome and genome. Indeed, while the former is quite static, the latter is quite dynamic, changing during the development of an organism and in response to external stimuli[17]. Therefore,

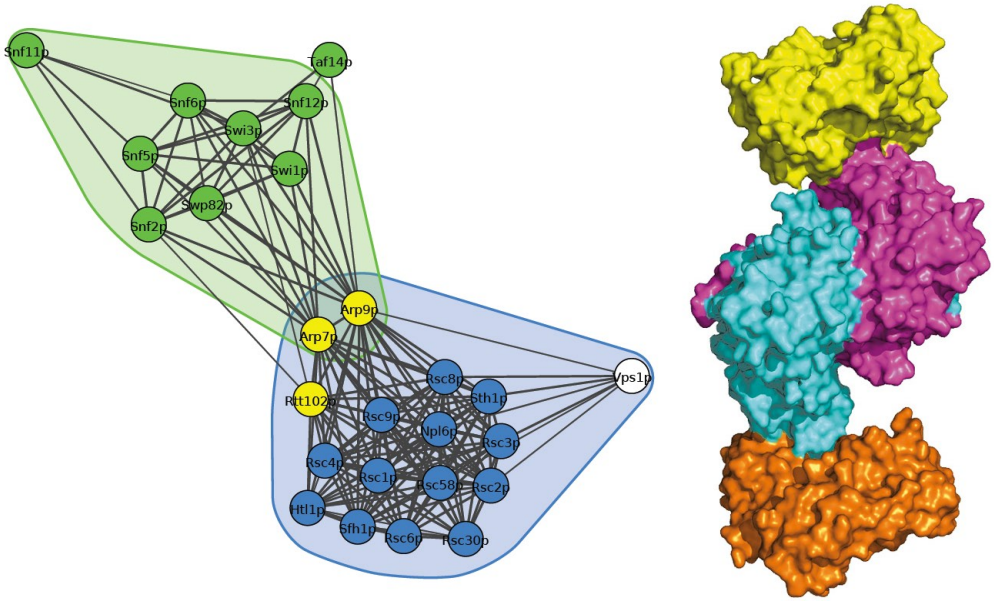

Figure 1.1: A PPI network and a protein complex[44]

Protein-Protein Interactions represent one of the most complex levels of structural organization in biological molecules.

1.3 Why to detect Protein Complexes from PPI networks

We previously highlighted that proteins rarely act alone and they frequently bind with each other in sophisticated groups called protein complexes, each of which performs one or more precise biological tasks.

The existence of a long list of biochemical phenomena performed by protein complexes in each living being makes clear why they are so important. Studying them in detail corresponds to understanding how they are built and how they work, allowing to expand our knowledge regarding biological systems.

Moreover, it is worth emphasizing that a better comprehension of their roles allows us to realize how a protein complex disorder can affect the biological processes in which it is involved, or vice versa. It is therefore not surprising that numerous recent research efforts have shown that proteins are strongly related to

diseases. Indeed, diseases are usually caused by an erroneous production of some protein complex.

For example, if a particular protein of a complex is not produced, or is produced in an incorrect amount, a fraction of all the complexes in which it is related could be affected negatively. Moreover, it is worth emphasizing that physical interaction between proteins depends on their physical structure. Hence, if for some reason a single protein of a complex has the wrong shape, it may not be able to bind with the other proteins in the complex. This issue can be further complicated by the fact that changes in shape can be position-specific. As a result, the proteins interacting with an altered protein can generate a protein complex that is unable to perform its original task in an appropriate manner. Minor corruptions of these sophisticated macromolecules may therefore lead to unpredictable results.

The great importance of protein complexes and their relation to (human) diseases has lead to the development of a new paradigm to deal with diseases called network medicine. Although correlations between protein complexes and complex diseases have been suggested in the past, only recently a sufficient amount of protein complexes have been identified.

This highlight the essentiality of studying and detecting protein complexes.

1.4 Yeast PPI networks

Protein-Protein Interaction networks represent networks where each node is a protein and there is an edge between a pair of proteins if they interact. Often, weights representing interaction confidence values between pairs of proteins are also provided.

Since each living organism has a different proteome, namely a different set of proteins, it is possible to outline a different PPI network for each of them. Typically, since more than one study can be performed on a single organism, for some living beings more than one PPI network is available.

However, the literature presents in depth studies for the proteomes of only very few organisms. The yeast *Saccharomyces cerevisiae* is probably the best studied of them. Nevertheless, even if all its 6,000 proteins are known, the total number of Protein-Protein Interactions is only estimated. In fact, of all the 28,000 inter-

actions estimated only a small part are contained in manually curated datasets. In this manuscript, as has been the case for most previous work, we will operate using datasets and gold standards of yeasts. [51].

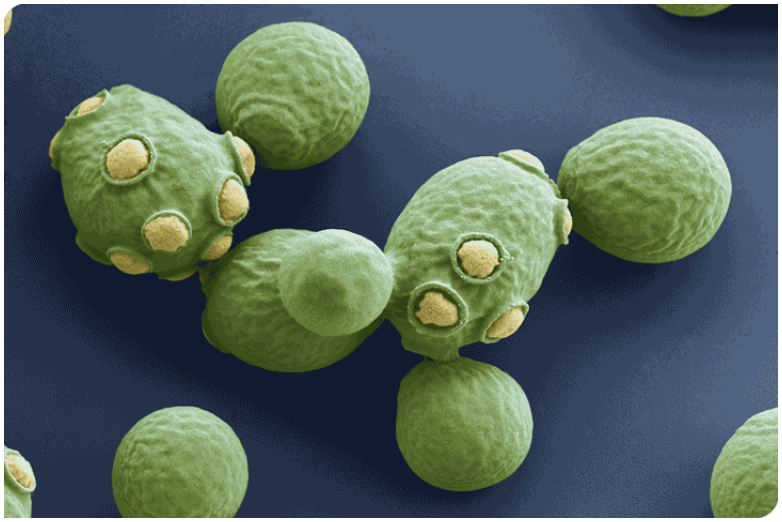

Figure 1.2: Yeasts are one of the most widely used organisms for genetic studies as cancer research. In the figure is reported a photo of the yeast Saccharomyces cerevisiae. Other species, such as Candida, are opportunistic pathogens and cause infections[46]

There are several reasons for using dataset of yeasts proteins rather than human ones to assess the quality of clustering algorithm for PPI. Firstly, the set of Protein-Protein Interactions in yeasts is more simple.

For example, while more than 95% of human genes are interrupted by an intron, only less than 5% of all yeast genes contain an intron[51]. We briefly recall that genes encode proteins and that introns and exons are portion of gene sequence. The various portions of a gene that encode for a protein are called exons, and they are separated by sequences called introns.

This gene structure allows to build different proteins merging different exons of a gene. This process is known as alternative splicing as opposed to the scenario where there are no introns and only one splice can be performed. When a gene has only one splice, a single gene codifies for a single particular protein. When, instead, a gene is subdivided into multiple coding sections, these can be merged in various ways. Consequently, single genes may encode for multiple proteins.

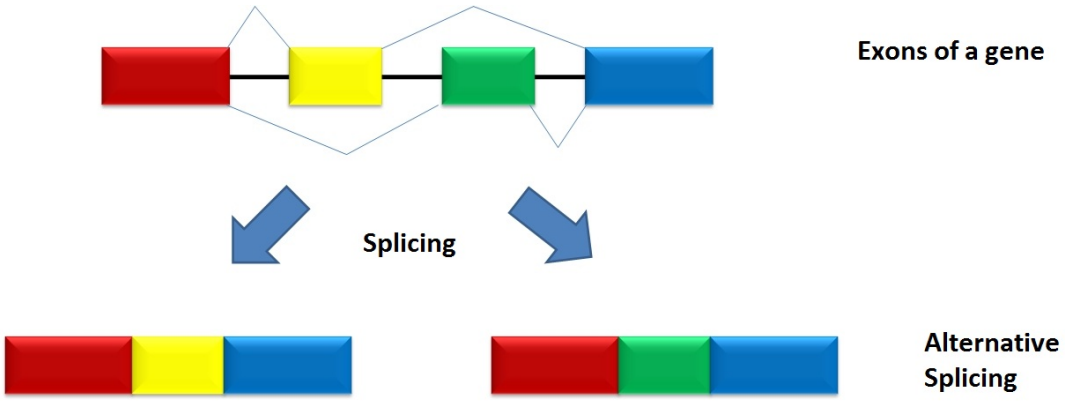

Figure 1.3: An example of alternative splicing[66]. In this figure the coloured pieces represent exons while the white gaps represent introns. As shown, different combination of the exons leads to alternative splices, and hence to different proteins

The presence of only a limited amount of introns within yeast genes with respect to those present in human genes, implies that alternative splicing is rarely seen to occur in this type of eukaryote. Therefore, in most cases, each yeast gene encodes for just one protein. This is one of the main reasons that often lead researchers to use yeast datasets to validate the results of a clustering algorithm.

1.5 Protein Protein Interactions

During the last few years we saw a huge development of high-throughput experimental technologies. This is one of the main reasons that lead to a considerable increase in the number of detectable Protein-Protein Interactions. At the same time remarkable progresses on large-scale technologies have taken place, and small-scale experimental datasets have been published. Moreover, public data repositories have been made available, with the aim of integrating information from both large and small scale experiments published in the literature.

Commonly with the term Protein-Protein Interaction we refer to physical interactions between proteins that occur in a cell of a living organism. However, in order to be included as a PPI, the physical contact between proteins has to be significant, otherwise a PPI network would include the set of all proteins that attach

each other by chance. It is also necessary to exclude all interactions involved in common scenarios during its life cycle, as when it is being made, folded, quality checked or degraded. For example, all proteins at one point "touch" the ribosome, many "touch" chaperones and most have at one point contact with degradation machinery[17].

As a result, the basic definition of protein interactions can be viewed as the physical interaction between proteins, while the definition used to build PPI networks has to consider a subset of more restricted cases. Usually, the interaction interface is considered in order to establish the nature of the interaction. Indeed, the interaction interface should be intentional and not accidental, namely the result of specific biomolecular events or forces. Moreover the interaction interface should be non-generic, namely the bind sites should be evolved for specific purposes, distinct from generic functions such as protein production or degradation.

Another key aspect for defining Protein-Protein Interactions is their biological context. Indeed, not all the possible interactions occur in a particular cell at any time. As previously mentioned, the proteome is dynamic and, also considering the subset of interactions discussed above, a physical bind between proteins can be static or permanent. It is well known that the cell machinery faces continuous turnover and reassembly, during which some proteins are folded and discarded. Hence, while some protein complexes having strong chemical bonds are stable and static, others are built only to perform transient actions. For example, some proteins are built only with the aim of activating the gene expression process[17].

Thus, protein interactions do not depend only on their interaction interface but also on cell type, cell cycle phase and state, developmental stage, environmental conditions, protein modifications, and presence of other binding partners.

1.6 Types of Protein-Protein Interactions

As pointed out before proteins attach each other physically depending on their shape. In order to fold together and build a macromolecule their shape has to be chemically complementary. Indeed, proteins bind to one another at specific sites through a combination of chemical bonds as hydrophobic bonding, van der Waals forces, and hydrogen bonds.

Moreover, the interaction interface can consist of small binding clefts of few peptides, or large surfaces made up by hundreds of amino acids. Clearly, the strength of the interactions depends on the type of chemical bonds involved, as well as on the size of the binding sites.

As a result protein interactions can be divided roughly into stable and transient interactions. Sets of proteins bound together in a protein complex by stable interactions are static and they perform their activities without being built depending on the scenario. Therefore they are always present in the cell.

On the other hand, protein complexes bound by transient interactions are built and discarded every time the cell needs to perform some particular task. Therefore, transient interactions are temporary and frequently require a specific set of conditions or cellular changes to occur.

Due to their flexibility, protein complexes using transient interactions control the majority of the cellular processes. Protein modification, transport, and cell cycling, are only few of cellular processes performed by protein complexes built by transient interaction between proteins.

1.7 Analysing and determining PPI: Binary and Co-Complex Methods

A combination of various methods and techniques is usually necessary to detect and validate protein interactions. In this section we present a brief review of some of the most relevant methods currently applied.

As we pointed out before, the recent development of the field of proteomics was also due to the large amount of innovative methods for detecting Protein-Protein Interactions developed. Several methods and techniques to discover and validate protein interactions are available, each of which has strengths and weaknesses.

We can classify these approaches in more than one way. For example, they can be classified based on the scale of the experimentation. Indeed, we can determine proteins interactions by using large or small scale experiments.

Another common way to classify them is by looking at the type of PPI data

produced. Using this criteria two main technologies have been proposed: binary methods and co-complex methods.

A technique is called binary if it measures the direct physical interaction between pairs of proteins. On the other hand a technique is a co-complex method if it measures the physical interactions among groups of proteins, without providing a measure of interaction between each pair of proteins. Hence, it is worth emphasizing that the latter considers groups of interacting proteins without specifying if the interaction of each protein with another is direct or indirect[17].

Intuitively, the results produced by a co-complex method are conceptually different from those detected by a binary method. It turns out that a binary interpretation cannot be assigned directly to data derived by a co-complex technique. Usually, an algorithm or a complex model has to be applied in order to translate data measuring the interaction among groups of protein into pairwise interactions[26].

Marc Vidal, a geneticist at the Dana-Farber Cancer Institute in Boston, used the simple example of a football match in order to point out the discrepancy of these two approaches. Lets image a set of referees watching a football match.

> "The pull-down mass-spectrometry approach will show you the players, referees and field, but not who is passing to whom and in what direction the ball is travelling," he says. "This is where a binary approach comes in."[8].

In the figure 1.4 is shown the difference between binary and co-complex method for Protein-Protein Interaction detection. In the left side of the picture are reported the true interactions occurring between 6 protein in a cell. In the right side of the figure are represented two network, one on top derived by a binary method, and the other returned by a co-complex method.

By finding pairwise interactions the binary method produces a network where each link really occurs. While, the group detected in the right bottom panel by a co-complex method is composed by a group of proteins that interact. However it does reflect if two proteins of the group interact directly or indirectly. Giving a binary interpretation of this group of interacting proteins we mark with a X the deduced links that do not occur.

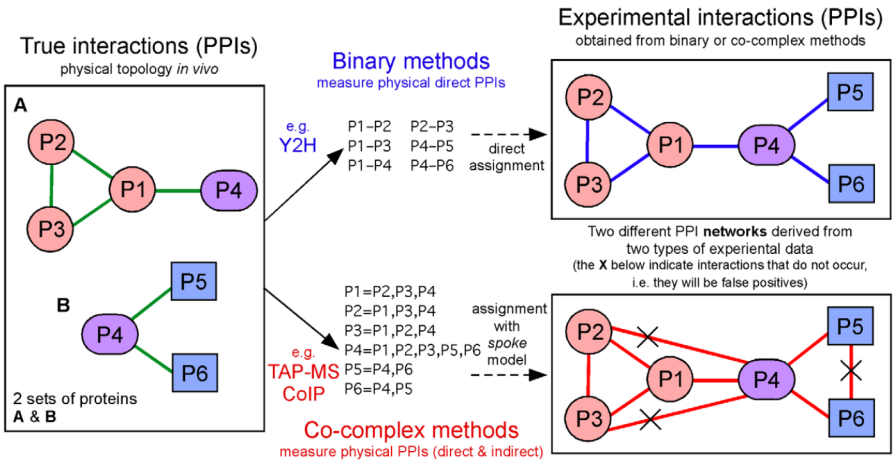

Figure 1.4: Binary and Co-Complex methods to analyse Protein-Protein Interaction[17]

The most widely used binary methodology for finding Protein-Protein Interactions is the *Y*east-*T*wo-*H*ybrid method, also called Y2H. While the most used co-complex technique is *T*andem *A*ffinity *P*urification coupled to *M*ass *S*pectrometry (TAP-MS). These two techniques are applied in large scale investigations. Another common co-complex approach, based on protein antibody recognition, is co-immunoprecipitation (Co-IP).

1.7.1 Binary Methods

Yeast Two-Hybrid (Y2H) Method

One of the best known binary approaches for finding protein interactions is the *Y*east-*T*wo-*H*ybrid method. This molecular biology technique is widely used with the aim of testing physical interactions between proteins or a single protein and a DNA molecule. Its name is due to the fact that a genetically modified yeast strain is used.

In order to verify an interaction between two proteins, A and B, the method proceeds as follows. A functional transcription factor is divided in two pieces, referred as DNA-binding domain (BD) and activation domain (AD).

In the next step the BD is fused to protein A, often referred as bait protein. It is worth emphasizing that the bait protein is usually a known protein for which we

are seeking new interaction partners. Protein B, the prey protein, can be a single protein or a library of proteins for which we are testing the interactions with the protein A. This B protein is fused to the AD.

As pointed out before this technique is applied inside the nucleus of yeast, thus both proteins fused with the relative piece of the transcription factor are introduced in the nucleus. If those two proteins will interact they will bind each other. Given that AD and BD are fused to proteins A and B, respectively, the two pieces of the transcription factor will be indirectly connected by the binding between bait and pray.

The key point of this scheme is that, in most eukaryotic organisms, transcription factors can function in proximity of each other without direct binding. This means that even if the transcription factor was split into two fragments, it can still activate the transcription process when its two fragments are indirectly connected.

Therefore, if the two proteins interact, the two pieces of the transcription factor will be close together, leading the expression of a particular reporter gene included in the yeast stand. This process can be detected because the transcription of the reporter gene will result in a specific phenotype. In this way, a successful interaction between the fused protein is linked to a change in the cell phenotype[71].

A high-throughput variant of this method is also available to study protein interactions. Unfortunately, the application of this technique in large-scale experiments is very expensive in terms of time and cost.

Protein-Fragment Complementation Assay - PCA

Another approach for identifying Protein-Protein Interactions, similar to the Y2H method, is the *P*rotein-fragment *C*omplementation *A*ssay, also known as PCA.

In this method each one of the proteins of interest, called again bait and prey, is covalently linked to an incomplete fragment of a third protein called reporter. The interaction between the bait and the pray proteins brings the fragments of the reporter protein close together, allowing them to form a functional reporter protein.

The activity of this functional reporter can be measured. Moreover, differently from the Y2H method, it provides a direct read-out that is not dependent on

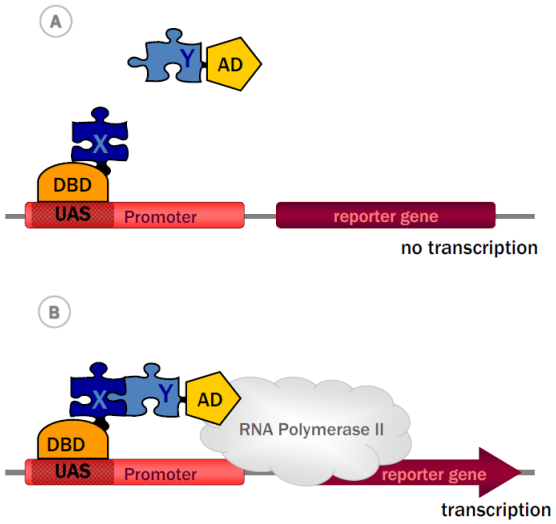

Figure 1.5: Yeast Two Hybrid System(Y2H)[10]. **A.** The pray protein Y is attached to the transcription factor known as Activation Domain (AD), while the bait protein X is attached to the DNA-binding Domain (DBD; referred to as BD in text). **B.** If the two proteins bind with each other the RNA polymerase II can bind the transcription factors and transcribes the reporter gene.

the transcription of another gene. Many different reporters can be applied. One example of function reporter protein used is the Yeast Gal4, and it is also used in classical yeast two-hybrid system. Therefore, the Y2H method is, in practise, an archetypical PCA assay[70].

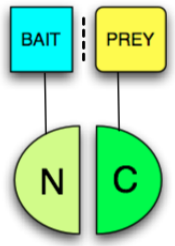

Figure 1.6: Protein Complementation Assay (PCA)[70]

1.7.2 Co-complex approaches

Tandem Affinity Purification - (T)AP-MS

*T*andem *A*ffinity *P*urification, also called TAP method, allows high throughput identification of protein interactions. In contrast to the Y2H approach its accuracy can be compared to that of small-scale experiments. Another important difference with the Y2H technique is that the interactions are detected within the correct cellular environment[17].

However, following the TAP method it is necessary to apply two steps of protein purification. These two steps avoid the detection of transient Protein-Protein Interactions[69].

It is worth emphasizing that TAP experiments were performed with the aim of building the Krogan et al. (2006) [34] and Gavin et al. (2006) [22] datasets. Such datasets, which we also used in this work, used the TAP method to a genome-wide scenario with the purpose of providing updated protein interaction data for yeast organism.

In the TAP method a protein, called bait protein and tagged with a particular molecular marker, is used to catch or "fish out" a group of proteins called pray proteins. In the following step a biochemical technique, known as "pull-down", is used to separate them from the mix. What takes place is a co-purification of protein groups.

It is worth emphasizing that the pull-down assay is used for two purposes. Indeed, it can be used to confirm the existence of a Protein-Protein Interaction predicted by other research techniques, or it can be applied as an initial screening assay for identifying previously unknown Protein-Protein Interactions.

Co-immunoprecipitation - Co-IP

Once having identified that two proteins interact with each others, using one of the methods presented in the previous sections, it is often necessary to verify their binding by applying additional methods. Indeed, all the methods presented so far count a certain amount of false positives and false negatives during the detection of Protein-Protein Interactions.

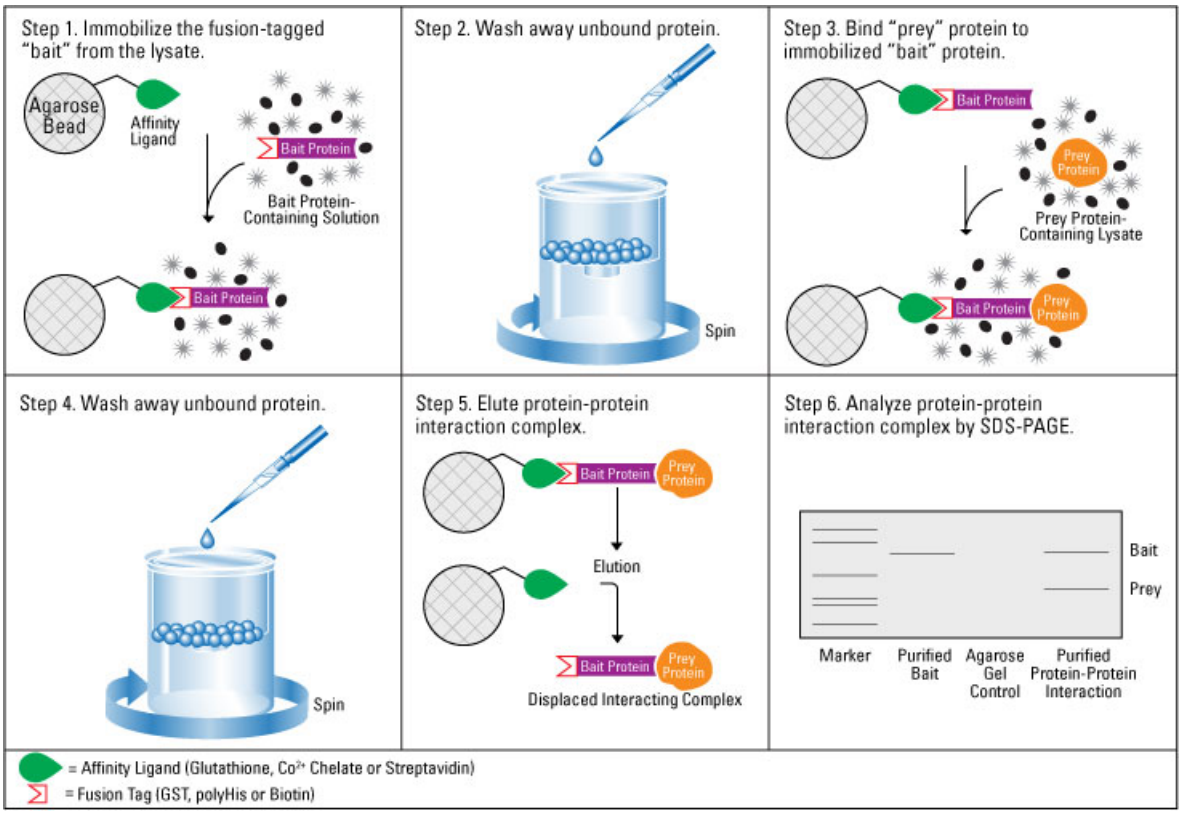

Figure 1.7: Pull-Down technique[61]

Co-immunoprecipitation (Co-IP) is one of the most widely used methods for verifying PPIs. Therefore, it is worth emphasizing that it is not a screening approach, and it only checks interactions between suspected interaction partners.

Moreover, immunoprecipitation experiments reveal direct and indirect interactions. Therefore, given two proteins, a positive result does not specify if those proteins interact directly or indirectly[7].

This assay is very similar to that of TAP, with the difference that in CO-IP an antibody is used instead of a bait protein.

With the aim of understanding better the principles of co-immunoprecipitation (Co-IP), we introduce the immunoprecipitation (IP) method.

Its principle is very straightforward and it is shown in figure 1.8. A particular antibody is used together with the first input protein, called the target protein. This antibody and the first protein are used joined together in a sample, such as

a cell. The cell acts like a proteins mixture, and if the first protein interacts with some other proteins in the mixture these will attach to one another. The resulting complex is called immune complex and it is also attached to the antibody.

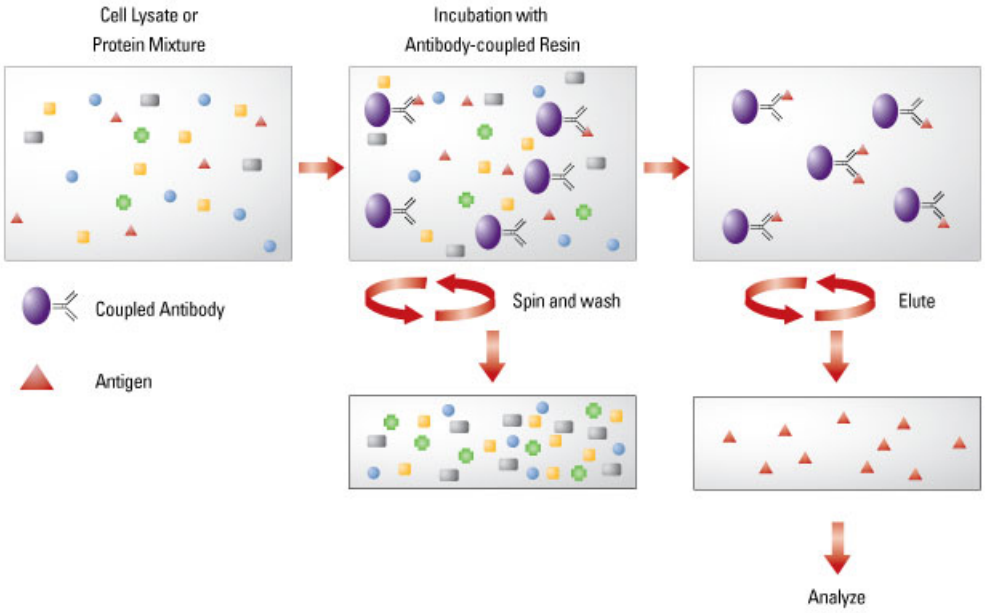

Figure 1.8: Immunoprecipitation Technique [61]

The immune complex is then captured, or precipitated, on a beaded support. The antibody-binding protein is immobilized on that support, and any proteins not precipitated on the beads are washed away. After washing of the beads, the antibody, the bait protein and all the proteins associated to that bait are eluted by boiling. The bound proteins can then be identified by Mass Spectrometry (MS).

Co-immunoprecipitation is an extension of IP that differs from it because, while the latter is focused on an antigen, the former is focused on the interacting proteins [61].

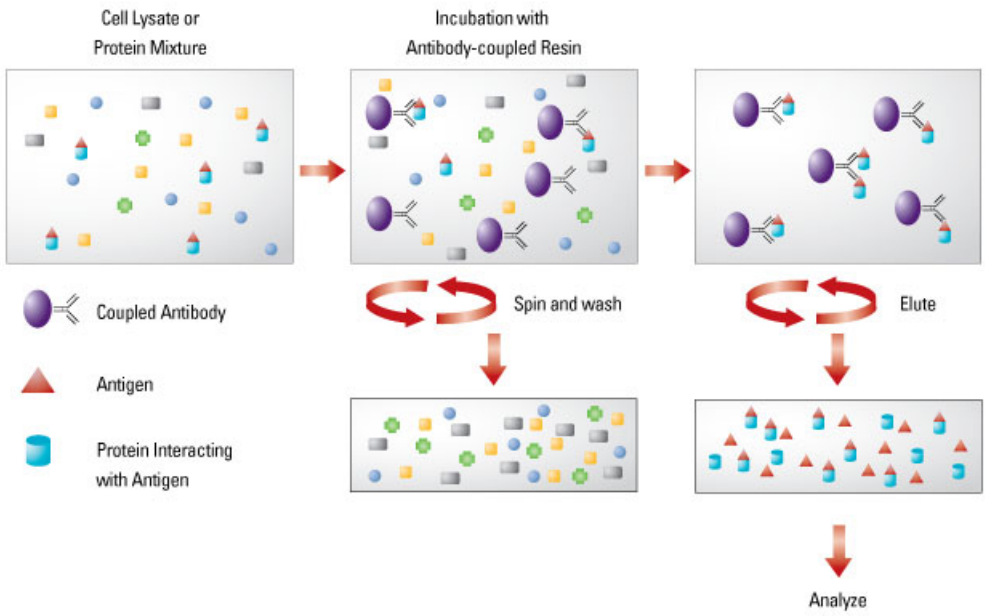

Figure 1.9: Co-Immunoprecipitation Technique(Co-IP) [61]

1.8 Gene Ontology

The *Gene Ontology* is one of the most relevant projects in bioinformatics. It started in 1998 with the aim of providing a consistent description of gene products in different databases. Indeed, one of the main issues in biology is that the nomenclature for genes and their products is often divergent, even when the experts appreciate their similarities. Such issue is further complicated by the fact that more than one name or identifier is frequently associated to each gene.

A simple example made by the authors of Gene Ontology involves a group of biologists searching for all the proteins involved in bacterial protein synthesis. Some databases could describe those proteins as molecules concerning "translation", while others could use a different, but functionally equivalent, terminology as "protein synthesis"[16].

In order to provide interoperability among genomic databases the *Go consortium* developed the Gene Ontology (GO). The goal of this consortium is to produce three structured species-independent vocabularies, called ontologies, describing the

roles of genes and gene products in any organism. GO describes genes and proteins in terms of the biological process in which they are involved, the cellular components in which they act and their molecular functions.

Therefore, genes and proteins are described in three vocabularies using the three following categories, also known as *Categories of GO* or *GO Domains*:

- *Cellular component*;
- *Molecular function*;
- *Biological process*.

The molecular function of a gene product is its set of biochemical activities at the molecular level. It describes the function performed by a protein without any spatial or temporal specifications. Examples of broad functional terms of this ontology are "enzyme" and "transporter".

Cellular component refers the part of a cell, or its extracellular environment, where a gene product performs its activities.

A biological process represents a list of biological events. It may sometimes be difficult distinguish biological process from molecular function, but in general the former has more than one step.

It is worth emphasizing that the Gene Ontology project does not have the aim of unifying biological databases. Indeed, GO is not a database of gene sequences, but rather it provides an abstract nomenclature in order to describe how proteins act in a cellular context.

1.8.1 Ontologies and GO Terms

The structure of an ontology resembles the current representation of our biological knowledge. Each one of the three ontologies presented so far is composed by a set of *GO Terms*. Each GO Term can be seen as an attribute of some particular gene or gene product. For example, the GO Term "GO:0006412" is a term of the ontology *Biological Process* and refers the process known as "translation". Each GO Term has several related fields as ID, name, description, synonymous, and relationships with other GO Terms.

Hence, the ontologies describe terms and relationships between terms. GO Terms are connected as nodes of a directed acyclic graph, thus all the connections between a term and its parents and children are known. The relationships used are directed because, for example, a mitochondria is an organelle but not all organelles are a mitochondria.

Each ontology has a hierarchical structure in which child terms are more specialized than parent terms. An example of relationship between GO Term is presented in figure 1.10.

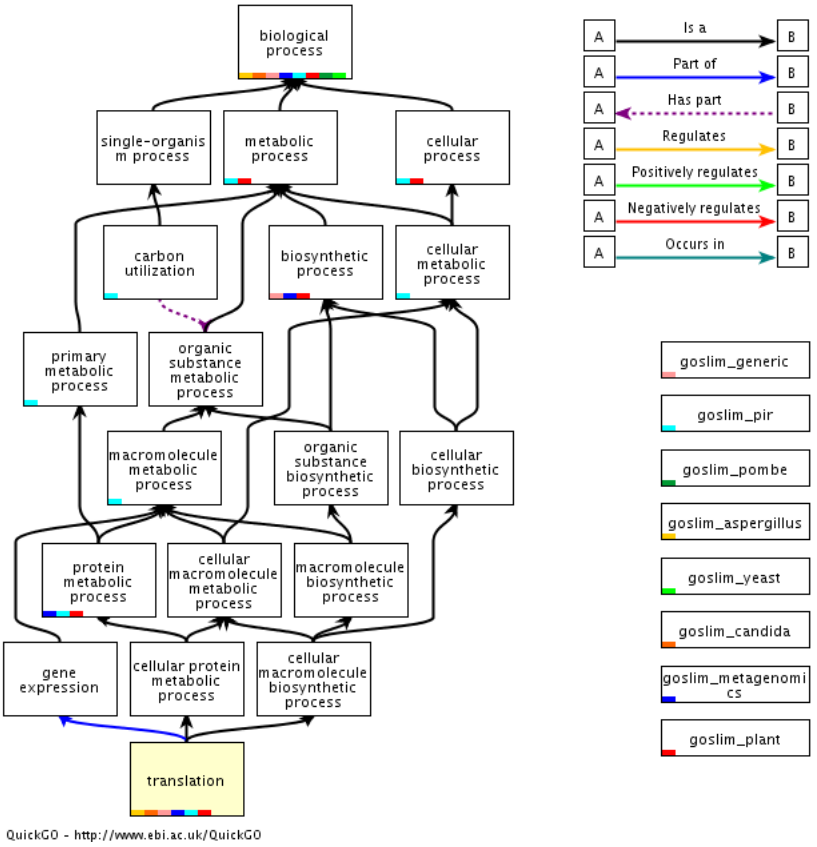

Figure 1.10: Chart of the GO Term GO:0006412[16]

Using the various terms form each Gene Ontology, a particular gene or gene product can be annotated to several levels depending on the knowledge available. In this way GO provides high flexibility, allowing to narrow and widen the focus of the user's query.

Moreover, for each gene and gene product annotated with a GO term the *true*

path rule holds. The *true path rule* states that "the pathway from a child term all the way up to its top-level parent(s) must always be true"[16]. Hence, a protein annotated with a specific GO Term would also be annotated with all its ancestors.

For example, a protein annotated as "oxygen binding' is also annotated as "binding" because the latter GO Term is a parent of the former.

1.9 Gold Standard for protein complexes

With the aim of assessing the performance of a clustering algorithm applied to PPI networks it is necessary to compare its predicted protein complexes with a set of known interactions. These kind of sets are called gold standards. Some of them are manually curated and widely applied in the literature, while others contain both manually curated and computationally predicted complexes.

During the past years the *M*unich *I*nformation *C*enter for *P*rotein *S*equences (MIPS) *S. cerevisiae* Protein-Protein Interaction dataset has been applied in several analysis as gold standard reference[73] because of its quality and comprehensiveness. Therefore, we consider the MIPS interactions as a gold standard. Moreover, following the guidelines of [44], we used a version the information of *S*accharomyces *G*enome *D*atabase (SGD) to provide an additional gold standard.

We compared the predicted complexes detected by the various clustering algorithms with these two reference complexes. A brief review reporting the versions used is reported in the table 1.1.

Source Gold Standard	**Version**	**Notes**
MIPS [41]	18 May 2006	We kept all MIPS categories containing at least two proteins as protein complexes
SGD [29]	11 Aug 2010	Gene Ontology (GO)-based protein complex annotations from SGD

Table 1.1: Gold standards used for protein complexes

1.9.1 MIPS

The MIPS dataset is a catalog organized hierarchically. Each category can contain subcategories extending at most five hierarchy levels. An example of protein complex that extends five hierarchy levels report the identifier 510.190.10.20.10 and it is a multifunctional coactivator that regulates transcription by RNA polymerase II.

It is worth emphasizing that not all the MIPS categories correspond to complexes; in certain cases they may be a set of related complexes. For example, the MIPS identifier 510.180 does not represent a complex, but rather it corresponds to all "DNA-repair complexes".

As suggested in [44], in order to avoid selection bias, we consider all the MIPS categories containing at least three and at most 100 proteins as protein complexes. Moreover, we excluded the MIPS category 550 and all its descendants, because this category correspond to unconfirmed protein complexes that were predicted by computational methods.

1.9.2 SGD

The *S*accharomyces *G*enome *D*atabase (SGD) provides biological information for the yeast Saccharomyces cerevisiae. It reports GO annotations for all the yeast proteins. Providing these annotations for each protein it is possible to know its function and the cellular component where the protein performs its activities. Following the approach used in [38] and [44], we used these annotations to create a dataset usable as gold standard.

The basic idea is mapping each protein into a protein complex using GO annotation. Three tools are necessary for this task:

- the mapping of yeast genes and proteins to GO terms[29];
- the GO structure [3];
- an inference engine to find proteins using a set of GO inference rules;

The engine used is available on [68], and taking as input the standard GO inference rules it allows to find all the terms that are descendants of a particular GO

term. Hence, running it using the relation "is_a" with the GO term corresponding to "protein complex" is it possible to find all the annotations related to protein complexes. Therefore it is possible to extract from SGD all the yeast proteins with those GO terms. However, it is necessary to remove all the proteins that are supported only from evidence code inferred from electronic annotation (IEA). Finally, these proteins are grouped in protein complexes using their GO annotations. Annotations with modifiers "not" or "colocalizes_with" have been ignored [44].

1.10 Protein-Protein Interaction Datasets

As Protein-Protein Interaction datasets for yeast we use five dataset widely used in the literature. Four of these dataset are weighted, reporting for each pair of proteins a value between zero and one. In table 1.2 are reported all the datasets used and relative details.

Details	**Collins**	**Krogan Core**	**Krogan Extended**	**Gavin**	**BioGRID**
Reference	[15]	[34]	[34]	[22]	[58]
Number of proteins	1622	2708	3672	1855	5640
Number of interactions	9074	7123	14317	7669	59748
Weighted	yes	yes	yes	yes	no

Table 1.2: Properties of the protein-protein interaction datasets used

Chapter 2

Dominant Sets

2.1 Introduction

In this chapter we shall see the details of Dominant Sets framework, a recent method whose purpose is to provide a general formulation of the clustering problem and an elegant way to address it.

The formal definition of cluster proposed is called "dominant-set", and it generalises the classical graph-theoretic notion of a maximal clique to edge-weighted graphs. It addresses the clustering problem from a novel point of view, which is provided by game theory. The advantages of this game-theoretic perspective are many, making Dominant Sets a very general framework, which can be applied in a wide range of scenarios. This game theoretic set-up allows the handling weighted and unweighted data. Furthermore, by considering overlapping clusters, it permits an object to belong to more than one cluster.

These features allow detecting overlapping protein complexes from any type of Protein-Protein Interaction making Dominant Set a candidate for this purpose.

2.2 Overview and Motivation

As the other clustering algorithms, Dominant Sets tries to extract coherent groups from the set of objects given as input. However, even if there is no shortage of clustering algorithms in the literature, they have often been developed trying to

address a specific instance of the problem. For the sake to exploit the powerful ideas and the elegant mathematical and algorithmic treatments from such sophisticated fields as linear algebra, graph theory, optimization and statistics they often overlook the real essence of the clustering issue.

One of the limits of this approach is that it usually leads to either very specific algorithms or to very general algorithms, which unfortunately produce groups that are considerable clusters only for some particular applications. Other limitations, frequently unbearable for general purposes, are the fact that the number of clusters must be known in advance, or that the clustering process consists in practice in a partitioning operation.

It is clear that there are many scenarios for which to force all the input elements to belong to some group has little sense, leading in some cases to the creation of extra classes which represent noise or inconsistent clusters. Moreover, partitioning the set of objects implies that each element cannot belong to more than one cluster.

These intrinsic limitations of the are too restrictive in many applications, such as clustering Protein-Protein Interactions network. It is in fact well-known that proteins rarely work alone, but rather they attach each other dynamically during the time to build protein complexes. Therefore, acting like gear of complex machineries, a protein can participate to more then one complex and hence it can belong to more than one cluster.

To solve these limitations the Dominant Sets framework has been developed starting from the essence of the clustering problem: the definition of cluster. It suggests a new way to characterize it starting from the idea that a cluster should satisfy two fundamental conditions: it should have high internal homogeneity, and there should be high inhomogeneity between the entities in the cluster and those outside of it. The formal formulation of cluster proposed is called "dominant-set", and its generalises the classical graph-theoretic notion of a maximal clique to edge-weighted graphs.

However, the framework provides much more than a new characterization of the concept of cluster. It establishes a correspondence between Dominant Sets and the extrema of a quadratic form over the standard simplex. Computationally, this allows to find clusters using straightforward continuous optimization techniques

such as payoff-monotonic game dynamics, a class of dynamical systems arising in evolutionary game theory.

The useful features of this game-theoretic perspective are various. It makes no assumptions on the underlying data representation, in fact it does not require that the objects to be clustered be represented as points in a vector space. It makes no assumptions on the structure of the affinity matrix, being it able to work with asymmetric and even negative similarity functions alike. It does not require *a priori* knowledge on the number of clusters. It leaves clutter elements unassigned and it allows extraction of overlapping clusters[50].

All these features make the Dominant Sets a very general framework applicable in a wide rage of scenarios, and a suitable candidate to detect protein complexes from Protein-Protein Interaction networks.

2.3 The Clustering problem

Cluster Analysis is an unsupervised learning process aimed to group a set of objects, according to their characteristics or to their relationships, in order to have similar objects grouped together and dissimilar ones into different groups. This activity is often also viewed as a particular case of the classification problem, called unsupervised classification. What is roughly referred to as classification is actually supervised classification, as a set of labels is also provided in addition to the set of objects to classify. Clearly in cluster analysis the labels, which define the belonging of a set of objects to a specific class, are initially unknown. Therefore, clustering is an unsupervised classification, given that the grouping of objects corresponds to assigning an initially unknown label to each object [60].

However, independently from the point of view, the clustering problem is not easy to address. In fact, sometimes it is impossible to proceed to this asset without any sort of bias. This concept is developed in an informal theorem called *The ugly duckling theorem*[64]; it gets its name from a famous story of Hans Christian Andersen because it shows that an ugly duckling is just as similar to a swan as two swans are to each other. Figure 2.1 illustrates this point.

Looking at the object traits it is possible to group the elements reported in the figure in more than one way. It is possible to group them according to their

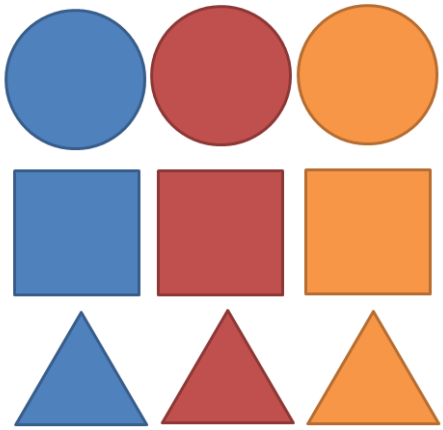

Figure 2.1: Example of clustering ambiguity

colour, or according to their shape obtaining completely different clusters. It is even possible to cluster them using both the characteristics ending up with six clusters overlapped, three by colour and three by shape. It is thus necessary to provide further information about the nature of the clusters. This is one of the main reasons for the existence of so many clustering algorithms in the literature.

By the same token algorithms employed for different problems show therefore a large divergence amongst themselves. Indeed, starting from a specific instance of a problem, particular techniques designed to find a convenient way to partition the input elements have often been developed. Unfortunately, this leads not only to a high fragmentation in the clustering algorithms scenario, which now counts very specific algorithms for every specific task, but also to miss the essence of the clustering problem, namely the definition of cluster.

The Dominant Sets framework introduced by Pavan and Pelillo [48, 49] aims to reverse the terms of the problem. Instead of determining a suitable way to partition the input data, it proposes a rigorous formulation of the notion of cluster and an elegant way to apply it.

Before the Dominant Sets framework is presented at length we shall see a common way to classify clustering algorithms, their data representation and their limits.

2.4 Central and Pairwise Clustering

As pointed out before there are many clustering algorithms present in the literature, and many ways to classify them. One of the most common classification divides them into two main types: central clustering, or feature based clustering, and pairwise clustering.

2.4.1 Central Clustering

In central clustering algorithms, also called feature-based algorithms, the elements are viewed as objects with a list of features each with its own value. According to this, each object is described in terms of vector of numerical attributes. In this way it is possible to map each object to a point in Euclidean (geometric) vector space.

It turns out that the distance between the points reflect the observed similarities or dissimilarity between the respective objects. One of the advantages of this representation is the existence of powerful analytical and computational tools not available in others. Indeed, classical pattern recognition methods tightly related to geometrical concepts have been developed during the past few decades, such as linear discriminant analysis, perceptrons, and kernel machines.

However, the geometric approach suffers from intrinsic limitations. First of all it is necessary to conduct a features selection or a features extraction step in order to characterize the objects. A non-trivial step is also often required in order to reduce the number of features, because vectors with high dimensionality reduce the algorithm's performance in terms of time.

Nonetheless, the characterization by features is not always possible. In fact, the biggest issue of this representation is the presence of a huge quantity of application domains where it is not possible to find satisfactory features or they are inefficient for learning purposes. This situation arises quite commonly when objects are described in terms of structural properties, such as parts and relations between parts. This lack led scientists to develop methods based on other structural representations such as trees or graphs.

2.4.2 Pairwise Clustering

The difficulty to express data that are explicitly related to one another, such as the elements of a graph, leads us to the branch of algorithms which deal directly with similarity between objects: the proximity-based algorithms.

In the pairwise clustering, the algorithm, instead of taking in input a set of feature vectors, takes the pairwise similarities among the objects. Often these similarities are represented as a matrix, where the element (i, j) expresses the similarity between object i and object j.

Clearly this approach does not change only the prospective of the problem, but it allows to overcome the limits cited in the previous section, allowing to cluster objects which could be difficult to represent by features. Moreover this setting is very general. In fact, using a convenient similarity measure, any feature-based clustering problem can be turned into a similarity-based one [12].

Furthermore, the similarity matrix representing the input data can be seen as the adjacency matrix of a graph. Indeed, mapping input objects to nodes of a weighted graph, where weights represents the similarity relations between them, is quite natural. Following this approach, a proximity-based algorithm is able to exploit concepts and results from graph theory to extract the clusters from the input data.

The largest part of these algorithms search combinatorial structures in the similarity graph, such as a minimum spanning tree [74] or a minimum cut [23, 56, 72]. Some other authors [4, 54] argue that the maximal clique is the strictest definition of a cluster, and several algorithms searches complete subgraphs, namely a clique, have been proposed.

2.5 The formulation of the notion of cluster

In this section we shall see the reasons that led to develop the Dominant Sets framework and hence how it tackles the limits of the other clustering algorithms. In fact, as we shall see, many of them have been designed in a way too devote to exploit the powerful mathematical treatments from graph theory, often forgetting what is a cluster from a more general point of view. Several intrinsic limitations

frequently arise due to the attempt to fit well known techniques without a rigorous characterization of the problem. Being the essence of the clustering problem related to the definition of cluster, Pavan and Pelillo proposed a new way of characterizing it.

Hence, in this section we present the basic idea of Dominant Sets and their rigorous formulation. Its authors showed a strong connection between the optimization problem usable to find them and evolutionary game theory, exhibiting the possibility to exchange them and take the most from both.

2.5.1 Why a new formulation

The clustering Dominant Sets algorithm is a graph-based pairwise clustering technique. As pointed out in the previous section, the pairwise clustering graph-based algorithms present in the literature are roughly divided into those that try to partition the graph with some technique and those that seek for cliques.

Graph-based clustering algorithms based on the former approach look for a partition of the input graph in order to exploit the powerful ideas and the elegant mathematical and algorithmic treatments from graph theory, such as minimum cut or the minimum spanning tree. Looking for partitioning the input data into coherent classes, these methods deal with a very specific version of the clustering problem.

Moreover, the partitioning approach used by graph-based clustering algorithms often leads to very well known limitations: the number of clusters must be known in advance, all the input data have to be assigned to some class and each element cannot belong to more than one cluster. Unfortunately, although probabilistic model-based approaches do not suffer from some of these problems, this oversimplified formulation of the clustering problem does not lead only to several restrictions but also to miss that the real problem is more general.

In contrast with this approaches others methods search in the input data complete subgraphs, namely a clique. They can be considered more suitable for scenarios in which overlapped clusters are allowed.

Unfortunately, it is worth emphasizing that while the minimum cut and the

minimum spanning tree are notions that are explicitly defined on edge-weighted graphs, the concept of a maximal clique is defined on unweighted graphs.

To take the advantages of the existing maximal clique algorithms some authors [30, 4, 25] proposed to work on unweighted graph obtained by a threshold operation performed on the original weighted graph. Once the threshold is set the elements of the matrix below the threshold are considered zeros while the others are considered ones.

Although such threshold operation can be used in some scenarios, there are other cases in which the performance is significantly impaired. This is due to the fact that setting the threshold is not always a simple task. Moreover it is clear that binarization leads to loss of information.

As shown by Paccanaro et al in [44], the detection of protein complexes in protein-protein interaction networks is one of the cases where the use of the weights allows to obtain protein complexes more accurately. This is probably due to the fact that the weights which measure the connections between proteins are obtained synthetically and often they do not reflect their real tightness. Binarizing these data often leads to discarding links with weights below the threshold, while to deal with them permits to have a more accurate vision of the global structure of the protein-protein interactions networks.

Pavan and Pelillo generalise the classical graph-theoretic notion of a maximal clique to edge-weighted graphs [48, 49], with the aim of overcoming the issues associated to the thresholding operation necessary to apply clique-based algorithms, and those related to the partitioning approach.

They tried to characterise the original problem and to address the real essence of it, namely, what is the definition of cluster and how to characterize it. Translating the reasonable and simple idea of cluster into a rigorous formulation, called "dominant-set", they present an elegant way to find clusters and a framework which overcomes the limitations mentioned above.

Moreover they established a connection between the problem of finding Dominant Sets and quadratic programming, allowing to use dynamics from evolutionary game theory to find them. In this way the Dominant Sets framework provide the advantages of the pairwise clustering exploiting also the solid theory and the the long list of tools available from the graph theory, optimization and game theory.

2.5.2 The Idea

An informal definition states that "a cluster is a set of entities which are alike, and entities from different clusters are not alike" [30]. Hence a cluster should satisfy two fundamental conditions[49]:

1. **Internal Criterion**: it should have high internal homogeneity;
2. **External Criterion**: there should be high inhomogeneity between the entities in the cluster and those outside.

Recalling that Dominant Sets is a pairwise clustering algorithm graph-based it is clear that in this context the entities are represented as nodes of a weighted graph. Therefore the criteria reported above express the following concept. The weights on the edges in a cluster should be large, while the weights on the edges that connect the cluster's nodes to nodes outside the cluster should be small.

Before we present the formal definition of cluster used in the Dominant Sets framework it is important to understand the general idea behind it. The intuitive idea is to percolate the weights from the edges to the nodes of the graph.

2.5.3 Formal definition of the input data

Before we present the rigorous formulation of the notion of cluster proposed by the Dominant Sets framework it is necessary to present formal definitions used to represent the input data. The input data to be clustered is represented as an undirected edge-weighted graph without self-loops $G = (V, E, w)$, where $V = \{1, ..., n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set, and $w : E \to R^*_+$ is the (positive) weight function. Vertices in G correspond to input data points and edges represent relationships. The weights on the edges reflect similarity between pairs of linked vertices. As usual, the graph G is represented with its **weighted adjacency** (or **similarity**) matrix, which is the $n \times n$ symmetric matrix $A = (a_{ij})$ defined as:

$$a_{ij} = \begin{cases} w(i,j), & \text{if } (i,j) \in E \\ 0, & otherwise \end{cases}$$

Clearly, the absence of selfloops, means that all the elements on the main diagonal of A are zero.

2.5.4 Formal definition of Dominant Sets

Before we see how the **internal** and the **exteral criterion** presented in the previous section have been formalized it is necessary to introduce some useful definitions.

Let $S \subseteq V$ be a non-empty subset of vertices and $i \in V$. The **(average) weighted degree** of i w.r.t. S is defined as:

$$awdeg_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij}$$

Observe that $awdeg_i(i) = 0$ for any $i \in V$. This term averages all the absolute similarity a_{ij} between i and all the other nodes j in S.

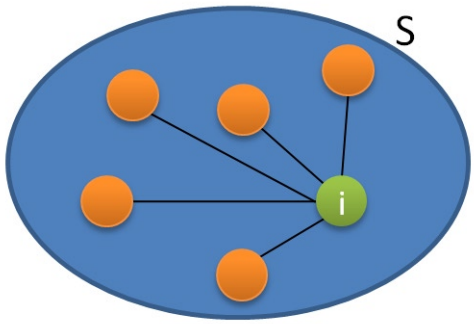

Figure 2.2: Average weighted degree. The figure shows a set (S) of elements and the edges connecting i and all the other nodes of S

Moreover, if $j \notin S$ we define:

$$\phi_S(i, j) = a_{ij} - awdeg_S(i)$$

Note that $\phi_i(i, j) = a_{ij}$, for all $i, j \in V$ with $i \neq j$. Intuitively, $\phi_S(i, j)$ measures the similarity between nodes j and i, with respect to the average similarity between node i and its neighbours in S. Note that $\phi_S(i, j)$ can be either positive or negative.

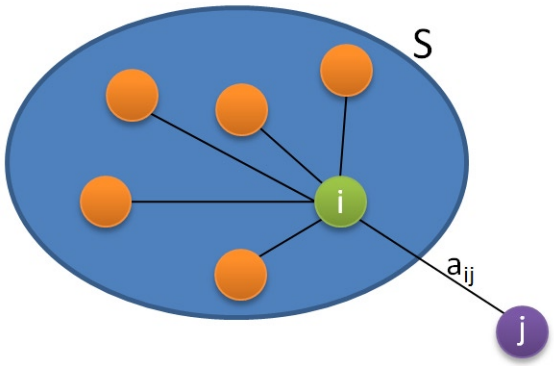

Figure 2.3: $\phi_S(i,j)$ measures the absolute similarity between j and i, with respect to the average similarity between node i and its neighbours in S

We are now in a position to formalize the notion of "percolation" of node-weights, which is captured by the following recursive definition.

Definition 2.1. Let $S \subseteq$ V be a non-empty subsets of vertices and $i \in S$. The weight of i w.r.t S is

$$\mathrm{w}_S(i) = \begin{cases} 1, & if |S| = 1 \\ \sum_{j \in S \setminus \{i\}} \phi_{S\setminus\{i\}}(j,i)\ \mathrm{w}_{S\setminus\{i\}}(j), & otherwise \end{cases}$$

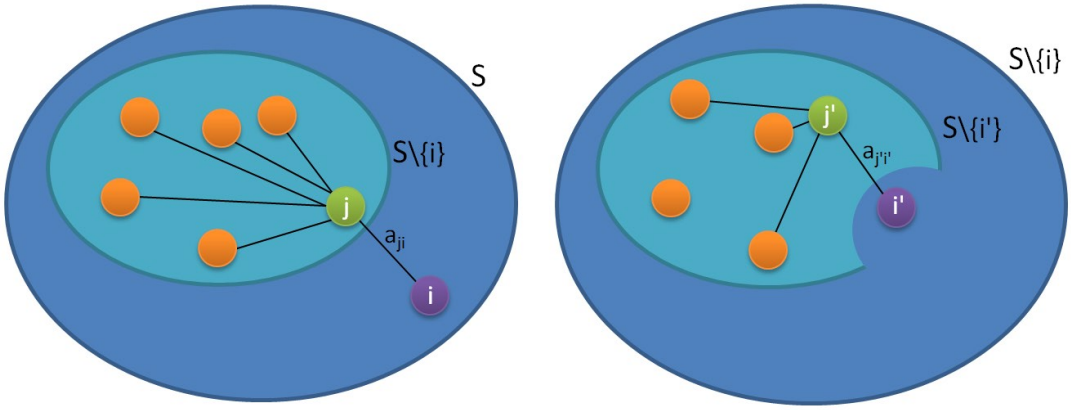

Figure 2.4:
To the left is reported an example of a set S and an element i which represents the input of $\mathrm{w}_S(i)$. The orange elements represent the elements iterated by the sum at the first step of the recursion. In order to express $\phi_{S\setminus\{i\}}(j,i)$ we used the chart in figure 2.4. On the right side is represented the recursive step. The node j on the left side become the node i of the recursive step on the right. The nodes signed as prime are the nodes of the first recursive step.

Moreover, the total weight of S is defined to be:

$$W(S) = \sum_{i \in S} \mathrm{w}_S(i)$$

Note that $\mathrm{w}_{\{i,j\}}(i) = \mathrm{w}_{\{i,j\}}(j) = a_{ij}$, for all $i, j \in V$ $(i \neq j)$. Also, observe that $\mathrm{w}_S(i)$ is calculated simply as a function of the weights on the edges of the subgraph induced by S.

Intuitively, $\mathrm{w}_S(i)$ gives us a measure of the overall similarity between vertex i and the vertices of $S\setminus\{i\}$ with respect to the overall similarity among the vertices in $S\setminus\{i\}$.

Now that all the necessary definitions have been reported it is possible to provide the formal definition of the two criterions presented before, which characterize a cluster. The following definition represents the formalization used by Dominant Sets framework of the concept of cluster in an edge-weighted graph.

Definition 2.2. A nonempty subset of vertices $S \subseteq V$ such that $W(T) > 0$ for any nonempty $T \subseteq S$, is said to be dominant if:

1. $\mathrm{w}_S(i) > 0$ **Internal Homogeneity**

2. $w_{S\cup\{i\}}(i) < 0$, for all $i \notin S$ **External Inhomogeneity**

For example, looking the graph of figure 2.5 it turns out that $w_{\{4,5,6,7\}}(4) < 0$ and $w_{\{8,9,10,11\}}(8) > 0$. Indeed, this can be intuitively grasped just looking at the values of the edges associated at the vertexes 4 and 8. We can notice that the weight associated to vertex 4 is smaller than that of subset $\{5, 6, 7\}$; conversely, the weight associated to vertex 8 is greater than that of subset $\{9, 10, 11\}$.

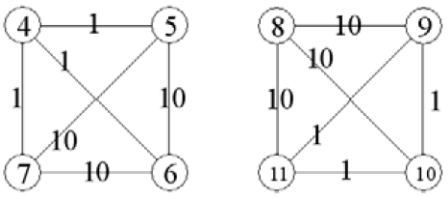

Figure 2.5: Example of edge-weighted graphs[49]

As other example consider the graph in figure 2.6. The subset of vertices $\{1, 2, 3\}$ is dominant. This is explained by observing that the weights of the edges internal of $\{1, 2, 3\}$ are 60, 70 and 90. While the weights of the edges that connect internal nodes with external ones have values between 5 and 25. Hence, the values of weights in the former set is larger then the second one.

Therefore, from this example it is clear that a dominant set has the property of having an overall similarity among internal nodes higher than that between internal nodes and external ones. This is the reason of considering Dominant Sets as clusters of nodes.

By definition Dominant Sets capture compact structures, hence it is not surprising that the definition of a dominant set is equivalent to the definition of (strictly) maximal clique when applied to unweighted graphs [48]. Since maximal cliques are a classic formalization of the notion of a cluster [1], [6], [8], [18] this represents a further motivation to consider Dominant Sets as clusters.

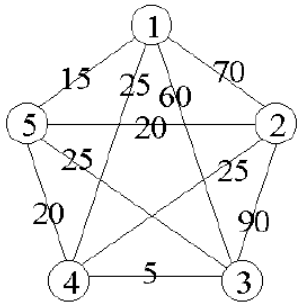

Figure 2.6: Example of dominant subset of vertices - {1,2,3}[49]

2.6 How to find Dominant Sets - Optimization Problem

The authors of this framework proposed much more than a new way to present and model clusters on graphs. They transform the combinatorial problem of finding Dominant Sets in a graph into a pure optimization problem. Establishing a correspondence between Dominant Sets and the extrema of a continuous quadratic form over the standard simplex allows them to use continuous optimization techniques to find them [49].

Consider a similarity graph $G = (V, E, w)$ with n vertices, and its adjacency matrix A. A vector with n components can be associated to a cluster of vertexes to represent it. Each component of this vector is a real value that express how the relative node is associated to the cluster. If a node is weakly associated to a cluster the relative component of the vector of that cluster is a small value. Conversely, if that node is highly connected to that cluster it is a large value.

In a good cluster its elements are strongly associated with each other, having large values in the similarity matrix. Hence, a natural way of defining the cohesiveness of a cluster is given by the following quadratic form:

$$f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x}$$

This allows us to formulate the pairwise clustering problem as the problem of finding a vector $\mathbf{x}$ that maximizes f. Imposing to this objective function probability

constraints we yield to the following standard quadratic program.

$$\begin{aligned} &maximize \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \Delta \end{aligned} \tag{2.1}$$

where

$$\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq 0 \wedge \mathbf{e}^T\mathbf{x} = 1\}$$

is the standard simplex of the n-dimensional Euclidean space $\mathbb{R}^n$.

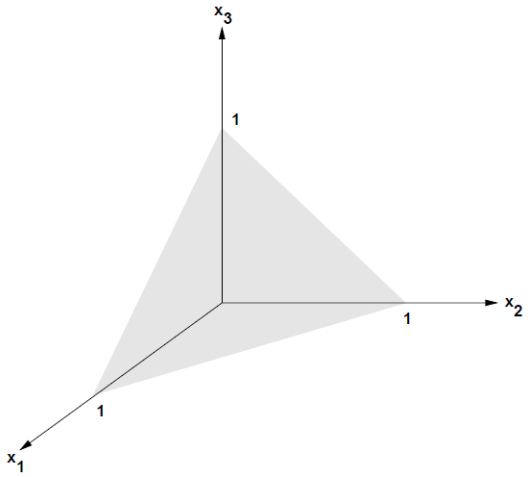

Figure 2.7: Standard Simplex Δ in R^3 [12]

It is worth emphasizing that the problem (2.1) is a generalisation of the Motzkin-Straus problem from graph theory [42], and that alternatively is it possible to write $\mathbf{x}^T A\mathbf{x}$ as $\sum_i \sum_j a_{ij}x_i x_j$.

According to this formulation, a maximally cohesive cluster corresponds to a local solution of program (2.1). However, it is necessary show how this notion of cluster is intimately related to Dominant Sets.

As reported in [49] a point $\mathbf{x} \in \Delta$ satisfies the Karush-Kuhn-Tucker (KKT) conditions for problem (2.1) i.e., the first-order necessary conditions for local optimality (2.2), if there exist $n+1$ real constants (Lagrange multipliers) $\mu_1, ..., \mu_n$ and λ, with $\mu_i \geq 0$ for all $i = 1, ..., n$, such that:

$$(A\mathbf{x})_i - \lambda + \mu_i = 0 \tag{2.2}$$

for all $i = 1, ..., n$ and $\sum_{i=1}^{n} x_i \mu_i = 0$

Note that, since both x_i and μ_i are non negative for all $i = 1, .., n$, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ implies $\mu_i = 0$. Hence, the KKT conditions can be rewritten as:

$$(A\mathbf{x})_i = \begin{cases} = \lambda, & \text{if } i \in \sigma(\mathbf{x}) \\ \leq \lambda, & otherwise \end{cases} \tag{2.3}$$

for some real constant λ (indeed, it is immediate to see that $\lambda = \mathbf{x}^T A\mathbf{x}$). A point $\mathbf{x} \in \Delta$ satisfying (2.3) will be called a KKT *point* throughout.

Before to see the next definition we remember that given a vector $\mathbf{x} \in \mathbb{R}^n$, the support of $\boldsymbol{x}$ is defined as the set of indices corresponding to its non-zero components, that is:

$$\sigma(\mathbf{x}) = \{i \in V : x_i \neq 0\} \tag{2.4}$$

We can now introduce a definition and a lemma useful for our purposes.

Definition 2.3. We say that a nonempty subset of vertices S admits weighted characteristic vector $\mathbf{x}^S \in \Delta$ if it has nonnull total weight W(S), in which case, we set:

$$x_i^S = \begin{cases} \frac{\mathrm{w}_S(i)}{W(S)}, & \text{if } i \in S \\ 0, & otherwise \end{cases}$$

Notice that, by definition, Dominant Sets always admit a weighted characteristic vector. Moreover $\mathbf{x}^S$ has n components where n is the number of the vertices of S. Summing up all the elements $\frac{\mathrm{w}_S(i)}{W(S)}$ we obtain 1, because $\forall i\ \mathrm{w}_S(i) \geq 0$ and $\sum_i \mathrm{w}_S(i) = 1$.

The next two results establish useful connections between KKT points of program (2.1) and weighted characteristic vectors.

Lemma 2.6.1. *Let $\sigma = \sigma(\boldsymbol{x})$ be the support of a vector $\boldsymbol{x} \in \Delta$ which admits weighted characteristic vector $\boldsymbol{x}^\sigma$. Then, $\boldsymbol{x}$ satisfies the KKT equality conditions in* (2.3) *if and only if $\boldsymbol{x} = \boldsymbol{x}^\sigma$. Moreover, in this case, we have:*

$$\frac{W_{\sigma\cup\{j\}}(j)}{W(\sigma)} = (A\boldsymbol{x})_j - (Ax)_i = -\mu_j \tag{2.5}$$

for all $i \in \sigma$ and $j \notin \sigma$, where the μ_js are the (nonnegative) Lagrange multipliers of program (2.1).

Proposition 2.6.2. *Let $x \in \Delta$ be a vector whose support $\sigma = \sigma(\boldsymbol{x})$ has positive total weight $W(\sigma)$ and, hence, admitting weighted characteristic vector $\boldsymbol{x}^\sigma$. Then, $\boldsymbol{x}$ is a KKT point for* (2.1) *if and only if the following conditions hold:*

1. $\boldsymbol{x} = \boldsymbol{x}^\sigma$
2. $w_{\sigma\cup\{j\}}(j) \leq 0, for\ all j \notin \sigma$

A formal proof of the lemma 2.6.1 and of the preposition 2.6.2 is presented in [49]. We can now introduce the core theorem of the section. One of the main results presented in [48, 49] is an important theorem which establishes an interesting connection between Dominant Sets and local solutions of program (2.1). A a formal proof of the theorem is presented in [48] and in [49].

Theorem 2.6.3. *If S is a dominant subset of vertices, then its weighted characteristics vector $\boldsymbol{x}^S$ is a strict local solution of program* (2.1).

Conversely, if $\boldsymbol{x}^*$ is a strict local solution of program (2.1) *then its support $\sigma = \sigma(\boldsymbol{x}^*)$ is a dominant set, provided that $w_{\sigma\cup\{i\}}(i) \notin 0$ for all $i \neq \sigma$.*

The theorem 2.6.3 establishes a one-to-one correspondence between strict local maximizers of $\mathbf{x}^T A\mathbf{x}$ over Δ and Dominant Sets. Hence, to the support of the weighted characteristic vector $\mathbf{x}^S$, with $S \subseteq V$, corresponds to find a set of vertexes which is a dominant set in V, because $\mathbf{x}^S$ is a strict local maximizer of the optimization problem $max\ \mathbf{x}^T A\mathbf{x}$ (with A adjacency matrix of G).

By virtue of theorem 2.6.3 Dominant Sets are in correspondence with (strict) solutions of quadratic program 2.1. This is very important because once the clustering problem is formulated as a continuous optimization problem, we can use any optimization technique to solve it. In the next section we shall see that payoff-monotonic dynamics from evolutionary game theory lend themselves well to this task.

2.7 From Local Optima to Game Theory

Before seeing how is it possible to exploit evolutionary game theory for our purposes, showing that first-order discrete-time replicator equations are a useful heuristic for finding Dominant Sets, this section presents some principles of game theory and evolutionary game theory.

2.7.1 Introduction to Game Theory

The goal of game theory is to model situations where two or more agents, called players, have to take some decision with the purpose of maximizing their utility. It is a wide field with mathematical models and techniques which change according to the characteristics of the game. For a complete insight of the topic we refer to [43][45][21]. Each player has several strategies and at each time it decides between different options in order to maximize a payoff which depends on the moves played by the co-players, which in turn try to maximize their payoff.

In this context with the term **strategy** we refer one of the options that the player can choose. It is worth emphasizing that the decisions of each player depends not only on on the options available for that particular player but also on the actions of others.

Frequently, the concept of strategy is confused with that of move. An action performed by a player during a game is a **move**, while a strategy specify the action that a player take at any step. For example,in chess moving white's Bishop from one place to another is a move. While a strategy is a complete algorithm from playing a game, declaring for a player what to do for every possible situation during the game.

Supposing to have only two players with m and n moves respectively, $A_1, ..., A_m$ and $B_1, ..., B_m$. Each time the players repeat the same game choosing a strategy without knowing what the co-player does. If the players choose the moves A_i and B_j respectively, the player A will receive an utility a_{ij}. The matrix $m \times n$, whose components are a_{ij}, is called **payoff matrix A**. A game of this type is known as **zero-sum** game because what a player wins is equal to what the other

loose. Moreover in this specific example the game is **non-cooperative** because the players basically are in competition.

The strategies can be pure or mixed. In **pure strategies** the players play each time the same move, let's say A_i and B_i. This leads us to the concept of game equilibrium. An **equilibrium** for a particular game is reached when none of the two players takes advantages to change its move if the opponent does not change its own.

Definition 2.4. A Nash equilibrium is a profile of strategies such that each player's strategy is an optimal response to the other player's strategies[21].

There more than one way to check if a game has an equilibrium in the space of pure strategies. Unfortunately some games does not admit equilibria in pure strategies, namely at a certain point some players will prefer to change its move.

The equilibria is guaranteed for each game only in the case of mixed strategies. A **mixed strategy** is an assignment of a probability to each pure strategy.

As pointed out before a mixed strategy is a probability distribution on the whole set of possible moves. For a player A for example a mixed strategy is a vector $\mathbf{x} \in \mathbb{R}^m_+$ such that $\sum_{i=1}^m x_i = 1$. Hence, for the player A each component x_i is the probability to play the move A_i. By the same token for a player B a mixed strategy is a vector $\mathbf{y} \in \mathbb{R}^n$ such that $\sum_{i=1}^n y_i = 1$. Given a couple (x, y) for the players the average payoff for A, namely the loss for B, is:

$$E(x, y) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j$$

. The notion of equilibria is the same of the one given for the pure strategies, which can be expressed formally by the following statement. A pair of strategy $(\bar{\mathrm{x}},\bar{\mathrm{y}})$ is an equilibria for the game if $E(x, \bar{y}) \leq E(\bar{x}, \bar{y}) \leq E(\bar{x}, y)$ for every strategy x of A and every strategy y of B.

Going forward we will always assume that the game is a two-player symmetric game where each player has the same payoff function. We define the expected payoff that a player obtains by playing the strategy i against an opponent playing

a mixed strategy $\mathbf{x}$ as

$$\pi(\mathbf{e}^i|\mathbf{x}) = (A\mathbf{x})_i = \sum_j a_{ij}x_j$$

where $\mathbf{e}^i$ is the vector with all components equal to zero excepting the i^{th}-component which is equal to 1. Sometimes this is identified with the pure strategy i.

The expected payoff received by adopting a mixed strategy $\mathbf{y}$ is thus

$$\pi(\mathbf{y}|\mathbf{x}) = \sum_{i\in S} y_i\pi(\mathbf{e}^i|\mathbf{x}) = \mathbf{y}^T A\mathbf{x}$$

The exprected payoff of the entire population is given by

$$\pi(\mathbf{x}) = \pi(\mathbf{x}|\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$$

We will also use the following notations:

$$\pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) = \pi(\mathbf{y}|\mathbf{x}) - \pi(\mathbf{x})$$

and

$$\pi(\mathbf{y}-\mathbf{x}) = \pi(\mathbf{y}-\mathbf{x}|\mathbf{y}) - \pi(\mathbf{y}-\mathbf{x}|\mathbf{x})$$

The *best replies* $\beta(\mathbf{x})$ against a mixed strategy $\mathbf{x}$ is the set of mixed strategies that maximize the expected payoff when played against $\mathbf{x}$, namely

$$\beta(\mathbf{x}) = \arg\max_{\mathbf{z}\in\Delta} \pi(\mathbf{z}|\mathbf{x})$$

With this formulation the notion of Nash equilibrium can be expressed as follows. A pair $(\mathbf{x},\mathbf{y}) \in \Delta^2$ is a Nash equilibrium if $\mathbf{x} \in \beta(\mathbf{y})$ and $\mathbf{y} \in \beta(\mathbf{x})$ Since we will consider symmetric games, namely games where two players are undistinguishable, only symmetric pairs $(\mathbf{x},\mathbf{x})$ of strategies are of interest. Hence, by abuse of language we call Nash equilibrium

$$\forall \mathbf{y} \in \Delta, \quad \pi(\mathbf{y}-\mathbf{x}|\mathbf{x}) \leq 0$$

This implies that $\forall i \in S,\ \pi(\mathbf{e}^i|\mathbf{x}) \leq \pi(\mathbf{x})$ Notice that the payoff of every strategy in the support of a Nash equilibrium $\mathbf{x}$ is constant, while the payoff is less or equal

than $\pi(\mathbf{x})$ for all strategies outside the support of $\mathbf{x}$.

W can also formulate the KKT conditions (2.3) presented in the previous chapter as follow

$$\pi(\mathbf{e}^i|\mathbf{x}) = (A\mathbf{x})_i = \begin{cases} = \lambda, & \text{if } i \in \sigma(\mathbf{x}) \\ \leq \lambda, & otherwise \end{cases} \tag{2.6}$$

and hence $\lambda = \pi(\mathbf{x})$ because

$$\pi(\mathbf{x}) = \sum_{i \in \sigma(\mathbf{x})} x_i \pi(\mathbf{e}^i|\mathbf{x}) = \sum_{i \in \sigma(\mathbf{x})} x_i \lambda = \lambda$$

Therefore, if $\mathbf{x}$ satisfies the KTT $\forall i = 1, .., n$ this corresponds to $\pi(\mathbf{e}^i|\mathbf{x}) \leq \pi(\mathbf{x})$ which is the Nash equilibrium condition reported above for symmetric payoff matrices. Hence, for this particular case, the Nash condition is equivalent to the necessary condition for local optimality in (2.1). This shows the connection between our optimization problem and game theory.

In the next chapter we shall introduce some basic concepts of evolutionary game theory and the *Evolutionary Stable Strategy* (ESS) notion.

2.7.2 Evolutionary Game Theory

Evolutionary game theory, is a discipline introduce in 1973 by J. Maynard Smith [57] with the aim to model the evolution of animal behaviour using the principles of noncooperative game theory. During these past years the evolutionary game theory allowed to explain many complex aspects of biology. However, even if its original purpose was to model the evolutionary Darwinian process, recently it has been applied also in economy, sociology and philosophy. For a complete overview of the topic we refer to [65, 55, 28].

At the beginning, evolutionary game theory was born with the aim to explain the animal behaviour in a conflict situation. The concept of strategy is analogous to that applied in classical game theory, however its success depends on several factors, such as the alternative strategies and the frequency with which they are applied by the other members of the population. Moreover, it is worth emphasizing

that in this context it is relevant also how effective a strategy is against itself. Indeed, if the individuals of a specie plays an effective strategy able to dominate the other species they will end up to compete against each other.

Other conceptual differences are:

- in order to model animal behaviour players the agents are supposed to be rational;
- the payoff in this context is in unit of fitness, namely it represents the reproductive success;
- it is a game with more than one player played in a large population of individuals which compete for a limited resource;
- players do not choose their strategy or have the ability to change it, they are born with that strategy preprogrammed from nature, and their offspring will inherit that same identical;

These variances are necessary to model the Darwin's theory of evolution. However, as in the classical game-theoretic approach, the results of a game prove how effective is each strategy. This schema models exactly the evolution process, testing the capability of surviving and reproducing of the various strategies applied by the individuals of a population. This is the main reason for which it is widely applied to explain some of the most important biology questions, such as the group selection, the altruism dynamics and the co-evolution [67].

A key concept in evolutionary game theory is that of evolutionary stable strategy [57, 40, 39], which represents a strategy robust to the evolutionary selection scheme. Let $\mathbf{x} \in \Delta$ the strategy played by a large population of individuals. Suppose that in this population appears a small group $\varepsilon \in (0, 1)$ of mutants designed by nature to play a strategy $\mathbf{y}$. We call $\mathbf{x}$ the **incumbent strategy** and $\mathbf{y}$ the *mutant strategy*. The payoff in a match of this population is the same as in a match with an individual who plays the mixed strategy $\mathbf{w} = \varepsilon\mathbf{y} + (1 - \varepsilon)\mathbf{x} \in \Delta$. The payoff of the incumbent strategy is thus $\pi(\mathbf{x}|\mathbf{w})$, and that of the mutant strategy is $\pi(\mathbf{y}|\mathbf{w})$.

The intuition behind the evolutionary process suggests that the evolution will select the incumbent strategy against the mutant one if its payoff is greater that

the mutant strategy

$$\pi(\mathbf{x}|\mathbf{w}) > \pi(\mathbf{y}|\mathbf{w})$$

Hence, the definition of *evolutionary stable strategy* (ESS) arise quite naturally: a strategy $x \in \Delta$ is said to be an ESS if the mutants remain a small portion of the population, namely if the previous inequality holds for any mutant strategy $\mathbf{y} \neq \mathbf{x}$.

Likewise the Nash equilibrium, the ESS is a property of the game and it does not suggest how the population can arrive at such strategy. Indeed, it only expresses the necessary condition for a strategy in order to be robust to the evolutionary pressure.

As pointed out before a process representing the selection mechanism that governs the dynamical behaviour of the population over time is necessary. **Replicator equations** are the most common approach to study the dynamics in a game played in an evolutionary setting, determining the growing rate of the set of individuals that are playing a given strategy. As we will see the growing rate of a given strategy is obtained subtracting to average payoff of the entire population the average payoff of that particular strategy.

It is worth emphasizing that evolutionary game theory is a model representing the entire Darwinian process, figure 2.8. Indeed, the game played models the natural selection process while the replicator dynamics models the hereditary dynamics applied after each generation.

Formally, let be $J = \{1, ..., n\}$ the set of n strategies and $\mathbf{W} = (w_{ij})$ the payoff matrix $n \times n$, where w_{ij} is hence the payoff when the strategy i is played versus the strategy j. At the time t we denote with $x_i(t)$ the proportion of population's players which play strategy i. Remembering that we have n strategies we have a state vector defined as $\bar{x}(t) = (x_1(t), x_2(t), ..., x_n(t)) \in \Delta$. Belonging to the standard simplex it is clear a probability distribution.

Now it is possible to define the **average payoff** of the strategy i as:

$$\pi_i = (\mathbf{W}\mathbf{x})_i = \sum_j w_{ij} x_j$$

where x_j represents the probability of picking the strategy j, and w_{ij} the payoff

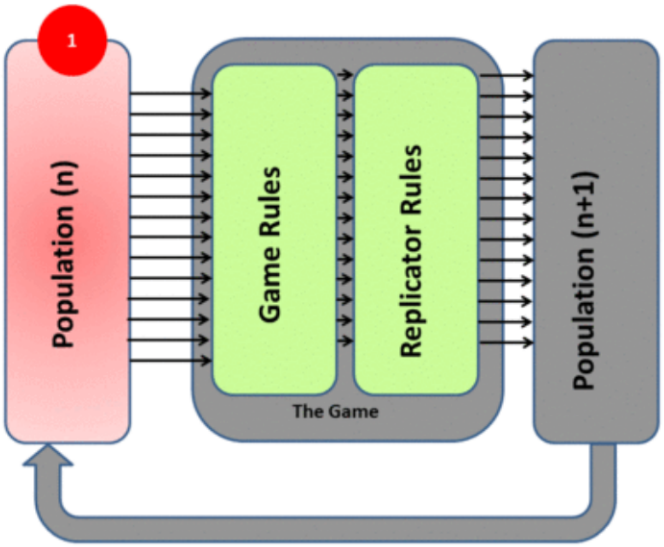

Figure 2.8: Evolutionary Game Theory Model [67]

when the strategy i is played against the strategy j. The **average fitness of the entire population** is hence $\pi = \sum_i x_i \pi_i$.

In evolutionary game theory the fact that the game is played over and over, generation after generation, models the natural section process which result in the evolution of the fittest strategies. The basic idea behind replicator dynamics is that good strategies, namely strategies better than the average, will spread over time, while bad strategies will get extinct. With the aim of describing the evolution of behavioral phenotypes a set of differential equations have has been proposed. A general class of evolution equations is given by:

$$x_i = \frac{dx_i}{dt} = x_i(\pi_i(x) - \pi(x)) \quad \text{continuous time} \tag{2.7}$$

$$x_i(t+1) = \frac{x_i(t)\pi_i(t)}{\pi(t)} = \frac{x_i(t)\pi_i(t)}{\sum_j x_j(t)\pi_j(t)} \quad \text{discrete time} \tag{2.8}$$

Note that if $\pi_i > \pi$, or respectively $x_i(t+1) > x_i(t)$ the strategy i spread over time.

The simplex Δ is invariant under these dynamics, which means that every trajectory starting in Δ will remain in Δ for all future times. Moreover holds the

following theorem

Theorem 2.7.1. *If $M = M^T$ then the function $\boldsymbol{x}(t)^T M\boldsymbol{x}(t)$ is strictly increasing with increasing t along any non-stationary trajectory $\boldsymbol{x}(t)$ under discrete-time* (2.8) *replicator dynamics. Furthermore, any such trajectory converges to a stationary point.*

Finally, a vector $\boldsymbol{x} \in \Delta$ is asymptotically stable under (2.8) *if and only if $\boldsymbol{x}$ is a strict local maximizer of $\boldsymbol{x}^T M\boldsymbol{x}$ on Δ.*

The previous result is known as the *fundamental theorem of natural selection* [28, 65] and it was formulated by R. A. Fisher in 1930, while that all trajectories of the replicator dynamics converge to a stationary point has been proven more recently [36, 37].

Hence, first-order discrete-time replicator equations are a simple and useful heuristic for finding Dominant Sets. Indeed, let A denote the weighted adjacency matrix of an edge weighted graph G. By letting $M = A$ we know that the replicator dynamical systems 2.8, starting from an arbitrary initial state, will iteratively maximize the function $\mathbf{x}^T M\mathbf{x}$ over Δ and will eventually be attracted with probability 1 by the nearest asymptotically stable point. By virtue of Theorem 2.7.1 this will then correspond to a strict local maximizer of $\mathbf{x}^T M\mathbf{x}$ in Δ and hence, for what said in the relative section , to a dominant set.

Since the process cannot leave the boundary of Δ, it is usual to start out the relaxation process from some interior point, a common choice being the barycenter of Δ. This prevents to be biased in favor of any particular solution during the research.

2.8 Infection and immunization

Unfortunately, as we will emphasize in section 5.1, replicator dynamics, as do more or less the other standard evolutionary dynamics, are afflicted by some computational problems. In order to overcome this issues we review briefly some principles of a new class of evolutionary dynamics, inspired by infection and immunization processes. For a complete overview of the topic we refer to [13, 14].

These dynamics are built upon a central paradigm of evolutionary game theory called invasion barrier.

The main concept is the following. Consider the set of all the populations not in equilibrium. For any of these $\mathbf{x}$ there exists at least one mixed strategy $\mathbf{y}$ that is a better response to $\mathbf{x}$ than $\mathbf{x}$ to itself. In this case we say that $\mathbf{x}$ has no invasion barrier against $\mathbf{y}$. Therefore, if small share of mutant agents "infect" the current population, namely play an "infective strategy" $\mathbf{y}$, they will spread until the invasion barrier against them becomes positive. This amount to say until the new population turns out to be "immune" against the "infective strategy" $\mathbf{y}$.

This process remembers how a vaccine works: a small share of virus is introduced in a body in order to lead its immune system to prevent future infections. The authors of this recent class of evolutionary game dynamics propose to iterate this process of infection and immunization in order to obtain a population for which no infective strategy can be found anymore, because in that case a Nash equilibrium has been reached. In their work they provide a formal proof that fixed points of these dynamics are Nash equilibria and vice versa, independently from the way they select infective strategies at each iteration.

Chapter 3

Clustering methods for PPI networks

3.1 Introduction

In this chapter we present some of the most well known clustering methods applied to PPI network present in literature. For each algorithm we present a brief introduction with the main characteristics.

In addition to ClusterONE, the state of the art for detecting protein complexes from Protein-Protein Interaction networks, were used Affinity Propagation, CFinder, CMC, MCL, MCODE, RNSC and RRW.

Only some of these algorithms support the use of edge weights, particularly Affinity Propagation, MCL, RRW and ClusterONE. In order to run algorithms not supporting directly weights, namely similarity values between pairs of input objects, (MCODE, RNSC, CMC and CFinder), we pre-binarized the input networks using the threshold values originally suggested by the authors of the datasets. Interactions with a weight smaller than the proposed threshold were ignored; interactions with a weight larger than the proposed threshold were kept. The suitability of these thresholds have been checked using the heuristic proposed by Apeltsin et al [2].

Only some of them could handle overlapping clusters, we allude to MCODE, CFinder, CMC, RRW and ClusterONE. This amount to say that only ClusterONE

Algorithm	Version	Weighted	Overlapping	Reference
ClusterONE	0.93	yes	yes	[44]
Affinity Propagation	5 Dec 2007	yes	no[a]	[20]
CFinder	2.0.5	no	yes	[47, 1]
CMC	2.0	yes[b]	yes	[35]
MCL	10-201	yes	no	[19, 63]
MCODE	1.31c	no	yes	[5]
RNSC	2004	yes	no	[33]
RRW	8 Aug 2011	yes	yes	[38]

[a] CFinder has a weighted variant, but it turned out to be too slow for the protein complex detection task;

[b] Only in the initial stage;;

Table 3.1: Details of the clustering algorithms applied

and RRW have the capability to handle at the same time weighted PPI data and overlapping clusters.

The table 3.1 presents all the clustering algorithms evaluated in this review with the relative version of the implementation used, the features and the original paper's authors.

3.2 ClusterONE

ClusterONE, also known as clustering with *O*verlapping *N*eighborhood *E*xpansion, is a method for detecting potentially overlapping protein complexes from protein-protein interaction data. As shown in the original paper [44] and summarized in section 5.5 this method exhibits better performance than the other clustering methods present in literature.

The most important feature provided by it is the capability to handle at the same time weighted PPI data and overlapping clusters. Indeed, due to the fact that proteins may have multiple functions, they therefore correspond to nodes may belong to more than one cluster; for example, 207 of 1,628 proteins in the CYC2008 hand-curated yeast complex data set[52] participate in more than one complex. Addressing in an explicit way this problem ClusterOne represent at the moment the better method for PPI data clustering.

3.2.1 The Algorithm

The ClusterONE algorithm uses a greedy approach with the aim of calculating a score called cohesiveness and detecting groups of proteins in Protein-Protein Interaction networks that corresponds to protein complexes. Cohesiveness measures how likely it is for a group of proteins to form a protein complex, and it was defined as follows:
Let $w^{in}(V)$ denote the total weight of edges contained entirely by a group of proteins V, and let $w^{bound}(V)$ denote the total weight of edges that connect that group with the rest of the network.

The cohesiveness of V is then given by

$$f(V) = \frac{w^{in}(V)}{w^{in}(V) + w^{bound}(V) + p|V|}$$

Before we proceed, it is interesting to emphasize the role of p. $p|V|$ is a penalty term modelling the uncertainty in the input data. It is included in the cohesiveness formula based on the hypothesis that, due to the limitations in the experimental procedure, each Protein-Protein Interaction network can contain a certain amount of yet undiscovered interactions .

Setting $p > 0$ corresponds to increasing the boundary weight $w^{bound}(V)$ by p$|V|$, assuming that every protein in V has p additional boundary edges. Therefore, an user can use different values of p for different proteins, depending on its biological assumption. A well-studied protein may have a lower p value assigned because it is less likely to have undiscovered interactions.

Comparing the cohesiveness of two groups of protein it is possible to assess which of them represents the best protein complex. Indeed, it is clear that a subgraph with many reliable edges has a high w^{in}, while a well-separated subgraph has a low w^{bound}. Both of these characteristics increase $f(V)$ making cohesiveness an easy way to asses the quality of groups of proteins.

First Step - Clusters construction

The algorithm consists of three steps. In the first step, ClusterONE grows groups with high cohesiveness from selected seed proteins. It starts selecting as the first

seed the protein with the highest degree, and grows a cohesive group from it using a greedy procedure. When the growth process finishes the algorithm selects the next seed. The next seed is chosen looking at the highest degree from the the proteins not already included in any protein complex. The process ends when there are no remaining proteins to consider.

A description of the greedy process used is reported below starting from v_0[44].

1. let $V_0 = v_0$. Set the step number $t = 0$.
2. calculate the cohesiveness of V_t and let $V_{t+1} = V_t$
3. for every external vertex v incident on at least one boundary edge, calculate the cohesiveness of $V' = V_t \cup \{v\}$. If $f(V') > f(V_{t+1})$, let $V_{t+1} = V'$.
4. for every internal vertex v incident on at least one boundary edge, calculate the cohesiveness of $V'' = V_t \setminus \{v\}$. If $f(V'') > f(V_{t+1})$, let $V_{t+1} = V''$.
5. if $V_t \neq V_{t+1}$, increase t and return to step 2. Otherwise, declare V_t a locally optimal cohesive group.

It is worth emphasizing that during the growth process any vertex, including the seed, can be removed from the group that the algorithm is building. However, if the seed is not included in its group it will not longer be a seed in the next steps, and hence no other groups can be built from it. Nevertheless it can be included in a group if that group growing from a different seed absorbs it.

To clarify the above procedure, we consider the following graph. Seven of the eleven nodes are marked by letters from A to G in figure 3.1. Assuming that no unknown connections exist in our graph, i.e. $p = 0$, the cohesiveness of the highlighted set is $\frac{10}{15}$.

In steps 3 and 4, ClusterONE can extend or contract the current set of vertexes. The nodes C, F and G can be added, while the nodes A, B, D and E can be removed. In this case the best choice is to add C to the set. Indeed, following this way three boundary edges will be "converted" to internal ones without bringing in any new boundary edges. After this step the group's cohesiveness increases to $\frac{13}{15}$ and making it locally optimal. This is clear, given that computing the cohesiveness of the groups obtained adding F or G, it decrease to $\frac{14}{17}$ and $\frac{14}{18}$ respectively.

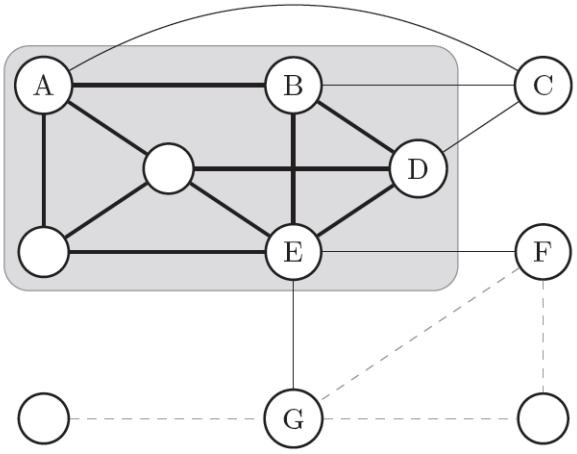

Figure 3.1: ClusterONE, example of execution[44]

Second Step - Merge

A pair of cohesive groups computed in the first step is merged in the second one according to their overlap. By default, highly overlapped groups are merged if their overlap score ω is larger than 0.8.

The overlap score of two protein sets A and B is defined as follows[35]:

$$\omega(A, B) = \frac{|A \bigcap B|^2}{|A||B|}$$

This step can be performed in two ways: merging one cluster after another, re-computing the overlap score at each step; concurrently building a graph of overlaps. The current implementation of clusterONE uses the following efficient procedure.

Firstly, given a set of clusters, it compute the overlap score for each pair of clusters. After that, it builds a graph in which each node represents a cluster and two nodes are connected if they overlap more than a certain threshold. Clusters connected to each other by a path of edges are merged. The results of this step are called protein complex candidates. Clearly, clusters isolated are promoted to protein complexes candidate without any merging step.

Third Step - Discard

In the final step of the algorithm, complex candidates that contain less than three proteins are discarded. This approach is quite common in literature and it is due to the fact that protein complexes of size two are difficult to detect. However, an input parameter regulates the minimum size of the protein complexes returned as output.

Protein complex candidates with a density below a given threshold σ are also discarded. The density of a group of n proteins is defined as the sum of its weighted internal edges, divided by $\frac{n(n-1)}{2}$.

3.3 MCL

The MCL clustering algorithm is a graph clustering algorithm introduced by Stijn van Dongen in [18].

Graph clustering is an important unsupervised learning technique widely studied in literature. This technique is very useful every time that a clear structure lies on the input data. In fact, in those cases, the classical methods which treat each object to be clustered as a point in a n-dimensional space have some difficult to catch the underling structure. Therefore, sometimes addressing the clustering problem modelling the input data as element of a graph may result more natural.

MCL uses the input similarity matrix as the adjacency matrix the of graph in which nodes are the input objects and edges represent similarity between objects. The edges could be weighted or unweighed, but overlapped output clusters are not allowed.

MCL translates the clustering process to the problem of finding dense regions of the input graph, that is there are many links within a cluster and fewer links between clusters. From this perspective the authors of MCL model the fact that starting from a cluster's node and randomly traversing the graph, it should be be more likely to stay within a cluster than travelling between clusters.

In order to exploit this evidence MCL uses Random Walks on the input graph adopting Markov Chains. In this way it tries to discover where the flow tends to gather and therefore where clusters are. For example, considering the figure 3.2,

in one time step, a random walker at node 1 has a 33% chance of going to node 2,3 or 4, and 0% chance to nodes 5,6, or 7. From node 2, 25% chance for 1, 3, 4 or 5 and 0% for 6 and 7.

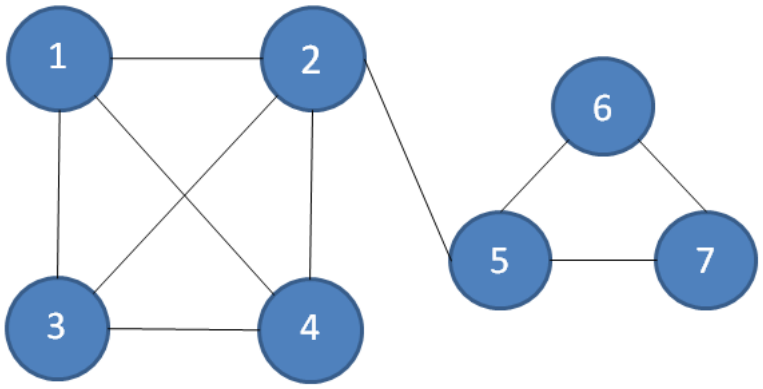

Figure 3.2: MCL example

It is then possible to build a set of transition matrices in which the columns look like probability vectors. Each transition matrix k represent the probability to move from one node i to another node j of the graph in k steps. The Markov Chains ensure that, given the present state, the past and future states are independent, and hence the probabilities for the next step only depend on the current probabilities. Calculating successive powers of the associated adjacency matrix the Markov Cluster Algorithm simulates a flow on the input graph.

It is worth emphasizing that often the input graph requires the adding of a self loop on each graph's node. In fact, if a transition matrix is multiplied for itself k times emerges the problem so-called k-path clustering.

Basically the problem becomes clear, for example, when one node x is directed connected to other two nodes i and j, which neighbours are connected to x in a different manner. Indeed, lets suppose that, while i is highly connected with other nodes, a turn connected to x, j is not.

At the beginning, the first transition matrix express in a proper manner the difference between the connection among x and i and the connection among x and j. But, intrinsically after each matrix multiplication, the resulted matrix does not take into account the fact that i and j are connected to x in a different

manner. The self loop supply to this lack demoting less popular neighbors and thus permitting to the k-th matrix to be more representative.

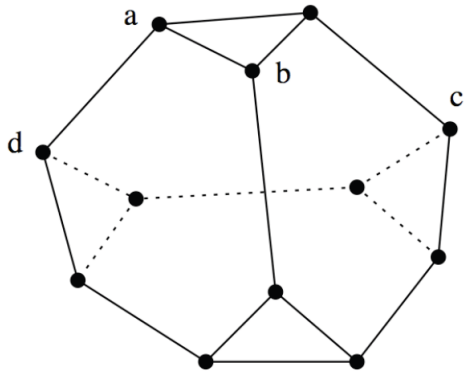

Figure 3.3: MCL k-path clustering problem[19]. Considering Z_2 as the set of paths of length 2 from one node to another. We notice that $Z_2(a, b) = 1$ and $Z_2(a, c) = 1$ but a and b are more closely couple than a and c. Moreover considering Z_3 we have that $Z_3(a, b) = 2$ and $Z_3(a, d) = 2$ but again, a and b are more closely coupled that a and d

Adding self loops solves the problem only temporarily, indeed in the long run their effect disappears. For this reason some adjustments have to be performed on the transition matrix. At each iteration it is necessary to raise each single column to a non-negative power, and then re-normalizing. This operation is named "Inflation". This step enhances the contrast between regions of strong or weak flow in the graph.

The inflation is the only parameter required by MCL and it tunes the granularity of the clustering. Larger inflation values result in smaller clusters, while smaller inflation values generate only a few large clusters. The range of possible inflation values for the MCL algorithm [63, 19] was sampled uniformly with a step size of 0.1 until to reach a value which corresponds to better performance.

3.4 RSNC

RSNC, also known as *R*estricted *N*eighborhood *S*earch *C*lustering, is a graph clustering algorithm introduced by A.D King et al in [33].

It requires as an input the number of clusters to be extracted and, even though, it was designed with the precise purpose of clustering Protein-Protein Interaction networks, it does not support the detection of overlapped clusters and it cannot handle real values as similarity between pairs of proteins.

The lack of these features represents a remarkable limitation in protein complexes detection. Indeed, PPI networks often provide a weight for each edge and, as Paccanaro at al. show in [44], clustering algorithms achieve better performance taking them into account. Moreover is well known that a protein could belong to more than one protein complex.

As argued in the previous section, the process of clustering a graph $G(V, E)$ can be seen as the decomposition of its node set into subsets of nodes, each of which is highly interconnected. Hence, it can be seen as the detection of subsets of nodes which induce dense subgraphs.

With the aim of finding dense subgraphs RNSC calculates the value of a particular cost function, which depends on its edges. Being unable to deal with weighted edges this cost function depends on their quantity.

It uses two types of score: the *naive score* and the *scaled score*. The value of the former, for a node i, is the sum between the number of neighbours that are not in the same cluster of i, and the number of node that are not neighbours of i but that belong to the same cluster.

$$naive\_score(i) = \big|\{x \in V \backslash C, (i, x) \in E\}\big| + \big|\{x \in C, (i, x) \notin E\}\big|$$

where $C \subseteq V$ represent the cluster of node i.

The scaled score, for a node i that belong to a cluster C, is its naive score divided by the number of nodes in C plus the number of neighbours of i.

The algorithm computes a summarizing value for assessing a cluster summing the values of these scores for each node in that cluster. An example of this measure is proposed in figure 3.4.

$$scaled\_score(i) = \frac{naive\_score(i)}{|C| + |\{x \in V, (i, x) \in E\}|}$$

RNSC starts from a random solution. At each iteration the algorithm tries to move a vertex from one cluster to another. If the value of the cost function

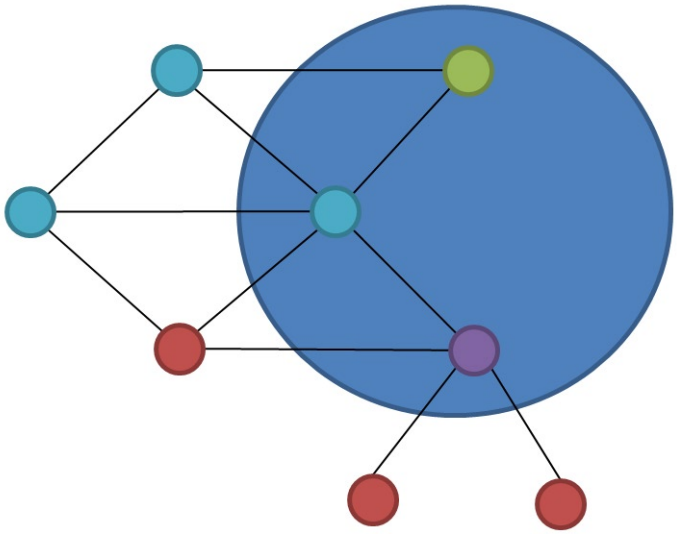

Figure 3.4: RNSC example: we consider the purple node. The number of its neighbours not in the highlighted cluster is 3 (red nodes); the number of its not neighbours in the cluster is 1 (green node). The naive score is hence 4. The number of its neighbours is 3 and the size of the cluster is 3, hence the scaled score is $\frac{4}{3+3}$

decreases after one of these attempts the new cluster is kept. Therefore it is a cost-based local search algorithm.

Moreover, in order to prevent a cycle it maintains a list of tabu moves[24], also called forbidden moves, and it terminates when a specified number of moves, which do not decrease the cost function, has been reached.

Unfortunately, being randomized, different runs of this algorithm, on the same input data, can result in different clusters.

Furthermore, as other local search algorithms, RNSC is prone to find poor local minima. With the aim of avoiding this problem, it makes some diversification moves, mixing the contents of a cluster at random.

RNSC algorithm has a large number of tunable parameters. Following the approach chosen by Brohee et al [11], all the possible combinations of the following parameter values have been tried:

- Shuffling diversification length: 3, 5, 9
- Diversification frequency: 10, 20, 50

- Number of experiments: 1, 3, 10
- Naive stopping tolerance: 10, 20, 50
- Scaled stopping tolerance: 1, 5, 15
- Tabu length: 1, 10, 50, 100
- Tabu tolerance: 1, 3, 5

The total number of parameter combinations tried was 2916. Since, as was said before, the RNSC algorithm is randomized, each combination was tried 5 times for each dataset and the one resulting in the best maximum matching ratio was kept.

3.5 Affinity Propagation

The Affinity Propagation is a clustering algorithm proposed by B.J. Frey and D. Dueck in 2007[20]. It handles real-valued similarities between the input object but it cannot detect overlapped cluster.

AP is a k-centers clustering techniques using the input data to learn a set of centers such that the sum of squared errors between each input object and its nearest center is minimum. In this type of algorithm the centers are often selected from the input data points. In this case they are referred as "exemplars".

Often, this methods begin with an initial set of exemplars randomly selected refining this set iteratively with the aim of decreasing the sum of squared errors. Unfortunately, in this manner the clustering is sensitive to the initial selection of exemplars.

Affinity Propagation tries to deal with this limitation taking a quite different approach. It considers each data point as a node in a network, each of which sends messages to all the other communicating its relative attractiveness for them. Each node receiving this attractiveness communication responds sending back messages about their relative availability. Given an availability message a node restart the process responding again with an attractiveness value. Hence, two type of messages are exchanged: the "responsibility" information $r(i, k)$, and the "availability" information $a(i, k)$.

AP uses this message passing procedure between nodes with the aim of examining all the object as potential exemplars. Indeed, by exchanging messages each object is trying to identify its best representative, namely its best exemplar, with respect to a particular function. The name of this algorithm is due to the fact that the magnitude of each message exchanged during the time resembles the affinity that a particular data point has for another data point as its exemplar.

3.5.1 The Algorithm

With the aim of detecting the centers, Affinity Propagation uses three main matrices, a similarity matrix, a responsibility matrix and an availability matrix, storing the results in a criterion matrix. AP handles non-positive real number as similarity values, hence, the similarity matrix is usually built by negating the distances between objects given as input. The algorithm is reported below:

1. Initialize the availability matrix $A_{N\times N}$ to zero
2. Updating all responsibilities $r(i,k)$:

$$r(i,k) \leftarrow s(i,k) - max_{k's.t.k'\neq k}\{a(i,k') + s(i,k')\}$$

3. Updating all availabilities $a(i,k)$:

$$a(i,k) \leftarrow min\{0, r(k,k) + \sum_{i's.t.i'\notin\{i,k\}} max\{0, r(i',k)\}\}$$

$$a(k,k) \leftarrow \sum_{i's.t.i'\neq k} max\{0, r(i',k)\}$$

4. $c(i,k) \leftarrow (i,k) + a(i,k)$. For point i, the point k that maximizes $c(i,k)$ is its exemplar;
5. If decisions made in step 4 did not change for a certain times of iteration or a fixed number of iteration reaches, go to step 6. Otherwise, go to step 2
6. Each data point has its own exemplar, and the points with the same exemplar constitute a community.

The availability matrix is initialized with all the elements set to zero, while the responsibility of a node i to j is the subtraction between the similarity among i and j minus the maximum of the remaining similarities of the column j. At the next step the criterion matrix is updated combining availabilities and responsibilities. For each input object its exemplar is given by looking at the criterion matrix by row. Indeed, the column of the criterion matrix with the highest value for each row corresponds to the exemplar for the item of that row. Clearly, if a set of rows share a particular exemplar, this amount to say that the objects related to that row are in the same cluster.

Notice that $s(i, i)$ is called preference value. The number of identified exemplars is influenced by the values of the input preferences emerging also from the message-passing procedure. At the beginning, if all the objects are equally suitable as exemplars, the preference value is set to the same value. However this value can be changed in order to produce a different number of clusters.

For the purpose of protein-protein interaction networks clustering, the preference value was set equal for all data points. It was determined by sampling the interval $[-1; 1]$ uniformly with step size of 0.1 and settling on the preference value that results in the best quality score.

3.6 CFinder

CFinder is one of the first overlapping clustering methods for PPI clustering published in the literature[47, 1]. The standard implementation is divided in two parts, a core part written in C++ used to cluster the input network, and a component written in Java used to visualize the output protein complexes. The original version of the algorithm operates on undirected, unweighted networks.

Taking as input a parameter k, CFinder detect all the k-cliques of the input network. We recall that a k-clique is a complete subgraph of k nodes. It builds a k-clique accessibility graph where two k-cliques are connected if they are adjacent, namely if they share exactly $k - 1$ nodes. It is worth to emphasizing that large values of k correspond to be very strict during the detection of dense regions, and hence to recognize small group with higher density of links.

A detailed description of the algorithm is to be found in [47]. Later on, other

versions of this algorithm have been proposed. One of the improvements added was the substitution of the k-cliques search with the enumeration of the maximal cliques with at least k vertexes of the input network. Indeed, each subset of a maximal clique is also a clique, therefore a maximal clique of size n will be mapped to a connected subgraph consisting of $\binom{n}{k}$ vertices in the k-clique accessibility graph. It is possible to shrink that subgraph into a single vertex that will represent the whole maximal clique without affecting the connectivity properties of the k-clique accessibility graph.

The current version provides the weighted extension of CFinder proposed in [59], but, as the original one, cannot detect overlapped clusters. Unfortunately this variant is computationally more prohibitive than the previous one. Indeed, the reference implementation of CFinder (as downloaded from http://www.cfinder.org on 8 Aug 2010) did not provide a result for the Collins dataset for a conservative setting of $k = 4$ and $I = 0.8$ in 48 hours, therefore we included the unweighted variant in our benchmarks.

3.7 CMC

CMC algorithm [35] is a clustering algorithm specifically designed to find protein complexes in protein-protein interactions network. It assesses the probability that two proteins are in the same protein complex using an iterative scoring algorithm followed by a maximal clique finding process. As most of the clique-based algorithms it is not able to handle weighted networks. However it allows as output overlapped protein complexes.

During the search process the cliques found are properly merged with the aim of building the final set of protein complexes. Two cliques are considered sufficiently overlapped using a certain overlap threshold, while a merge threshold determines when two cliques have to be merged together. It is worth emphasizing that two cliques are merged when the network between them is denser than the merge threshold. In the opposite case the smaller clique is discarded.

Both parameters can assume value beween zero and one. Obviously a low overlap threshold implies the detection of only few big protein complexes, while

high one will result in a large number of redundant complexes, with the degenerate case where none of them is able to merge with others

As result, the tested range of the overlap threshold was limited to real values between 0.2 and 0.8, sampled with a step size of 0.1. The merge threshold was tested on uniformly sampled real values between 0 and 1 with a step size of 0.1. The benchmarks have been obtained using the original implementation of the CMC software (version 2). According to the suggestions of the authors of the algorithm [35], a size limit of 4 was used instead of the default size limit.

3.8 MCODE

The MCODE algorithm, known as *M*olecular *C*omplex *D*etection [5] is a clustering algorithm for protein complexes. As the most of the algorithms exploiting the clique notion, it cannot handle weighted input networks. However it is able to detect overlapped protein complexes. The algorithm consists of three phases:

- vertex weighting;
- protein complex formation;
- post-processing.

During the first phase a particular score is assigned to each vertex with the aim of measuring the "clique-ness" of its neighborhood.

In the second step, starting from the node with highest degree, a protein complex is grown from each node. This growth process, used to establish protein complexes, is regulated by a parameter called *depth limit*, which regulates how far it has to continue from the seed node to the others. MCODE controls how much difference is allowed between the scores of each node in a particular complex using another parameter, the *vertex weight percentage*.

The post processing step applied at the end of the algorithm is divided in two complementary operations: the *haircut* and the *fluffing*. Given a certain protein complex, the *haircut* iteratively removes nodes that are connected by a single edge to the rest of protein complex, conversely, the latter tries to expand it with nodes outside the clusters highly connected with it.

Unfortunatelly, even if MCODE produces overlapping complexes during the fluffing phase, our experiments have shown that the algorithm performs better when fluffing is turned off.

All the possible combinations of the following parameters have been tried:

- Depth limit: 3, 4, 5
- Vertex weight percentage: 10% to 50% in steps of 5%
- Haircut: on or off
- Fluffing: on or off
- Fluffing percentage: 0, 10% or 20%

3.9 RRW

The clsutering algorithm *R*epeated *R*andom *W*alk[38], also known as RRW, is able to handle weighted and unweighted graph, allowing the detection of overlapped clusters. Starting from a node and using an affinity function, it grows a cluster.

The basic idea behind RRW is the following. Given a cluster of nodes the algorithm tries to expand it with the aim of including proteins with high proximity to that cluster. Random walks with restart are used to find the set of proteins near a certain cluster.

RRW takes as input a parameter k that regulate this process. Indeed, starting from a cluster of size one it iterates this expansion at most k times or until a stopping condition is reached. Usually, the stopping condition is related to the number of nodes in the cluster, allowing cluster of size $\leq k$.

The process presented above is applied to all the nodes and it is followed by a rank step which remove some cluster. Indeed, the clusters with a high overlap score are post processed with the aim of removing clusters with an overlap above a given threshold.

The RRW algorithm requires to specify the restart probability of the random walk at each step and two threshold parameters, the overlap threshold and the early cutoff. An other two parameters required are the minimum and maximum

size of the clusters, however it is worth to emphasize that the authors, in their original publication, recommend a maximum cluster size equal to eleven.

Therefore, the maximum size has been set equal to 11, tuning the remaining parameters by trying all possible combinations of the following values:

- Restart probability: 0.5 to 0.9 in steps of 0.1
- Overlap threshold: 0.05 to 0.3 in steps of 0.05
- Early cutoff: 0.5 to 0.9 in steps of 0.1

Chapter 4

Performance Evaluation

4.1 Introduction

In this chapter we shall present one of the most important steps that need to be addressed during the application of a clustering algorithm on Protein-Protein Interaction networks: the performance evaluation. Indeed, the quality assessment of the results obtained by a clustering algorithm is substantially different for each application.

Depending on the application the performance evaluation process can involve standard measures or set of images for image segmentation or pattern recognition. While in classic computer science applications standard methods to approach this problem exist, in computational biology often there is no standard way to proceed.

As a consequence all the methods and measures used to assess the quality of the predicted protein complexes have to take into account the major problem which afflicts computational biology data: the lack of a complete knowledge.

As we shall see, the partial understanding that we have of all the mechanisms present in a living cell, in particular of protein-protein interactions, forces us to develop specific techniques to deal with this deficit.

4.2 The performance evaluation problem

Assessing the performance of a clustering algorithm is one of the key steps during its application. The general term "performance" is usually used to refer to a set of parameters regarding the quality of an algorithm: output quality, time and memory cost, etc.

However, in this work, the performance evaluation of a clustering algorithm for protein complex detection corresponds to assessing the accuracy of the predicted complexes. Indeed, it is worth emphasizing that often, in computational biology and bioinformatics, the time and the memory cost of a program is not as crucial as the accuracy of the results. Usually, if the execution time is reasonable, biologists are more interested on the results' precision, rather than on the time needed to reach them.

Another important difference between the performance evaluation of an algorithm applied to classical computer science scenarios and one applied in a computational biology scenario, is the way used to proof the quality of the results obtained. For example, standard datasets of images are usually applied with the aim of assessing the quality of the results achieved by a clustering algorithm. Sometimes, the roughly quality of an image segmentation process is quite clear just looking at it.

Unfortunately, assessing the quality of biological data is, in most of the cases, not so trivial. In this sense, the lack of a complete groundtruth represents one of the biggest problems to validate an algorithm's results[32]. Indeed, as pointed out in 1.4, it is quite common to have only partial information concerning the biological system of organism.

Hence, not all the proteins taken as input from a clustering algorithm for PPI are present in the gold standards used. The overlap between the gold standards and the PPI networks available is therefore only partial. Having only partial information it becomes necessary to develop measures and methods which take into account this lack of knowledge during the process of performance evaluation.

If, for example, a predicted protein complex is not contained in the available gold standards this does not mean that the predicted complex is necessarily wrong.

Furthermore, methods to evaluate a predicted complex with only partial overlap with some reference complex are necessary.

In the next sections we shall see the gold standards used, the measures and methods applied to assess the the quality of the protein complexes predicted.

4.3 Quality Measures for protein complexes

To assess the performance of all the clustering algorithms used in this work we needed to compare a set of predicted complexes with a set of gold standard protein complexes. Unfortunately, as pointed out before, the match between a predicted complexes and a gold standard is often only partial. This is one of the main issues arising during their comparison. Furthermore, the proteins in a gold standard complex can match proteins contained in more than one predicted complex and vice versa.

As proposed in [44] we used three independent quality measures to assess the similarity between a set of predicted complexes and a set of reference complexes:

- the fraction of protein complexes matched by at least one predicted complex;
- the geometric accuracy measure [35];
- the Maximum Matching Ratio [44].

It is worth emphasizing that all these measures try to assess the quality of a protein complex predicted comparing it with the protein complexes present in a specific gold standard. Hence, also if there is more than one gold standard in literature, it is not possible to establish the quality of a predicted complex if it does not match, at least partially one gold standard complex.

4.3.1 Fraction

The first measure used is the fraction of pairs between predicted and reference complexes with an overlap score ω larger than 0.25. We chose this value as suggested in [44]. It is worth emphasizing that, with a threshold larger than 0.25, if

two complexes have the same size this implies that their intersection is at least half of the complex size.

Recall that the overlap score between two protein sets A and B is defined as follows:

$$\omega(A, B) = \frac{|A \bigcap B|}{|A||B|} \tag{4.1}$$

4.3.2 Geometry Accuracy

The second measure used is the geometric accuracy, introduced by Brohee and van Helden [11]. It is is the geometric mean of two other measures: the clustering-wise sensitivity (Sn), and the clustering-wise positive predictive value (PPV). Both are based on the confusion matrix $\mathbf{T} = [t_{ij}]$ of the complexes.

Lets consider n reference and m predicted complexes. The element t_{ij} of the confusion matrix refers the number of proteins that are found both in reference complex i and predicted complex j. Moreover, defining n_i as the number of proteins in reference complex i, Sn and PPV are:

$$Sn = \frac{\sum_{i=1}^{n} max_{j=1}^{m} t_{ij}}{\sum_{i=1}^{n} n_i} \tag{4.2}$$

$$PPV = \frac{\sum_{j=1}^{m} max_{i=1}^{n} t_{ij}}{\sum_{j=1}^{m} \sum_{i=1}^{n} t_{ij}} \tag{4.3}$$

It is worth to emphasize that the the clustering-wise sensitivity can be inflated putting every protein in the same cluster. While the positive predictive value can be maximized putting every protein in its own cluster. For these reason these two measures are balanced computing the geometric mean of the clustering-wise sensitivity and the positive predictive value:

$$Acc = \sqrt{Sn \times PPV}$$

4.3.3 MMR

In [44] Paccanaro et al proposed a measure called the maximum matching ratio (MMR) to evaluate a set of predicted protein complexes with respect to a set of reference complexes.

The MMR represents the two sets of predicted complexes as a bipartite graph where in one side there are the predicted complexes and in the other the reference ones. Each node of this graph represents a protein complex and each edge connecting two nodes has a weigh reflecting the overlap between those complexes, where the overlap score between two protein complexes is computed by (4.1).

To assess the quality of the predicted complexes it is necessary to select the maximum weighted bipartite matching on this graph. Namely, a subset of edges such that each predicted protein complex and each reference complex is incident on at most one selected edge. Moreover, the sum of their weights have to be maximal. The value of the MMR is given by the total weight of this particular subset of edges, divided by the number of reference complexes.

Therefore this measure expresses how well the predicted complexes represent the reference complexes. MMR offers an easy way to compare predicted complexes with a gold standard, penalizing all those cases when a reference complex is predicted as two pieces by the clustering algorithm. Indeed, in those cases only one piece, namely only one of the two predicted complexes, can match the respective reference complex.

Motivations for the MMR measure

It is worth emphasizing that a component of the accuracy score tends to be lower if there are predicted complexes which overlap significantly each others. For this reason, clustering algorithms supporting overlapped clusters tends to be penalized, and hence disadvantaged with respect to the others. In the following few sections, we will show in detail this property of the geometric accuracy measure, motivating the use of MMR.

Problems of PPV

Looking at (4.3) is it possible to notice that the value of PPV can be misleading if some proteins in reference complex i appear in either more than one predicted complex or in none of them.

Indeed, as shown in [44], in this case n_i is not equal to the sum of row i in the confusion matrix $\mathbf{T}$. In general, n_i may be larger, smaller or equal to the sum of row i. To be clear we denote the sum of row i with t_{i*}.

Lets us to consider the case when the whole set of reference and predicted complexes is the same. In this case, $t_{ii} = n_i$ for every i. However, if complex a i is overlapped with a complex j there will be other non-zero elements in $\mathbf{T}$, as $t_{ij} > 0$. Nonetheless, these non-zero elements cannot exceed t_{ii}. This amount to saying that in all the cases $max_{j=1}^{m} t_{ij} = max_{i=1}^{n} t_{ij} = n_i$. The Sn and PPV measures are then as follows:

$$Sn = \frac{\sum_{i=1}^{n} n_i}{\sum_{i=1}^{n} n_i} = 1$$

$$PPV = \frac{\sum_{i=1}^{n} n_i}{\sum_{i=1}^{n} t_{i*}} \leq 1$$

The consequence of this fact is that a perfect clustering algorithm, that always returns the reference complexes clustering the input data, may have a positive predictive value lower than a silly algorithm which places every protein in a separate cluster. Indeed, assuming that we have k proteins, and the protein j is a member of the complex c_j, the positive predictive value for such a dummy algorithm is:

$$PPV = \frac{k}{\sum_{j=1}^{k} c_j} = 1$$

Let us to point out a concrete example regarding the MIPS catalog used to evaluate our predicted complexes.

The MIPS catalog contains 1189 unique proteins, hence $k = 1189$, but its total

size, counting also the duplicated protein contained in each complex, is 2451. This amount to saying that $\sum_{j=1}^{k} c_j = 2541$. Hence, the dummy algorithm, which places every protein in a single cluster, will obtain a PPV score equal to $\frac{1189}{2451} = 0.468$. While the algorithm that detects all the complexes in a perfect way, returning as output the MIPS catalog itself, will obtain a PPV score equal to 0.3475. Hence, it results clear that the PPV scores of an algorithms should be interpreted with care.

Moreover, the geometric accuracy measure assumes that influences negatively the protein complexes evaluation for our purposes. Indeed, it explicitly penalizes predicted complexes that do not match any of the reference complexes. Unfortunately, as pointed out at the beginning of this chapter, gold standard sets of protein complexes are usually incomplete [32].

Hence, a predicted complex may not match any reference complex. However, a predicted complex that does not match any reference complex is not necessarily an undesired result, indeed it could still an undiscovered complex. Therefore, trying to optimize the geometric accuracy might lead to do not detect any new complexes from a PPI network.

With the aim of avoiding this problem the MMR divides the total weight of the maximum matching with the number of *reference* complexes. However, using only this measure to assess the quality of a set of predicted complexes, it is necessary to quantify the functional homogeneity of the detected complexes with alternative methods.

Problems of clustering-wise separation

In order to solve some of the problems related to PPV and Sn, Brohee and van Helden [2] suggested the *clustering-wise separation* measure as an alternative metric.

Unfortunately this separation measure is also not suitable for our purposes. The reasons are similar to the ones reported in the previous section, and are related with overlapped clusters.

Lets us present some useful definitions before we introduce the clustering-wise separation measure.

The relative frequencies of the confusion matrix with respect to the marginal row-wise or column-wise sums are:

$$F_{ij}^{r} = \frac{t_{ij}}{\sum_{j=1}^{m} t_{ij}}$$

$$F_{ij}^{c} = \frac{t_{ij}}{\sum_{j=1}^{n} t_{ij}}$$

The *separation* of predicted complex i and reference complex j is then given by:

$$Sep_{ij} = F_{ij}^{r} F_{ij}^{c}$$

The *complex-wise* and the *cluster-wise separation* scores are then calculated for the whole set of references and predicted complexes as:

$$Sep_{co} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} Sep_{ij}}{m}$$

$$Sep_{cl} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} Sep_{ij}}{n}$$

The *clustering-wise separation* is then calculated as the geometric mean of Sep_{co} and Sep_{cl}:

$$Sep = \sqrt{Sep_{co} Sep_{cl}}$$

In order to underline the limits of this measure for our problem, we remark its interpretation quoting Brohee and van Helden [2]:

"The maximal value $Sep_{ij} = 1$ indicates a perfect and exclusive correspondence between complex j and cluster i: it indicates that the cluster contains all the members of the complex and only them"

Unfortunately, in a similar manner of PPV, this measure presents a problem when the gold standard and/or the set of predicted complexes contains overlapped protein complexes.

Taking the same example reported before, we can compute the clustering-wise separation of the MIPS catalog with themselves. It is equal to 0.3260, and even if it is correct because the MIPS complexes are not well separated, it proposes a misleading result. Indeed, the MIPS complexes match themselves perfectly, and according with what quoted above it should be equal to 1.

We can notice that the MMR measure is, in some sense, similar to the clustering-wise separation score. It also starts by calculating a single quality score for every reference-predicted complex pair, the match score. However in a next step it finds the maximum matching between reference and predicted complexes without penalizing overlaps.

Chapter 5

Experimental Results

5.1 Implementation Used

We have previously defined the notion of Dominant Sets. There are several ways to approach the problem of finding them; it is possible to tackle it from a combinatorial point of view, or to transform it into a purely optimization problem. Moreover, we saw also that the latter corresponds to a game where Dominant Sets are derived as a result of the competition of individuals playing a game involving two players. From this perspective each player simultaneously selects an object from the set of input objects and receives a payoff according to the graph similarity matrix. Hence the payoff depends clearly on the similarity of the selected objects. Moreover, remembering that clusters are sets of objects with high mutual similarity, the players are inclined to select objects belonging to a common cluster. Hence, the competition between the competitors induces the players to learn a common notion of cluster by reaching an equilibrium by the respective hypothesis of cluster membership. As already said, with the purpose to reach an equilibrium we play the clustering game in an evolutionary setting.

As a first attempt we applied the replicator dynamics reported in the previous chapter in a peel-off approach. Running this replicator equation directly on the input similarity matrix we obtain a dominant set for each step. At each step the nodes related to the equilibrium reached are removed, and the dynamics are run

on the submatrix related to the remaining nodes. Clearly following this way the output corresponds to a partition of the input nodes.

Algorithm 1 Patitional clustering algorithm

```
1: procedure PATITIONAL_CLUSTERING(G = (V, E, w))
2:     P ← ∅
3:     while V ≠ ∅ do
4:         S ← Dominant_Sets(G)
5:         P ← P ∪ {S}
6:         V ← V\S
7:     end while
8: end procedure
```

However approaching the detection of protein complexes from Protein-Protein Interaction Network by partitioning the input set of proteins leads to a set of clusters without any overlap. While this lack could be sometimes negligible for the measure indexes used to assess the quality of protein complexes, it is a considerable shortage from a biological point of view. Not allowing proteins to belong to more than one cluster the output clusters not resembling the real biological structure of protein complexes.

A pairwise clustering approach based on dominant sets and replicator dynamics that allows overlapping clusters has been proposed by Torsello et al. in [62]. In their work, in order to enumerate all the dominant sets, the authors iteratively render unstable the ESS reached at each step. This can be achieved adding new strategies that are best replies to the extracted ESSs. Unfortunately, replicator dynamics, as more or less the other standard evolutionary dynamics, turn out to be computationally inefficient. This is probably due to the fact that in literature these dynamics have been used on games with typically a small number of strategies. Unluckily clustering Protein-Protein Interaction networks may involve thousands strategies and hence efficiency cannot be overlooked. If the number of nodes is quite high, it requires a long time in order to converge, and the computational cost to reach a new ESS spreads over time because at each iteration new strategies are added.

For these reasons we applied an implementation of a recent class of evolutionary game dynamics, proposed in [13, 14] and inspired by infection and immunization

processes, that needs a linear number of iterations with respect to the number of strategies.

5.2 Refining of Dominant Sets to Protein Complexes

As pointed out before in order to detect Dominant Sets we used the implementation of the evolutionary dynamics inspired by infection and immunization processes proposed in [13]. Unfortunately this implementation looks for clusters that follow the Dominant Sets definition in very a strict way. We recall that Dominant Sets generalise the classical graph-theoretic notion of a maximal clique to edge-weighted graphs. Hence the Dominant Sets found tend to resemble the clique structure. However, the structure of protein complexes is a clique only from an ideally point of view. Indeed, a high number of false positive and false negative links, or unreliable weights, occurs in PPI network[10]. Therefore looking for structures that look exactly like cliques leads to detecting inaccurate clusters that often are subset of real protein complexes.

Observing the protein complexes detected we noticed a large amount of complexes composed by a single protein. Going forward when we talk about dominant sets of size one we will often call them singletons. From the point of view of the implementation used each of these nodes represent a cluster, namely they cannot form a coherent group with any other nodes. This leads to protein complexes showing low quality for each of three quality measure used. This phenomena is appreciable in the results reported in the figures 5.1 and 5.2.

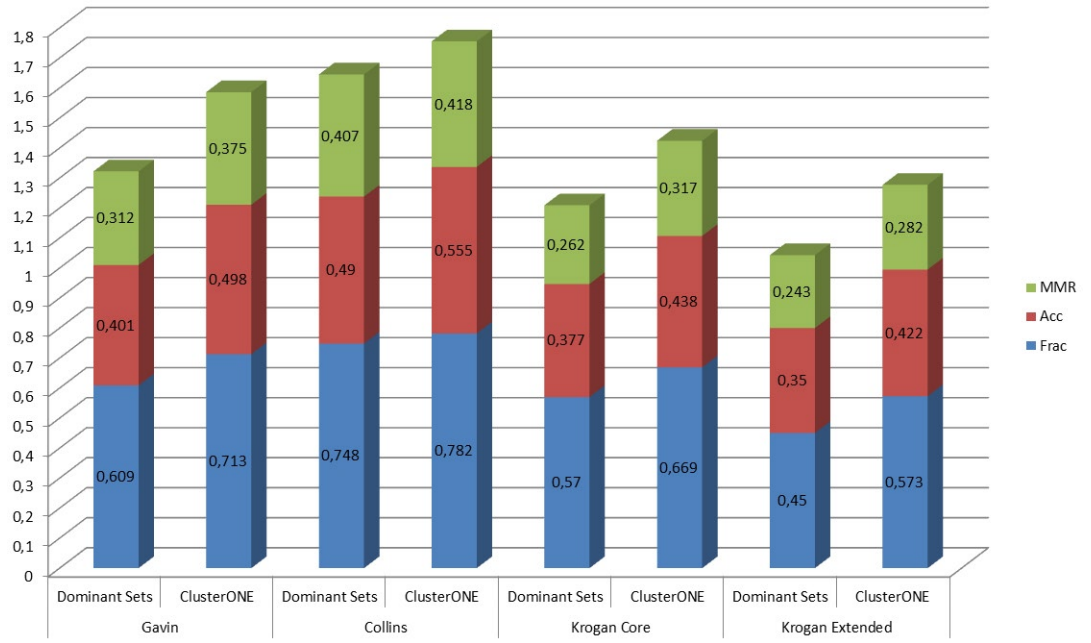

Figure 5.1: Quality of predicted complexes by Dominant Sets and ClusterONE with respect to the MIPS gold standard

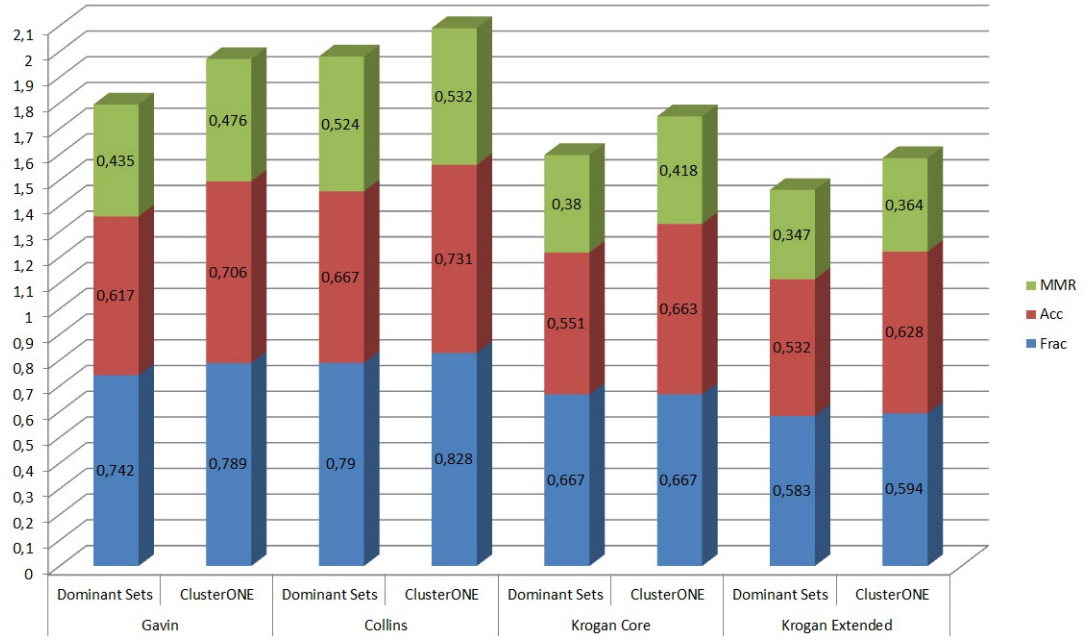

Figure 5.2: Quality of predicted complexes by Dominant Sets and ClusterONE with respect to the SGD gold standard

In order to relax the strict notion of dominant sets we introduced a further step of refinement. Since the notion of dominant set does not correspond to the one of protein complex, after the first step, where the dominant sets of the input network are found, it is necessary to determine which of them have to be promoted to protein complexes. The idea consists in refining the Dominant Sets found adding to the appropriate clusters the singletons.

It is worth emphasizing that joining dominant sets of size one to other clusters does not leads to loosing any significant clusters. Indeed, clusters with size one are meaningless in our biological context, because a protein complex is by definition an aggregate of more than one protein. Moreover, we recall that all the other algorithms applied discard the protein complexes of size less than three. This is a quite common approach in literature and is due to the fact that protein complexes of size two are intrinsically hard to detect.

However, we have not yet defined a suitable and meaningful way to implement this refinement step. Indeed, in order to add dominant sets of size one to other clusters, it necessary to find out how to determine the appropriate clusters, and what is the appropriate way to join them. The basic idea of our approach is to measure iteratively the cohesiveness [44] of the dominant sets, and then, in order to increase it, we try to add a singleton to each of them .

We recall that the cohesiveness of a set of vertexes V is given by

$$f(V) = \frac{w^{in}(V)}{w^{in}(V) + w^{bound}(V)}$$

where $w^{in}(V)$ denotes the total weight of edges contained entirely by a group of proteins V, and $w^{bound}(V)$ denotes the total weight of edges that connect the group with the rest of the network. Clearly a subgraph with many reliable edges has a high w^{in}, and a well-separated subgraph has a low w^{bound}, both having the effect of increasing $f(V)$.

Therefore cohesiveness provides an easy and efficient way to assess if the singleton that we are trying to add to a given dominant set is increasing or decreasing the cohesion of that particular dominant set. Applying iteratively this procedure it is possible to refine the dominant sets adding to them protein complexes that otherwise we should discard.

However, measuring only the cohesiveness of a set of vertexes before and after the addition of an extra node, we have no way of knowing how many links connect the singleton to the set of vertexes. In this way all the nodes linked by few links to our dominant sets will be included in it. Lets consider the following scenario: a set of vertexes V which the total sum of edges weight contained entirely in V is 4, and an extra vertex connected to only one of those nodes by a weak link (figure 5.3). In this case $f(V) = \frac{4}{4+\epsilon}$ while $f(V \cup \{singleton\}) = \frac{4+\epsilon}{4+\epsilon}$, hence

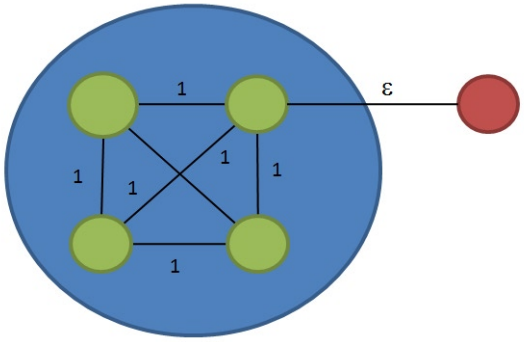

Figure 5.3: Example of join between a dominant set and a singleton

$f(V) < f(V \cup \{singleton\})$. Therefore cohesiveness suggest us to add the extra vertex to V. Nonetheless, it is worth emphasizing that V is a coherent group in which all the nodes are highly similar (e.g similarities equal to 1) while the singleton is connected to only one of those nodes by a very weak link that represents the low similarity between them. In this particular case the cohesiveness is deceived by that fact that the extra vertex that we are trying to add is isolated except for the link to the dominant set.

For this reason we set a threshold of minimum connections needed between the extra vertex and a dominant set. The threshold was empirically determined to be around the 30%. This means that, before proceeding with the cohesiveness test, we check if the singleton that we are trying to add to a dominant set is connected at least to 20% of its nodes. If both of the requirements are satisfied the dominant set is promoted to protein complex.

However, it is worth to emphasize that the Dominant Set originally found cannot be discarded or modified as described so far without keeping into account their importance. Indeed, to be a Dominant Sets is a strict requirement, and in

some way it suggest a core of proteins which with high probability work together in a complex. We recall that proteins often contribute to more than one complex, and that sometimes complexes acts like gears that a biological system reuse for some particular function into other complexes. By the same token we decide to not modify directly detected the Dominant Sets as the first step, but rather to automatically promote all of them to protein complexes adding new complexes based on the latter except for the addition of a certain number of singletons.

Following the procedure pointed out before we significantly improved the quality of the predicted protein complexes, as shown in figure 5.4 and 5.5.

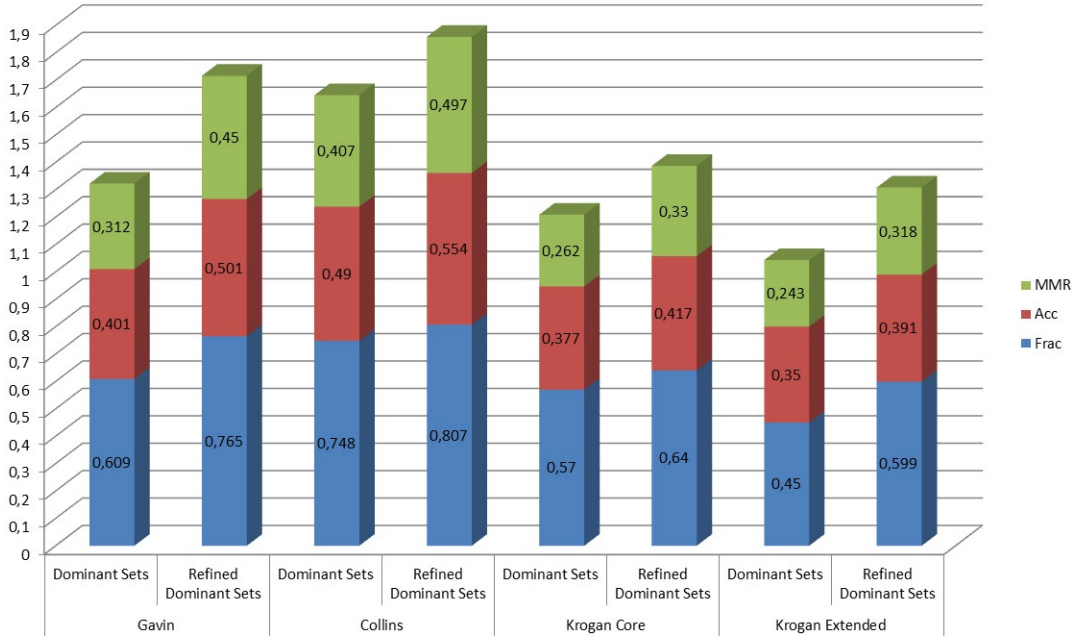

Figure 5.4: Quality of predicted complexes by Dominant Sets and Refined Dominant Sets with respect to the MIPS gold standard

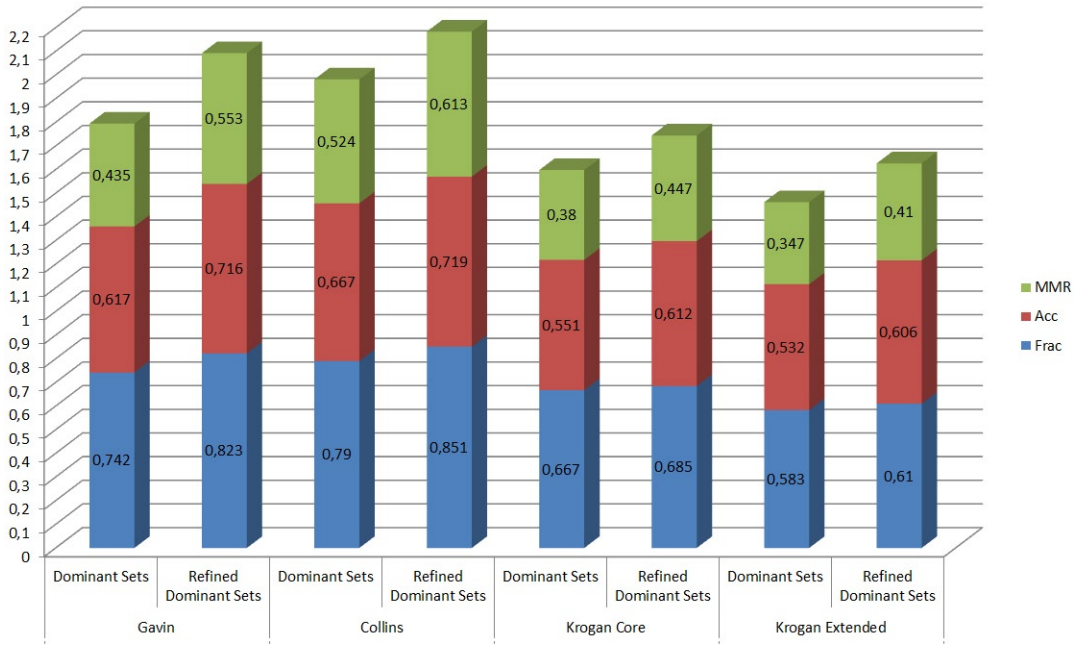

Figure 5.5: Quality of predicted complexes by Dominant Sets and Refined Dominant Sets with respect to the SGD gold standard

5.3 Testing - General Consideration

One of the most known procedure for evaluating the performance of a machine learning algorithm starts by dividing the data into a training and a test set. The parameters of the algorithm are tuned on the training set, and the optimal settings are then used to establish the performance score on the testing set.

It turns out that is possible to apply this approach only if the input data can be split into problem instances such that:

- each instance is a complete input for the learning algorithm on its own
- each instance is independent from the others

Unfortunately, for graph clustering algorithm neither of these two assumptions hold. Moreover, in biological contexts, where the input dataset is a single biological network, removing a fraction of edges from a network would change its structural properties. Indeed, biological networks cannot be easily decomposed

and the attempt to split them could substantially affect the outcome of the clustering algorithm.

Removing edges from a network is dissimilar to removing a set of problem instances from the input dataset in order to put them in the testing set. In some sense it is similar to adding noise to a feature vector in a standard machine learning algorithm.

Evaluating the performance of clustering algorithms on graphs is therefore a tricky problem. Nevertheless, it is important to avoid the typical biases in the evaluation of a well-known method applied to a new scenario. Substantial care should be taken in order to avoid over-optimization of algorithm parameters to a given dataset or to a given quality score[9].

To this end, we have decided on the following:

1. Each of the algorithms have been tested on five different datasets: three high-throughput experimental datasets [22, 34], a computationally derived network that integrates the results of these studies [15], and a compendium of all known yeast protein-protein interactions [58];

2. We used more than one quality score to assess quality of each set of protein complexes detected by the relative algorithm: the fraction of matched complexes with a given overlap score threshold, the geometric accuracy [11] and the maximum matching ratio[44];

3. We used two different gold standards: the MIPS compendium of protein complexes [41] and a set derived from the Gene Ontology annotations of the Saccharomyces Genome Database [29];

4. For each clustering algorithm tested, except for Dominant Sets, we tuned the input parameters in order to achieve the best performance for protein complexes detection in Protein-Protein Interaction Network. This procedure has been conducted in a separate way for each measure, for each dataset and with respect to each gold standard. In this way we obtain different input parameters for each scenario, with the purpose to obtain the most optimistic score. On the other hand, we avoid this approach for Dominant. Namely our results were obtained without tuning the parameters. Hence, the

score of Dominant Sets represent its real performance when it is adapted for detecting overlapping protein complexes form high-throughput experimental PPI network, while the scores of generic clustering algorithm used measure their performance when they are optimized on a specific dataset with respect to a specific gold standard.

5.4 Parameter settings for each algorithm

We establish the parameters running each algorithm several times with a different combination of settings unless the authors of the original algorithm suggested some particular settings for detecting protein complexes in Protein-Protein Interaction Network. The set of parameter combinations tried for each algorithm and its details is reported in the relative section of chapter 3.

It is worth emphasizing that, since the MIPS gold standard and the SGD gold standard are not entirely consistent with respect to the membership of some proteins in some complexes, we decided to test these two gold standards separately as done in [44].

It is quite interesting to notice that, as we will show in the section 5.5, the optimal parameter values for non-overlapping algorithms seem to vary wildly between dataset. Indeed, parameters that work well for a given PPI network may not be suitable at all for others. Algorithms that allow overlapping clusters seem to show more stable performance if their parameters remain within a certain range.

As we will see the classical clustering algorithms tested use often several input parameters. If this is not so relevant to the purpose of comparing our algorithm with the others it is a remarkable lack of flexibility. Indeed, in a real scenario it is not possible to asses the quality of the results in order to tune the input parameter.

One of the advantages of Dominant Sets approach is its needing of only one input parameter, which depends on the size of the network.

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Preference	-0.9	0.35	0.4	-0.15	-0.15

Table 5.1: Affinity Propagation parameter settings for MIPS

5.4.1 The MIPS Gold Standard

Affinity Propagation

CFinder

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
k-clique template size	3	3	3	4	N\A

Table 5.2: CFinder parameter settings for MIPS. N/A for the BioGRID dataset indicates that even the unweighted CFinder implementation did not give any result within 24 hours.

CMC

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Overlap threshold	0.7	0.7	0.7	0.7	N\A
Merge threshold	0.5	0.4	0.5	0.5	N\A

Table 5.3: CMC parameter settings for MIPS. N/A for the BioGRID dataset indicates that the algorithm produced a prohibitively large number of clusters (more than 6000) for all parameter settings we have tried

MCODE

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Depth limit	3	3	3	3	3
Vertex weight percentage	20%	20%	10%	10%	10%
Fluff complexes	no	no	no	no	no
Fluff threshold	N\A	N\A	N\A	N\A	N\A
Haircut complexes	yes	yes	yes	yes	yes

Table 5.4: MCODE parameter settings for MIPS

MCL

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Inflation	4.9	2.3	2.3	3.2	3.3

Table 5.5: MCL parameter settings for MIPS

RNSC

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Shuffling diversification length	5	9	9	9	9
Diversification Frequency	50	20	20	20	20
Number of experiments	3	3	3	3	3
Naive stopping tolerance	10	10	20	20	20
Scaled stopping tolerance	5	5	1	5	5
Tabu length	100	10	50	100	10
Tabu tolerance	1	3	1	5	1

Table 5.6: RNSC parameter settings for MIPS

RRW

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Restart probability	0.5	0.5	0.5	0.6	0.9
Overlap threshold	0.2	0.2	0.2	0.1	0.2
Early cutoff	0.5	0.6	0.7	0.6	0.6

Table 5.7: RRW parameter settings for MIPS

ClusterONE

Even if ClusterONE allows a large set of parameters it was designed explicitly for detecting overlapping protein complexes form high-throughput experimental PPI

network. Its authors, in [44], suggest the default parameters of their implementation. Since their benchmarks used the same datasets and the same gold standards used in this work, we applied this algorithm without adjusting any parameters. Hence the merging and the density threshold ware left to 0.8 and 0.3 respectively.

However it is worth emphasizing that, for unweighted networks, the implicit parameter density threshold is automatically set to 0.6 or 0.5 depending on the network transitivity. This is due to the fact that some datasets are obtained with particular methods used to detect protein-protein interaction, while others are obtained by a mixture of these methods. In our case for example while the datasets Collins[15], Krogan[34] and Gavin[22] have been built by TAP tagging experiments only, the BioGRID dataset [58] contains a mixture of TAP tagging and Y2H low-throughput experimental results. This makes the latter network structurally very different, with a high fraction of star-like structures. Differences in structural properties of these networks are pointed out in [44].

Counting the probability of triangles given three proteins connected by at least two edges is it possible to quantify this phenomena. This measure is known as transitivity or global clustering coefficient. In other words, transitivity express the probability of finding a third edge among triplets of proteins where at least two of the possible three connections exists. Hence, if the network contains many star-like structures the transitivity has a low value. In that cases, in order to discard trivial clusters, ClusterONE uses a high value for the density threshold.

Dominant Sets

The implementation of Dominant Sets used in this work takes only one input parameter. This parameter is called tolerance and it regulates the convergence process. In particular when the solution reached in a particular iteration differs less then this parameter from the one reached at the previous iteration the convergence process is interrupted. In practice this process is related to the precision of the clusters detected.

Generalising the classical graph-theoretic notion of a maximal clique to edge-weighted graphs, Dominant Sets considers compact clusters as accurate. However the notion of protein complexes is not so strict, indeed protein complexes are not

cliques. For this reason we used a relatively high tolerance, interrupting the convergence process quite early. In this way we detect groups of object which represent "roughly cliques" resembling the real biological structure of protein complexes.

In practice this parameter is related to the size of the network. Indeed, the higher the number of nodes in the network the higher is the probability to have an interaction between a pair of proteins. This aspect could lead to detect clusters too large and inaccurate. The relationship between the size of the network and the tolerance parameter allows us to determine the latter simply by looking at the structure of the network.

The values used in our benchmark are reported below. We used these settings also for the SGD standard. It is possible to notice that the higher the number of edges in the dataset the lower the tolerance value becomes.

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Tollerance	10^{-2}	10^{-2}	10^{-4}	10^{-2}	10^{-7}

Table 5.8: Dominant Sets parameter settings for MIPS and SGD

5.4.2 The SGD Gold Standard

Affinity Propagation

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Preference	0.4	0.35	0.3	-0.6	-0.7

Table 5.9: Affinity Propagation parameter settings for SGD

CFinder

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
k-clique template size	3	3	4	4	N\A

Table 5.10: CFinder parameter settings for SGD. N/A for the BioGRID dataset indicates that CFinder did not give any result within 24 hours.

CMC

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Overlap threshold	0.7	0.7	0.7	0.7	N\A
Merge threshold	0.5	0.4	0.3	0.5	N\A

Table 5.11: CMC parameter settings for SGD. N/A for the BioGRID dataset indicates that the algorithm produced a prohibitively large number of clusters (more than 6000) for all parameter settings we have tried

MCODE

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Depth limit	3	3	3	3	3
Vertex weight percentage	20%	20%	10%	10%	10%
Fluff complexes	no	no	no	no	no
Fluff threshold	N\A	N\A	N\A	N\A	N\A
Haircut complexes	yes	yes	yes	yes	yes

Table 5.12: MCODE parameter settings for SGD

MCL

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Inflation	4.6	2.0	2.6	4.7	3.2

Table 5.13: MCL parameter settings for SGD

RNSC

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Shuffling diversification length	9	3	9	9	9
Diversification Frequency	50	20	50	10	10
Number of experiments	3	3	10	3	1
Naive stopping tolerance	50	50	50	20	20
Scaled stopping tolerance	5	5	1	15	15
Tabu length	100	50	50	100	1
Tabu tolerance	1	1	1	1	1

Table 5.14: RNSC parameter settings for SGD

RRW

	Collins	Krogan Core	Krogan Extended	Gavin	BioGRID
Restart probability	0.5	0.5	0.5	0.6	0.9
Overlap threshold	0.2	0.2	0.2	0.1	0.2
Early cutoff	0.5	0.6	0.7	0.6	0.6

Table 5.15: RRW parameter settings for SGD

5.5 Quality of the Predicted Complexes

We tested all eight algorithms presented in the section 4 and the one based on Dominant Sets, on the five large scale yeast PPI datasets presented in section 1.10. In order to assess the quality of the protein complexes predicted we compared them to two reference set: one derived from the MIPS catalog of protein complexes[41], and another from Gene Ontology-based complex annotations in the SGD[44]. For each set of protein complexes predicted we computed three different scores based on three different measures: the fraction of protein complexes matched by at least one predicted complex, the geometric accuracy measure and the maximum matching ratio.

Our results show that the Dominant Sets framework is a suitable candidate to detect protein complexes from protein-protein interaction networks. The quality of protein complexes predicted by the Dominant Sets algorithm from the most reliable datasets is more accurate, than that obtained by ClusterONE -the state of the art for detecting protein complexes. Remarkable results are also achieved for unweighted datasets. Dominant Sets is proven to be quite susceptible to those datasets which are known to be more noisy, where those less accurate weights on the graph's edges have lead to results comparable with those obtained with other standard clustering techniques. However it is important to keep in mind that most of the alternative clustering algorithm cannot deal with overlapped clusters and weighted datasets at the same time.

5.5.1 Quality score by MIPS gold standard

In this section are reported some benchmark results obtained comparing the protein complexes predicted by each clustering algorithm with the gold standard obtained from the MIPS catalog of protein complexes.

For each dataset is reported a plot with all the clustering algorithms in the $x-axis$ and the individual quality scores of the predicted complexes with respect to the MIPS catalog in the $y-axis$. The total height of each bar is the value of the composite score. Numbers are the value for each score. The clustering algorithms are grouped in two main groups in order to separate clustering algorithms that cannot handle overlaps (RNSC, AP, MCL) from those that can.

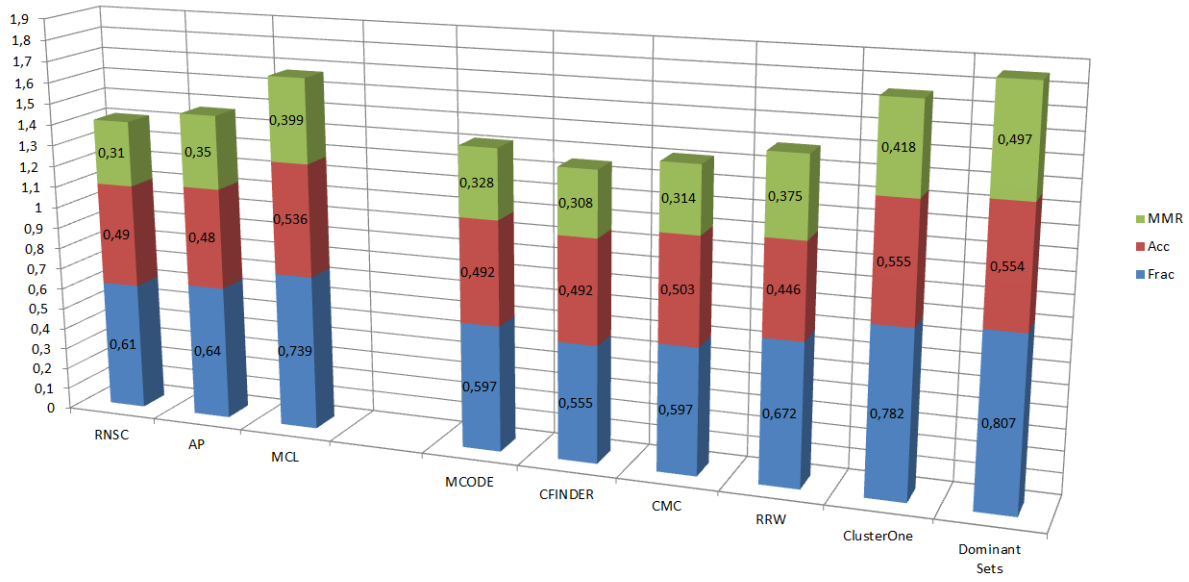

Figure 5.6: Quality of the predicted protein complexes from Collins dataset w.r.t the MIPS gold standard

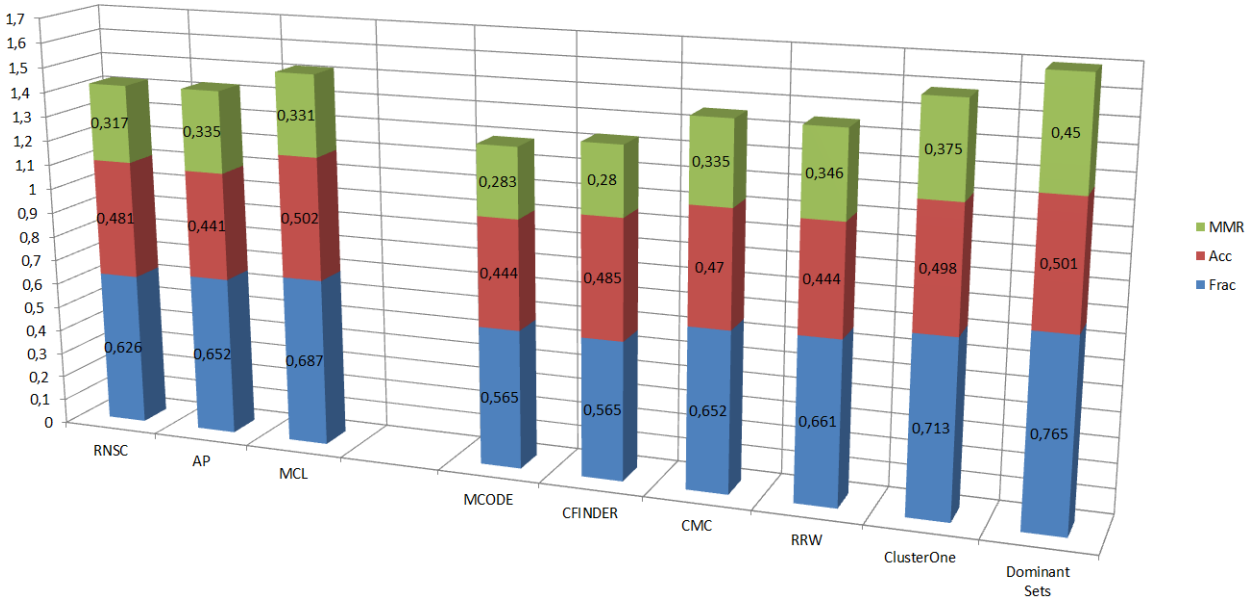

Figure 5.7: Quality of the predicted protein complexes from Gavin dataset w.r.t the MIPS gold standard

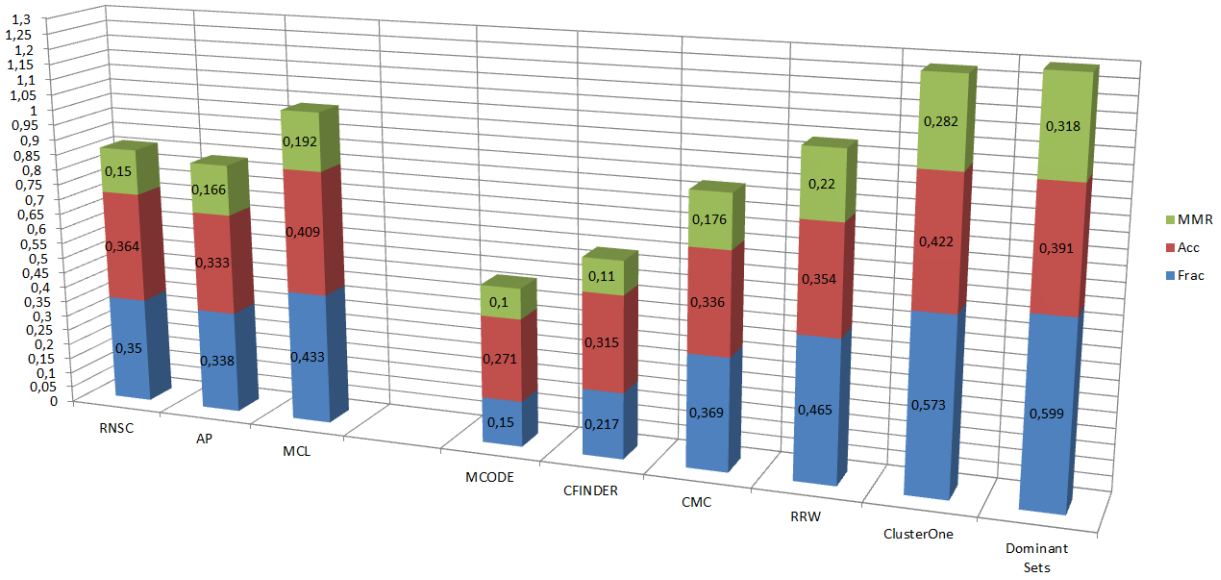

Figure 5.8: Quality of the predicted protein complexes from Krogan Extended dataset w.r.t the MIPS gold standard

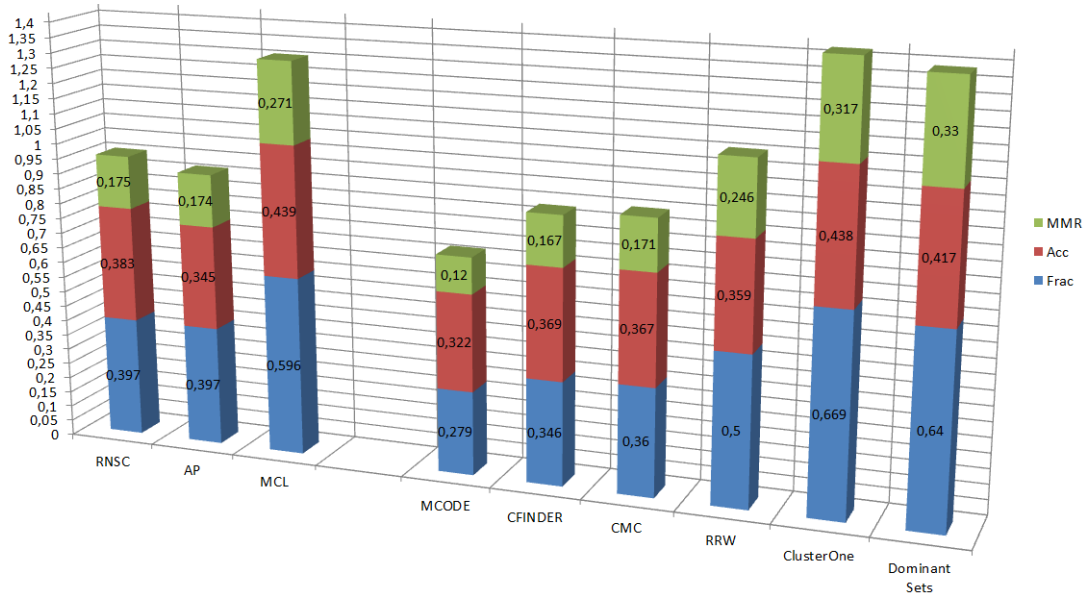

Figure 5.9: Quality of the predicted protein complexes from Krogan Core dataset w.r.t the MIPS gold standard

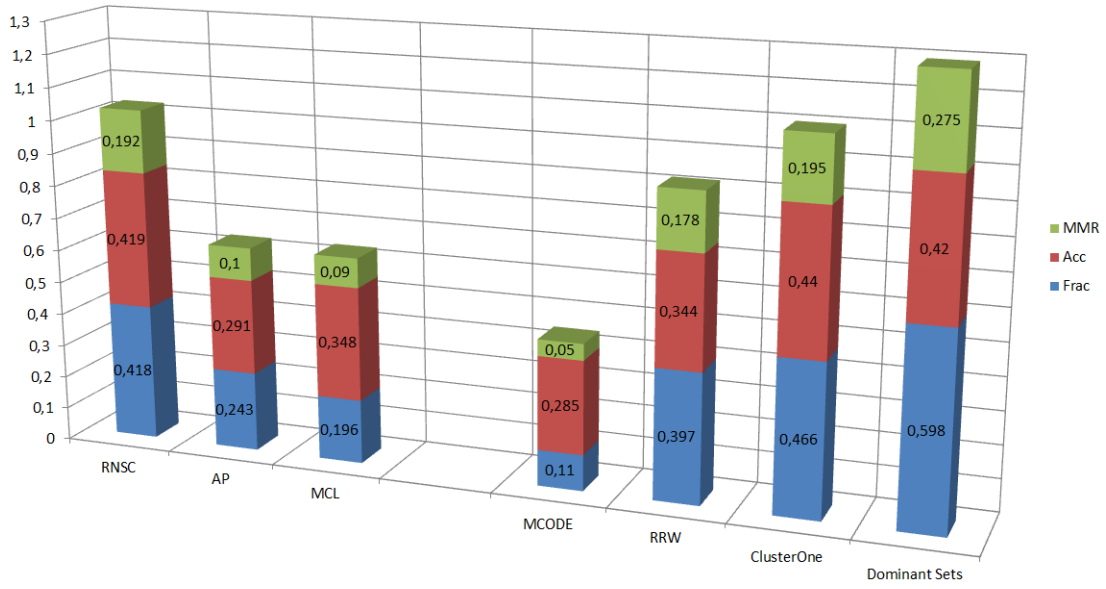

Figure 5.10: Quality of the predicted protein complexes from BioGRID dataset w.r.t the MIPS gold standard

From the previous figures it is clear that the clustering algorithm based on Dominant Sets outperforms the other approaches in almost all datasets.

Consider the composite score obtained by summing all the three quality score as a summary score for the protein complexes predicted by each algorithm. It possible to notice that for all the datasets, except for Krogan Core, the best results are always achieved by Dominant Sets. In particular, we obtained significative improvements for the unweighted dataset BioGRID (figure 5.10). In order to remark this behaviour tables 5.16, 5.17, 5.18 and 5.20 show the composite score values and its percentage increase with respect to Dominant Sets for each approach.

Algorithm	**Frac**	**Acc**	**MMR**	**Composite Score**	***Percentage increase***
RNSC	0.418	0.419	0.192	1.029	+25.7%
AP	0.243	0.291	0.1	0.634	+103.9%
MCL	0.196	0.348	0.09	0.634	+103.9%
MCODE	0.11	0.285	0.05	0.445	+190.6%
RRW	0.397	0.344	0.178	0.919	+40.7%
ClusterONE	0.466	0.44	0.195	1.101	+17.4%
Dominant Sets	0.598	0.42	0.275	**1.293**	

Table 5.16: Quality of the predicted protein complexes from BioGRID dataset w.r.t the MIPS gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	**Frac**	**Acc**	**MMR**	**Composite Score**	***Percentage increase***
RNSC	0,626	0,481	0,317	1,424	+20,5%
AP	0,652	0,441	0,335	1,428	+20,2%
MCL	0,687	0,502	0,331	1,52	+12,9%
MCODE	0,565	0,444	0,283	1,292	+32,8%
CFINDER	0,565	0,485	0,28	1,33	+29,0%
CMC	0,652	0,47	0,335	1,457	+17,8%
RRW	0,661	0,444	0,346	1,451	+18,3%
ClusterONE	0,713	0,498	0,375	1,586	+8,2%
Dominant Sets	0,765	0,501	0,45	**1,716**	

Table 5.17: Quality of the predicted protein complexes from Gavin dataset w.r.t the MIPS gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,61	0,49	0,31	1,41	+31,8%
AP	0,64	0,48	0,35	1,47	+26,4%
MCL	0,739	0,536	0,399	1,674	+11,0%
MCODE	0,597	0,492	0,328	1,417	+31,1%
CFINDER	0,555	0,492	0,308	1,355	+37,1%
CMC	0,597	0,503	0,314	1,414	+31,4%
RRW	0,672	0,446	0,375	1,493	+24,4%
ClusterONE	0,782	0,555	0,418	1,755	+5,9%
Dominant Sets	0,807	0,554	0,497	**1,858**	

Table 5.18: Quality of the predicted protein complexes from Collins dataset w.r.t the MIPS gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,397	0,383	0,175	0,955	+45,2%
AP	0,397	0,345	0,174	0,916	+51,4%
MCL	0,596	0,439	0,271	1,306	+6,2%
MCODE	0,279	0,322	0,12	0,721	+92,4%
CFINDER	0,346	0,369	0,167	0,882	+57,3%
CMC	0,36	0,367	0,171	0,898	+54,5%
RRW	0,5	0,359	0,246	1,105	+25,5%
ClusterOne	0,669	0,438	0,317	1,424	-2,6%
Dominant Sets	0,64	0,417	0,33	1,387	

Table 5.19: Quality of the predicted protein complexes from Krogan Core dataset w.r.t the MIPS gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,35	0,364	0,15	0,864	+51,4%
AP	0,338	0,333	0,166	0,837	+56,3%
MCL	0,433	0,409	0,192	1,034	+26,5%
MCODE	0,15	0,271	0,1	0,521	+151,1%
CFINDER	0,217	0,315	0,11	0,642	+103,7%
CMC	0,369	0,336	0,176	0,881	+48,5%
RRW	0,465	0,354	0,22	1,039	+25,9%
ClusterONE	0,573	0,422	0,282	1,277	+2,4%
Dominant Sets	0,599	0,391	0,318	**1,308**	

Table 5.20: Quality of the predicted protein complexes from Krogan Extended dataset w.r.t the MIPS gold standard and percentage increase of the composite quality score achieved by Dominant Sets

As emphasized by the previous tables, except for Krogan Core, Dominant sets shows an overall quality improvement ranging from 2,4% to 17,4% in protein complexes detection from Protein-Protein Interaction Networks.

In particular, focusing our attention to the single components of this overall quality, results that Dominant Sets outperform all the other clustering approaches in two of three quality measures. Indeed, appreciable improvements concern the fraction of protein complexes matched by at least one predicted complex and the maximum matching ratio (the blue and green measures in all the plots).

The geometric accuracy is instead slightly lower than the one achieved by ClusterONE. In order to explain this behaviour it is important to emphasize that, as explained in details in the section 4.3.2 and 4.3.3, the geometric accuracy is the geometric mean of two other measures, the PPV and the Sn. Unfortunately, one of this two component, the PPV, tends to be lower if a protein appears in more than one predicted cluster. Hence this measure is affected in the wrong way if several predicted complexes are overlapped, puting overlapping clustering algorithms at a disadvantage. For this reason Paccanaro at. el in [44] proposed the MMR. The maximum matching ratio was explicitly designed to assess the quality of overlapped protein complexes.

The results achieved by Dominant Sets show a MMR significantly better than the one obtained by the other approaches. This aspect is crucial and leads to reconsider our results for the better.

Dataset	Algorithm	Frac	MMR	Composite Score	Percentage Increse
Collins	MCL	0,739	0,399	1,138	+14,59%
	ClusterONE	0,782	0,418	1,2	+8,67%
	Dominant Sets	0,807	0,497	**1,304**	
Gavin	MCL	0,687	0,331	1,018	+19,35%
	ClusterONE	0,713	0,375	1,088	+11,66%
	Dominant Sets	0,765	0,45	**1,215**	
Krogan Extended	RRW	0,465	0,22	0,685	+33,87%
	ClusterONE	0,573	0,282	0,855	+7,25%
	Dominant Sets	0,599	0,318	**0,917**	
Krogan Core	MCL	0,596	0,271	0,867	+11,88%
	ClusterONE	0,669	0,317	0,986	-1,62%
	Dominant Sets	0,64	0,33	**0,97**	
BioGRID	RNSC	0,418	0,192	0,61	+43,11%
	ClusterONE	0,466	0,195	0,661	+32,07%
	Dominant Sets	0,598	0,275	**0,873**	

Table 5.21: Frac and MMR of the 3 algorithms that achieved the highest score for each dataset with respect to MIPS gold standard. The percentage increase achieved by Dominant Sets is also reported

The table 5.21 reports for each dataset the three algorithms that achieved the best fraction of protein complexes matched by at least one predicted complex and the maximum matching ratio with respect to the MIPS gold standard. The composite score has been obtained summing these two measures. It is worth emphasizing that, except for Krogan Core, Dominant sets shows for this composite score an improvement ranging from 7,25% to 32,7% in protein complexes detection.

5.5.2 Quality score by SGD gold standard

In this section are reported some benchmark results obtained comparing the protein complexes predicted by each clustering algorithm with the gold standard SGD. All the plots reported in this section use the same format describe in the section 5.5.1.

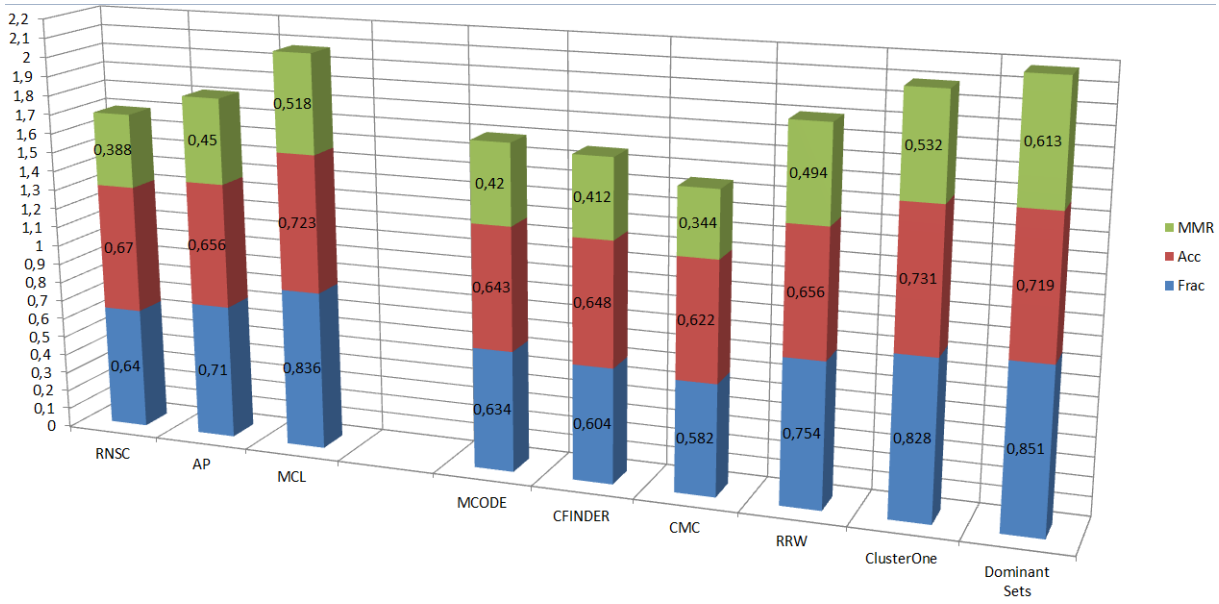

Figure 5.11: Quality of the predicted protein complexes from Collins dataset w.r.t the SGD gold standard

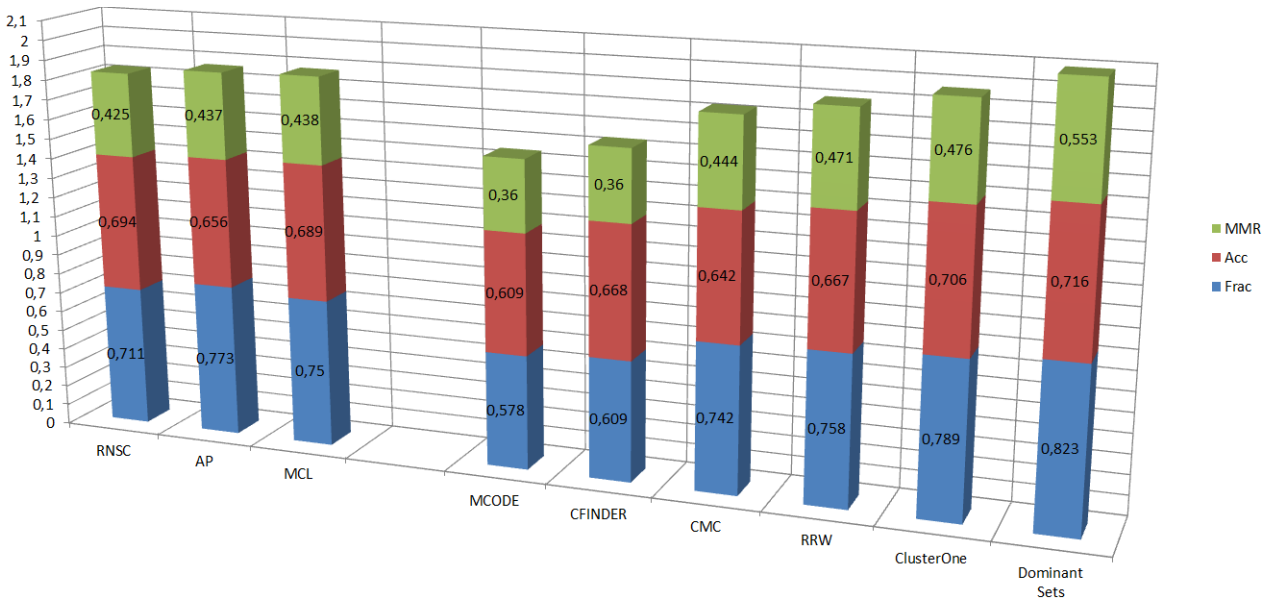

Figure 5.12: Quality of the predicted protein complexes from Gavin dataset w.r.t the SGD gold standard

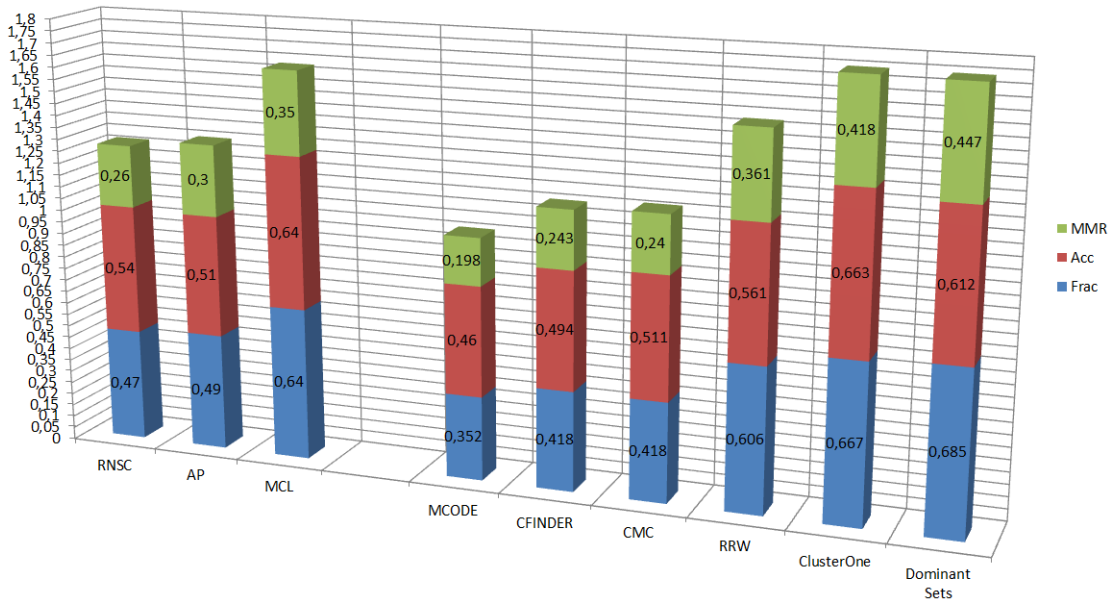

Figure 5.14: Quality of the predicted protein complexes from Krogan Core dataset w.r.t the SGD gold standard

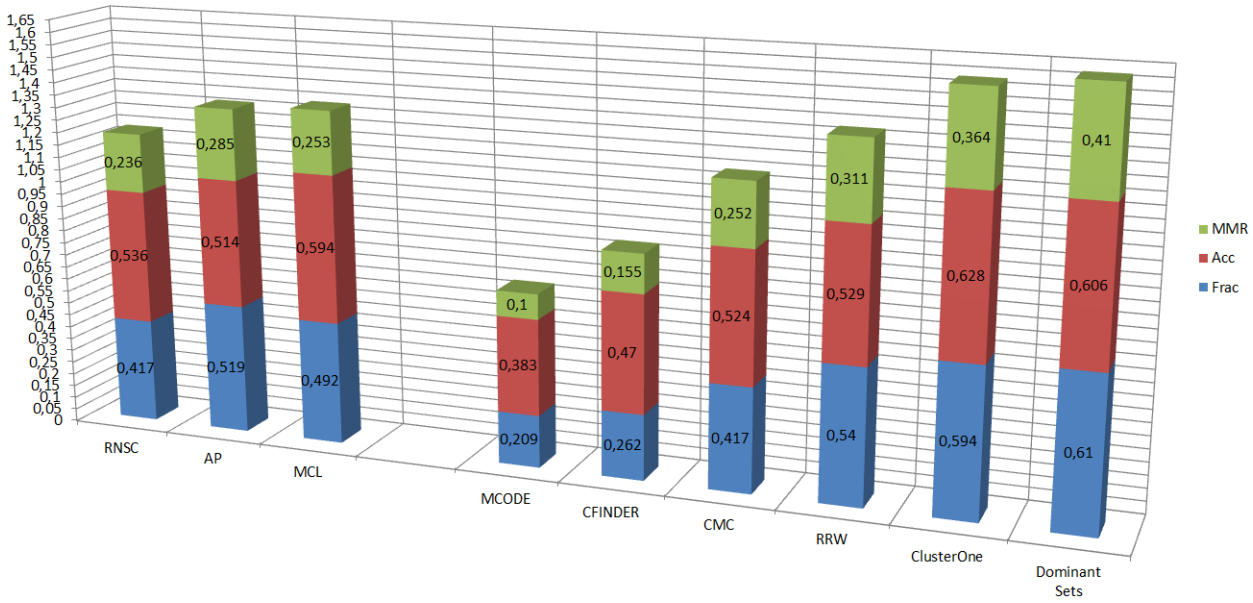

Figure 5.13: Quality of the predicted protein complexes from Krogan Extended dataset w.r.t the SGD gold standard

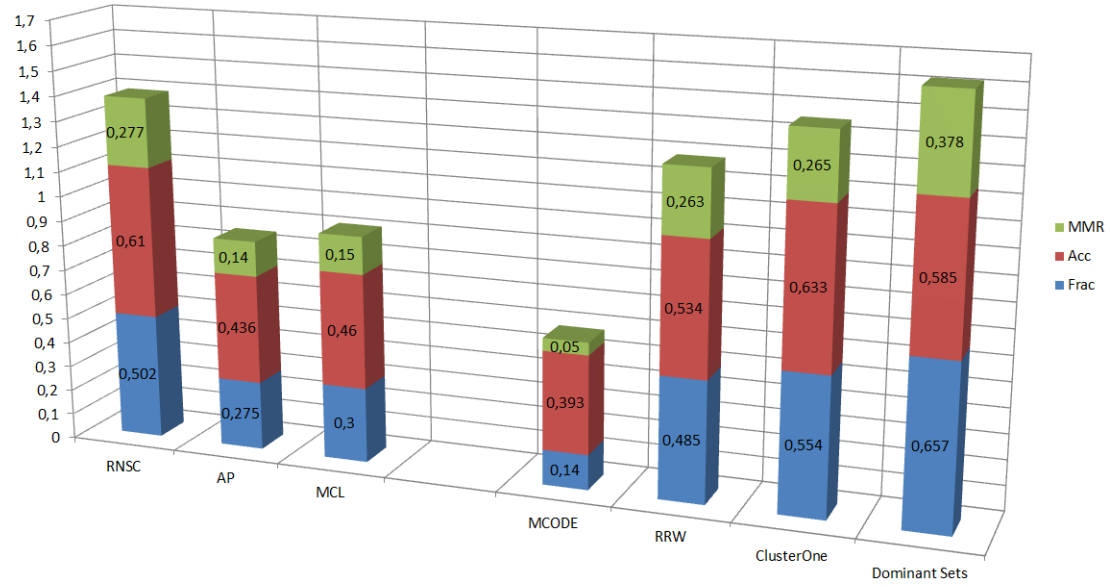

Figure 5.15: Quality of the predicted protein complexes from BioGRID dataset w.r.t the SGD gold standard

The quality of the protein complexes obtained comparing the predicted complexes by each algorithm to the gold standard SGD follows the same trend than

the one shown comparing them to the gold standard MIPS. Moreover, it is worth emphasizing the behaviour obtained on Krogan Core dataset. Recalling that protein complexes predicted from this dataset show an overall quality lower of 2.6% with respect to ClusterOne, we can see that this margin is reduced to -0.2% if we compare our predicted complexes with the SGD gold standard. This lead us to reconsider for the better the disadvantage between our algorithm and ClusterOne on this particular dataset.

Algorithm	**Frac**	**Acc**	**MMR**	**Composite Score**	***Percentage increase***
RNSC	0,502	0,61	0,277	1,389	+16,6%
AP	0,275	0,436	0,14	0,851	+90,4%
MCL	0,3	0,46	0,15	0,91	+78,0%
MCODE	0,14	0,393	0,05	0,583	+177,9%
RRW	0,485	0,534	0,263	1,282	+26,4%
ClusterOne	0,554	0,633	0,265	1,452	+11,6%
Dominant Sets	0,657	0,585	0,378	1,62	

Table 5.22: Quality of the predicted protein complexes from BioGRID dataset w.r.t the SGD gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	**Frac**	**Acc**	**MMR**	**Composite Score**	***Percentage increase***
RNSC	0,711	0,694	0,425	1,83	+14,2%
AP	0,773	0,656	0,437	1,866	+12,0%
MCL	0,75	0,689	0,438	1,877	+11,3%
MCODE	0,578	0,609	0,36	1,547	+35,0%
CFINDER	0,609	0,668	0,36	1,637	+27,6%
CMC	0,742	0,642	0,444	1,828	+14,3%
RRW	0,758	0,667	0,471	1,896	+10,2%
ClusterOne	0,789	0,706	0,476	1,971	+6,0%
Dominant Sets	0,82	0,716	0,553	2,089	

Table 5.23: Quality of the predicted protein complexes from Gavin dataset w.r.t the SGD gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,64	0,67	0,388	1,698	+28,6%
AP	0,71	0,656	0,45	1,816	+20,2%
MCL	0,836	0,723	0,518	2,077	+5,1%
MCODE	0,634	0,643	0,42	1,697	+28,6%
CFINDER	0,604	0,648	0,412	1,664	+31,2%
CMC	0,582	0,622	0,344	1,548	+41,0%
RRW	0,754	0,656	0,494	1,904	+14,7%
ClusterOne	0,828	0,731	0,532	2,091	+4,4%
Dominant Sets	0,851	0,719	0,613	2,183	

Table 5.24: Quality of the predicted protein complexes from Collins dataset w.r.t the SGD gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,47	0,54	0,26	1,27	+37,3%
AP	0,49	0,51	0,3	1,3	+34,2%
MCL	0,64	0,64	0,35	1,63	+7,0%
MCODE	0,352	0,46	0,198	1,01	+72,7%
CFINDER	0,418	0,494	0,243	1,155	+51,0%
CMC	0,418	0,511	0,24	1,169	+49,2%
RRW	0,606	0,561	0,361	1,528	+14,1%
ClusterOne	0,667	0,663	0,418	1,748	-0,2%
Dominant Sets	0,685	0,612	0,447	1,744	

Table 5.25: Quality of the predicted protein complexes from Krogan Core dataset w.r.t the SGD gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Algorithm	Frac	Acc	MMR	Composite Score	*Percentage increase*
RNSC	0,417	0,536	0,236	1,189	+36,8%
AP	0,519	0,514	0,285	1,318	+23,4%
MCL	0,492	0,594	0,253	1,339	+21,4%
MCODE	0,209	0,383	0,1	0,692	+135,0%
CFINDER	0,262	0,47	0,155	0,887	+83,3%
CMC	0,417	0,524	0,252	1,193	+36,3%
RRW	0,54	0,529	0,311	1,38	+17,8%
ClusterOne	0,594	0,628	0,364	1,586	+2,5%
Dominant Sets	0,61	0,606	0,41	1,626	

Table 5.26: Quality of the predicted protein complexes from Krogan Extended dataset w.r.t the SGD gold standard and percentage increase of the composite quality score achieved by Dominant Sets

Dataset	Algorithm	Frac	MMR	Composite Score	Percentage Increse
Collins	MCL	0,836	0,518	1,1353	+8,12%
	ClusterONE	0,828	0,532	1,36	+7,65%
	Dominant Sets	0,851	0,613	**1,464**	
Gavin	RRW	0,758	0,471	1,229	+11,96%
	ClusterONE	0,789	0,476	1,265	+8,77%
	Dominant Sets	0,823	0,553	**1,376**	
Krogan Extended	RRW	0,54	0,311	0,851	+19,86%
	ClusterONE	0,594	0,364	0,958	+6,47%
	Dominant Sets	0,61	0,41	**1,02**	
Krogan Core	MCL	0,64	0,35	0,99	+14,34%
	ClusterONE	0,667	0,418	1,085	4,33%
	Dominant Sets	0,685	0,447	**1,132**	
BioGRID	RNSC	0,502	0,277	0,779	+32,86%
	ClusterONE	0,554	0,265	0,819	+26,37%
	Dominant Sets	0,657	0,378	**1,035**	

Table 5.27: Frac and MMR of the 3 algorithms that achieved the highest score for each dataset with respect to SGD gold standard. The percentage increase achieved by Dominant Sets is also reported

5.6 Concluding Remarks

The results achieved show that the Dominant Sets framework is a suitable candidate to detect protein complexes from protein-protein interaction networks. Our benchmarks showed that the clustering algorithm based Dominant Sets outperformed the other approaches both on weighted and unweighted PPI networks. It matches more complexes and provides a better one-to-one mapping with the reference complexes used in almost all the data sets.

MCL and ClusterONE yielded the closest score to Dominant Sets. However it is important to notice that MCL cannot handle overlaps and ClusterONE, the state of the art for detecting protein complexes, has been designed explicitly for protein complexes detection in Protein-Protein Interaction.

Chapter 6

Assessing the Quality of Putative Dimers

6.1 Introduction

During our experiments the Dominant Sets algorithm showed a high tendency to detect dimers, namely protein complexes of size two. Figure 6.1 compares the quantity of dimers predicted by Dominant Sets and ClusterONE for each datasets.

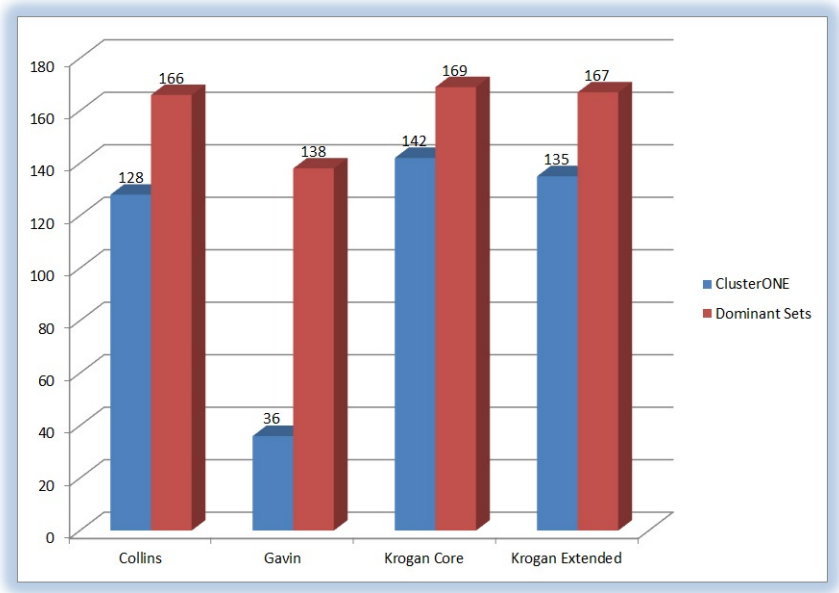

Figure 6.1: Quantity of predicted dimers

The higher number of dimers detected by the Dominant Sets algorithm is of remarkable interest. Indeed, some authors argue that the clustering algorithms currently available have substantial difficulties to recognize protein complexes of size two[27]. Their detection is hampered by the structure itself of this particular complexes.

As pointed out in [27], also the state of the art for detecting protein complexes in PPI networks underestimates the number of dimers. This is due to the fact that this particular type of complex has only one edge connecting two proteins, and in some sense there is a lack of association evidence that makes their extraction form a PPI network very difficult. The figure 6.2 shows the complex size distribution detected by ClusterOne, highlighting the underestimation of complexes of size two and three.

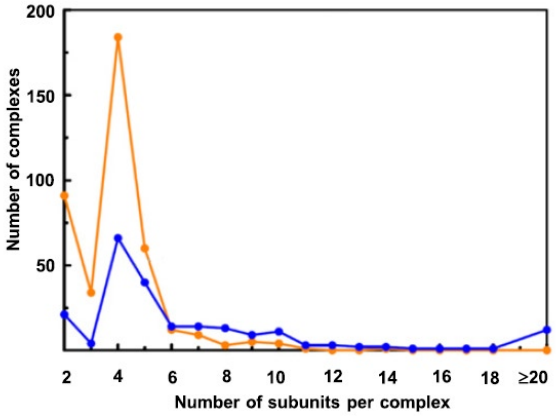

Figure 6.2: Complex size distribution. The blue line represents the complexes curated in literature, while the orange line represent the novel complexes predicted using ClusterOne in [27]

This particular kind of complex has important biological roles but finding and validating them is still an open problem. To the best of our knowledge, this is the first attempt at defining a method with the express aim of finding dimers, which raises the need to define an evaluation methodology to validate the putative dimers produced by Dominant Sets.

6.2 Dimers Evaluation through Gold Standards

As a first attempt for evaluating a set of putative dimers predicted by any clustering algorithm we tried to quantify how many of them are known dimers, and hence included in some protein complex gold standard. For this step we used the gold standards reported in the table 6.1.

Source Gold Standard	Version	Notes
MIPS [41]	18 May 2006	We kept all MIPS categories containing at least two proteins as protein complexes
SGD [29]	11 Aug 2010	Gene Ontology (GO)-based protein complex annotations from SGD
CYC2008 [53]	v.2 1 Nov 2010	Comprehensive catalogue of 408 manually curated protein complexes from small-scale experiments reported in literature

Table 6.1: Gold standards used for evaluating dimers

Unfortunately these gold standards and our PPI networks overlap only partially, namely only some proteins are shared between them. For this reason we cannot verify all the putative dimers predicted, but only the fraction shared with the gold standard. We can determine the proteins shared before the process of dimers detection. With this aim we apply three preprocessing steps on the gold standards. By computing the set of proteins shared between our PPI networks and the gold standard we are implicitly quantifying the best prediction that any clustering algorithm for finding dimers can achieve in that particular gold standard. Lets see in detail these three steps.

Firstly, we removed from the gold standards all the protein complexes with size greater than two. We kept only the dimers reported in each gold standard, namely pairs of proteins linked by a single edge. It turns out that, in this context, the set of protein complexes in our gold standards correspond to a set of edges. Going forward we will always refer to the gold standards as the results of this filtering step.

Source Gold Standard	Number of Complexes	Number of Dimers	Percentage of dimers
MIPS	266	63	23.7%
SGD	323	68	21%
CYC2008	408	172	42%

Table 6.2: Properties of the gold standard datasets used for dimers evaluation

After this first step we checked how many of these edges are also present in our PPI networks. We refer a dataset set ds as a graph

$$G_{ds} = (V_{ds}, E_{ds})$$

and a gold standard gs as a graph $G_{gs} = (V_{gs}, E_{gs})$. The intersection between the gold standards gs and the set of edges E_{ds} of the dataset ds is hence defined as

$$E_{gs}(ds) = E_{ds} \cap E_{gs}$$

and it represents the set of all the dimers contained in our dataset set ds with respect to to the gold standard gs. Therefore the cardinality of $E_{gs}(ds)$ is the maximum quantity of dimers of ds that we can verify with the gold standard gs. The Figure 6.3 represents the procedure described.

However, in the set of edges $E_{gs}(ds)$ there may be a set of protein pairs connected but isolated with respect to all the other nodes. These isolated pairs of proteins are clearly detected by any clustering algorithm applied because they are isolated clusters of size 2. We will refer with AP_{ds} the set of *A*lways *P*redicted complexes of the dataset ds. Hence, as third step we remove from all the datasets all the proteins connected to only one other protein. Removing these proteins also from $E_{gs}(ds)$ we obtain the set that represents the best dimer prediction that a clustering algorithm can achieve in a dataset ds with respect to a gold standard gs. We summarize the results of these steps in the tables 6.3,6.4 and 6.5.

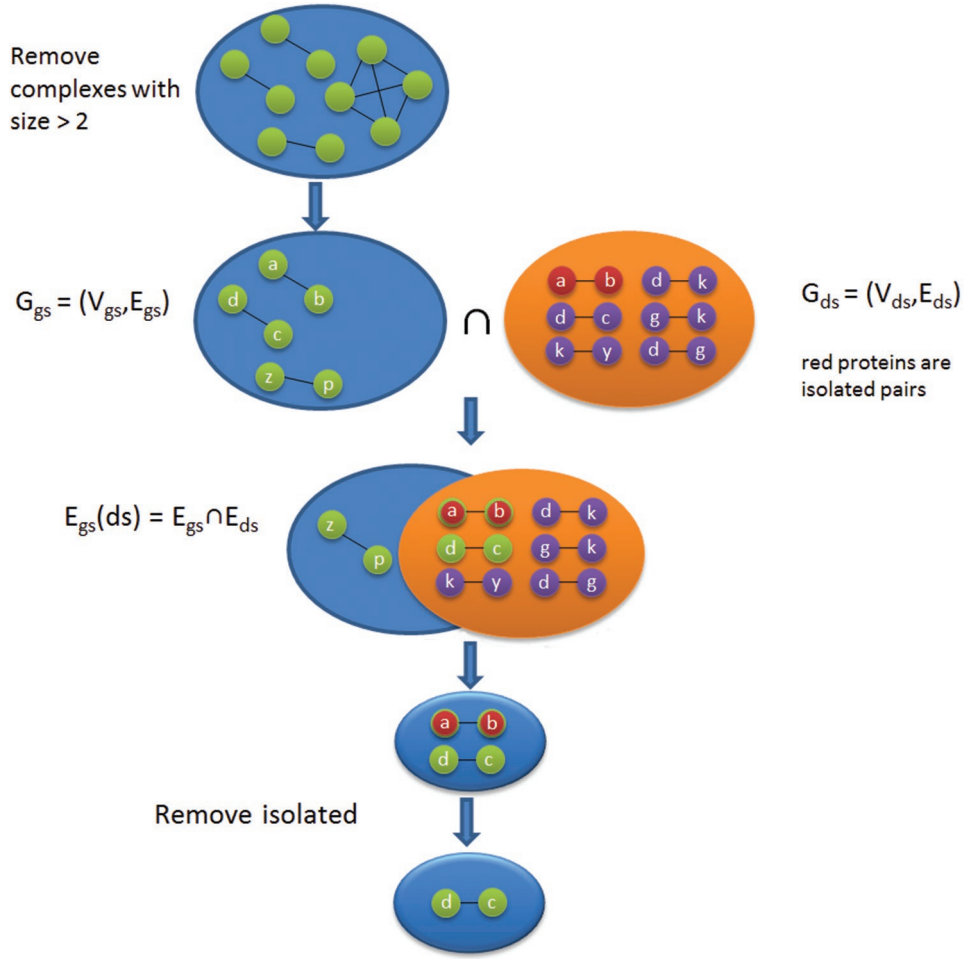

Figure 6.3: Filtering of gold standard

Dataset	Dimers predictable	Dimers always predicted	Best prediction
Collins	34	12	22
Gravin	26	0	26
Krogan Core	32	2	30
Krogan Extended	33	0	33

Table 6.3: MIPS dimers shared with the datasets

Dataset	Dimers predictable	Dimers always predicted	Best prediction
Collins	42	20	22
Gravin	34	2	32
Krogan Core	35	3	32
Krogan Extended	38	0	38

Table 6.4: SGD dimers shared with the datasets

Dataset	Dimers predictable	Dimers always predicted	Best prediction
Collins	84	40	44
Gravin	70	5	65
Krogan Core	84	5	79
Krogan Extended	88	0	88

Table 6.5: CYC2008 dimers shared with the datasets

After these three preprocessing steps we cluster the PPI networks with the aim of detecting dimers. For this step we applied Dominant Sets and ClusterONE.

The entire set of clusters returned by each method has been filtered keeping only protein complexes of size two, namely the putative dimers. For a given dataset ds we refer the output graph returned by a clustering method as

$$G_{ds}^{c} = (V_{ds}^{c}, E_{ds}^{c})$$

and its variant with only clusters of size two as

$$G_{ds}^{c=2} = (V_{ds}^{c=2}, E_{ds}^{c=2})$$

Therefore, to verify which of these putative dimers are true positives with respect to a gold standard gs it is necessary to intersect $E_{ds}^{c=2}$ with $E_{gs}(ds)$

$$TP = E_{ds}^{c=2} \cap E_{gs}(ds)$$

.

Clearly, directly comparing the quantity of true positive predicted dimers (TP) with the total number of dimers predictable ($E_{gs}(ds)$) in a given dataset ds with respect to a gold standard gs does not tell to us so much. Indeed, this fraction could be misleading. Let us consider, for example, a clustering algorithm to detect dimers which returns as output every single pair of proteins in the network. Clearly this algorithm will achieve a perfect score, maximizing the fraction $\frac{TP}{E_{gs}(ds)}$. This is due to the fact that it returns all the edges, including all the dimers shared with every gold standard, and hence $TP = E_{gs}(ds)$.

For this reason, in order to asses the quality of our results, we compute 2 quality measures: *precision* and recall. A third composite measure, known as *F-measure* or F_1, is also computed. We recall that these measures, widely applied for clustering analysis, are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{6.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

$$F - measure = \frac{(\beta^2 + 1) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{6.3}$$

Notice that when $\beta = 0$ the equation (6.3) coincides with (6.1), and hence the recall does not affect the *F-measure* leading to $F - measure = Precision$. Increasing β gains the weight of recall. For our purpose we set $\beta = 1$ assigning to *recall* the same weight of *precision*.

In particular, for a given dataset ds the measures became:

$$Precision = \frac{TP + |AP_{ds}|}{|E_{ds}^{c=2}|}$$

$$Recall = \frac{TP}{|E_{gs}(ds)|}$$

where, as explained before, $E_{gs}(ds)$ refers the best prediction achievable on the dataset ds with respect to the gold standard ds, and $E_{ds}^{c=2}$ is the set of all the putative dimers predicted on ds by a clustering algorithm. As pointed out before,

there may be in ds a set of dimers isolated in the network, namely a set of proteins each of which is connected to only one other protein. We refer this set as AP_{ds}.

In the tables 6.6, 6.7 and 6.8 are presented the results obtained applying the measure described above.

Dataset	**Total predicted**		**TP+AP**		**TP**		**Precision**		**Recall**		**F_1**	
	One	DS	One	DS	One	DS	One	DS	One	DS	One	DS
Collins	128	166	15	22	3	10	0.117	0.133	0.136	0.455	0.126	0.206
Gavin	36	138	3	12	3	12	0.083	0.087	0.115	0.462	0.096	0.146
KroganC	142	169	11	15	9	13	0.077	0.089	0.3	0.433	0.123	0.148
KroganE	135	167	9	14	9	14	0.067	0.084	0.273	0.424	0.108	0.14

Table 6.6: Dimers Evaluation by MIPS gold standard

Dataset	**Total predicted**		**TP+AP**		**TP**		**Precision**		**Recall**		**F_1**	
	One	DS	One	DS	One	DS	One	DS	One	DS	One	DS
Collins	128	166	22	28	2	8	0.172	0.169	0.091	0.364	0.119	0.231
Gavin	36	138	7	15	5	13	0.194	0.109	0.156	0.406	0.173	0.172
KroganC	142	169	13	19	10	16	0.092	0.112	0.313	0.5	0.142	0.183
KroganE	135	167	12	17	12	17	0.089	0.102	0.316	0.447	0.139	0.166

Table 6.7: Dimers Evaluation by SGD gold standard

Dataset	**Total predicted**		**TP+AP**		**TP**		**Precision**		**Recall**		**F_1**	
	One	DS	One	DS	One	DS	One	DS	One	DS	One	DS
Collins	128	166	43	54	3	14	0.336	0.325	0.068	0.318	0.113	0.321
Gavin	36	138	12	28	7	23	0.333	0.203	0.108	0.354	0.163	0.258
KroganC	142	169	24	31	19	26	0.169	0.183	0.241	0.329	0.199	0.235
KroganE	135	167	25	26	25	26	0.185	0.156	0.284	0.295	0.224	0.204

Table 6.8: Dimers Evaluation by CYC2008 gold standard

6.3 Dimers Evaluation through Yeast Two Hybrid Experiments

Assessing the quality of a set of dimers is a tricky task. Comparing the putative dimers given as output by a clustering algorithm with a set of manually

curated complexes could be considered the first step of quantifying their quality. Our benchmarks support the intuition that clustering algorithms based on Dominant Sets are suitable candidates for finding dimers, providing a better one-to-one matching with reference complexes in all the data sets. Unfortunately the PPI networks and the gold standards overlap only partially. In particular, only few proteins involved in some dimers are shared between them. With this lack of ground truth concerning dimers we cannot proceed with some relevant statistical test. The tables 6.3, 6.4 and 6.5 underline this issue.

For this reason we try to asses the quality of a set of dimers also using Y2H experiments. As reported in the section 1.7.1 the Y2H method allows us to verify the physical interaction between a pair of proteins. This approach to detect protein-protein interaction is widely used in literature and many researches have been published on yeast proteins. Large and small scale experiments are available.

A common way to access these data is using the *Bio*logical *G*eneral *R*epository for *I*nteraction *D*atasets, an online interaction repository that counts 37.439 publications from major model organism species. We used the most recent version of BioGRID (3.2.96 - December 25th, 2012) extracting 1566 studies on yeast Saccharomyces cerevisiae proteins based on Y2H method. Of these studies 1552 are small scale experiments while 14 are large scale experiments.

It is worth emphasizing the difference between small and large scale experiments. Indeed, their different nature lead us to exploit them in a different manner.

Usually, in small scale experiments only few proteins are involved. Often, this is due to the fact that they concern a very specific biological event or scenario. (e.g."Valproic acid- and lithium-sensitivity in prs mutants of Saccharomyces cerevisiae"). Focusing only on few proteins, researchers provide reliable protein-protein interactions. However, if two proteins appear in a given small scale Y2H experiment but their interaction is not reported, this does not imply that those two proteins do not interact. Indeed, it could be the case that the topic of that publication was not the interaction between them. Hence, the interaction between that particular pair of proteins could be not even checked.

On the other hand, large scale experiments usually counts at least 100 interactions. Being so wide usually the interaction reported in this kind of studies are

less reliable. Nonetheless often their authors check every single possible interaction between all the proteins reported.

Therefore, in some sense, while the presence of a particular interaction is highly reliable in small scale Y2H experiments, in large scale experiments is the absence of interaction between a pair of protein reported on it that is highly reliable.

The idea here proposed uses yeast two hybrid experiments to assign a probability of physical interaction between the proteins of each putative dimers detected. Moreover, using Y2H data we compute also the probability that no other proteins interact with a given predicted dimer.

6.4 Estimating the reliability of an edge

As pointed out before we filtered the BioGRID dataset extracting only the physical interactions checked by Y2H and concerning the yeast Saccharomyces cerevisiae. Starting from this set of interactions we built the entire network. Unfortunately this large list of protein interactions comes from different sources, from different authors and has different accuracy. Moreover we recall that the 2YH method, as all the other methods for protein-protein interaction detection, has a large amount of false positives and false negatives[10].

Hence, with the aim of using this data it is necessary to establish a probability representing the reliability of each physical interaction. How we compute this network and the weight of its edges is the main purpose of this section. Having this set of probabilities we can assign an accuracy value also to the putative dimers detected by a clustering algorithm. Indeed, in order for a pair of proteins to be a dimer, they have to be connected together with high probability, while they have to be connected to others protein with low probability (ideally zero). Therefore, in order to evaluate the quality of the predicted dimers as first step we assign to each Y2H interaction a value reflecting its accuracy. After that, with these data, we asses the quality of each putative dimer.

It is worth emphasizing why we used data obtained by the Y2H method. Indeed, someone can argue that protein-protein interaction data obtained by many other approaches are available.

The causes of this choice are intrinsically related to the nature of the yeast

two hybrid approach. Indeed, this molecular biology technique is used to discover protein-protein interactions by testing for physical interactions between two proteins. While other methods, as (T)AP-MS, measure physical interactions between groups of proteins without distinguishing whether they are direct or indirect.

Therefore, checking pairs instead of groups of proteins, the Y2H method seems to be the most suitable approach for validating dimers.

6.4.1 Integration of Y2H datasets

In order to estimate the probability of a physical interaction between a any pair of proteins in our Y2H data we proceed as follows. Given a pair of proteins (a, b) and defining DS as the set of all yeast two hybrid experiments available on yeast as

$$DS = DS_{smalls} \cup DS_{larges} = ds_1 \cup ds_2 \cup ... \cup ds_n$$

where DS_{smalls} is the subset of all the small scale experiments and DS_{larges} is the one of large scale experiments, we define

$$P(a \curlywedge b) = \begin{cases} 1, & \text{if } a \curlywedge b \in ds_x \text{ where } ds_x \in DS_{smalls} \\ \frac{h}{n}, & \text{otherwise} \end{cases} \tag{6.4}$$

where h is the number of $ds_x \in DS_{smalls}$, namely the number of large scale yeast two hybrid publications, where both the proteins a and b appear. While n is the number of large scale publications where a and b interact. This represents the probability that the protein a and the protein b are linked together building a protein complex $a \curlywedge b$ of size 2.

It is worth emphasizing that the equation (6.4) models what we pointed out in the previous section: the interaction of a pair of proteins in a small scale experiment is more reliable but its absence cannot be interpreted as non-interaction. Hence, if the interaction of a pair of proteins cannot be proved by some small scale experiment, we compute its accuracy as the number of large scale datasets which support that hypothesis divided by the number of datasets that contain those proteins.

After the reliability estimation of each interaction, for a putative dimer $a \curlywedge b$

we compute the probability to be a real dimer as

$$P_d(a \nmid b) = P(a \nmid b) \cdot \prod_{\text{x incident to a}} (1 - P(x \nmid a)) \prod_{\text{x incident to b}} (1 - P(b \nmid x)) \quad (6.5)$$

Intuitively, equation (6.5) means that in order to be a dimer, with respect to a set of yeast two hybrid experiments, a pair of proteins have to be linked together with high probability and linked to any other proteins with low probability. Therefore, putative dimers with an high value of $P_d(a \nmid b)$ are more likely true dimers.

6.5 Experimental Results

For each putative dimer, predicted by ClusterONE and Dominant Sets, we compute its probability of being a true dimer using the physical interaction network built as pointed out before. Unfortunately, only a fraction of all the putative dimers predicted have been studied in some of Y2H experiments present in literature. The tables 6.9 and 6.10 show, for each dataset, the fraction of predicted dimers contained in at least one Y2H experiment.

Dataset	Total predicted	Validated on Y2H
Collins	128	54
Gavin	36	14
Krogan Core	142	46
Krogan Extended	135	40

Table 6.9: Dimer predicted by ClusterOne and validated on Y2H experiments

Dataset	Total predicted	Validated on Y2H
Collins	166	73
Gavin	138	37
Krogan Core	169	50
Krogan Extended	167	52

Table 6.10: Dimer predicted by Dominant Sets and validated on Y2H experiments

Using our Y2H network we were able to produce, for each clustering method, a list of dimers with an associated probability, that is a sublist of the predicted dimers.

Since the results predicted by a computational method are commonly validated in laboratory, we sort the predicted dimers according to their probability in a decreasing way. In this way we can provide to biologists only a portion of our predicted dimers, those we highest probability.

In order to build these portions of dimers that have to be validated by further steps we split this ordered list of dimers into buckets. The first bucket contains the first five elements of the putative dimers list, namely the dimers with the highest probability of being true dimers. The second bucket contains the first ten elements of the list. We iterate this procedure increasing the size of each bucket of five elements until to build a bucket containing all the dimers.

In figure 6.4, 6.5, 6.6 and 6.7 we plot, for each dataset, the average of the probabilities associated to each bucket. Hence, the x-axis represents the size of each bucket of dimers, while the y-axis represent their average probability of being true dimers.

Indeed, each segment of these plots compare the the average probability of being a true dimer of the first n-elements predicted by each method.

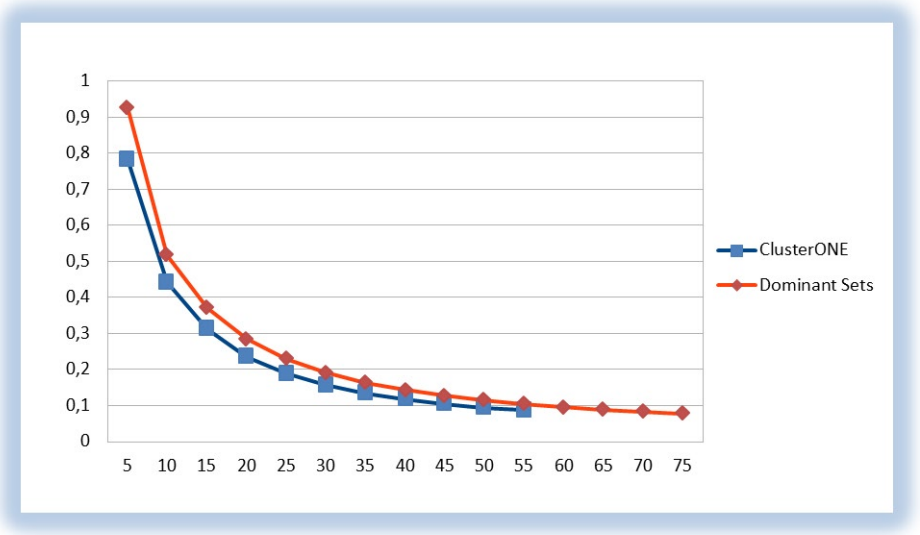

Figure 6.4: Average quality of the predicted dimers from the Collins dataset with respect to our Y2H network

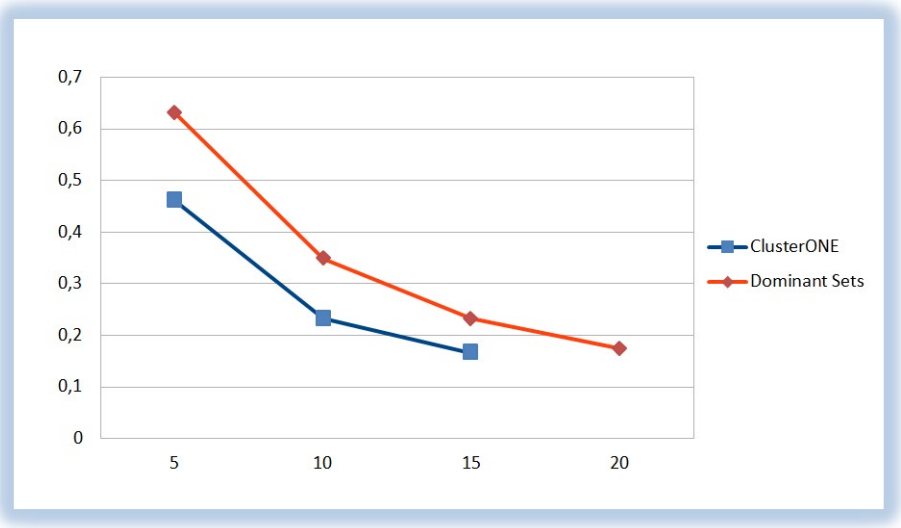

Figure 6.5: Average quality of the predicted dimers from the Gavin dataset with respect to our Y2H network

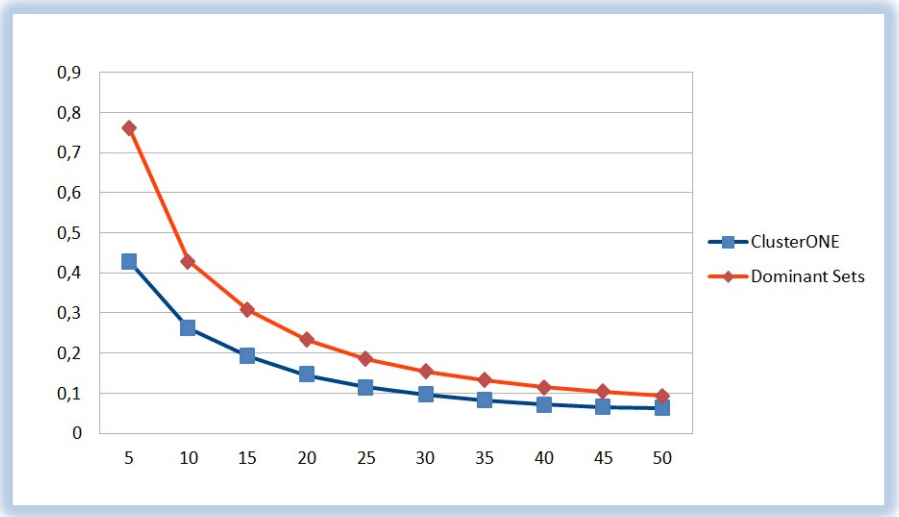

Figure 6.6: Average quality of the predicted dimers from the Krogan Core dataset with respect to our Y2H network

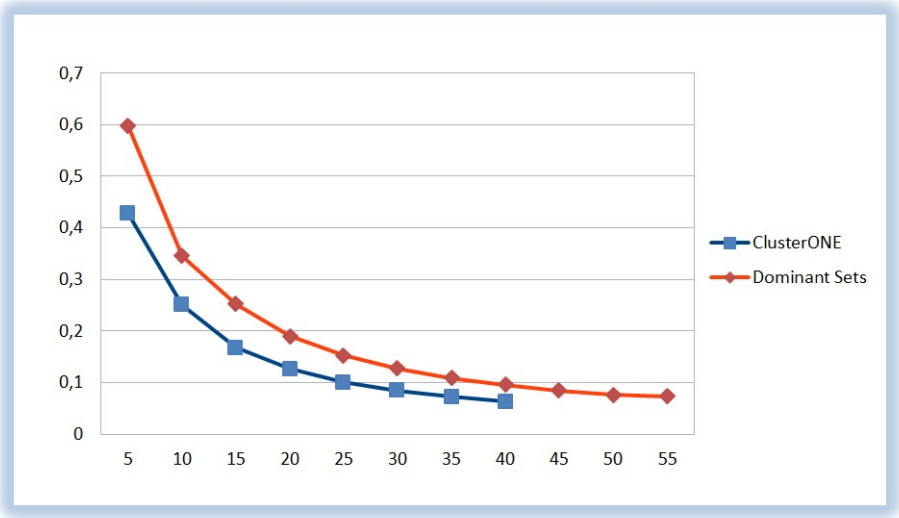

Figure 6.7: Average quality of the predicted dimers from the Krogan Extended dataset with respect to our Y2H network

As it is possible to see for all the datasets each bucket of dimers predicted by Dominant Sets shows an higher average probability of being a set of true dimers

than those predicted by ClusteONE. In particular, remarkable improvements are achieved for the Gavin, Krogan Core and Krogan Extended datasets.

Hence, this approach, as the one based on gold standards, supports the intuition that putative dimers detected by Dominant Sets have on average higher probability to be real dimers than those obtained with ClusterONE.

Chapter 7

Discussion and Conclusions

In this thesis, we applied a framework based on Dominant Sets in order to detect overlapping protein complexes from Protein-Protein Interaction networks. We have shown that the notion of Dominant Sets fits properly with the notion of protein complexes, allowing us to extract from PPIs datasets, sets of core protein complexes. Using further steps of refinement we promote the Dominant Sets to protein complexes adding to them the unclustered proteins in an appropriate way. We measure the cohesiveness of each Dominant Set, expanding it with clusters of size one if this operation conducts to remarkable improvements of cohesiveness.

Overlapped protein complexes detected in this fashion show noticeable quality with respect to those obtained with other standard clustering techniques. However, it is important to keep in mind that most of the alternative clustering algorithms cannot deal at the same time with overlapped clusters and weighted datasets.

Moreover, the quality of protein complexes predicted by the Dominant Sets algorithm is, in most of the cases, even more accurate than that obtained by ClusterOne -the state of the art for detecting protein complexes.

Our experiments highlighted also that Dominant Sets algorithm shows a high tendency to detect dimers, namely protein complexes of size two. This particular kind of complexes has important biological roles. To the best of our knowledge, this is the first attempt at defining a computational method with the express aim of finding dimers, which raises the need to define an evaluation methodology to validate the putative dimers.

Thus, we provide two different methodologies to validate putative dimers. The first attempt is based on matching them on three different gold standard datasets. Using this first approach our tests support the intuition that putative dimers detected by Dominant Sets have better quality with respect to those obtained with ClusterONE. Indeed, they show an higher F-measure for all the dataset and all the gold standards used.

However, the availability of relative small ground truth datasets led us to develop a second methodology for assessing the quality of putative dimers, which drives to a novel prospective. This approach exploits binary Y2H set of Protein-Protein Interactions.

We built a network of physical protein interactions aggregating the entire set of published Y2H interactions. We estimated the reliability of each edge of this network in order to establish, for each putative dimer, a probability related to the likelihood of being a dimer. Also this approach supports the intuition that putative dimers detected by Dominant Sets have on average higher probability to be real dimers than those obtained with ClusterONE.

In Chapter 6 we presented the issues related to the evaluation of putative dimers and a first attempt of defining an evaluation methodology to validate putative dimers. This renders our proposal a novel contribution. Unfortunately, the limited knowledge present in literature makes this problem difficult to address from a statistical point of view. For this reason, as a future work, we want estimate a reliability term for each publication based on the Y2H method. Adding this reliability term to the Y2H network built we expect a quality improvement of this physical Protein-Protein Interaction network. Progress on this wide network is expected to increase its predictive capability for evaluating putative dimers.

Bibliography

[1] B. Adamcsek, G. Palla, I. J. Farkas, et al. "CFinder: locating cliques and overlapping modules in biological networks". In: *Bioinformatics* 22.8 (2006), pp. 1021–1023.

[2] L. Apeltsin, J. H. Morris, P. C. Babbitt, and T. E. Ferrin. "Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution". In: *Bioinformatics* 27.3 (2011), pp. 326–333.

[3] M. Ashburner, C. A. Ball, J. A. Blake, et al. "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." In: *Nature genetics* 25.1 (May 2000), pp. 25–29. ISSN: 1061-4036.

[4] J.G. Auguston and J. Minker. "Ananalysis of some graph theoretical clustering techniques". In: *ACM* (1970).

[5] G. D. Bader and C. W. Hogue. "An automated method for finding molecular complexes in large protein interaction networks". In: *BMC Bioinformatics* 4 (2003), p. 2.

[6] Amos Bairoch, Rolf Apweiler, Cathy H. Wu, et al. "The Universal Protein Resource (UniProt)". In: *Nucleic Acids Research* 33 (2005).

[7] T. Berggard, S. Linse, and P. James. "Methods for the detection and analysis of protein-protein interactions". In: *Proteomics* 7.16 (2007), pp. 2833–2842.

[8] Nathan Blow. "Systems biology: Untangling the protein web". In: *Nature* 460 (2009), pp. 415–418.

[9] Anne-Laure Boulesteix. "Over-optimism in bioinformatics research". In: *Bioinformatics* 26.3 (2010), pp. 437–439.

[10] Anna BrÃ¼ckner, CÃ©cile Polge, Nicolas Lentze, et al. "Yeast Two-Hybrid, a Powerful Tool for Systems Biology". In: *International Journal of Molecular Sciences* 10.6 (2009), pp. 2763–2788. ISSN: 1422-0067.

[11] S. Brohee and J. van Helden. "Evaluation of clustering algorithms for protein-protein interaction networks". In: *BMC Bioinformatics* 7 (2006), p. 488.

[12] S. Rota Bulo. "A game-theoretic framework for similarity-based data clustering". PhD thesis. University of Venice, 2009.

[13] Samuel Rota Bulo and Immanuel M. Bomze. "Infection and immunization: A new class of evolutionary game dynamics". In: *Games and Economic Behavior* 71.1 (2011). ¡ce:title¿Special Issue In Honor of John Nash¡/ce:title¿, pp. 193 –211. ISSN: 0899-8256.

[14] Samuel Rota Bulo, Marcello Pelillo, and Immanuel M. Bomze. "Graph-based quadratic optimization: A fast evolutionary approach". In: *Computer Vision and Image Understanding* 115.7 (2011). ¡ce:title¿Special issue on Graph-Based Representations in Computer Vision¡/ce:title¿, pp. 984 –995. ISSN: 1077-3142.

[15] Sean Collins, Patrick Kemmeren, Xue-Chu Zhao, et al. "Towards a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae". In: *Molecular Cellular Proteomics* (Jan. 2007), pp. 600381–600200.

[16] Go Consortium. *Gene Ontology*. [Online; accessed 21-Dec-2012].

[17] Javier De Las Rivas and Celia Fontanillo. "Protein-Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks". In: *PLoS Comput Biol* 6.6 (June 2010), e1000807.

[18] Stijn van Dongen. "Graph Clustering by Flow Simulation". PhD thesis. University of Utrecht, 2000.

[19] A J Enright, S Van Dongen, and C A Ouzounis. "An efficient algorithm for large-scale detection of protein families". In: *Nucleic.Acids.Res.* 30 (2002), 1575–84.

[20] B. J. Frey and D. Dueck. "Clustering by passing messages between data points". In: *Science* 315.5814 (2007), pp. 972–976.

[21] Drew Fudenberg and Jean Tirole. *Game Theory*. Translated into Chinesse by Renin University Press, Bejing: China. Cambridge, MA: MIT Press, 1991.

[22] A. C. Gavin, P. Aloy, P. Grandi, et al. "Proteome survey reveals modularity of the yeast cell machinery". In: *Nature* 440.7084 (2006), pp. 631–636.

[23] Yoram Gdalyahu, Daphna Weinshall, and Michael Werman. *Self Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization*. 2001.

[24] Fred Glover, Manuel Laguna, and Rafael MartÃ. *Tabu Search*. 1997.

[25] Thore Graepel. "Statistical Physics of Clustering Algorithms". In: *DIPLOMARBEIT, TECHNIQUE UNIVERSITAT, FB PHYSIK, INSTITUT FUR THEORETISHE PHYSIK*. 1998.

[26] L. Hakes, D. L. Robertson, S. G. Oliver, and S. C. Lovell. "Protein interactions from complexes: a structural perspective". In: *Comp. Funct. Genomics* (2007), p. 49356.

[27] Pierre C. Havugimana, G. Traver Hart, Tamás Nepusz, et al. "A census of human soluble protein complexes." In: *Cell* 150.5 (Aug. 2012), pp. 1068–1081. ISSN: 1097-4172.

[28] Josef Hofbauer and Karl Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, May 1998. ISBN: 052162570X.

[29] Eurie L. Hong, Rama Balakrishnan, Qing Dong, et al. "Gene Ontology annotations at SGD: new data sources and annotation methods." In: *Nucleic Acids Research* 36.Database-Issue (Jan. 23, 2008), pp. 577–581.

[30] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN: 0-13-022278-X.

[31] Eric Jain, Amos Bairoch, Severine Duvaud, et al. "Infrastructure for the life sciences: design and implementation of the UniProt website". In: *BMC Bioinformatics* 10 (2009), p. 36.

[32] Ronald Jansen and Mark Gerstein. "Analyzing protein function on a genomic scale: the importance of gold-standard positives and negatives for network prediction". In: *Current Opinion in Microbiology* 7.5 (Oct. 2004), pp. 535–545. ISSN: 13695274.

[33] A. D. King, N. Przulj, and I. Jurisica. "Protein complex prediction via cost-based clustering". In: *Bioinformatics* 20.17 (2004), pp. 3013–3020.

[34] N. J. Krogan, G. Cagney, H. Yu, et al. "Global landscape of protein complexes in the yeast Saccharomyces cerevisiae". In: *Nature* 440.7084 (2006), pp. 637–643.

[35] G. Liu, L. Wong, and H. N. Chua. "Complex discovery from weighted PPI networks". In: *Bioinformatics* 25.15 (2009), pp. 1891–1897.

[36] V. Losert and E. Akin. "Dynamics of games and genes: Discrete versus continuous time". In: *Math. Biol.* (1983).

[37] Y. Lyubich, G.D. Maistrowskii, and Yu.G. Ol khovskii. "Selection-induced convergence to equilibrium in a single-locus autosomal population." In: ().

[38] K. Macropol, T. Can, and A. K. Singh. "RRW: repeated random walks on genome-scale protein networks for local cluster discovery". In: *BMC Bioinformatics* 10 (2009), p. 283.

[39] J. Maynard Smith and G. R. Price. "The Logic of Animal Conflict". In: *Nature* 246.5427 (1973), pp. 15–18.

[40] J. MaynardÂ smith. "The theory of games and the evolution of animal conflicts". In: *Journal of Theoretical Biology* 47.1 (Sept. 1974), pp. 209–221. ISSN: 00225193.

[41] H. W. Mewes, D. Frishman, K. F. X. Mayer, et al. "MIPS: Analysis and annotation of proteins from whole genomes". In: *Nucleic Acids Res* 32 (2004), pp. 41–44.

[42] T. S. Motzkin and E. G. Straus. "Maxima for graphs and a new proof of a theorem of Turán". In: *Canadian Journal of Mathematics* 17 (1965), pp. 533–540.

[43] John Nash. "Non-Cooperative Games". In: *The Annals of Mathematics*. Second Series 54.2 (Sept. 1951), pp. 286–295. ISSN: 0003486X.

[44] T. Nepusz, H. Yu, and A. Paccanaro. "Detecting overlapping protein complexes in protein-protein interaction networks". In: *Nat. Methods* 9.5 (2012), pp. 471–472.

[45] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. ISBN: 0691119937.

[46] Microbiology Online. *Microbiology Online (YEAST IMAGE)*. `http://www.microbiologyonline.org.uk/about-microbiology/introducing-microbes/fungi`. [Online; accessed 28-Dec-2012]. 2012.

[47] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. "Uncovering the overlapping community structure of complex networks in nature and society". In: *Nature* 435.7043 (2005), pp. 814–818.

[48] M. Pavan. "A New Graph-Theoretic Approach to Clustering, with Applications to Computer Vision". PhD thesis. UniversitÃ Ca' Foscari di Venezia, 2004.

[49] Massimiliano Pavan and Marcello Pelillo. "Dominant Sets and Pairwise Clustering". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29.1 (Jan. 2007), pp. 167–172. ISSN: 0162-8828.

[50] Marcello Pelillo. *What is a Cluster? Perspectives from Game Theory*.

[51] Jeffrey A Pleiss, Gregg B Whitworth, Megan Bergkessel, and Christine Guthrie. In: ().

[52] S. Pu, J. Wong, B. Turner, et al. "Up-to-date catalogues of yeast protein complexes". In: *Nucleic Acids Res.* 37.3 (2009), pp. 825–831.

[53] Shuye Pu, Jessica Wong, Brian Turner, et al. "Up-to-date catalogues of yeast protein complexes". In: *Nucleic Acids Research* 37.3 (Feb. 2009), pp. 825–831. ISSN: 1362-4962.

[54] Vijay V. Raghavan and C. T. Yu. "A Comparison of the Stability Characteristics of Some Graph Theoretic Clustering Methods". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 3.4 (Apr. 1981), pp. 393–402. ISSN: 0162-8828.

[55] William H. Sandholm. *Population games and evolutionary dynamics*. Economic learning and social evolution. Cambridge, Mass. MIT Press, 2010. ISBN: 978-0-262-19587-4.

[56] Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation". In: *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE* 22.8 (2000), pp. 888–905.

[57] John Maynard Smith. *Evolution and the Theory of Games*. Cambridge, UK: Cambridge University Press, 1982.

[58] Chris Stark, Bobby joe Breitkreutz, Teresa Reguly, et al. *BioGRID: a General Repository for Interaction Datasets*. 2006.

[59] Peng Gang Sun and Lin Gao. "Fast algorithms for detecting overlapping functional modules in protein-protein interaction networks". In: *Proceedings of the 6th Annual IEEE conference on Computational Intelligence in Bioinformatics and Computational Biology*. CIBCB'09. Nashville, Tennessee, USA: IEEE Press, 2009, pp. 247–254. ISBN: 978-1-4244-2756-7.

[60] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN: 0321321367.

[61] *Thermo Scientific*. `http://www.piercenet.com/browse.cfm?fldID=9C471132-0F72-4F39-8DF0-455FB515718F`. Accessed: 15/10/2012.

[62] A. Torsello, S.R. Bulo, and M. Pelillo. "Beyond partitions: Allowing overlapping groups in pairwise clustering". In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. 2008, pp. 1 –4.

[63] Stijn Van Dongen. "Graph Clustering Via a Discrete Uncoupling Process". In: *SIAM J. Matrix Anal. Appl.* 30.1 (Feb. 2008), pp. 121–141. ISSN: 0895-4798.

[64] S. Watanabe. *Theorem of the ugly duckling*. New York: Wiley, 1969, pp. 376–377+.

[65] Jorgen W. Weibull. *Evolutionary game theory*. Cambridge, Mass. [u.a.]: MIT Press, 1995. XV, 265. ISBN: 0262231816.

[66] Wikipedia. *Alternative Splicing (IMAGE)*. `http://en.wikipedia.org/wiki/File:Splicing_overview.jpg`. [Online; accessed 27-October-2012]. 2012.

[67] Wikipedia. *Evolutionary Game Theory (IMAGE)*. [Online; accessed 5-Dec-2012].

[68] Wikipedia. *Fork of Biopython used to build the SGD gold standard*. [Online; accessed 5-Dec-2012].

[69] Wikipedia. *Methods to investigate protein-protein interaction*. `http://en.wikipedia.org/wiki/Methods_to_investigate_protein%E2%80%93protein_interactions`. [Online; accessed 1-Jenuary-2013]. 2012.

[70] Wikipedia. *Protein Fragment complementation Assay (IMAGE)*. `http://en.wikipedia.org/wiki/Protein-fragment_complementation_assay`. [Online; accessed 15-October-2012]. 2012.

[71] Wikipedia. *Two-hybrid screening*. `http://en.wikipedia.org/wiki/Yeast_two-hybrid_system`. [Online; accessed 1-October-2012]. 2012.

[72] Zhenyu Wu and Richard Leahy. "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1993).

[73] Haiyuan Yu, Nicholas M Luscombe, Hao Xin Lu, et al. "Annotation transfer between genomes: protein-protein interologs and protein-DNAregulogs". In: *Genome Research* 14 (2004), 1107–18.

[74] C. T. Zahn. "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters". In: *IEEE Trans. Comput.* 20.1 (Jan. 1971), pp. 68–86. ISSN: 0018-9340.