Giovanni Scardoni

Computational Analysis of biological networks

Ph.D. Thesis

June 30, 2010

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
prof. Roberto Giacobazzi

Series N°: **TD-07-10**

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

ad Alice e alla sua gioia di vivere,
ad Anita e ai suoi sorrisi

Acknowledgments

First of all, I want to thank my advisor Roberto Giacobazzi, the open minded professor who supported my will of get ahead with my project in my own way and who trusted me also when results were far to come. I thank also Carlo Laudanna for having given me the opportunity of exploring the fantastic world of computational biology, bioinformatics and network science. I'm also grateful to him for our interesting debates about biology, science and life. I'm thankful to Vincenzo Manca and Giuditta Franco for the interest they showed in my research project. Thanks to Michele Petterlini for his precious collaboration and to all the people I met during this research experience, they always left something to me, among all Fausto, Samir, Alessio, Isabella and Mila.

Contents

1 Introduction 1
- 1.1 Part I: Topological analysis of biological networks 2
  - 1.1.1 Contributions 2
- 1.2 Part II: Dynamic analysis of metabolic pathways 4
  - 1.2.1 Contributions 5
- 1.3 Publications 6

2 Basic notions 7
- 2.1 Graphs 7
- 2.2 Biological networks 8

Part I Topological analysis of biological networks

3 Centralities definition and description 13
- 3.1 Preliminary definitions 13
- 3.2 Centralities 14
  - 3.2.1 Degree ($deg(k)$) 14
  - 3.2.2 Diameter (Δ_G) 15
  - 3.2.3 Average Distance (AvD_G) 16
  - 3.2.4 Eccentricity ($C_{ecc}(v)$) 16
  - 3.2.5 Closeness ($C_{clo}(v)$) 17
  - 3.2.6 Radiality ($C_{rad}(v)$) 19
  - 3.2.7 Centroid value ($C_{cen}(v)$) 20
  - 3.2.8 Stress ($C_{str}(v)$) 21
  - 3.2.9 S.-P. Betweenness ($C_{spb}(v)$) 22
- 3.3 Normalization and relative centralities 23
- 3.4 Conclusions 23

4 CentiScaPe 25
- 4.1 System overview 26
- 4.2 Algorithm and implementation 27
- 4.3 Using CentiScaPe 28

4.3.1 CentiScaPe Results Panel . . . 29
4.3.2 Graphic output . . . 29
4.4 Conclusions . . . 32

5 A real world example: Centralities in the human kino-phosphatome . . . 35
5.1 Centralities analysis . . . 35
5.2 Phosphoproteomic analysis of chemoattractant stimulated human PMNs . . . 40
5.2.1 Human primary polymorphonuclear cells isolation . . . 40
5.2.2 Human primary polymorphonuclear cell stimulation . . . 41
5.2.3 Evaluation of protein phosphorylation . . . 41
5.3 Combining topological analysis and experimental data . . . 41
5.4 Conclusions . . . 43

6 Network centralities interference . . . 45
6.1 Interference notion . . . 46
6.2 Betweenness interference . . . 47
6.3 Interference centralities definition . . . 50
6.4 A real word example: interference in the human kino-phosphatome 54
6.5 Further consideration for network centralities interference . . . 56
6.6 Nodes centrality robustness, dependence and competition value . . . 57
6.7 Conclusions . . . 62

Part II Dynamic analysis of biological pathways

7 Abstract Interpretation for dynamic simulation of pathways . . . 65
7.1 Preliminaries . . . 66
7.1.1 Pathways . . . 66
7.1.2 Abstract interpretation . . . 67
7.2 Modeling pathways . . . 68
7.2.1 Pathway definition . . . 68
7.2.2 The pathway simulation . . . 70
7.2.3 Semantics for pathways . . . 70
7.2.4 Abstract semantics for pathways . . . 72
7.3 Abstract interpretation based analysis of pathways . . . 73
7.4 A real world example: the mitotic oscillator . . . 73
7.5 Abstract interference for biological pathways . . . 79
7.5.1 Abstract interference and mitotic oscillator . . . 81
7.6 Conclusions . . . 81

8 Conclusions . . . 83

References . . . 85

1

Introduction

Characterizing, describing, and extracting information from a network is by now one of the main goals of science, since the study of network currently draws the attention of several fields of research, as biology, economics, social science, computer science and so on. The main goal is to analyze networks in order to extract their emergent properties [10] and to understand functionality of such complex systems. This thesis concerns the analysis of biological networks and the two main approaches are treated: the first based on the study of their topological structure, the second based on the dynamic properties of the system described by the networks. Since "always structure affects function" [57], the topological approach wants to understand networks functionality through the analysis of their structure. For instance, the topological structure of the road network affects critical traffic jam areas, the topology of social networks affects the spread of information and disease and the topology of the power grid affects the robustness and stability of power transmission. The approach to networks dynamic, explores the variation of the network in time as for instance the reactants concentration in a metabolic pathway, a network describing the set of chemical reactions occurring within a cell. In this case the simulation of a metabolic pathway [29], [30] characterizes completely the behavior of each element of the pathway. But, since structure and dynamic are strictly related, some times the two approaches cannot be considered separately. This is the case of network motifs [45], [55], particular subnetworks which occur significantly more often in biological networks than in random networks and whose topological characteristics have been shown to subtend to particular functions. For instance a feedback loop motifs implies a possible oscillation behavior of the system.

This thesis concern both static and dynamic analysis of biological network, but most of the results can be applied to other kind of networks. It is divided into two parts, the first regarding the topological analysis of networks and the second facing some problem of dynamic simulations of metabolic pathways.

1.1 Part I: Topological analysis of biological networks

The topological analysis of networks, concerns the study and characterization of networks structure. Remarkable results have been reached in this field, and even if far from being complete, several key notions have been introduced, not only for biological networks. These unifying principles underly the topology of networks belonging to different fields of science. Fundamental are the notions of scale-free network [8], [37], cluster [46], network motifs [45], [55], small-world property [62], [61], [60] and centralities. Particularly, centralities have been initially applied to the field of social science [25] and then to biological networks [63]. Usually, works regarding biological networks rightly consider global properties of the network and when centralities are used, they are often considered from a global point of view, as for example analyzing degree or centralities distribution [37], [60], [63], [64], [38]. A node-oriented approach have been used analyzing attack tolerance of network, where consequences of central nodes deletion are studied [2], [22]. But also in this case the analysis have been concentrate on global properties of the network and not on the relevance of the single nodes in the network. Similarly, available software for network analysis is usually oriented to global analysis and characterization of the whole networks. To identify relevant nodes of a biological network, protocols of analysis integrating centralities analysis and lab experimental data are needed and the same for software allowing this kind of analysis. Cytoscape is an excellent visualization and analysis tool with the analysis features greatly enhanced by plug-ins. Plug-in such as NetworkAnalyzer [6] computes some node centralities but does not allow direct integration with experimental data. Applications such as VisANT [35], and Centibin [39] calculate centralities, although they either calculate fewer centralities or are not suitable to integration with experimental data. Starting from these general considerations, the first part of this thesis concern the application of network centralities analysis to protein interaction networks from a perspective oriented to identify relevant nodes in such networks. Necessary steps to do this are illustrated in the next section.

1.1.1 Contributions

The aim of the first part of this thesis is to face the centralities analysis of a protein interaction network from a node oriented point of view. We want to identify nodes that are relevant for the networks for both centralities analysis and lab experiments. To do this, the following steps have been done:

- Some centralities that we consider biologically significant have been detected. A biological meaning of these centralities have been hypothesized.
- A protocol of analysis for a protein network based on integration of centralities analysis and data from lab experiments (activation level) have been designed.
- A software (CentiScaPe) for computing centralities and integrating topological analysis results with lab experimental data set have been designed and implemented.
- A human kino-phosphatome network have been extract from a global human protein interactome data-set, including 11120 nodes and 84776 unique undirected interactions obtained from public data-bases.

- The software have been applied to this human kino-phosphatome network and activation level of each protein (in threonine and thyrosine) have been related to centrality values.
- Proteins important from both topological analysis and activation level have been easily identified: the attention of successive experiments and analysis should be focused on these proteins.

A further step have been introduced in this thesis. Once we have identified relevant proteins in a network, we are interested in identifying non-obvious relations between these and other proteins in the network. In any network structure, the role of a node depends, not only on the features of the node itself, but also on the topological structure of the network and on the other nodes features. So even if centralities are node properties, they depend also on other nodes. We know that in a protein network nodes can be added or deleted because of different reasons as for example gene duplication (adding) or gene deletion or drug usage (deleting). If we delete a relevant node in the network, the effects of the deletion have impact not only on the single node and its neighbors, but also on other parts of the network. For instance, if you are close friend of an important politician of your town, you have a central role in the social network of the town, and consequently your friends have a central role. But if this politician looses his central role, or if he is completely excluded from the political life of the town, for instance because they put him in prison (this correspond to a deletion on the social network), also you loose your central role in the network and the same for those people related to you. The idea is that the impact of an adding or deletion of a node can be measured through the variation of centrality values of the other nodes in the network. Such notion introduced in the thesis have been called "network centralities interference". It allows to identify those nodes that are more sensitive to deletion or adding of a particular node in the network. Interference have been applied to the human kino-phosphatome network with some interesting results: for example two different relevant proteins (Mapk1 and Prkca) have been shown to "interfere" with different nodes in the network. It means that if we remove Mapk1 from the network, nodes affected from this deletion are not the same nodes affected from Prkca deletion. Besides, Mapk1 and Prkca have been shown to "interfere" each other i.e. if one of them is deleted the other increases its central role in the network. In this sense they are "competitors" in the network. Complementary to this notion, we introduce the notion of "node centrality robustness". This notion measures how much the central role of a node in the network is due to the presence of another node. Focusing the attention to a node, we remove another node from the network and we measures variations of centrality values of the node of interest. We repeat the process but removing another node, an so on for all the nodes in the network. If the node of interest is "robust", the variation is low for all the nodes removed, and central role of the node does not depend on other nodes of the network.

Chapter 2 contains some basic notions of network science used in the rest of the thesis. Chapter 3 consists in a review of some centralities considered important from a biological point of view in a protein network. For each centrality a possible biological meaning have been treated and some examples illustrate the significance of each centrality. Chapter 4 introduce the CentiScaPe software, the Cytoscape plug-in we implemented for computing network centralities. Main fea-

ture of the software is the possibility of integrating experimental data-set with the topological analysis. In CentiScaPe, computed centralities can be easily correlated between each other or with biological parameters derived from the experiments in order to identify the most significant nodes according to both topological and biological properties. In chapter 5 the protocol of analysis is introduced through an example of analysis of a human kino-phosphatome network. Most relevant kinases and phosphatases according to their centralities values have been extracted from the network and their phosphorylation level in threonine and tyrosine have been obtained through a lab experiment. Centrality values and activation (phosphorylation) levels have been integrated using CentiScaPe and most relevant kinases and phosphatases according to both centrality values and activation levels have been easily identified. In chapter 6 the notion of node centrality interference and node centrality robustness are introduced. They are discussed through some examples and interference is applied to the real example of the human kino-phosphatome network (section 6.4).

1.2 Part II: Dynamic analysis of metabolic pathways

Metabolic pathways are series of chemical reactions occurring within a cell. These reactions depend on some parameters such as concentration of reactants, functions and other parameters regulating the speed of reaction, and are organized in complex networks. Usually pathways are modeled by differential equations, that represents the changes in the concentration of the molecules of the pathway. This approach is useful and well-studied [56], [59], [30] and is essentially based on standard numerical techniques for solving differential equations, as for example the Eulero's method and similar. But also many different other models using a variety of computational formalisms and logics originally intended for modeling and analysis of computer systems have been used to model and analyze them. Much of the effort has been devoted to developing techniques to represent relevant biological concepts and to simulate their behavior. Model checking have been applied to program simulating pathway [14]. With this approach we can infer if from a starting state of the system it is possible to reach another state satisfying a particular property. Pathway logic [24], [58] have been similarly used to analyze pathways with standard temporal logic questions. Pathway logic approach is based on the rewriting system language MAUDE [15] and have been successfully used for modeling a pathway including more than 650 proteins and 500 rules then analyzed with the MAUDE model checker. Hybrid systems [4], [3] have been used to model those biological systems passing from discrete to continuous behavior. Petri nets [31], *pi*-calculus model [49] and its stochastic version [48], [43] have also been used. The P-systems [47] approach have also been applied successfully to several pathways. What is unifying all these approaches is that formal methods used to analyze software are applied to the analysis of biological systems. A pathway with thousands of reactions can be viewed as a computer program with thousands of instructions, so the design of formal tools for modeling biomolecular processes and for reasoning about their dynamics seems to be a mandatory research path to which the field of formal verification in computer science may contribute a lot. In this

thesis we propose to go beyond simulation and to focus on how abstract interpretation [18], [19], [20] can be used to analyze a dynamic simulation of a biological pathway. Abstract interpretation is a framework for software analysis independent from the model, and it is useful to extract numerical properties of program variables. We show that it can be used to extract properties of pathways simulations and can contibute to the solution of the problem of parameters estimation.

1.2.1 Contributions

In chapter 7 the abstract interpretation framework is applied to simulation of biological pathways. We suppose to have a program simulating a pathway, and we apply abstract interpretation techniques to this program, focusing the attention to the proteins concentration. Notably, abstract interpretation can be applied to any program and consequently to any model of simulation. In our examples we use differential equations and the Eulero's method. Particularly we use the abstract interpretation analysis based on constants and intervals domain [18], and congruence domain [32]. Constant domain is used to analyze reactants concentration in order to identify those reactants having concentration constant after a certain time. Interval domains are similarly used to identify the concentration range of the reactants. So analyzing the simulation we know if the concentration of a particular element remains in a certain interval of values for all the computation time. Congruence domain is shown to be useful for automatic identification of regular oscillations in a pathway simulation. Analyzing a simulation, we focus the attention on the time when the concentration of a reactant stop growing and start decreasing and we keep this time value in a variable. If the oscillation is regular, the variable assume regular values (for instance 3, 7, 11...) and its analysis on the congruence domain results in a congruence class (3 mod 4 in the example). All these properties can be easily inferred observing the graphic of the simulation, but what makes abstract interpretation so strong is that it is completely automated. So we can launch thousands of simulations with different parameters and the analysis will results in a completely characterization of those starting values resulting in a particular behavior (oscillation, constant concentration, concentration belonging to some range of values). Obviously is not possible to observe thousands of graphics to infer the same properties. In such a way we can easily answer to question as "Which are the starting values resulting in a simulation where the protein X has concentration value in the range [3 , 5]?". Or "Which are the starting values resulting in a simulation where the protein X has oscillating concentration values?. Or again "Which are the starting values resulting in a simulation where the protein X has concentration value that is constant after 10 seconds?". Besides, abstract interpretation is completely independent from the model used and can be applied to many different simulation methods. The method can also be applied to the problem of parameters estimation. For many pathways we don't know some parameters of the functions regulating the pathway reactions. But if we know the behavior of the pathway we can infer the parameters inducing that behavior. For instance, suppose that a parameter of a function regulating a pathway is missing, but suppose we know that the concentration of the reactant X is always in the range of [3 ,7], and the concentration of the reactant Y is in the range [2 ,8]. Using

abstract interpretation techniques, we start thousands of simulations for thousands different values of the missing parameter and we analyze the concentration value of X and Y. At the end we check which starting values lead to the proper ranges of concentration. This or these are the right values for the missing parameter. In section 7.2 the abstract interpretation method have been successfully applied to the Goldbeter [29], [30] mitotic oscillator pathway.

A further step have been done in abstract interpretation analysis of biological pathways focusing on an "interference" point of view. Similarly to the static approach of chapter 6, a notion of interference have been applied to pathways simulation. If we change the starting concentration value of a reactant, also the concentration variables of other reactants can be modified during the simulation. This is a problem of variable interference as introduced in [28], i.e. changing the value of a variable results in changing the value of another variable in another program point. So if we change a starting parameters of a pathways, this "interferes" with the concentration values of others reactants. We introduce such a notion of "interference for biological pathway" and we apply it to our simulations. Besides, also abstract interference have been recently introduced [26] in order to characterize interference not between program variables but between properties of program variables. We apply abstract interference to pathways simulation in order to find relations between properties of reactants. For example: "If the starting concentration value of the reactant X is in the interval [2.4 , 7] then the concentration value of reactant Y after 3 seconds is constant to the value 3.5". In section 7.5 abstract interference for biological pathway is introduced and discussed.

1.3 Publications

CentiScaPe software, implemented in collaboration with Michele Petterlini, have been released as a Cytoscape plug-in. Contents of chapter 3 have been released as a "centralities tutorial" with the plug-in. At present version 1.1 of CentiScaPe is available and it is downloaded with a rate of about 140 downloads for month from the Cytoscape website (see Cytoscape website for download statistics). First results using CentisCaPe have been presented at 48th ASCB annual meeting [11]. Contents of chapter 5 have been developed in collaboration with Carlo Laudanna and together with contents of chapters 4 they are part of a publication on Bioinformatics [52]. A preliminary work about abstract interpretation for dynamic simulation of pathways (chapter 7) was presented at PLID 2005 [50] and then presented at EAAI [51] and published on conference proceedings. The node centrality interference and robustness is unpublished. It is still a work in progress and will be submitted when completed with a proper software and integrated with data from lab experiments.

Other results using CentisCaPe have been published by different authors in [44] and [53]. CentiScaPe is also used at GlaxoSmithKline computational biology labs.

2

Basic notions

Some fundamental notions are needed to deal with biological networks. Here we briefly introduce some main concepts that are used in the rest of the thesis. An excellent and complete review of results in network science, facing all the main aspects of research can be found in [13].

2.1 Graphs

A network is mathematically represented by a graph $G = (N, E)$ where N is the set of nodes (or vertices) and E is the set of edges (or links), i.e a set of pair of nodes. If the pair of nodes are ordered the graph is directed, undirected otherwise. If a pair $(n_1, n_2) \in E$ then n_1 and n_2 are neighbors. A graph $G' = (N', E')$ is a subgraph of the graph G if $N' \subseteq N$ and $E' \subseteq E$.
Note. Graphs are the mathematical representation, networks are the real systems. Even if not properly correct in the rest of the thesis we refere to network or graph indifferently, since we apply mathematical concept to real systems.

Degree

The degree of a node is the number of neighbors of a node. The average degree of nodes for the whole network is used as an index to describe the "density" of a network. In networks in which each link has a selected direction, incoming (k in) and outgoing (k out) degrees need to be considered. The degree distribution ($p(k)$) gives the probability that a selected node has exactly k links.

Cluster

A cluster [46] is a group of nodes highly connected between them. A cluster can be identified by the clustering coefficient (C). It is a measure of the degree of interconnectivity in the neighbourhood of a node.

Shortest path

Shortest path (SP). The path between two nodes in a network with a smaller number of steps than the many alternative paths between the two nodes.

Scale-free networks

Many networks are characterized by a power law-like degree distribution [37] [8]. In a scale-free network, the probability that a node has k links follows $p(k) \sim k^{-\gamma}$, where γ is the degree exponent. Such distributions are seen as a straight line on a loglog plot. A relatively small number of highly connected nodes are known as hubs, and the probability of those hubs is statistically more significant than in a random network.

Small world property

The small world property is common to many real networks. It is the fact that most pairs of vertices in the networks seem to be connected by a short path through the network. A famous experiments carried out by Stanley Milgram in the 1960s, in which letters passed from person to person were able to reach a designated target individual in only a small number of steps, around six in the published cases. This result is one of the first direct demonstrations of the small-world effect. It have been recently mathematically studied and applied to different kinds of networks [62], [61].

2.2 Biological networks

Different kinds of biological networks have been proposed, depending on the biological process studied. We can distinguish proteins interaction networks, gene regulatory networks, and metabolic networks or pathways.

Proteins networks

Proteins networks are networks where the nodes are proteins and the edges are interaction between proteins. A first map of the yeast proteins interaction network have been introduced in [36]. A particular case of proteins interaction networks are signal transduction networks. These are proteins networks regulating the transmission of information within a cell. The edges are directed and can have an inhibition or activation role. Information about edges direction and role are not available in some cases.

Gene regulatory networks

A gene regulatory network or genetic regulatory network is a network governing gene expression. The nodes are genes, proteins, or mRNA. The edges represent activation or inhibitions of reactions or protein or mRNA production.

Metabolic networks or pathways

Pathways are network describing the set of reactions regulating a biological process. Even if they are usually represented in several ways, the following characteristics are common to all the pathways. The nodes can be reactants (substrates), products of the reactions enzymes, or reactant-enzyme complex. The edges reflect reactions or regulation of reactions. They can be directed edges from reactants/enzymes to complex or directed edge from complex to products/enzyme.

Part I

Topological analysis of biological networks

3

Centralities definition and description

In this chapter, some of the classical network centralities have been introduced. For each centrality, we present the mathematical definition, a brief description with some examples, and a possible biological meaning in a protein network. As known network centralities allow to categorize nodes for their relevance in the network structure. Usually, works concerning biological networks rightly consider global properties of the network and even if centralities are used, they are often considered from a global point of view, as for example analyzing degree or centralities distribution [37], [60], [63], [64], [38]. An interesting algorithm for finding cluster is also based on the betweenness centrality value [33] A node-oriented approach have been used analyzing attack tolerance of network, where consequences of central nodes deletion are studied [2], [22]. But also in this case the analysis have been concentrate on global properties of the network and not on the relevance of the single nodes in the network. The approach of this thesis consists in a further step: centralities can be used to identify single proteins and, combined with biological parameters coming from lab experiments, allow us to characterize proteins for their topological and biological relevance. This will become more significant in chapter 5 where we introduce a protocol of analysis where centrality values of proteins in a protein network are related to their activation level (phosphorylation level). In this chapter a brief centralities review is presented and some functional hypothesis of centralities role in a protein interaction network are introduced. A good and complete description of network centralities can be found in [42], where also some algorithms are presented. For many centralities indices it is required that network is connected, i.e. each node is reachable from all the others. If not, some centralities can results in infinity values or some other not properly correct computation. Besides some centralities are not defined for directed graph (except of trivial situation), so we will consider here and in the rest of the thesis only connected undirected graph.

3.1 Preliminary definitions

Let $G = (N, E)$ an undirected graph, with $n = |N|$ vertexes. $deg(v)$, indicate the degree the vertex. $dist(v, w)$ is the shortest path between v and w. σ_{st} is the

number of shortest paths between s and t and $\sigma_{st}(v)$ is the number of shortest paths between s and t passing through the vertex v. Notably:

- Vertex = nodes; edges = arches;
- The distance between two nodes, $dist(v, w)$ is the shortest path between the two nodes;
- All calculated scores are computed giving to higher values a positive meaning, where positive does refer to node proximity to other nodes. Thus, independently on the calculated node centrality, higher scores indicate proximity and lower scores indicate remoteness of a given node v from the other nodes in the graph.

3.2 Centralities

3.2.1 Degree ($deg(k)$)

Is the simplest topological index, corresponding to the number of nodes adjacent to a given node v, where adjacent means directly connected. The nodes directly connected to a given node v are also called first neighbors of the given node. Thus, the degree also corresponds to the number of adjacent incident edges. In directed networks we distinguish in-degree, when the edges target the node v, and out-degree, when the edges target the adjacent neighbors of v. Calculation of the degree allows determining the degree distribution $P(k)$, which gives the probability that a selected node has exactly k links. $P(k)$ is obtained counting the number of nodes $N(k)$ with $k = 1, 2, 3 \ldots$ links and dividing by the total number of nodes N. Determining the degree distribution allows distinguishing different kind of graphs. For instance, a graph with a peaked degree distribution (Gaussian distribution) indicates that the system has a characteristic degree with no highly connected nodes. This is typical of random, non-natural, networks. By contrast, a power-law degree distribution indicates the presence of few nodes having a very high degree. Nodes with high degree (highly connected) are called hubs and hold together several nodes with lower degree. Networks displaying a degree distribution approximating a power-law, $P(k) \approx k^{-\gamma}$, where γ is degree exponent, are called scale-free networks [8]. Scale-free networks are mainly dominated by hubs and are intrinsically robust to random attacks but vulnerable to selected alterations [2], [36]. Scale-free networks are typically natural networks.

In biological terms

The degree allows an immediate evaluation of the regulatory relevance of the node. For instance, in signaling networks, proteins with very high degree are interacting with several other signaling proteins, thus suggesting a central regulatory role, that is they are likely to be regulatory hubs. For instance, signaling proteins encoded by oncogenes, such as HRAS, SRC or TP53, are hubs. Depending on the nature of the protein, the degree could indicate a central role in amplification (kinases), diversification and turnover (small GTPases), signaling module assembly (docking proteins), gene expression (transcription factors), etc. Signaling networks have typically a scale-free architecture.

3.2.2 Diameter (Δ_G)

ΔG is the maximal distance (shortest path) amongst all the distances calculated between each couple of vertexes in the graph G. The diameter indicates how much distant are the two most distant nodes. It can be a first and simple general parameter of graph compactness, meaning with that the overall proximity between nodes. A high graph diameter indicates that the two nodes determining that diameter are very distant, implying little graph compactness. However, it is possible that two nodes are very distant, thus giving a high graph diameter, but several other nodes are not (see figure 3.1). Therefore, a graph could have high diameter and still being rather compact or have very compact regions. Thus, a high graph diameter can be misleading in term of evaluation of graph compactness. In contrast a low graph diameter is much more informative and reliable. Indeed, a low diameter surely indicates that all the nodes are in proximity and the graph is compact. In quantitative terms, high and low are better defined when compared to

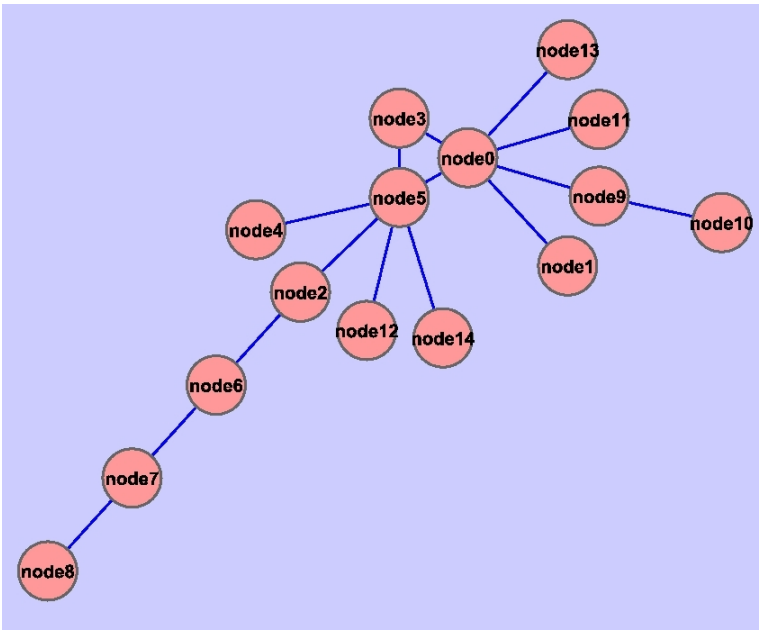

Fig. 3.1. A network where high diameter is due to a low number of nodes

the total number of nodes in the graph. Thus, a low diameter of a very big graph (with hundreds of nodes) is much more meaningful in term of compactness than a low diameter of a small graph (with few nodes). Notably, the diameter enables to measure the development of a network in time.

In biological terms

The diameter, and thus the compactness, of a biological network, for instance a protein-signaling network, can be interpreted as the overall easiness of the proteins to communicate and/or influence their reciprocal function. It could be also a sign of functional convergence. Indeed, a big protein network with low diameter may suggest that the proteins within the network had a functional co-evolution. The diameter should be carefully weighted if the graph is not fully connected (that is, there are isolated nodes).

3.2.3 Average Distance (AvD_G)

$$AvD_G = \frac{\sum_{i,j \in N} dist(i,j)}{n(n-1)}$$

where n is the number of nodes in G. The average distance (shortest path) of a graph G, corresponding to the sum of all shortest paths between vertex couples divided for the total number of vertex couples. Often it is not an integer. As for the diameter, it can be a simple and general parameter of graph "compactness", meaning with that the overall tendency of nodes to stay in proximity. Being an average, it can be somehow more informative than the diameter and can be also considered a general indicator of network "navigability". A high average distance indicates that the nodes are distant (disperse), implying little graph compactness. In contrast a low average distance indicates that all the nodes are in proximity and the graph is compact (figure 3.2). In quantitative terms, high and low are better defined when compared to the total number of nodes in the graph. Thus, a low average distance of a very big graph (with hundreds of nodes) is more meaningful in term of compactness than a low average distance of a small graph (with few nodes).

In biological terms

The average distance of a biological network, for instance a protein-signaling network, can be interpreted as the overall easiness of the proteins to communicate and/or influence their reciprocal function. It could be also a sign of functional convergence. Indeed, a big protein network with low average distance may suggest that the proteins within the network have the tendency to generate functional complexes and/or modules (although centrality indexes should be also calculated to support that indication).

3.2.4 Eccentricity ($C_{ecc}(v)$)

$$C_{ecc}(v) := \frac{1}{max\{dist(v,w) : w \in N\}}$$

The eccentricity is a node centrality index. The eccentricity of a node v is calculated by computing the shortest path between the node v and all other nodes in the graph, then the longest shortest path is chosen (let (v, K) where K is the

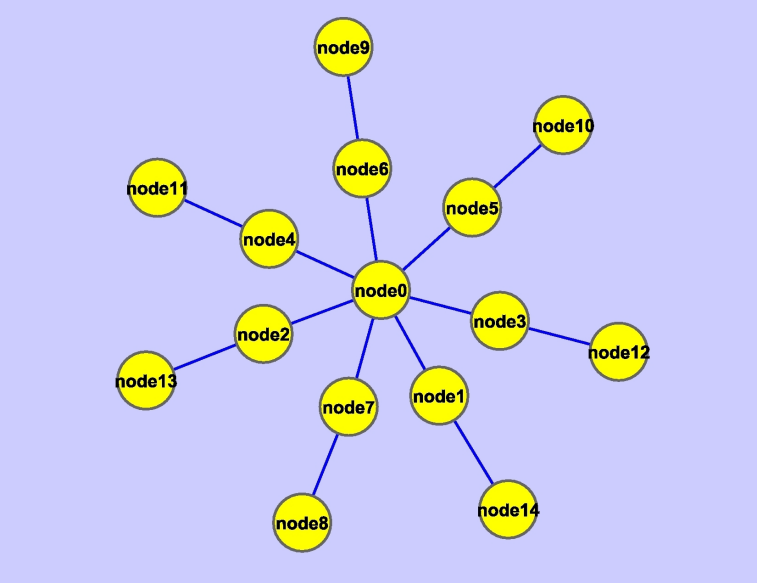

Fig. 3.2. A network with low diameter and average distance. The network is "compact"

most distant node from v). Once this path with length $dist(v, K)$ is identified, its reciprocal is calculated ($1/dist(v, K)$). By doing that, an eccentricity with higher value assumes a positive meaning in term of node proximity. Indeed, if the eccentricity of the node v is high, this means that all other nodes are in proximity. In contrast, if the eccentricity is low, this means that there is at least one node (and all its neighbors) that is far form node v. Of course, this does not exclude that several other nodes are much closer to node v. Thus, eccentricity is a more meaningful parameter if is high. Notably, high and low values are more significant when compared to the average eccentricity of the graph G calculated by averaging the eccentricity values of all nodes in the graph.

In biological terms

The eccentricity of a node in a biological network, for instance a protein-signaling network, can be interpreted as the easiness of a protein to be functionally reached by all other proteins in the network. Thus, a protein with high eccentricity, compared to the average eccentricity of the network, will be more easily influenced by the activity of other proteins (the protein is subject to a more stringent or complex regulation) or, conversely could easily influence several other proteins. In contrast, a low eccentricity, compared to the average eccentricity of the network, could indicate a marginal functional role (although this should be also evaluated with other parameters and contextualized to the network annotations).

3.2.5 Closeness ($C_{clo}(v)$)

$$C_{clo}(v) := \frac{1}{\sum_{w \in N} dist(v, w)}$$

The closeness is a node centrality index. The closeness of a node v is calculated by computing the shortest path between the node v and all other nodes in the graph, and then calculating the sum. Once this value is obtained, its reciprocal is calculated, so higher values assume a positive meaning in term of node proximity. Also here, high and low values are more meaningful when compared to the average closeness of the graph G calculated by averaging the closeness values of all nodes

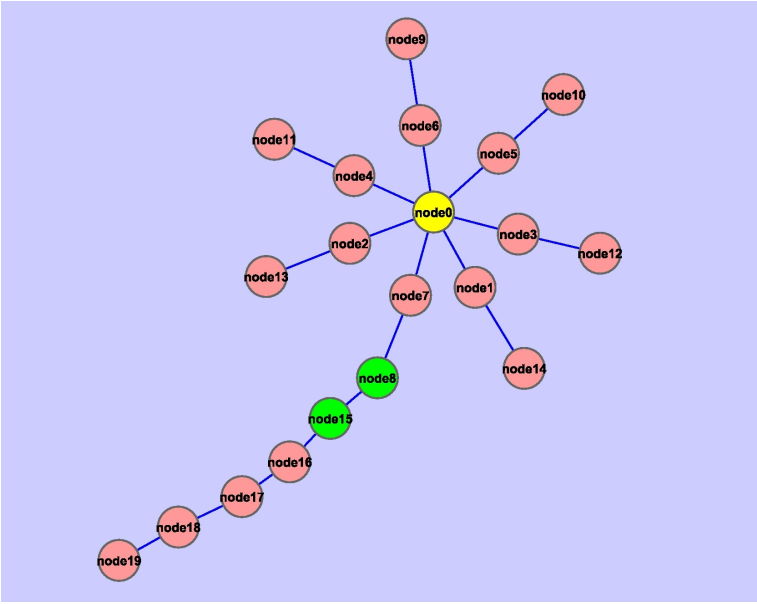

Fig. 3.3. The network shows the difference between eccentricity and closeness. The values of eccentricity are node0=0.14, node8=0.2, node15=0.2. The closeness values are node0=0.021, node8=0.017, node15=0.014. In this case node0 is closer than node8 and node15 to the most of nodes in the graph. Eccentricity value of node0 is smaller than value of node8 and node15, but this is due only to few nodes. If they are proteins this probably mean that node0 is fundamental for the most of reaction in the network, and that node8 and node15 are important only in reactions between few proteins.

in the graph. Notably, high values of closeness should indicate that all other nodes are in proximity to node v. In contrast, low values of closeness should indicate that all other nodes are distant from node v. However, a high closeness value can be determined by the presence of few nodes very close to node v, with other much more distant, or by the fact that all nodes are generally very close to v. Likewise, a low closeness value can be determined by the presence of few nodes very distant from node v, with other much closer, or by the fact that all nodes are generally distant from v. Thus, the closeness value should be considered as an average tendency to node proximity or isolation, not really informative on the

specific nature of the individual node couples. The closeness should be always compared to the eccentricity: a node with high eccentricity + high closeness is very likely to be central in the graph. Figure 3.3 shows an example of difference between closeness and eccentricity.

In biological terms

The closeness of a node in a biological network, for instance a protein-signaling network, can be interpreted as a measure of the possibility of a protein to be functionally relevant for several other proteins, but with the possibility to be irrelevant for few other proteins. Thus, a protein with high closeness, compared to the average closeness of the network, will be easily central to the regulation of other proteins but with some proteins not influenced by its activity. Notably, in biological networks could be also of interest to analyze proteins with low closeness, compared to the average closeness of the network, as these proteins, although less relevant for that specific network, are possibly behaving as intersecting boundaries with other networks. Accordingly, a signaling network with a very high average closeness is more likely organizing functional units or modules, whereas a signaling network with very low average closeness will behave more likely as an open cluster of proteins connecting different regulatory modules.

3.2.6 Radiality ($C_{rad}(v)$)

$$C_{rad}(v) := \frac{\sum_{w \in N}(\Delta_G + 1 - dist(v,w))}{n-1}$$

The radiality is a node centrality index. The radiality of a node v is calculated by computing the shortest path between the node v and all other nodes in the graph. The value of each path is then subtracted by the value of the diameter $+1$ ($\Delta G+1$) and the resulting values are summated. Finally, the obtained value is divided for the number of nodes -1 ($n-1$). Basically, as the diameter is the maximal possible distance between nodes, subtracting systematically from the diameter the shortest paths between the node v and its neighbors will give high values if the paths are short and low values if the paths are long. Overall, if the radiality is high this means that, with respect to the diameter, the node is generally closer to the other nodes, whereas, if the radiality is low, this means that the node is peripheral. Also here, high and low values are more meaningful when compared to the average radiality of the graph G calculated by averaging the radiality values of all nodes in the graph. As for the closeness, the radiality value should be considered as an average tendency to node proximity or isolation, not definitively informative on the centrality of the individual node. The radiality should be always compared to the closeness and to the eccentricity: a node with high eccentricity + high closeness+ high radiality is a consistent indication of a high central position in the graph.

In biological terms

The radiality of a node in a biological network, for instance a protein-signaling network, can be interpreted as the measure of the possibility of a protein to be

functionally relevant for several other proteins, but with the possibility to be irrelevant for few other proteins. Thus, a protein with high radiality, compared to the average radiality of the network, will be easily central to the regulation of other proteins but with some proteins not influenced by its activity. Notably, in biological networks could be also of interest to analyze proteins with low radiality, compared to the average radiality of the network, as these proteins, although less relevant for that specific network, are possibly behaving as intersecting boundaries with other networks. Accordingly, a signaling network with a very high average radiality is more likely organizing functional units or modules, whereas a signaling network with very low average radiality will behave more likely as an open cluster of proteins connecting different regulatory modules. All these interpretations should be accompanied to the contemporary evaluation of eccentricity and closeness.

3.2.7 Centroid value ($C_{cen}(v)$)

$$C_{cen}(v) := min\{f(v,w) : w \in N \setminus \{v\}\}$$

Where $f(v,w) := \gamma_v(w) - \gamma_w(v)$, and $\gamma_v(w)$ is the number of vertex closer to v than to w. The centroid value is the most complex node centrality index. It is computed by focusing the calculus on couples of nodes (v,w) and systematically counting the nodes that are closer (in term of shortest path) to v or to w. The calculus proceeds by comparing the node distance from other nodes with the distance of all other nodes from the others, such that a high centroid value indicates that a node v is much closer to other nodes. Thus, the centroid value provides a centrality index always weighted with the values of all other nodes in the graph. Indeed, the node with the highest centroid value is also the node with the highest number of neighbors (not only first) if compared with all other nodes. In other terms, a node v with the highest centroid value is the node with the highest number of neighbors separated by the shortest path to v. The centroid value suggests that a specific node has a central position within a graph region characterized by a high density of interacting nodes. Also here, high and low values are more meaningful when compared to the average centrality value of the graph G calculated by averaging the centrality values of all nodes in the graph.

In biological terms

The centroid value of a node in a biological network, for instance a protein-signaling network, can be interpreted as the probability of a protein to be functionally capable of organizing discrete protein clusters or modules. Thus, a protein with high centroid value, compared to the average centroid value of the network, will be possibly involved in coordinating the activity of other highly connected proteins, altogether devoted to the regulation of a specific cell activity (for instance, cell adhesion, gene expression, proliferation etc.). Accordingly, a signaling network with a very high average centroid value is more likely organizing functional units or modules, whereas a signaling network with very low average centroid value will behave more likely as an open cluster of proteins connecting different regulatory modules. It can be useful to compare the centroid value to algorithms detecting

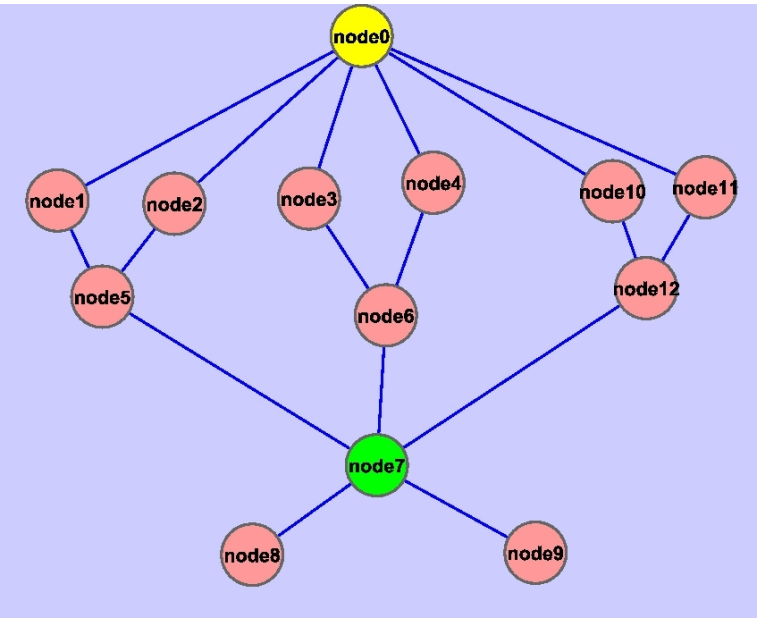

Fig. 3.4. The network shows the difference between centroid and closeness. Here node0 has highest centroid value (centroid=1, closeness=0,04) and node7 has highest closeness value (centroid=-1, closeness= 0,05).

dense regions in a graph, indicating protein clusters, such as, for instance, MCODE [7].

3.2.8 Stress ($C_{str}(v)$)

$$C_{str}(v) := \sum_{s \neq v \in N} \sum_{t \neq v \in N} \sigma_{st}(v)$$

The stress is a node centrality index. Stress is calculated by measuring the number of shortest paths passing through a node. To calculate the stress of a node v, all shortest paths in a graph G are calculated and then the number of shortest paths passing through v is counted. A stressed node is a node traversed by a high number of shortest paths. Notably and importantly, a high stress values does not automatically implies that the node v is critical to maintain the connection between nodes whose paths are passing through it. Indeed, it is possible that two nodes are connected by means of other shortest paths not passing through the node v. Also here, high and low values are more meaningful when compared to the average stress value of the graph G calculated by averaging the stress values of all nodes in the graph.

In biological terms

The stress of a node in a biological network, for instance a protein-signaling network, can indicate the relevance of a protein as functionally capable of holding

together communicating nodes. The higher the value the higher the relevance of the protein in connecting regulatory molecules. Due to the nature of this centrality, it is possible that the stress simply indicates a molecule heavily involved in cellular processes but not relevant to maintain the communication between other proteins.

3.2.9 S.-P. Betweenness ($C_{spb}(v)$)

$$C_{spb}(v) := \sum_{s \neq v \in N} \sum_{t \neq v \in N} \delta_{st}(v)$$

where

$$\delta_{st}(v) := \frac{\sigma_{st}(v)}{\sigma_{st}}$$

The S.-P. Betweenness is a node centrality index. It is similar to the stress but

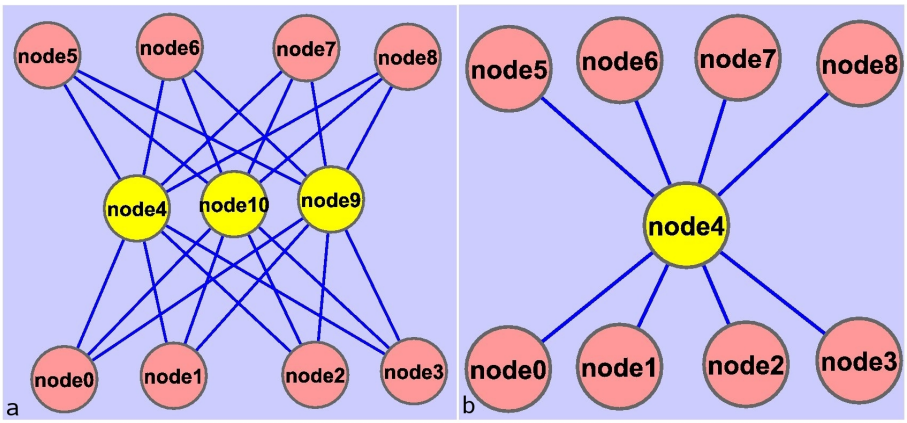

Fig. 3.5. Betweenness vs Stress. In fig. a node4, node10, and node9 present high value of stress (= 56), and the same value of betweenness (=18.67). In fig.b, node4 presents the same value of stress of fig.a and higher value of betweenness(=56). This is because the number of shortest paths passing through node4 is the same in the two network. But in the second network node4 is the only node connecting the two parts of the network. In this sense betweenness is more precise than stress giving also information on how the node is fundamental in the network. If we remove node4 in fig.a, the connection between the node in the network don't change so much. If we remove node 4 from fig.b the network is completely disconnected.

provides a more elaborated and informative centrality index. The betweenness of a node n is calculated considering couples of nodes $(v1, v2)$ and counting the number of shortest paths linking $v1$ and $v2$ and passing through a node n. Then, the value is related to the total number of shortest paths linking $v1$ and $v2$. Thus, a node can be traversed by only one path linking $v1$ and $v2$, but if this path is the only connecting $v1$ and $v2$ the node n will score a higher betweenness value (in the stress computation would have had a low score). Thus, a high S.-P. Betweenness score

means that the node, for certain paths, is crucial to maintain node connections. Notably, to know the number of paths for which the node is critical it is necessary to look at the stress. Thus, stress and S.-P. Betweenness can be used to gain complementary information. Further information could be gained by referring the S.-P. Betweenness to node couples, thus quantifying the importance of a node for two connected nodes. Also here, high and low values are more meaningful when compared to the average S.-P. Betweenness value of the graph G calculated by averaging the S.-P. Betweenness values of all nodes in the graph.

In biological terms

The S.-P. Betweenness of a node in a biological network, for instance a protein-signaling network, can indicate the relevance of a protein as functionally capable of holding together communicating proteins. The higher the value the higher the relevance of the protein as organizing regulatory molecule. The S.-P. Betweenness of a protein effectively indicates the capability of a protein to bring in communication distant proteins. In signaling modules, proteins with high S.-P. Betweenness are likely crucial to maintain functionally and coherence of signaling mechanisms.

3.3 Normalization and relative centralities

Once centralities have been computed, the question that arise immediately is what does it means to have a centrality of, for example, 0.4 for a node? This clearly depends on different parameters as the number of nodes in the network, the maximum value of the centrality and on the topological structure of the network. In order to compare centrality scores between the elements of a graph or between the elements of different graphs some kind of normalization of centrality values is needed. Common normalizations applicable to most centralities are to divide each value by the maximum centrality value or by the sum of all values. We will use the second in the rest of the thesis defining it as the relative centralities value. So, given a centrality C, $C(G, n)$ is the value of the centrality of node n in the network G. We define the relative centrality value of node n as:

$$relC(G, n) = \frac{C(G, n)}{\sum_{i \in N} C(G, i)}$$

So a relative centrality of 0.4 means that the node has the 40% of the total centrality of the network. This definition can be applied to all centralities except of centroid value, since it can have negative values. Relative centrality value will be used for definition of node centrality interference in chapter 6.

3.4 Conclusions

A review of nodes centralities have been presented. The centralities introduced have been chosen for their biological relevance, and a possible biological meaning for each centrality have been hypothesized. Normalization of centralities, useful for comparison between nodes in a network and between nodes of different networks have also been considered.

4

CentiScaPe

In this chapter we describe the CentiScaPe software [52], a Cytoscape [16], [54] plugin we implemented to calculate centrality values and integrating topological analysis of networks with lab experimental data. The vast amount of available experimental data generating annotated gene or protein complex networks has increased the quest for networks analysis tools. Biological networks are usually represented as graphs, where the nodes are biological entities (such as cells, genes, proteins or metabolites) and the edges are functional and/or physical interactions between them. Visualization and analysis tools are needed to understand individual node functions masked by the overall network complexity. Several techniques suitable to network structural analysis exist, such as the analysis of the global network structure [1], network motifs [45], network clustering [34] and network centralities [63]. Particularly, centralities are node parameters that can identify nodes having a relevant position in the overall network architecture (see chapter 3). Cytoscape is an excellent visualization and analysis tool with the analysis features greatly enhanced by plug-ins. Plug-in such as NetworkAnalyzer [6] computes some node centralities but does not allow direct integration with experimental data. Applications such as VisANT [35], and Centibin [39] calculate centralities, although they either calculate fewer centralities or are not suitable to integration with experimental data. Figure 4.1 shows a comparative evaluation of CentiScaPe and other applications. CentiScaPe is the only Cytoscape plug-in that computes several centralities at once. In CentiScaPe, computed centralities can be easily correlated between each other or with biological parameters derived from the experiments in order to identify the most significant nodes according to both topological and biological properties. Functional to this capability is the scatter plot by value options, which allows easy correlating node centrality values to experimental data defined by the user. Particularly this feature allows a new way to face the analysis of biological networks, integrating topological analysis and lab experimental data. This new approach is described in chapter 5. At present version 1.1 is available and it is downloaded with a rate of about 140 downloads for month (see Cytoscape website for download statistics). First results using CentiScaPe have been published in [44] and [53] and presented at 48th ASCB annual meeting [11]. CentiScaPe is also used at GlaxoSmithKline computational biology labs.
Availability: CentiScaPe can be downloaded via the Cytoscape web site:

Parameter for single node	Centiscape	Network analyzer	Visant	Centibin
Degree	Yes	Yes	Yes	Yes
Radiality	Yes	Yes	No	Yes
Closeness	Yes	Yes	No	Yes
Stress	Yes	Yes	No	Yes
Betweenness	Yes	Yes	No	Yes
Centroid value	Yes	No	No	Yes
Eccentricity	Yes	Yes	No	Yes
Scatter plot between centralities	Yes	No	No	No
Scatter plot with experimental data	Yes	No	No	No
Highlighting node according to centralities values (more/less than a threshold, and/or combination of query)	Yes	No	No	No
Plot by node (graphical output of centralities values for single node)	Yes	No	No	No

Fig. 4.1. Features of CentiScaPe versus Network Analyzer, Visant, Centibin

http://chianti.ucsd.edu/cyto_web/plugins/index.php.
Tutorial, centrality descriptions and example data are available at:
http://profs.sci.univr.it/scardoni/centiscape/centiscapepage.php

4.1 System overview

CentiScaPe computes several network centralities for undirected networks. Computed parameters are: Average Distance, Diameter, Degree, Stress, Betweenness, Radiality, Closeness, Centroid Value and Eccentricity. Plug-in help and on-line files are provided with definition, description and biological significance for each centrality (see chapter 3). Min, max and mean values are given for each computed centrality. Multiple networks analysis is also supported. Centrality values appear in the Cytoscape attributes browser, so they can be saved and loaded as normal Cytoscape attributes, thus allowing their visualization with the Cytoscape mapping core features. Once computation is completed, the actual analysis begins, using the graphical interface of CentiScaPe.

4.2 Algorithm and implementation

To calculate all the centralities the computation of the shortest path between each pair of nodes in the graph is needed. The algorithm for the shortest path is the well known Dijkstra algorithm [23]. There are no costs in our network edges, so in our case the algorithm keeps one as the cost of each edge. To compute Stress and Betweenness we need all the shortest paths between each pair of nodes and not only a single shortest path between each pair. To do this the Dijkstra algorithm has been adjusted as follows. Exploring the graph when calculating the shortest path between two nodes s and t, the Dijkstra algorithm keep for each node n a predecessor node p. The predecessor node is the node that is the predecessor of n in one of the shortest paths between s and t. So in case of the Dijkstra algorithm, only one predecessor for each node is needed. To have all the shortest paths, we replace the predecessor p with a set of predecessors for each node n. The set of predecessors of the node n is the set of all the predecessors of the node n in the shortest paths set between s and t, i.e. one node is in the set of predecessors of n if it is a predecessor of n in one of the shortest paths between s and t containing n. Once the predecessors set of each node n has been computed, also the tree of all the shortest paths between s and t can be easily computed. Once we have computed all the shortest paths between each pairs of nodes of our network, the algorithm of each centralities comes directly from the formal definition of each centrality. n the case of betweenness, to decrease the computational complexity of the algorithm, further considerations can be done. A vertex v is in the shortest path betweenn s and t if $d(s,t) = d(s,v) + d(v,t)$. If this is the case, the number of shortest paths using v is computed as $\sigma_{st}(v) = \sigma_{sv}\sigma_{vt}$. Computational complexity for each centrality value is shown in table 4.1. A well done description of this and

Centrality	Computational complexity
Diameter	O($mn + n^2$)
Average distance	O($mn + n^2$)
Degree (deg(v))	O(n)
Radiality (rad(v))	O($mn + n^2$)
Closeness (clo(v))	O($mn + n^2$)
Stress (str(v))	O($mn + n^2$)
Betweenness (btw(v))	O(n^3)
Centroid Value (cen(v))	O($mn + n^2$)
Eccentricity (ecc(v))	O($mn + n^2$)

Table 4.1. Computational complexity for each centrality value. n is the number of nodes and m is the number of edges in the network.

other centralities algorithms can be found in [42]. CentiScaPe is written in Java as a Cytoscape plugin, in order to exploit all the excellent features of Cytoscape and to reach the larger number of users. The Java library JFreechart [27] has been used for some graphic features.

4.3 Using CentiScaPe

Once CentiScaPe have been started, the main menu will appear as a panel on the left side of the Cytoscape window as shown in figure 4.2. The panel shows to the

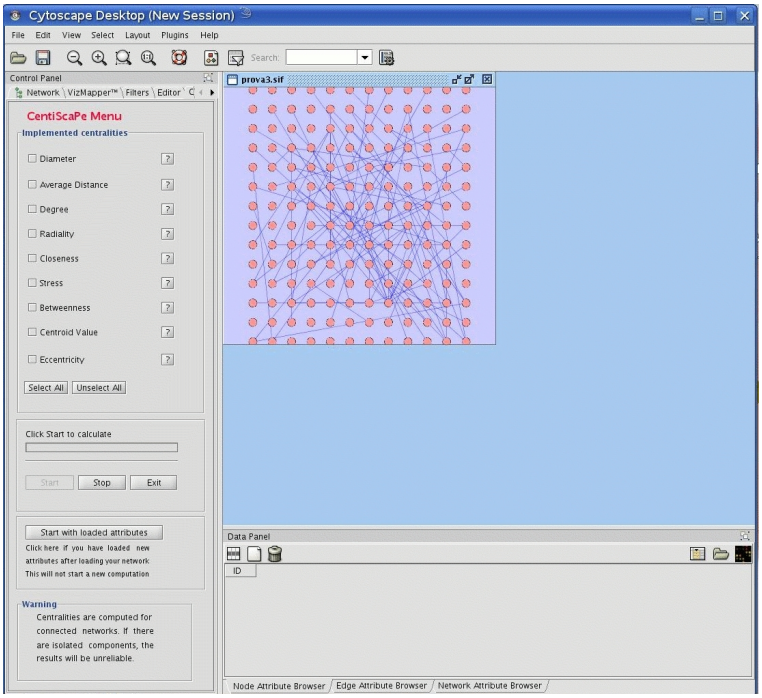

Fig. 4.2. CentiScaPe starting panel. On the left side the main menu appears, to select the centralities for computation.

user the list of centralities and the user can select all the centralities or some of them. A banner and a node worked count appear during the computation to show the computation progress. The numerical results are saved as node or network attributes in the Cytoscape attributes browser, depending on the kind of parameters, so all the Cytoscape features for managing attributes are supported: after the computation the centralities are treated as normal Cytoscape attributes. The value of each centrality is saved as an attribute with name "CentiScaPe" followed by the name of the centrality. For example the eccentricity is saved in the Cytoscape attributes browser as "CentiScaPe Eccentricity". Since the Cytoscape attributes follow the alphabetical order this make it easy to find all the centralities in the attributes browser list. There are two kind of centralities: network centralities, and node centralities.

Network Centralities

The network centralities concern the entire network and not the single nodes. They are the Diameter and the Average Distance. They will appear on the data panel selecting the Cytoscape network attribute browser.

Node Centralities

All other centralities are node parameters and refer to the single nodes. So they will appear on the attribute browser as node attributes. Using the Node attribute browser the user can select one or more of them as normal attributes. CentiScaPe also calculates the min, max and mean value for each centrality. Since they are network parameters they appear on the Network attribute browser. As for the other attributes the user can save and load network and node parameters to/from a file. If an attribute is already loaded or calculated and the user try to recalculate it, a warning message will appear.

4.3.1 CentiScaPe Results Panel

If one or more node centralities have been selected, a result panel will appear on the right side of the Cytoscape window (figure 4.4). The first step of the analysis is the Boolean logic-based result panel of CentiScaPe (figure 4.3). It is possible, by using the provided sliders in the Results Panel of Cytoscape, to highlight the nodes having centralities values that are higher, minor or equal to a threshold value defined by the user. The slider threshold is initialized to the mean value of each centrality so all the nodes having a centrality value less or equal to the threshold are highlighted by default in the network view with a color depending on the selected visual mapper of Cytoscape (yellow in figure 4.4). So if one centrality has been selected, all the nodes having a value less or equal the threshold for that centrality are highlighted. If more than one centralities has been selected they can be joined with an AND or an OR operator. If the AND operator is selected, the nodes for which all the values are less or equal the corresponding threshold are highlighted. If the OR operator is selected the nodes for which at least one value is less or equal the corresponding threshold are highlighted. The possibility of highlighting also the nodes that are more/equal than the threshold is supported. So the user can select the more/equal option for some centralities, the less/equal option for others and can join them with the AND or the OR operator. If necessary, one or more centralities can be deactivated. This feature can immediately answer to questions as: Which are the nodes having high Betweenness and Stress but low Eccentricity? Notably, the threshold can also be modified by hand to gain in resolution. In figure 4.4 are highlighted all the nodes having centralities values more/equal than the corresponding threshold (AND operator). Once the nodes have been selected according to their node-specific values, the corresponding subgraph can be extracted and displayed using normal Cytoscape core features.

4.3.2 Graphic output

Two kind of graphical outputs are supported: plot by centrality and plot by node, both allowing analysis that are not possible with other centralities tools. The user

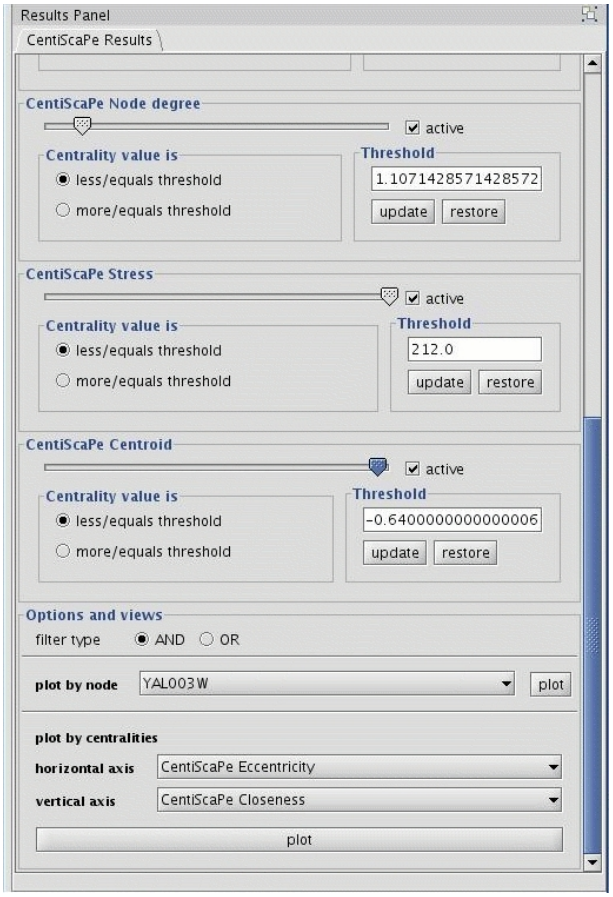

Fig. 4.3. CentiScaPe results panel. Here the less/equals option for Node degree, Stress and Centroid have been joined with the AND operator. and

can correlate centralities between them or with experimental data, such as, for example, gene expression level or protein phosphorylation level (plot by centrality), and can analyze all centralities values node by node (plot by node). Example of plot by node and plot by centrality are shown in figure 4.5. Graphics can be saved to a jpeg file.

Plot by centrality

The plot by centrality visualization is an easy and convenient way to discriminate nodes and/or group of nodes that are most relevant according to a combination of two selected parameters. It shows correlation between centralities and/or other quantitative node attributes, such as experimental data from genomic and/or proteomic analysis. The result of the plot by centrality option is a chart where each individual node, represented by a geometrical shape, is mapped to a Cartesian axis. In the horizontal and vertical axis, the values of the selected attributes are

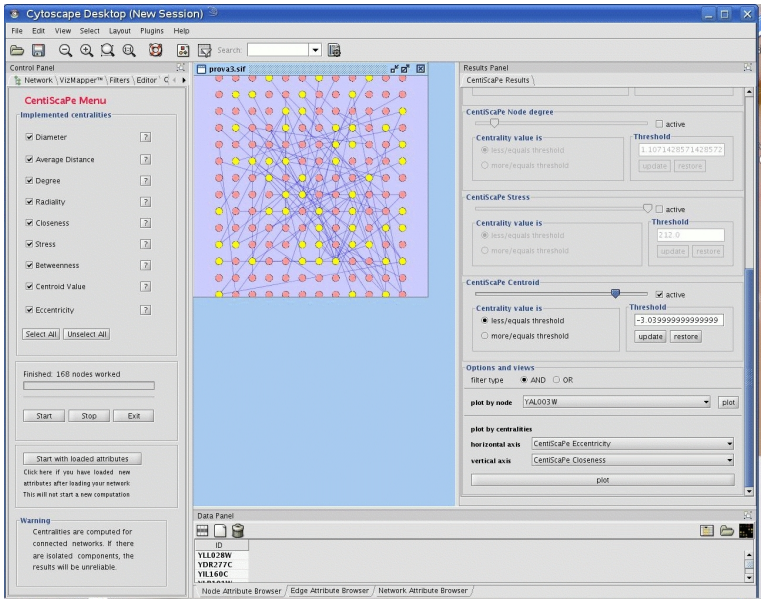

Fig. 4.4. A computation results of CentiScaPe. All nodes having centrality values more/equal than the corresponding threshold (AND operator) are highlighted.

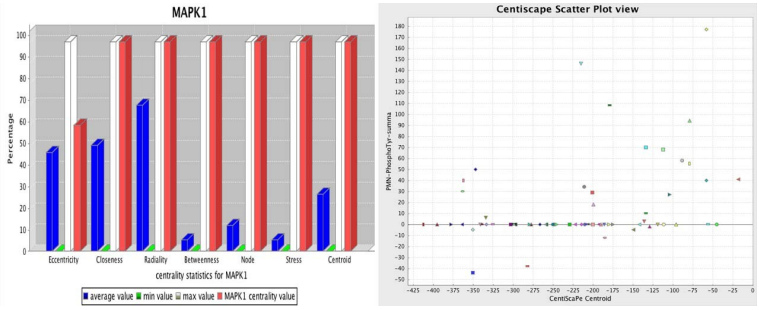

Fig. 4.5. Example of plot by node (left) and plot by centrality (right) with CentiScaPe.

reported. Most of the relevant nodes are easily identified in the top-right quadrant of the chart. Figure 4.6 shows a plot of centroid values over intensity of protein tyrosine phosphorylation in the human kino-phosphatome network derived from the analysis of human primary polymorphonuclear neutrophils (PMNs) stimulated with the chemoattractant IL-8 (see chapter 5). The proteins having high values for both parameters likely play a crucial regulatory role in the network. The user can plot in five different ways: centrality versus centrality, centrality versus experimental data, experimental data versus experimental data, a centrality versus itself and an experimental data versus itself. Notably, a specific way to use the plot function is to visualize the scatter plot of two experimental data attributes. This is an extra function of the plug-in and can be used in the same way of the

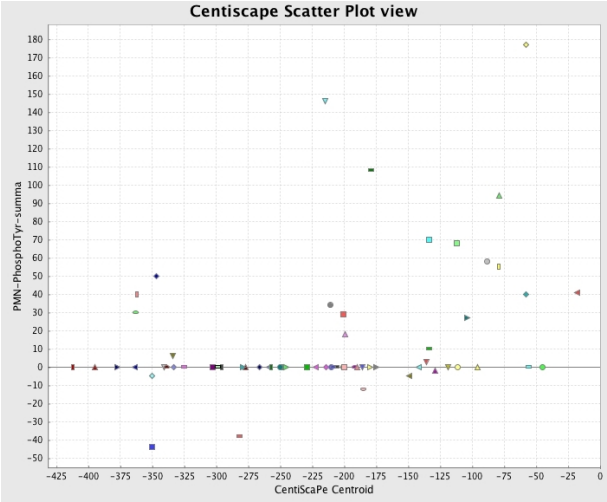

Fig. 4.6. Integration of topological analysis with experimental data. Centroid values are plotted over protein phosphorylation levels in tyrosine. Relevant nodes are easily identified in the top-right quadrant.The centralities values and the node identifier appear in CentiScaPe by passing with the mouse over each geometrical shape in the plot.

centrality/centrality option and centrality/experimental attribute option. If the plot by centrality option is used selecting the same centrality (or the same experimental attribute) for both the horizontal and the vertical axis, result is an easy discrimination of nodes having low values from nodes having high values of the selected parameter (figure 4.7). Thus, the main use of the plot by centrality feature is to identify group of nodes clustered according to combination of specific topological and/or experimental properties, in order to extract sub-networks to be further analyzed. The combination of topological properties with experimental data is useful to allow more meaningful predictions of sub-network function to be experimentally validated.

Plot by node

The plot by node option, another unique feature of CentiScaPe, shows for every single node the value of all calculated centralities represented as a bar graph. The mean, max and min values are represented with different colors. To facilitate the visualization, all the values in the graph are normalized and the real values appear when pointing the mouse over a bar. Figure 4.8 shows, as an example, the values for the MAPK1 calculated from the global human kino-phosphatome (see chapter 5).

4.4 Conclusions

CentiScaPe is a versatile and user-friendly bioinformatic tool to integrate centrality-based network analysis with experimental data. CentiScaPe is completely inte-

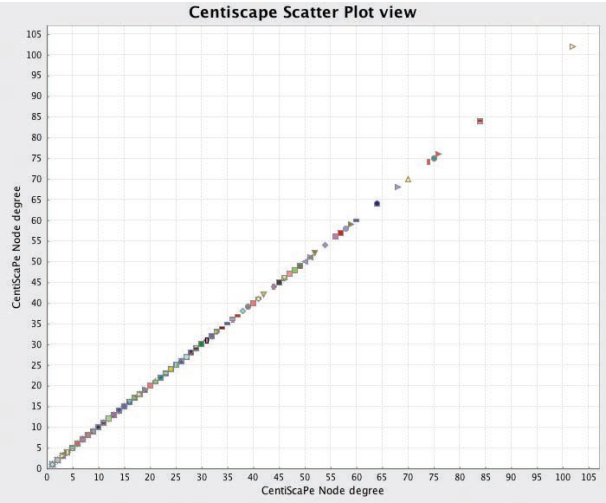

Fig. 4.7. A scatter plot degree over degree. As expected this generate a linear distribution: nodes having low values are easily identified in the bottom/left quadrant of the graph. Nodes with high degree values are in the top/right quadrant.

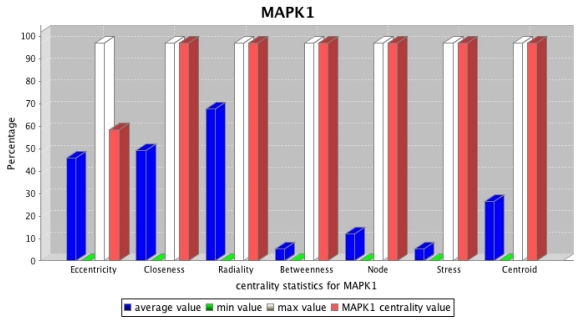

Fig. 4.8. A plot by node example. For each centrality the specific node value (red), the mean value (blue), the min value (green), and the max value (white) is shown.

grated into Cytoscape and the possibility of treating centralities as normal attributes permits to enrich the analysis with the Cytoscape core features and with other Cytoscape plug-ins. The analysis obtained with the Boolean-based result panel, the plot by node and the plot by centrality options give meaningful results not accessible to other tools and allow easy categorization of nodes in large complex networks derived from experimental data.

5

A real world example: Centralities in the human kino-phosphatome

In this chapter a new protocol of analysis of protein interaction networks is introduced through an example of analysis of the human kino-phosphatome. The analysis starts with the extraction of known interactions from a protein interactome. In our case we consider kinases and phosphatases interaction i.e. those interaction regarding activation and inhibition of proteins in the network. Kinases and phosphatases are enzymes involved in the phosphorilation process: they transfer or remove phosphate groups to/from a protein regulating in this way its activity. Substantially kinases and phosphatases activate or inhibit other proteins. In a kino-phosphatome network this process generates a cascade of activations and inhibitions of proteins corresponding to the transmissions of signals and to the control of complex processes in cells. The approach to the kino-phosphatome network is to identify the most important proteins for their centrality values and then to analyze with a lab experiment their activation level. After this, using the CentiScaPe feature of integrating topological analysis and data from lab experiments, those values are integrated and those nodes important for both centralities value and activation level are easily identified. This introduce a new way of facing the analysis of a protein interaction network based on the Strogatz assertion that in a biological network "Structure always affects function" [57]. Instead of concentrating the analysis, as usual, on the global properties of the network (such degree distribution, centralities distribution, and so on) we consider in a cause-effect point of view single nodes of the network relating their centrality values (cause) with activation level (effects). Most of the contents of this chapter have been published on [52].

5.1 Centralities analysis

The protocol used for the analysis of the human kino-phosphatome network is the following:

- The nodes of interest are extracted from the global network, resulting in a subnetwork to analyze (in our example the subnetwork of human kinases and phosphatases have been extracted from a human proteins interactome).

- The centralities values are computed with CentiScaPe. A subnetwork of proteins with all centrality values over the average is extracted.
- The lab experiment identifies which of these proteins present high phosphorilation level (in our example in tyrosine and threonine).
- Using CentiScaPe, lab experimental data and centrality values are integrated, so proteins with high level of activation and high centralities values are easily identified.
- Further experiments and analysis should be focused on these proteins.

This protocol have been applied as follows. A global human protein interactome data-set (Global Kino-Phosphatome network), including 11120 nodes and 84776 unique undirected interactions (IDs = HGNC), was compiled from public data-bases (HPRD, BIND, DIP, IntAct, MINT, others; see [52] on-line file GLOBAL-HGNC.sif) between human protein kinases and phosphatases. The re-

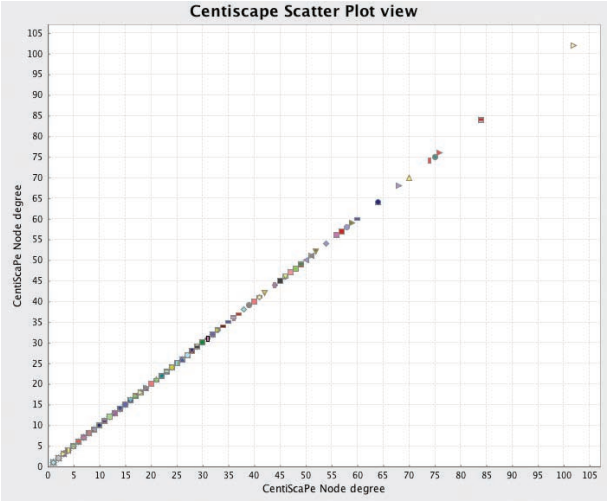

Fig. 5.1. A scatter plot degree over degree. As expected this generate a linear distribution. Notably the distribution is not uniform: many nodes display low degree and only few nodes with high degree, according to the scale-free architecture of biological network.

sulting sub-network, a kino-phosphatome network, consisted of 549 nodes and 3844 unique interactions (see [52] on-line files Table S4 and Kino-Phosphatome.sif), with 406 kinases and 143 phosphatases. The kino-phosphatome network did not contain isolated nodes. We used CentiScaPe to calculate centrality parameters. A first general overview of the global topological properties of the kino-phosphatome network comes from the min, max and average values of all computed centralities along with the diameter and the average distance of the network (table 5.1). These data provide a general overview of the global topological properties of the kino-phosphatome network. For instance, an average degree equals to 13.5 with an average distance of 3 may suggest a highly connected network in which proteins are strongly functionally interconnected. Computation of network centralities allowed

CentiScaPe Average Distance	3.0292037280789224
CentiScaPe Betweenness Max value	20159.799011925716
CentiScaPe Betweenness mean value	1112.0036429872616
CentiScaPe Betweenness min value	0.0
CentiScaPe Centroid Max value	18.0
CentiScaPe Centroid mean value	-393.07285974499086
CentiScaPe Centroid min value	-547.0
CentiScaPe Closeness Max value	8.771929824561404E-4
CentiScaPe Closeness mean value	6.175318530305184E-4
CentiScaPe Closeness min value	3.505082369435682E-4
CentiScaPe Diameter	8.0
CentiScaPe Eccentricity Max value	0.25
CentiScaPe Eccentricity mean value	0.18407494145199213
CentiScaPe Eccentricity min value	0.125
CentiScaPe Radiality Max value	6.91970802919708
CentiScaPe Radiality mean value	5.970796271921072
CentiScaPe Radiality min value	3.7937956204379564
CentiScaPe Stress Max value	210878.0
CentiScaPe Stress mean value	11537.009107468124
CentiScaPe Stress min value	0.0
CentiScaPe degree Max value	102.0
CentiScaPe degree mean value	13.5591985428051
CentiScaPe degree min value	1.0

Table 5.1. Global values of the kino-phosphatome network computed using CentiScaPe. The table includes min, max and mean value for each centrality and also the global parameter Diameter and Average Distance.

a first ranking of human kinases and phosphatases according to their central role in the network (see [52] on-line files Table S6 reporting all node-by-node values of different centralities). To facilitate the identification of nodes with the highest scores we applied the "plot by centrality" feature of CentiScaPe. A first plotting degree over degree generated a linear distribution, as expected (see fig. 5.1). However, it is evident that the distribution is not uniform, with the majority of nodes having a similar low degree and very few having very high degree. This is consistent with the known scale-free architecture of biological networks [37]. The scale-free topology of the kino-phosphatome network was also confirmed with Network Analyzer [6]. A total of 186 nodes (164 kinases and 22 phosphatases) displayed a degree over the average. The top 10 degrees (64 to 102) were all kinases, with MAPK1 showing the highest degree (102). Notably, MAPK1 displayed the highest score for most of the computed centralities (fig. 5.2), suggesting its central regulatory role in the kino-phosphatome. In contrast, PTPN1 had the highest degree, 46, between all phosphatases (top 31 among all nodes) and had a rather high score also for other centralities (fig. 5.3). Thus, degree analysis suggests that MAPK1 and PTPN1 are the most central kinase and phosphatase, respectively. To further support this suggestion we analyzed the centroid. Plotting centroid over centroid provided a linear distribution, as expected and as for the degree, also here the distribution was not uniform (fig 5.1 and fig. 5.4). Average centroid was -393. 242 nodes (206

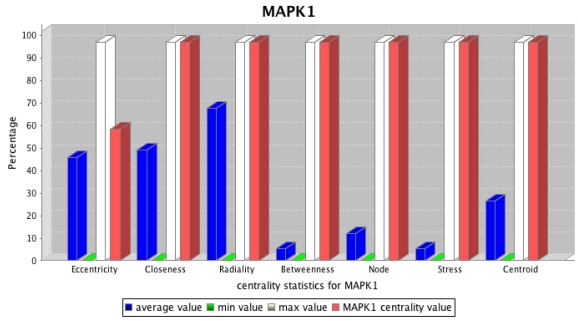

Fig. 5.2. Network analysis of human kino-phosphatome. The protein kinase MAPK1 shows high centralities values for most of the computed centralities suggesting its central role in the network structure and function. For each centrality the specific node value (red), the mean value (blue), the min value (green), and the max value (white) is shown.

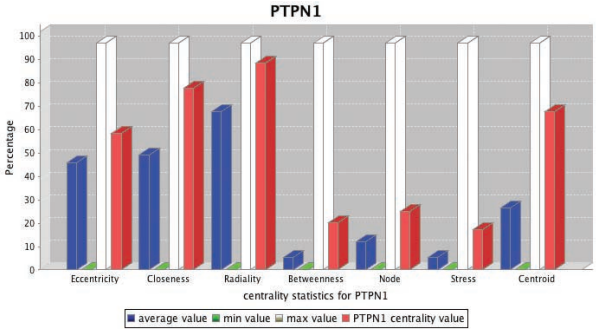

Fig. 5.3. The plot by node representation for PTPN1. The phosphatase PTPN1 presents the highest degree between all the phosphatases and a rather high score for other centralities. This suggests that PTPN1 may play a central regulatory role in the network. For each centrality the specific node value (red), the mean value (blue), the min value (green), and the max value (white) is shown.

kinases and 36 phosphatases) displayed a centroid over the average. The top 10 centroid (-79 to 8) were all kinases, with MAPK1 showing the highest centroid value (18). PTPN1 had the highest centroid value, -154, between all phosphatases (top 22 among all nodes). Thus, as for the degree, also the centroid value analysis suggests a possible scale-free distribution, with MAPK1 and PTPN1 being the most central kinase and phosphatase, respectively. This conclusion is also easily evidenced by plotting the degree over the centroid (fig. 5.5).

Here MAPK1 appears at the top right of the plot and PTPN1 is present in the top most dispersed region of the plot, thus suggesting their higher scores. Interestingly, from the analysis is evident a non-linear distribution of nodes, with few dispersed nodes occupying the top right quadrant of the plot (i.e. high degree and high centroid): these nodes can potentially represent particularly important regula-

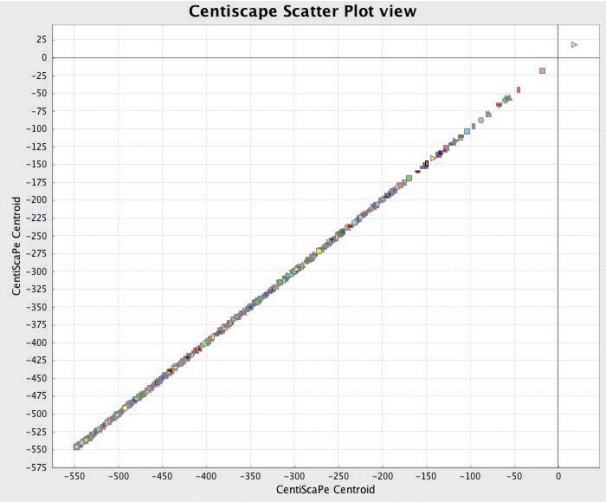

Fig. 5.4. A scatter plot of centroid over centroid. As expected, this generate a linear distribution. Notably, as for the degree, the distribution is not uniform: many nodes display low centroid whereas only few nodes have high centroid.

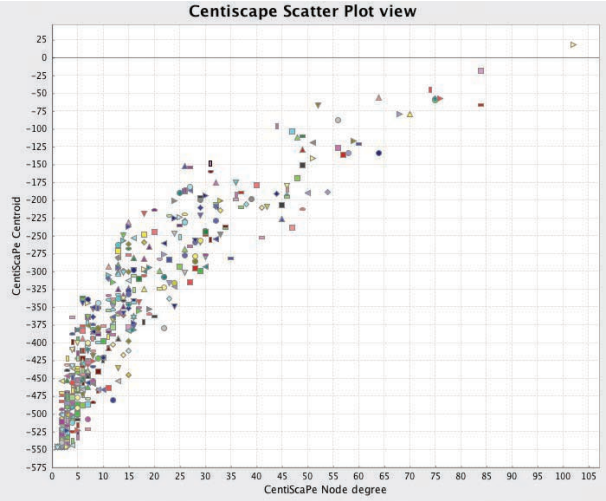

Fig. 5.5. A plot by centralities representation of degree over centroid. In the top right of the plot appear the nodes having high values of both degree and centroid (including MAPK1).

tory kinases and phosphatases. This kind of analysis can be iterated by evaluating all other centralities. To extract the most relevant nodes according to all centrality values we used CentiScaPe to select all nodes having all centrality values over the average. Upon filtering we obtained a kino-phosphatome sub-network (fig.5.6) con-

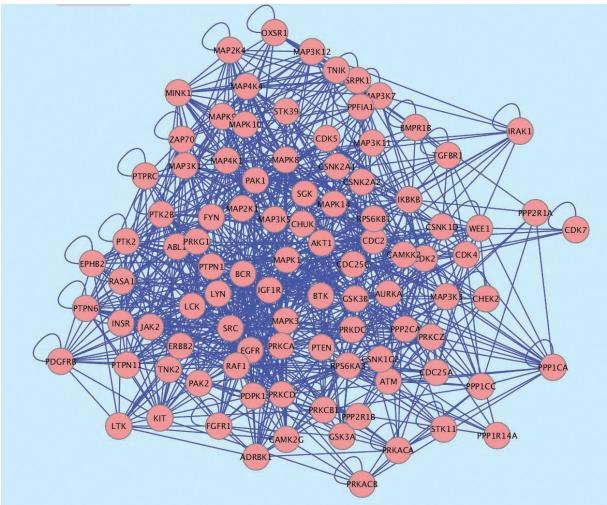

Fig. 5.6. The subnetwork resulting from the extraction of all nodes having all the centralities over the average. The network consist of 97 nodes (82 kinases and 15 phosphatases) and 962 interactions.

sisting of 97 nodes (82 kinases and 15 phosphatases) and 962 interactions (see [52] on-line files Table S7, and K-P sub-network.sif).

This sub-network possibly represents a group of highly interacting kinases and phosphatases displaying a critical role in the regulation of protein phosphorylation in human cells. Further analysis with CentiScaPe or other analysis tools, such as MCODE [7] or Network Analyzer [6], performing a Gene Ontology database search [5], or adding functional annotation data, may allow a deeper functional exploration of this sub network. The regulatory role of proteins belonging to the kino-phosphatome network may be also experimentally tested in a context-selective manner. Indeed, the centrality analysis by CentiScaPe can be even more significant by superimposing experimental data. To test this possibility, we focused the analysis on human polymorphonuclear neutrophils (PMNs).

5.2 Phosphoproteomic analysis of chemoattractant stimulated human PMNs

5.2.1 Human primary polymorphonuclear cells isolation

Human primary polymorphonuclear cells (PMNs) were freshly isolated form whole blood of healthy donors by ficoll gradient sedimentation. Purity of PMN preparation was evaluated by flow cytometry and estimated to about 95% of neutrophils. Isolated PMNs were kept in culture at 37^oC in standard buffer (PBS, $1mM\ CaCl_2$, $1mM\ MgCl_2$, 10% FCS, $pH7.2$) and used within 1 hour. Viability before the assays was more than 90%.

5.2.2 Human primary polymorphonuclear cell stimulation

Human neutrophils were resuspended in standard buffer at $10^7/ml$ and stimulated under stirring at 37^oC for 1 min. with the classical chemoattractant fMLP ($100nM$). Stimulation was blocked by directly disrupting the cells for 10 min. in ice-cold lysis buffer containing: $20mM$ MOPS, $pH7.0$, $2mM$ EGTA, $5mM$ EDTA, $30mM$ sodium fluoride, $60mM$ β-glycerophosphate, $20mM$ sodium pyrophosphate, $1mM$ sodium orthovanadate, $1mM$ phenylmethylsulfonylfluoride, $3mM$ benzamidine, $5\mu M$ pepstatin A, $10\mu M$ leupeptin, 1% Triton X-100. Lysates were clarified by centrifugation at $12.000xg$ for 10 min. and kept at 80^oC until further processing.

5.2.3 Evaluation of protein phosphorylation

Protein phosphorylation was evaluated both qualitatively and quantitatively by using the Kinexus protein array service (see [40]). Kinexus provides a complete service for high throughput proteomic and phosphoproteomic high sensitive analysis of cell lysed samples, allowing detection of more than 800 proteins, including about 200 phosphorylated proteins (about 350 phospho-sites) by means of in-house validated antibody microarrays (see [41]). $100\mu l$ of frozen samples of lysed PMNs (about $1mg/ml$ protein concentration) have been sent to Kinexus for the analysis. Phosphoproteomic antibody microarray data have been delivered by email and subsequently elaborated to extract values of protein phosphorylation of control versus agonist-triggered samples. (phosphorylation data files are available on-line: see [52] PMN-PhosphoSer.NA, PMN-PhosphoTyr.NA, PMN-PhosphoThr.NA).

5.3 Combining topological analysis and experimental data

Data about protein phosphorylation were used as bioinformatic probes and node attributes to extract, from the Global Kino-Phosphatome network, subnetworks of protein phosphorylation, to be analyzed with CentiScaPe Experimental data were loaded as node attributes in Cytoscape and the computed centrality values were plotted over values of protein phosphorylation. Here, every node is represented with two coordinates consisting of a computed centrality and of experimental data regarding protein phosphorylation induced in PMNs by fMLP. In figures 5.7 and 5.8 are shown plots of centroid values over intensity of protein phosphorylation in threonine or tyrosine residues induced by fMLP triggering in human PMNs. Notably, in the plot are shown only those proteins whose phosphorylation level was experimentally determined. The two plots allow immediately evidencing that proteins phosphorylated in threonine (fig. 5.7) or in tyrosine (fig. 5.8) have different topological position in the network, with proteins phosphorylated in tyrosine showing a higher centrality values. This could suggest that tyrosine phosphorylation induced in PMNs by chemoattractants involves signaling proteins regulating clusters of proteins, as the centroid value may suggest. Besides, the top/left quadrant is empty in both figures 5.7 and 5.8. So there are no nodes having low centroid value and high phosphorylation in threonine or tyrosine. This may suggest that

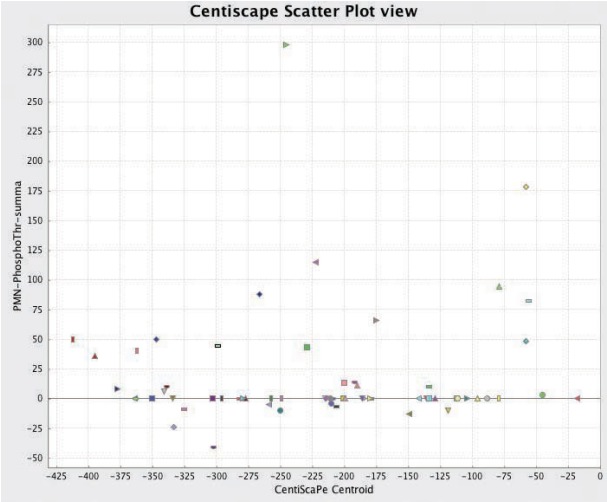

Fig. 5.7. Integration of topological analysis with experimental data. Centroid values are plotted over protein phosphorylation levels in threonine, experimentally determined as described in the text.

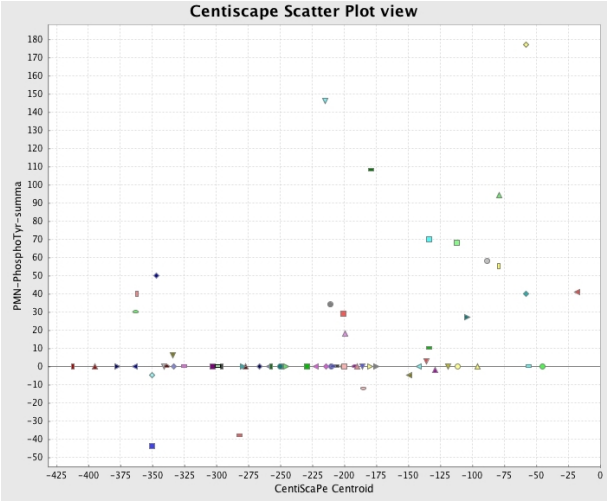

Fig. 5.8. The correlations between Centroid value and intensity of protein phosphorylation in Tyrosine. Proteins with high Centroid value and high level of phosphorylation are easy identified in the top/right quadrant of the graph. Pointing the mouse over the geometrical shapes in the plot shows the corresponding node ID and attribute values

centroid value and activation level are strictly related. Further hypotheses can be formulated by expanding the analysis to other centralities and by adding more phopshorylation data. From this type of plotting it is possible to further identify relevant nodes not only according to topological position but also to experimental

outputs. Thus, groups of nodes whose regulatory relevance is suggested by centrality analysis are further characterized by the corresponding data of biological activity.

5.4 Conclusions

In this chapter a protocol of analysis for protein network have been proposed. The key idea is that of identify most important proteins from both topological and biological points of view. Through the example of the kino-phosphatome network, we have seen how CentiScaPe can integrate the two kinds of analysis allowing an easy characterization of most relevant proteins. The topological analysis and experimental data do confirm each other's regulatory relevance and may suggest further, more focused, experimental verifications. Combination of CentiScaPe with other bioinformatics tools may help to analyze high throughput genomic and/or proteomic experimental data and may facilitate the decision process.

6

Network centralities interference

As seen in the previous chapter, network centralities allow to understand the role and the importance of each single node in a protein network. Next step we introduce in this chapter is to understand and measure changes to the topological structure of the network. The effects of mutation in the network structure have been studied from a global point of view: nodes are removed from the network and the effects on some global parameters, as for example diameter, average distance or global efficiency are evaluated [9], [36], [2], [22]. Our approach wants to answer to this question: "we remove or add one node in the network, how do other nodes modify their functionality because of this removal?". In biological network one or more nodes are removed or added frequently to the network; this can be due to several reasons:

- Gene deletion, a mutation in which a part of a chromosome or a sequence of DNA is missing. Deletion is the loss of genetic material and can results in the removal of one or more nodes in the network (proteins codified by deleted genes are missing in the network).
- Drug usage: a drug generally is used to inhibit a protein, this corresponds to remove the protein from the network.
- Gene duplication: is any duplication of a region of DNA that contains a gene. In this case the copy of the gene is very similar to the original and two similar proteins are codified. So the new protein corresponds to a new node in the graph with most of the edges of the first one.

Obviously we need to understand the consequences of these changes in the network structure. For instance in the case of a drug usage we can study its effects in the network functionality in order to prevent side effects of the drug in other parts of the network that are not directly involved. In the case of gene deletion or duplication we can better understand when network mutation can compromise network functionality. We are interested in exploring node by node modifications in the structure: a mutation can not be important in the whole network, but can modify completely the functionality of one or more nodes. Since centralities are related to nodes functionality, the effects of mutations can be understood analyzing modifications of centralities values due to the mutations. Clearly, the centrality value of one node is strictly dependent on the network structure, and

on the presence of other nodes in the network. So if we add or remove a node in the network, the modification on network structures are reflected on the centrality values of all the other nodes.

The notion we introduce is "nodes centralities interference". This notion measures variations of centrality values of single nodes as consequences of modification in the network structure, as node adding or deletion. It allow to characterize this consequences from a node-oriented point of view. Starting from a single node n we analyze the role of this node identifying the nodes that are strictly dependent on it measuring variations in their centralities when node n is removed from the network. Network centralities interference is introduced through some examples and then it is applied to the kino-phosphatome subnetwork (see chapter 5). Besides, the complementary measures of node centrality robustness, dependence and competition are introduced. These notions, given a node n, allows to identify the nodes whose presence in the network strictly affects the centrality values of node n.

6.1 Interference notion

In figure 6.1 effects of gene duplication on betweenness value are shown. In fig.6.1a

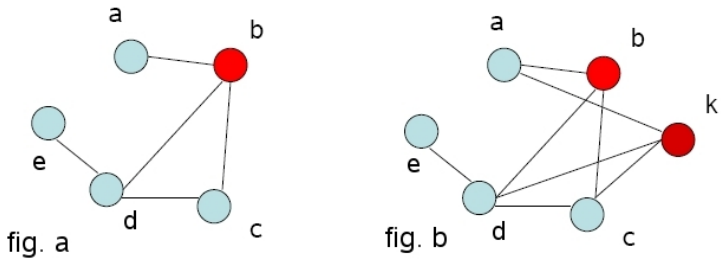

Fig. 6.1. **a**. Node b is in the shortest paths connecting all the other nodes to node a. Node b is essential for connecting node a to other nodes. **b**. Adding node k, with the same edges of node b, has effects on betweenness value of node b. Now also node k is in the shortest paths from node a to all the other nodes: betweenness value of node b will change. Node b is no more essential for connecting node a to the rest of the network.

node b is the only way to reach node a from nodes c,d and e. We suppose that because of gene duplication, a new node k is added to the network (fig.6.1b). Since it has the same edges of node b, now there are new shortest paths connecting node a to the rest of the network: the paths passing through node k. This causes a variation in the betweenness value of node b. To investigate variations of a centrality value due to modification in the network structure, we have been inspired by the notion of "variable interference" used in computer science: to change the value of a variable A in a computer program, can result in changing the value of a variable B in another point of the program. So we say that the variable A interferes with the variable B in the program. This notion have been introduced

and studied by Goguen [28] and can be defined as follow: Given h and l variables of a Program P, we can say that h interfere with l if:

$$\exists h_1, h_2 \in H, l_1 \in L, h_1 \neq h_2 :$$
$$[|P|]^l(h = h_1, l = l_1) \neq [|P|]^l(h = h_2, l = l_1),$$

where H is the domain of the variable h, L is the domain of l and where we mean with $[|P|]^l(h = h_1, l = l_1)$, the semantics of the program P restricted to the variable l starting from a generic starting state where h has h_1 as value and l has l_1 as value. So, if changes on starting values of variable h have effects on final value of variable l we have interference from h to l.

We want to introduce a similar notion for network centralities: in the previous example we can say that the introduction of node k in the network interferes with the betweenness value of node b. The aim is to characterize modifications in networks structures with modification in centrality values as "interference" between nodes in the network. We start analyzing changes in betweenness value and then we generalized the results to the other centralities. All definitions consider connected networks i.e. networks where each node is reachable from all the others. If, after a node deletion the network is not connected then centralities interference definitions cannot be applied. This case is not realistic in biological network because of the high size of node and edges and because of robustness property of such networks.

6.2 Betweenness interference

Consider a network $G = (N, E)$ where N is the set of nodes and E is the set of edges. $Btw(G, n)$ is the betweenness value of node n in the network. We consider $G_{|i}$ the network obtained from G removing node i and all its edges from the network. The betweenness value of node n in the new network is $Btw(G_{|i}, n)$. We define the "absolute betweenness interference" of node i with respect to node n in the network G as:

$$AbsInt_{Btw}(i, n, G) = Btw(G, n) - Btw(G_{|i}, n)$$

It is the difference of the betweenness value of node n in the network G and the betweenness value of node n in the network when the node i have been removed. Potentially the interference value suggests how the betweenness value of node n changes, depending on the presence of node i in the network. The interference value can be positive or negative.

- If this value is negative, it means that the role of node n in the network is higher when the node i is not present in the network. So we can say that node i has "negative interference" on node n, in the sense that the presence of node i in the network is "negative" for the node n to play a "central role" in the network.
- If the interference value is positive, it means that betweenness value of node n is higher if node i have been added to the network. In this case we say that i has "positive interference" on node n, in the sense that the presence of node i is "positive" for node n to play a "central role" in the network.

We can also use the modulus value of interference

$$ModInt_{Btw}(i, n, G) = |Btw(G, n) - Btw(G_{|i}, n)|$$

in order to evaluate if the interference of node i on n is high in absolute value regardless of its negative or positive value.

We introduce also another value, in order to relate interference value with the total value of betweenness. We use instead of betweenness its relative value as defined in section 3.3. The relative interference, or simply interference is defined as follow:

$$Int_{Btw}(i, n, G) = \frac{Btw(G, n)}{\sum_{j \in N} Btw(G, n)} - \frac{Btw(G_{|i}, j)}{\sum_{j \in N} Btw(G_{|i}, j)}$$

The relative interference shows which fraction of betweenness value a node loses or gains with respect to the rest of the network. It is the most precise value since the variation of betweenness is considered with respect to the total betweenness. A node can increase its absolute betweenness value but at the same time the relative betweenness value can decrease (as instance if the total value of betweenness increases). In this case the node loses its importance with respect of the network even if its absolute betweenness value increases. All this considerations will be clarified through the next example.

Example

In the table 6.1 are shown betweenness values for the network in figure 6.2. As

Node name	Betweenness value
node1	10.0
node0	2.0
node4	0.0
node2	1.33
node3	1.33
node5	1.33

Table 6.1. Betweenness values for the network in figure 6.2. Notice high value of node1 (10), and the same values of nodes2, node3 and node5 (1.33).

expected, the higher value is that of node1, which is the only way node4 can be connected to the rest of the network. Notice that node4 has 0 as betweenness value and that node2, node3, and node5, present the same betweenness. This is because the shortest paths connecting node0 to the rest of the network can pass through one of these nodes indifferently. We remove node5 from the network obtaining a new network and we calculate the new values of betweenness and then the interference of node5 with respect to the other nodes. Resulting network is shown in figure 6.3 and betweenness values in the table 6.2. Notice that betweenness value of node1 is lower. This is because node5 was part of a shortest path connecting node4 with

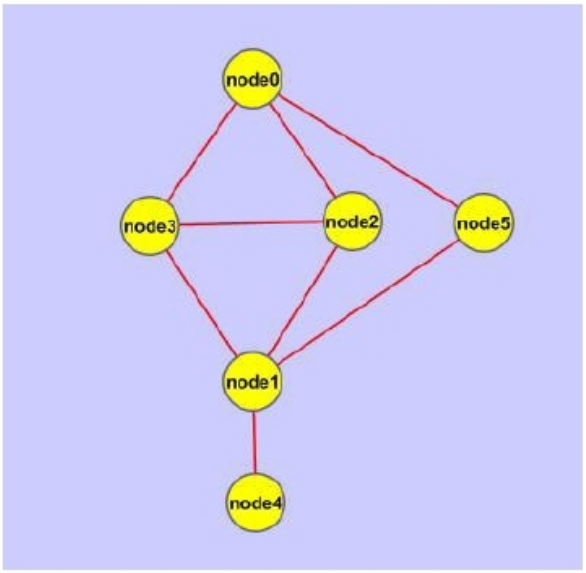

Fig. 6.2. In this network node2, node3, node5, are all possible ways to connect node 0 with the rest of the network. node1 is the only way to connect node4 to the rest of the network.

Node name	Betweenness value
node1	6.0
node0	0.0
node4	0.0
node2	2.0
node3	2.0

Table 6.2. Betweenness value for the network in figure 6.3. Node1 value is decreased (from 10 to 6), and values of nodes2 and node3 are increased (from 1.33 to 2).

node0, and node1 was also part of this path. Besides path connecting node5 to node4 passing through node1 is also missing. So the number of shortest paths passing through node1 is decreased. On the contrary, betweenness of node2 and node3 is higher, since shortest paths passing through node5 and connecting node1 to node0 and node4 to node1 are missing. Consequently the role of node2 and node3 are more relevant. If we remove another of these two nodes, for example node3, then node2 will become the only way to connect node0 with the rest of the network, and its betweenness value will increase again. Table 6.3 shows the absolute interference and relative interference value of node5 with respect to the entire network. As expected, the interference value of node5 with respect to node1 is positive, and the same for node0: they are more relevant in the network if also node5 is part of the network (positive interference). On the contrary, the interference of node5 with respect to node2 and node3 is negative: node2 and

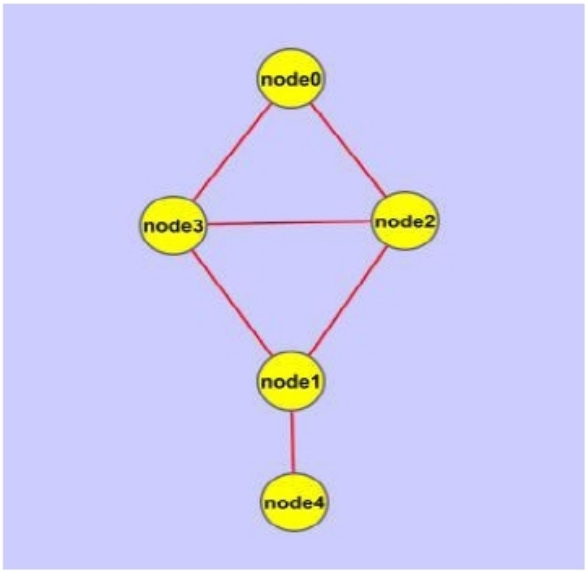

Fig. 6.3. Network obtained removing node5 from network in figure 6.2. Here node2 and node3 are the two possible ways to connect node0 with the rest of the network. The way for connecting node4 to node0 passing through node1 and node5 is missing.

Node	relative btw with node5	relative btw no node5	abs. interference	rel. interference
node1	0.63	0.6	4	0.03
node0	0.13	0	2	0.13
node4	0.0	0.0	0	0
node2	0.8	0.2	-0.67	-0.12
node3	0.8	0.2	-0.67	-0.12

Table 6.3. Interference values of node5 for the network in figure 6.2. Even if node1 has 4 as absolute interference its role in the network is still the same: the relative interference is 0.03 so it lost only 3% removing node5. Node2 and node3 present a negative interference of 0.12. They gain the 12% of betweenness if node5 is removed from the network.

node3 are more relevant if node5 is not part of the network, i.e the presence of node5 has negative interference with respect to node2 and node3.

6.3 Interference centralities definition

We can now generalize definition used for betweenness to the other centralities. Given a centrality C, we define the absolute interference of node i with respect to node n in the network G as follow:

$$AbsInt_C(i, n, G) = C(G, n) - C(G_{|i}, n)$$

Then we can define the modulus of interference:

$$ModInt_C(i,n,G) = |C(G,n) - C(G_{|i},n)|$$

Finally the relative interference (or simply interference) of the node i with respect to node n in the network G is:

$$Int_C(i,n,G) = \frac{C(G,n)}{\sum_{j\in N} C(G,j)} - \frac{C(G_{|i},n)}{\sum_{j\in N} C(G_{|i},j)}$$

Note The relative interference definition can be applied to all centralities defined in chapter 3 except of centroid value since it can have also negative values.

Next step for a complete analysis of interference is to quantify the interference of a single node with respect to the entire network. The question is: How node i is important for the functionality of the entire network? A node can interfere with high value with respect to few nodes and can have low interference value with respect to many others. Otherwise one node can interfere with significant values with respect to the most of the nodes in the network. In the second case the node can have importance for the entire network functionality and not only for one or few nodes. In order to quantify the interference with respect to the entire network we introduce the max interference centrality value and the global interference value. They can easily derived from the previous definitions. The interference max value of node i with respect to the network G is defined as follow:

$$maxInt_C(i,G) = \max_{n\in N_{|i}} \left\{ \frac{C(G,n)}{\sum_{j\in N} C(G,j)} - \frac{C(G_{|i},n)}{\sum_{j\in N} C(G_{|i},j)} \right\}$$

Then we define global interference value of node i

$$Int_C(i,G) = \sum_{n\in N_{|i}} \left(\frac{C(G,n)}{\sum_{j\in N} C(G,j)} - \frac{C(G_{|i},n)}{\sum_{j\in N} C(G_{|i},j)} \right)$$

and mean interference value

$$meanInt_C(i,G) = \left(\sum_{n\in N_{|i}} \left(\frac{C(G,n)}{\sum_{j\in N} C(G,j)} - \frac{C(G_{|i},n)}{\sum_{j\in N} C(G_{|i},j)} \right) \right) \frac{1}{|N|-1}$$

where $|N|$ is the number of nodes of the network.

Example

We shows an example of closeness interference for the network in figure 6.4. Observing the network we can easily identify two clusters having center respectively in node0 and in node4. We can also see that node12 is the one directly connecting node0 and node4 and that removing it the shortest path between the two nodes becomes the one passing for node1, node2 and node3. What we expect is that if we remove node12 then node0 and node4 become less central in the network (from a closeness point of view), since the resulting network is more dispersed. On the

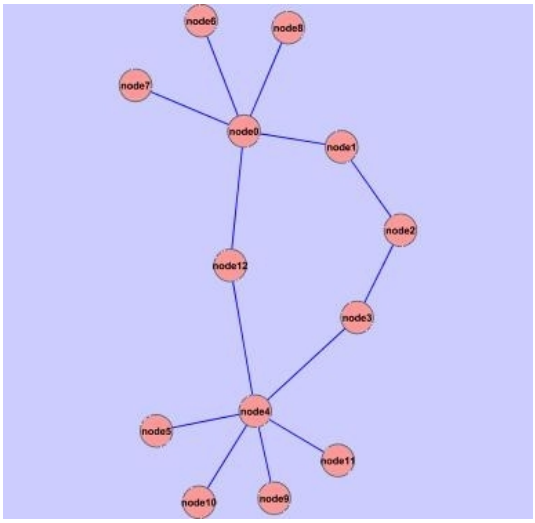

Fig. 6.4. Two cluster with center respectively in node4 and node0 are immediately identified in this network. They are connected through node12.

node name	closeness without node 12	closeness with node12	abs interference	interference
node0	0.0303	0.0417	0.0114	0.0076
node4	0.0345	0.0455	0.0110	0.0041
node6	0.0233	0.0286	0.0053	-0.0020
node7	0.0233	0.0286	0.0053	-0.0020
node8	0.0233	0.0286	0.0053	-0.0020
node10	0.0256	0.0303	0.0047	-0.0049
node11	0.0256	0.0303	0.0047	-0.0049
node5	0.0256	0.0303	0.0047	-0.0049
node9	0.0256	0.0303	0.0047	-0.0049
node1	0.0345	0.0323	-0.0022	-0.0260
node3	0.0370	0.0345	-0.0026	-0.0283
node2	0.0370	0.0333	-0.0037	-0.0310
node12		0.0435		

Table 6.4. Closeness interference values for the network in figure 6.4. Even if for some nodes the absolute interference value is positive, the relative interference value is negative. This mean that removing node12 this nodes gain in absolute closeness value but their relative closeness value decreases: they are more central with respect to the entire network if node12 is removed. Node1, node2 and node3 have the highest negative interference since they replace node12 connecting the two clusters of the network, so they are closer to most of the nodes.

contrary, node1, node2 and node3 should improve their "importance" if node12 is not part of the network.

We computed node12 closeness interference for the network. Values of node12 closeness interference with respect to the other nodes are shown in table 6.4. Notice

that node0 and node4 present positive values of interference. Their interference is high, because distance between the two clusters of the network is higher removing node12 from the network. On the contrary their relative interference is positive but not high because they remain central nodes for the network. So their reduction of closeness removing node12 is due to the reduction of the total closeness: the network without node12 is less "compact" but node0 and node4 remain central nodes in the network. Node1, node2 and node3 present negative interference values. This is because they are more central if node12 is not part of the network, since they become closer to the two clusters. In this case also the relative interference is high since they really become more central in the network. The importance of node12 is also confirmed by its high betweenness value (49).

To stress results of this analysis, we try to remove node11 from the network. Node11 is a peripheral node presenting betweenness value equals to 0. Interfer-

node name	interference	relative interference
node0	-0.0060	-0.0112
node1	-0.0048	-0.0090
node12	-0.0041	-0.0071
node2	-0.0037	-0.0066
node8	-0.0037	-0.0068
node7	-0.0037	-0.0068
node6	-0.0037	-0.0068
node3	-0.0026	-0.0040
node4	-0.0022	-0.0025
node9	-0.0020	-0.0028
node10	-0.0020	-0.0028
node5	-0.0020	-0.0028

Table 6.5. Closeness interference values for node11. All values are low, since node11 is a "peripheral node". Removing it from the network has low effects on the rest of the network

ence values are shown in 6.5. Notice that, as expected, all nodes present similar values. Particularly node1, node2, node3 don't have high interference values. Unlike node12, whose deletion cause important changes for node1 node2 and node3, node11 has low interference value with respect to the entire network: if removed or added to the network it does not cause relevant changes for any node. Similarly to results for attack tolerance in networks [22] the nodes with high betweenness are best candidate to attack network structure. Notice that this example could seem trivial if referred to figure 6.4. The results are more impressive if the same network is drawn randomly as in figure 6.5: an interference analysis in necessary in this case and always if we treat biological network having hundreds or thousands of nodes, as in the next example where we analyze the human kino-phosphatome.

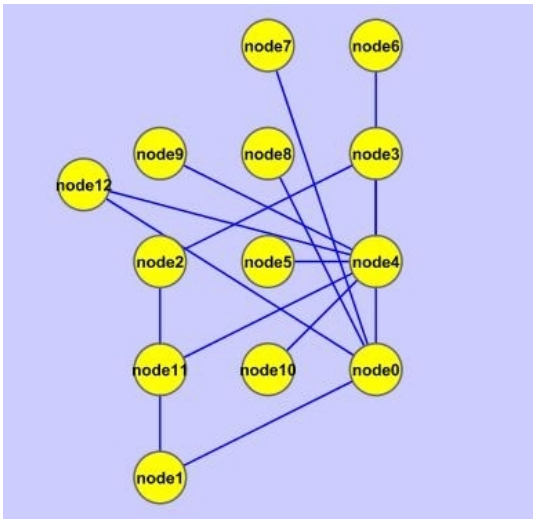

Fig. 6.5. The same network of figure 6.4. Consideration that can be done looking the figure are not possible in this situation. The interference analysis is absolutely necessary.

6.4 A real word example: interference in the human kino-phosphatome

As example we calculate interference in the human kino-phosphatome subnetwork analyzed in chapter 5. The network is shown in the figure 6.6. From a centralities

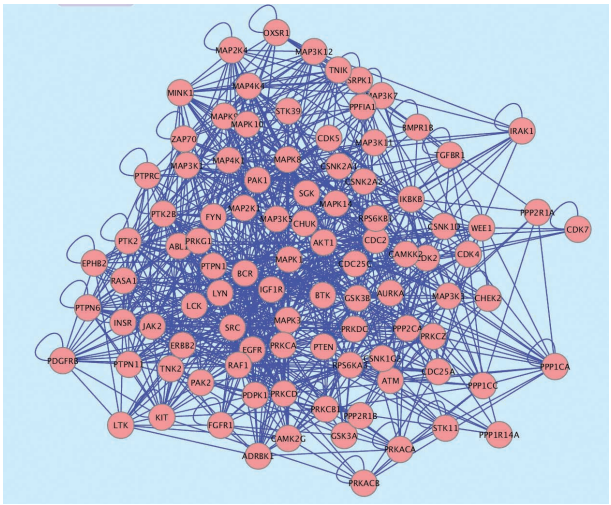

Fig. 6.6. The subnetwork resulting from the analysis in chapter 5. It consist of 97 nodes (82 kinases and 15 phosphatases) and 962 interactions.

analysis we found that Mapk1 and Prkca present high centrality values [52]. The question is how much they interfere with other proteins in the network and also if they interfere with the same groups of proteins. We try to calculate and compare betweenness interference of Mapk1 and Prkca. In table 6.6 first ten values of interference are shown, for both Mapk1 and Prkca. First of all, interference values are

First ten positive interference value of Mapk1 and Prkca

node	Mapk1 interf.	Mapk1 rel interf.	node	Prkca interf.	Prkca rel interf.
PPP1CA	7	0,0008	PPP2CA	10,35	0,0012
PPP2CA	3,87	0,0005	MAP3K3	7,41	0,0009
CAMK2G	3,46	0,0004	AURKA	7,07	0,0008
PTEN	3,02	0,0004	MAPK9	6,2	0,0007
RPS6KA3	2,92	0,0003	PTPRC	5,32	0,0006
PTPN11	2,62	0,0003	PTPN1	3,93	0,0004
JAK2	2,61	0,0003	CHEK2	3,68	0,0004
CDC25C	2,3	0,0003	CAMK2G	3,2	0,0004
PDPK1	2,28	0,0003	PPP1R14A	3,12	0,0004
ZAP70	2,25	0,0003	STK11	2,31	0,0003

Table 6.6. First ten positive interference values of Mapk1 and Prkca. Only PPP2CA and CAMK2G have high value with both Mapk1 and Prkca

not high. Notably, highest between interference relative values are around 0.001. This means that the variation of betweenness values is only around 0.1% of the global values and that removing one node does not change so much for the entire network. This agrees with results on robustness and attack tolerance of biological networks [2], [9], [36]. Then we can notice that only PPP2CA and CAMK2G are

First ten (absolute value) negative interference values of Mapk1 and Prkca

node	Mapk1 interf.	Mapk1 rel interf.	node	Prkca interf.	Prkca rel interf.
MAPK3	-29,13	-0,0034	RPS6KA3	-40,04	-0,0047
AKT1	-26,51	-0,0031	GSK3B	-24,65	-0,0030
MAPK8	-22,22	-0,0026	MAPK1	-24,65	-0,0030
PRKCA	-19	-0,0022	PRKDC	-19,94	-0,0024
CDC2	-14,65	-0,0017	PRKCB1	-18,4	-0,0022
SRC	-14,19	-0,0017	MAPK8	-16,81	-0,0020
PTK2B	-14,11	-0,0017	CDC2	-16,45	-0,0020
FYN	-12,78	-0,0015	PRKCZ	-16,41	-0,0019
IGF1R	-11,04	-0,0013	PTK2B	-16,31	-0,0019
MAPK14	-9,49	-0,0011	PRKACA	-14,65	-0,0017

Table 6.7. First ten negative interference values for Mapk1 and Prkca. Notice reciprocal high interference between Mapk1 and Prkca

present in both tables. This is really interesting since it means that the effects of removing Mapk1 or Prkca are not the same: Mapk1 remotion affects a part

of the network and Prkca remotion affects another part. Potentially this means that different functionality in the kino-phosphatome network are affected by the two proteins. Further analysis and lab experiments could confirm this hypothesis. Other interesting considerations can be done for table 6.7 where the ten most relevant negative interference value are shown. Here absolute values are higher than for positive interference indicating that negative interference of Mapk1 and Prkca is more significant. MAPK8, CDC2, PTK2B are present in both tables, they depends on both Mapk1 and Prkca. Curiously there is negative interference of Mapk1 with respect to Prkca (-19). This means that if Mapk1 is inhibited then Prkca assume a more relevant role in the network. Similarly Prkca has negative interference with respect to Mapk1 (-24,65): the role of Mapk1 is more relevant if Prkca is not part of the network. This should suggest, according with betweenness meaning, that some functionality of Mapk1 can be replaced by Prkca if Mapk1 is inhibited and conversely Prkca can be replaced by Mapk1 if inhibited. Such hypothesis should be confirmed by lab experiments, but surely some shortest paths connecting proteins in the network and passing through Mapk1 are replaced by shortest paths passing through Prkca if Mapk1 is inhibited and similarly for shortest paths passing through Mapk1 when Prkca is inhibited. In this sense Mapk1 and Prkca are "competitors" on having a central role in the network. Finally, observing both positive and negative interference, we can see that Mapk1 has positive interference with respect to RPS6KA3 and that Prkca has negative interference with the same protein. So, when Mapk1 is present in the network then RPS6KA3 has a more central role, on the contrary when Prkca is part of the network RPS6KA3 is less relevant. Also this hypothesis should be confirmed with further lab experiment.

6.5 Further consideration for network centralities interference

Other consideration about network interference can be done:

- As argued above, interference naturally induces cluster of proteins that are similar for their interference values due to the same node. A new clusterization algorithm can be derived if we group nodes depending on their interference value: given a node we compute its interference value and we put all the nodes having high interference in the same cluster. This interference-based modular decomposition of a network characterizes nodes for their answer to the inhibition (or adding) of a certain node in the network. If deletion of the node in a protein network is due to drug usage, the cluster of nodes having high interference value is the set of proteins where the drug has its greatest effects. In pharmacology this should permit to predict which proteins are more affected from the inhibition of another protein in the network. We can so prevent side effects of the inhibition of a node due to a drug usage.
- We know that biological networks are not easily affected from the removal of a single node [2], [9], [36]. So a possible scenario is that of removing or adding more than one node in a network. Definition of interference can be easily adapted to such a situation, where a set of node is considered. Given a subset S of the network nodes N ($S \in N$) and a centrality measure C, we

define the absolute interference of the set of nodes S with respect to a node n as follow:

$$AbsInt_C(S, n, G) = C(G, n) - C(G\backslash S, n)$$

Then we can define the modulus of interference:

$$ModInt_C(S, n, G) = |C(G, n) - C(G\backslash S, n)|$$

Finally the relative interference of the set of nodes S with respect to node n in the network G is:

$$Int_C(S, n, G) = \frac{C(G, n)}{\sum_{j \in N} C(G, j)} - \frac{C(G\backslash S, n)}{\sum_{j \in N} C(G\backslash S, j)}$$

- Max interference, global interference and mean interference are similarly defined. Besides the definitions can be easily adapted if we are interested in removing or adding one or more edges in a network.

6.6 Nodes centrality robustness, dependence and competition value

As just seen, centralities interference give answer to the question:"which are the nodes whose functionality is affected by node n?". Similarly we can analyze the effects of a node with respect to the entire network using the mean interference value. But another question can be of interest: if we are interested in a particular protein we'd like to know if its functionality can be affected by other proteins and how much. The question is, conversely to interference: "which are the nodes affecting node n?". To answer to this question we introduce the notion of robustness, competition and dependence value of a node with respect to a particular centrality. Given a network $G = (N, E)$ a centrality measure C and a node $n \in N$, we define the centrality robustness of node n as follow:

$$Rob_C(n, G) = \frac{1}{\max_{i \in N_{|n}} \{|Int_C(i, n, G)|\}}$$

Robustness depends on the maximum interference value that can affect the centrality value of the node. If it is low, the node can be easily "attacked" by removing or adding particular nodes. If it is high, the node is "robust", i.e. there is no node removal or adding that can affect its centrality value and consequently functionality. Notice that we consider absolute value of interference. To consider only the positive interference we define positive robustness value as:

$$PosRob_C(n, G) = \frac{1}{\max_{i \in N_{|n}} \{Int_C(i, n, G)\}}$$

where

$$Int_C(i, n, G) \geq 0$$

If low, this value means that the node is "central" because of the presence of at least another node in the network, if high the central role of the node is not

dependent on other nodes. Similarly we consider negative interference defining negative robustness as

$$NegRob_C(n,G) = \frac{1}{\max_{i \in N_{|n}}\{|Int_C(i,n,G)|\}}$$

where

$$Int_C(i,n,G) \leq 0$$

Low negative robustness means that the central role of the node can be "improved" removing a particular node from the network. In this sense the two nodes (node considered and the removed one) are "competitors" in the network. If negative robustness is high, the central role of the node cannot be improved removing a particular node from the network.

In some cases it is more intuitive to use the reciprocal of negative and positive robustness. We define the reciprocal of positive robustness as the *dependence value*:

$$Dep_C(n,G) = \max_{i \in N_{|n}}\{Int_C(i,n,G)\}$$

where

$$Int_C(i,n,G) \geq 0$$

If high this value means that node n is dependent on another node to have a central role in the network, i.e. if that node is removed than node n looses a consistent part of its central role (its centrality measures decreases). If low, node n does not strictly depends on the presence of other nodes in the network. Similarly we define the *competition value* as the reciprocal of negative robustness:

$$Comp_C(n,G) = \max_{i \in N_{|n}}\{|Int_C(i,n,G)|\}$$

where

$$Int_C(i,n,G) \leq 0$$

If high this value means that node n can consistently improve its central role if another node is removed from the network, i.e. if that node is removed than node n improves its central role (its centrality measures increases). In this sense the two nodes are "competitors" in the network.

Considering robustness, competition and dependence, we are interested in a single node. In this case the variation due to robustness (or competition or dependence) can be related to the centrality value of the node in the starting network (the network with no node deletion). Such notions of *relative robustness*, *relative dependence* and *relative competition* are defined as the fraction of the variation of the centrality value with respect to the starting centrality value. Given the centrality C and a node n the relative centrality value in the network G is defined as in section 3.3.

$$relC(n,G) = \frac{C(n,G)}{\sum_{i \in N} Int_C(i,G)}$$

The relative robustness is

$$relRob_C(n,G) = \frac{relC(n,G)}{\max_{i \in N_{|n}}\{|Int_C(i,n,G)|\}}$$

Similarly for dependence value:

$$relDep_C(n,G) = \frac{Dep_C(n,G)}{relC(n,G)}$$

and competition value:

$$relComp_C(n,G) = \frac{Comp_C(n,G)}{relC(n,G)}$$

Also the total robustness dependence and competition value can be used in order to characterize the entire network. Total robustness of a node n with respect to the centrality C in the network G is:

$$TotRob_C(n,G) = \frac{1}{\sum_{i \in N_{|n}} Int_C(i,n,G)}$$

If low the central role of the node depends on the presence in the network of other nodes. If high, the central role of the node is quite independent from other nodes (but the node can have high dependence from one or few nodes). Similarly are defined the total dependence value:

$$TotDep_C(n,G) = \sum_{i \in N_{|n}} Int_C(i,n,G)$$

where

$$Int_C(i,n,G) \geq 0$$

and the total competition value:

$$TotComp_C(n,G) = \sum_{i \in N_{|n}} |Int_C(i,n,G)|$$

where

$$Int_C(i,n,G) \leq 0$$

The role of node centrality robustness, dependence and competition value is shown in the next example.

Example

Consider the network in figure 6.7 and its betweenness values reported in table 6.8. Node3 and node6 have the highest values of betweenness (25.64), node4 and node5 presents the third highest value (12). As expected they have a central role in the network from a "betweenness" point of view. But does their values depend on other nodes or they are not affected by node deletion? Robustness analysis of node3 and node4 can answer to this question and are reported respectively in tables 6.9 and 6.10.

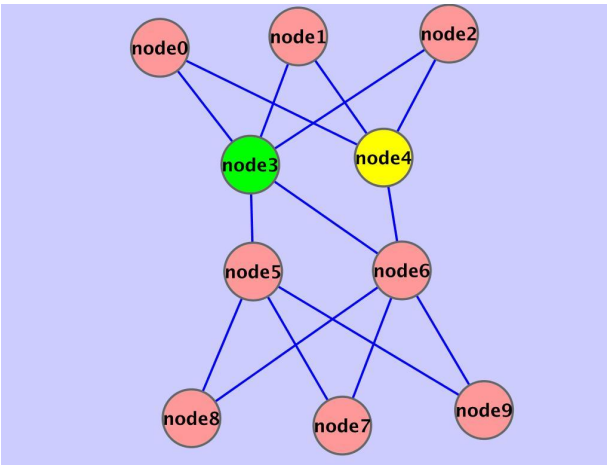

Fig. 6.7. Node3 and node6 have highest betweenness (25.64). Betweenness value of node4 and node5 is 12.

node name	betweenness	relative betweenness
node0	0.79	0.0098
node1	0.79	0.0098
node2	0.79	0.0098
node3	25.64	0.3205
node4	12.00	0.1500
node5	12.00	0.1500
node6	25.64	0.3205
node7	0.79	0.0098
node8	0.79	0.0098
node9	0.79	0.0098
total	80.00	

Table 6.8. Betweenness value for the network in figure 6.7.

Node3 has higher robustness value (1.4824) than node4 (0.5385). This agree with considerations that can be done observing the figure 6.7. We can see that node4 is in the shortest paths connecting node0, node1 and node2 with node7, node8 and node9. But if we remove node6, node4 looses this role and become a "peripheral node" connecting only node0, node1, node2 between them. This is shown in figure 6.8. This can not happen to node3 since it is connected to both node6 and node5. Node3 has highest dependence on node5 equals to 0.09999. The relative dependence value is 0.3118 indicating that node3 looses about the 31% of its starting betweenness value if node5 is removed from the network. Indeed, if we delete node5 the betweenness value of node3 become the same of node4, since they connect the same nodes through the same paths: those passing through node6. But dependence of node4 on node6 is higher (0.1143, with relative dependence 0.7619 i.e it looses about 76% of its starting betweenness value if node6 is removed from the

Node3 robustness dependence and competition values

Removed node	none	nodes 7,8,9	nodes 0,1,2	node4	node5	node6
Betweenness	25.6429	22.0000	16.5000	36.5000	15.0000	35.0000
Relative betweenness	0.3205	0.3667	0.2750	0.5368	0.2206	0.4167
Interference (relative)		-0.0461	0.0455	-0.2162	0.0999	-0.0961
Dependence (node5)	0.0999	Rel. dependence	0.3118			
Competition(node4)	0.2162	Rel. competition	0.6746			
Robustness	4.6247	Rel. robustness	1.4824			

Table 6.9. Robustness, dependence and competition values of node3 6.7.

Node4 robustness dependence and competition values

Removed node	none	nodes 7,8,9	nodes 0,1,2	node3	node5	node6
Betweenness	12.0000	10.0000	7.0000	36.0000	15.0000	3.0000
Relative betweenness	0.1500	0.1667	0.1167	0.4286	0.2206	0.0357
Interference (relative)		-0.0167	0.0333	-0.2786	-0.0706	0.1143
Dependence (node6)	0.1143	Rel. dependence	0.7619			
Competition (node3)	0.2786	Rel. competition	1.8571			
Robustness	3.5897	Rel. robustness	0.5385			

Table 6.10. Robustness, dependence and competition values of node4.

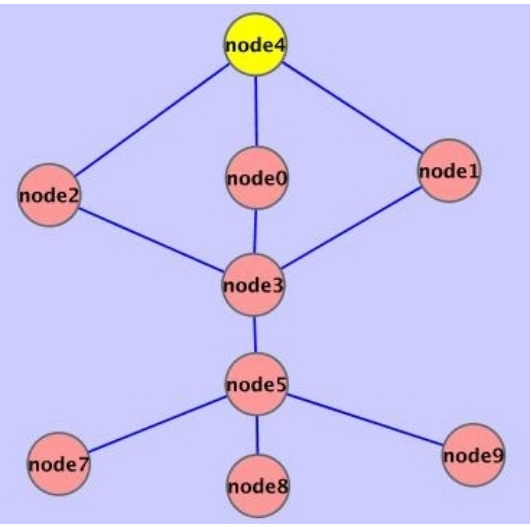

Fig. 6.8. Network of figure 6.7 after removing node6. Node4 becomes a peripheral node

network): as previous seen if we remove node6 then node4 becomes a "peripheral" node and node3 become the only way to connect the "top" of the network with the "bottom".

Also the competition value of both nodes is very informative. The highest values of node3 depends on deletion of node4 and the highest value of node4 depends on

node3. In this sense they are really "competitors" in the network. If we remove node3 then node4 becomes the only connection for the "top" and "bottom" of the network. The same for node3 if we remove node4. But node4 competition values is higher (1.875). This is due to the fact that starting betweenness value of node4 is lower (12) than node3 value. So the increase of betweenness of node4 is higher, the 185% of the starting value.

6.7 Conclusions

The notion of node centralities interference have been introduced in this chapter. The main idea is that of considering the relevance of a node with respect to other nodes measuring the impact that a node has on the centrality values of the others. If a node is important in the network its remotion cause high variation in the centrality values of the other nodes. This approach allows to identify nodes whose role is strictly dependent on another node in the network. Node centralities interference answers to the question: "Which are the nodes affected by node n?". Interference have been illustrated with some examples and an interference analysis have been applied to Mapk1 and Prkca in the kino-phosphatome network. Similarly the notion of robustness, dependence and competition value of a node have been introduced. These notions allow to identify if the central role of a node is dependent on some other nodes in the network, or if the node is structurally central in the network (robust). Robustness answer to the question: "Which are the nodes affecting node n?". Further analysis and lab experiments should investigate the role of this notion in a protein network.

Part II

Dynamic analysis of biological pathways

7

Abstract Interpretation for dynamic simulation of pathways

In this chapter we describe the use of abstract interpretation [18], [19], [20] to analyze dynamic simulations of a biological pathways, through an example of analysis of the mitotic oscillator pathway [30], [29]. A pathway is a biological network that consists of a series of chemical reactions occurring within a cell, catalyzed by enzymes, to achieve in either the formation of a metabolic product to be used or stored by the cell, or the initiation of another metabolic pathway. Many of these pathways are elaborate, and involve a step by step modification of the initial substance to shape it into the product with the exact chemical structure desired. First approach to pathways is to simulate their behaviour in order to find properties of interest [10] but, because of their growing complexity some models that were proposed in the past in computer science for modeling and analyzing software and concurrent systems, can be used to deal with pathways. We present how some well-known abstract domains, classically used in static analysis of software can be used to extract informations from a simulation program. The congruence domain [32] is used to automatically find oscillations in the proteins concentration, intervals and constants domains [18], are used to characterize the range of proteins concentration. This allow to execute simulations with a wide range of different starting parameters, as starting proteins concentration, function parameters and so on, and to automatically extract the properties of interests. The main result, which open new possibilities in the field of simulation of biological systems, is that thousands of simulations with different starting parameters can be done automatically in a few time extracting information in order to characterize oscillation conditions, and to guarantee important properties about proteins concentration in such a big amount of starting values. This can be important for two reasons:

- We can completely characterize starting values of a simulation, identifying the condition leading the system to particular behaviour as oscillation or concentration values in some range.
- We have a possible solution to the problem of parameters estimation: for most of pathways kinetics parameters are not or partially available but the behavior of the system in particular condition is known. So the method can be used to evaluate with a brute force algorithm some parameters of the system in order to identifying those parameters satisfying the final behavior.

Since abstract interpretation has been successful used for analyzing industrial software of more than hundreds thousands of statements [12], it seems an appropriate method for approaching also pathways of similar size. In order to show which can be the role of abstract interpretation in doing that, we introduce a simple pathways formalization and we apply it and our analysis to the mitotic oscillator pathway. Some considerations about the model: we know that abstract interpretation is a technique that is independent of the model used, since we can apply it to programs written in different languages and described by different models as automata, hybrid systems and so on. So, as in software analysis, where abstract interpretation can be applied to any model, in system biology the idea is that abstract interpretation should be something for giving more power to all the models just cited. It is to say, that we can use abstract interpretation for extracting numerical properties from a pathway, and it is not important if we formalize it using as model a rewriting systems, a Petri-net, π-calculus, P-systems or something else, because in a certain sense abstract interpretation can work in an "higher level": the only thing we need is a well defined semantics for the model we use. The main idea is that of starting from a program simulating a pathway behaviour, we extract the properties of interest by analyzing the program by abstract interpretation: practically we analyze the simulation program. A preliminary work about this was presented at PLID 2005 [50] and then presented at EAAI [51] and published on conference proceedings.

7.1 Preliminaries

7.1.1 Pathways

Cells are considered the fundamental unit of the living organisms. In complex being, they are divided in classes and have different functions for the system they belong to. For doing that, the cells necessarily need the ability of interacting with the environment and among them, and in a certain sense, they are able to receive and to transmit informations. The only way they have to interact with the outside is clearly through biochemical reactions between the molecules compounding them. Each different input coming from the environment, produces a set of chemical reactions in a cell, that are the "answer" of the cell to the input. Those reactions depend on some parameters, such as the concentration of the reactants and the functions that regulate the speed of a reaction, and they are organized in very complex networks, that are called pathways. Usually pathways are modeled by differential equations, that represents the changes in the concentration of the molecules of the pathway. This approach is useful and well-studied [56] [59] [30], and is essentially based on standard numerical techniques for solving differential equations, as for example the Eulero's method and similar. Different problems concern the pathways simulation: the first is that we want to do simulations with different starting values in order to characterize the pathway's behavior with respect to these parameters, and such simulations result in a too large amount of data to be treated without automatic methods; The second is that not always all the parameters characterizing the pathway are known and we need to infer this

missing parameters through software simulation; The third is that many examples of pathways are built by several thousands of molecules and reactions, and we need new techniques allowing to treat this kind of objects, for automatically simulate and analyze them. Since a similar amount of data was in the past treated in software analysis, it is a diffuse opinion that some models that were proposed in the past in computer science for modeling and analyzing software and concurrent systems, can be modified for analyzing, modeling and testing pathways, solving in such a way the complexity problems. All the main models belonging to theoretical computer science have been proposed for this purpose, for qualitative and quantitative analysis. Model checking have been successfully [14] applied to programs that simulate cells behaviour. This approach can answer to classical question of model checking tools as "Is it possible that a cell starting from a state satisfying the property P can reach a state that satisfies the property S?", or "What are all the initial states that would lead to a particular final state satisfying the property P?". In [24] and [58] the rewrite-system based language MAUDE [15] is used for modeling a pathway including more than 650 proteins and 500 rules, and the MAUDE model-checker is used for analyzing the pathways with standard temporal logic questions. In the same way hybrid systems and automata have been proposed [4] [3] since they seem to be a good model for representing the typical behaviour of some biological systems passing from discrete states to other depending on continuous function. Some other mathematical structures that are used for studying computer networks and concurrent systems are of large use for modeling biological systems as the π-calculus model [49], and also its stochastic version [48], [43], Petri-nets [31] and P-systems [47]. The open question about these models is only whether they will be able to face the complexity of biological pathways; it seems that at least some powerful techniques of abstractions will be needed. Our approach based on simulation of differential equations with numerical methods uses the abstract interpretation approach to simplify the analysis in order to facing more complex pathways and a greater numbers of simulation.

7.1.2 Abstract interpretation

The software analysis method we propose, to help the efforts in the direction of analyzing complex systems as biological pathways, is abstract interpretation [18], [19], [20]. It is a general theory aiming to approximate the properties of discrete and continuous dynamical systems. The idea is that of systematically deriving, from a complex model, a simpler approximate model preserving the properties that become salient during a simulation or a predictive analysis. It is based on a simple mathematical structure, that is the Galois insertion: given two Poset $(C, \leq)$, $(A, \leq)$ we say that (A, C, α, γ) is a Galois insertion with $\alpha : C \to A$, $\gamma : A \to C$ if for all $a \in A$ and for all $c \in C$ we have $\alpha(c) \leq a \Leftrightarrow c \leq \gamma(a)$ and $\alpha \circ \gamma = id$. C is called the concrete domain, A is called the abstract domain, α and γ are respectively the abstraction and concretization functions. The basic concept is this: instead of calculating the behaviour of a program in a concrete domain, that is not possible for the well known results about calculability, we calculate this behaviour in an abstract domain that is defined for the specific property we want to know. The structure of the Galois insertion is used for passing from concrete to abstract

domain and back ensuring some useful properties as soundness and some-times completeness. For pathways simulation and analysis, we use a simplified abstract interpretation framework, based not on the abstract computation but on the abstraction of concrete computation. Abstract computation is needed to ensure the termination of the analysis. But since we analyze finite simulation the problem of termination is avoided and abstracting concrete simulations we gain in precision. A lot of specific domain were presented, for equally specific properties, as finding the signs of variables, finding if a variable belongs to an interval and find if a variable belong to a congruence class (for example if the variable $x = 5 \ mod \ 8$). Other relational abstractions are also able to find numerical relation between variables in a program. A little review about the most used abstract domains can be found in [17] remanding to other more detailed works. The power of abstract interpretation have been demonstrated by some concrete successful example at industrial level [12], where abstract interpretation was used for analyzing programs of hundred thousand of statements in a software for airplanes control. This example shows how abstract interpretation is a well-founded techniques and more than a simple mathematical theory.

7.2 Modeling pathways

7.2.1 Pathway definition

We said that a pathway consists of a series of chemical reactions occurring within a cell. These chemical reaction can be divided in activation and inhibition reactions. We consider an activation reaction when one enzyme (p_1) works as a catalyst of a reaction that transforms a molecule (that is called a substrate) s in a product (p_2). Some times p_1 is not an enzyme but it is a molecule (substrate) whose reaction with another molecule s produce the protein p_2. p_1 can also works to inhibit a reaction that transforms the substrate s in the product p_2, so we say that p_1 inhibits p_2. Graphically, pathways can be represented as direct graphs, in which the nodes are the molecules, and the edges are the reactions between them. The edges are represented as continuous arrow if the role of the reaction is activation and with dashed arrow if its role is of inhibition. Formally we can define a pathway as a mathematical structure, defined by a set P of proteins (nodes) and by a set of ordered pairs $A \subseteq P \times P$ called edges or actions. Every edge $a = (p_1, p_2) \in A$ represents a chemical reaction between two molecules. Each reaction represented by an edge is provided with a reaction speed, that is the speed of production of a molecule p depending on some functions that can be summarized following this scheme [56]:

- zero-order reaction: converts substrate into product at a fixed rate independent of substrate concentration.
- first-order reaction: converts substrate into product at a rate proportional to substrate concentration.
- second-order reaction: creates product at a rate proportional to either the product of two substrate concentrations or the square of a single substrate concentration.

- higher order reactions are similarly defined.

In other cases, the rate of production of the result of the reaction p depends on some differential equations, that are in the most of cases of the standard form of the Michaelis-Menten equation:

$$\frac{d[p]}{dt} = \frac{V_{max}[s]}{[s] + k_m}$$

The square brackets represents concentrations, so $[s]$ is the concentration of the substrate s, end $d[p]$ is the variation of the concentration of the protein p, depending on the constant parameters V_{max} and k_m that have to be estimated from experimental data. These values comes from laboratory studies about the reaction considered and they are different with respect to different reactions. Some other reactions are well modeled by the Hill functions:

$$\frac{d[p]}{dt} = \frac{[s]^n}{[s]^n + K_H^n}.$$

As previously, the values of n and K_H depend on the reaction and comes from laboratory studies.

Example 7.1. We can see in figure 7.1 a tipical example of pathway with feedback, where a product inhibits one of its activators. The protein X_3 indeed inhibits

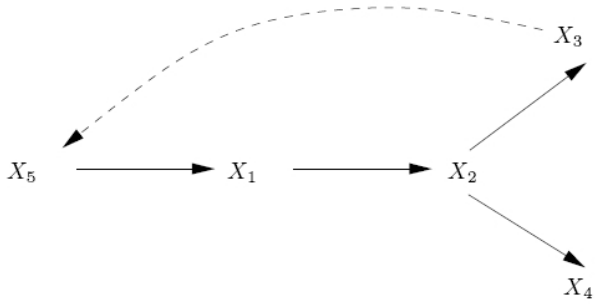

Fig. 7.1. An example of pathway with feedback. Nodes are the molecules, the edges are the reactions. Continuous arrow are activation reactions and dashed arrow are inhibition reactions.

the reaction between X_5 and X_1. This is a typical example of pathway that can subtend with the proper condition to oscillatory behaviour.

We have an oscillatory behaviour when the concentration amount of a molecule in a biological phenomenon does not reach a steady value but it oscillates between a min and a max value. Oscillations have a great importance in biology, since a lot of systems present this behaviour (for instance the circadian rhythm, the

heart pulsation, or the insulin production). In our model to each protein $p \in P$ is associated a variable X_p representing the concentration of p that is activated (usually expressed in Moles[1]), i.e. that quantity of protein p that has been activated by an enzyme and can be used as activator of some other reactions. Sometimes it is indicated also with the name of the protein between square brackets (ex.: $[X_p]$); The value X_p changes according to the differential equation of the pathway. A configuration or a state of the pathway is given by an assignment of values to the variables, and can be viewed as a "snapshot" of the system. Formally:

Definition 7.2. ***State of a pathway.*** *Given a $n-$protein pathway PW a configuration or a state of PW is represented as a vector $s =< h_1, \ldots, h_n >$ where $h_i, i = 1 \ldots n$ are the concentration values assigned respectively to the protein $p_1 \ldots p_n$ of the pathway.*

7.2.2 The pathway simulation

We suppose that the system is provided with a counter increasing at each turn from a configuration to another in order to have information about the temporal evolution of the model. To simulate the behaviour of the system we use the well known Euler's method: We start from an initial state s_0, corresponding to a reasonable initial assignment of values to the variables, and with the correct values for the constants of the differential equations coming from experimental data. Then, at each small step in time, the concentrations are substituted in the differential equations for calculating the rate of production of each protein p_i of the pathway. The rate of change is multiplied by the size of the time step and the results are added to the respective concentrations obtaining the new values of concentration for each protein p_i, and so the successive configuration. The procedure is repeated and if the time step chosen is sufficiently small we obtain a sequence of states:

$$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$$

simulating the behaviour of the system. The symbol $\rightarrow$ indicates that the state s_i is obtained from s_{i-1} through one step of the method described above. At each step, we have to calculate the new concentrations and to update the respective variables of the system. In figure 7.2 is represented the mitotic oscillator pathway and the result of a simulation with the Eulero's method, according with results in [29]. In the rest of the chapter, if not differently specified, the starting values of the examples are that of this figure.

7.2.3 Semantics for pathways

For correctly applying abstract interpretation to biochemical pathways we have to formalize some concepts. We need a collecting semantics for the pathway so, first of all, we define the lub of two state of a pathway.

[1] The mole is the SI term identifying the number of particles in a given amount of matter. It is a dimensionless quantity (meaning a number without units) numerically equal to Avogadro's number (= 6.02214151023).

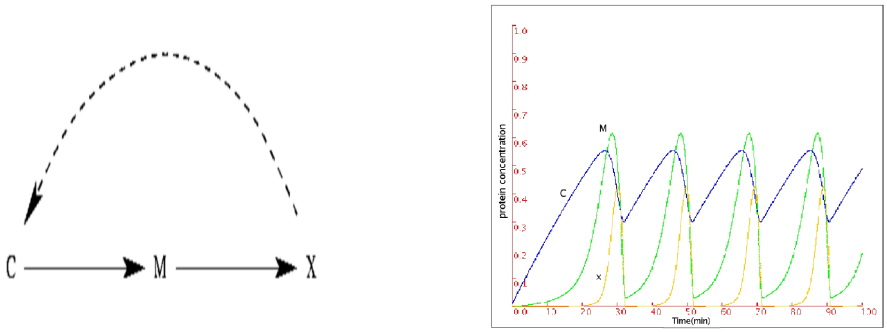

Fig. 7.2. The mitotic oscillator pathway. C represents cycline, M is cdc2-kinase and X is the protease. The negative feedback structure is clearly visible. In the right the behaviour of the mitotic oscillator pathway, with starting values: $v_i = 0.025\mu M min^{-1}, v_d = 0.25\mu M min^{-1}, K_d = 0.02\mu M, k_d = 0.01 min^{-1}, V_{M1} = 3 min^{-1}, V_2 = 1.5 min^{-1}, V_{M3} = 1 min^{-1}, V_4 = 0.5 min^{-1}, K_c = 0.5\mu M, K_1 = 0.01, K_2 = 0.01, K_3 = 0.01, K_4 = 0.01; C = 0.01\mu M, M = 0.01, X = 0.01$. The method used is Eulero's with a step of 0.01.

Definition 7.3. *Given two states* $s_i =< h_{1,i}, \ldots, h_{n,i} >$ *and* $s_l =< h_{1,l}, \ldots, h_{n,l} >$ *we define*

$$s_i \vee s_l =< h_{1,i} \cup h_{1,l}, \ldots, h_{n,i} \cup h_{n,l} >$$

Definition 7.4. *Given a pathway* PW *and a starting state (or configuration)* s_0 *we can define the semantics of pathway* $[\![PW]\!] : \mathbb{R}^n \to \wp(\mathbb{R})^n$ *as:*

$$[\![PW]\!]^{t_0,t_f}(s_0) = \bigvee_{i=t_0}^{i=t_f} s_i, \text{ where } s_i \to s_{i+1}.$$

$t_0, t_f \in \mathbb{N}$ *are respectively the starting and ending time of the semantics. It means that program semantics is considered only from time* t_0 *to time* t_f *and not for the entire simulation time. If* L *is a set of variable of the pathway, we use* $[\![PW]\!]_L^{t_0,t_f}(s_0)$ *meaning the semantics of the pathway restricted to* L.

Obviuosly, if we are interested in the values of the variables during the complete simulation, we have $t_0 = 0$ and t_f equals to the simulation time, so they can be omitted. Therefore $[\![PW]\!](s_0)$ is the set of all the values that each variable can have during the simulation. If $t_0 = 0$ and $t_f = T$ the semantics defined above correspond to the collecting semantics of the following simulation program in the program point P1:

```
s0=h1...hn; time := 0;
While time <= T do
   -calulate the new values for h1...hn
    (according with the simulation method)
   -increment time
   -update h1...hn
P1:
    endWhile
```

Note. Because of our definition of pathway simulation our semantics based model is completely independent from the simulation method used. In our example we use the Eulero's method but also Runge-Kutta can be used. Besides this framework can also applied to methods not based on ordinary differential equations. Some preliminary tests have been successfully done using P-systems [47]. Since abstract interpretation can be applied to any computer programs, our abstract interpretation based analysis can be applied to any simulation program.

7.2.4 Abstract semantics for pathways

Once the semantics have been defined, we approximate it through abstract interpretation techniques. Usually we need only to choose the proper abstract domain and the abstraction function α. Classically, the concrete values and the concrete operations are respectively replaced by abstract values and abstract operations. This is the normal use of abstract interpretation theory which guarantees the correctness of the abstract operations used and the termination of the analysis. Correctness of an abstract function $f^{\#}$ is so defined: $\alpha(f(x)) \subseteq f^{\#}(\alpha(x))$, where x is the element of the concrete domain, f is the concrete function, and α is the abstraction function. Analyzing pathways we don't need to use abstract interpretation in such classical way, i.e. we don't need to use abstract operators. The reason is that this kind of approximation looses to many informations i.e. it is not enough precise [51]. Thereby, in order to make our analysis more precise, we will abstract concrete computations. This is justified by the fact that our simulation are finite in time and we do not need to use abstract interpretation to avoid the termination problem of the program analysis. The new way we use abstract interpretation is that of having an intelligent reader that is able to observe and analyze data coming from simulations that are too complex for human mind. This happens in two cases: the first is when we have a simulation of a pathway with thousand of nodes and we want to extract properties for many of this nodes, the second is when we have pathway not so large, as the mitotic oscillator pathway of figure 7.2, but we want to infer properties for thousands of different starting values. In both cases is not possible to extract information from the model without automatic tools. At present, the strategy used is to simulate a pathway and to show with a graphical output the concentration behaviour with respect to time as in figure 7.2. This not allow to characterize completely the pathway behaviour since the number of simulation we can do is limited. This limit is avoided with the abstractions we are going to use. The abstract semantics we consider, obtained abstracting the collecting semantics, is formalized in the next definitions.

Definition 7.5. *Let $\alpha : \mathbb{R} \to A$ be the abstract function defined on the abstract domain A. Given a state $s_i =< h_{1,i}, \ldots, h_{n,i} >$ we define $\alpha : \mathbb{R}^n \to A^n$ as:*

$$\alpha(s_i) =< \alpha(h_{1,i}), \ldots, \alpha(h_{n,i}) >$$

Definition 7.6. *Given two states $s_i =< h_{1,i}, \ldots, h_{n,i} >$ and $s_l =< h_{1,l}, \ldots, h_{n,l} >$ and the abstract function $\alpha : \mathbb{R}^n \to A^n$ we define:*

$$\alpha(s_i) \vee \alpha(s_l) =< \alpha(h_{1,i}) \cup \alpha(h_{1,l}), \ldots, \alpha(h_{n,i}) \cup \alpha(h_{n,l}) > .$$

So the abstract semantics of the pathway PW can be defined:

Definition 7.7. *Given a starting state s_0 we define*

$$\alpha([\![PW]\!]^{t_0,t_f}(s_0)) = \bigvee_{i=t_0}^{i=t_f} \alpha(s_i), \text{ where } s_i \to s_{i+1}.$$

Also this semantics can be restricted to a set of variables L. This semantics is equivalent to execute the concrete calculus and then to abstract the collecting semantics of the variables at each step. The difference with respect to the abstract semantics classicaly used is that here no abstract functions are used. As consequence of the properties of abstract interpretation we have the correctness properties for the simulation from a starting state s_0:

$$[\![PW]\!](s_0) \subseteq \alpha([\![PW]\!](s_0))$$

This ensures that abstracting the semantics we loose informations in a "good" way, since we will never find wrong properties about the concrete computation but at most a superset of properties. So we can be not precise, but never wrong.

7.3 Abstract interpretation based analysis of pathways

Through some simple examples, we can now explore how abstract interpretation is useful in analyzing biochemical pathways, that is the kernel of this chapter. The results are obtained through an analyzer built in the Java language. The analyzer, take in input a Java class describing the pathway and a simulation step (in our case a single step of Eulero's computation for the pathway), it computes the concrete semantics (i.e. the simulation) and the output semantics as defined in the previous section (definition 7.7). To sum up, the steps of the analysis are:

- The simulation is computed according to the simulation method used
- The collecting semantics of the computation is stored
- The collecting semantics from time t_0 to time t_f of the simulation is abstracted according to the chosen domain.

The abstract analysis we consider are: congruence analysis, interval analysis, constant analysis.

7.4 A real world example: the mitotic oscillator

We apply the abstract interpretation based analysis to the mitotic oscillator pathway. The mitotic oscillator pathway is the one governing the crucial process of the cell division. This pathway is important for its regular oscillatory behaviour meaning that the concentration amount of molecules does not reach a steady state value, but it oscillate between a min and a max value with a constant period, regulating the fact that in dividing cells mitosis recurs at regular intervals. All the data about the mitotic oscillator comes from the works of Goldbeter [30] [29].

Oscillations have a great importance in biology, since there are a lot of systems that presents this kind of behaviour, as for example the circadian rythm, the heart pulsation, or the insuline production. We can see in figure 7.2 a simple rapresentation of the mitotic oscillator pathway. Cycline (C) activates cdc2-kinase (X) which activate the protease (M) which inhibits cycline. This kind of structure where a product inhibits one of its activators is called negative feedback structure and usually subtends to an oscillator behaviour.

According with the Goldbeter analysis, the corresponding differential equations regulating the pathway are:

$$\frac{d[C]}{dt} = v_i - v_d X \frac{C}{K_d + C} - k_d C, \tag{7.1}$$

$$\frac{d[M]}{dt} = V_1 \frac{1 - M}{K_1 + (1 - M)} - V_2 \frac{M}{K_2 + M}, \tag{7.2}$$

$$\frac{d[X]}{dt} = V_3 \frac{1 - X}{K_3 + (1 - X)} - V_4 \frac{X}{K_4 + X} \tag{7.3}$$

with

$$V_1 = \frac{C}{K_c + C} V_{M1}, V_3 = M V_{M3}.$$

Example 1: congruence domain, finding pathways oscillations

The first example introduces the use of the congruence domain [32]. Formally, if A is a set of natural numbers we say that $A \in [k + n\mathbb{Z}],\ k, n \in \mathbb{N}$ if $\forall a \in A$ we have $a = k\ mod\ n$. The abstraction function α abstract the set A in the corresponding congruence class where n is as large as possible. If no congruence class can be found for the set A it is abstracted in the $\top$ element meaning that A does not belong to a congruence class. The mitotic oscillator we consider is well known for the regularity precision of its oscillations, and an abstract interpretation analysis based on the congruence domain can automatically capture such a behavior, that is strictly related to a periodic changing on the concentrations values. The idea is simple but powerful. For each protein P we introduce a new variable called $Time_P$ representing at each moment the last time in which the derivative of the function representing concentration of P passes from positive to negative. In this moment the concentration of P stops its growing and begin to decrease and an oscillation occurs. If the period of oscillation is constant then the collecting semantics of the variable $Time_P$ belong obviously to a congruence class $[k + n\mathbb{Z}]$ where n is the period of oscillation. A possible oscillatory behaviour of the mitotic oscillator pathway is shown in figure 7.3. The starting values are the same of figure 7.2 except for parameter $V_2 = 0.5$. We focus the attention to cycline concentration. The collecting semantics of the variable $Time_C$, as in definition 7.4, starting at time 1 until time 100, results on the next set of values:

$$[\![PW]\!]^{1,100}_{|Time_C}(s_0) = \{7, 19, 31, 43, 55, 67, 79, 91\}$$

If we abstract this sequence according to the congruence domain abstract function, the result is the next congruence class:

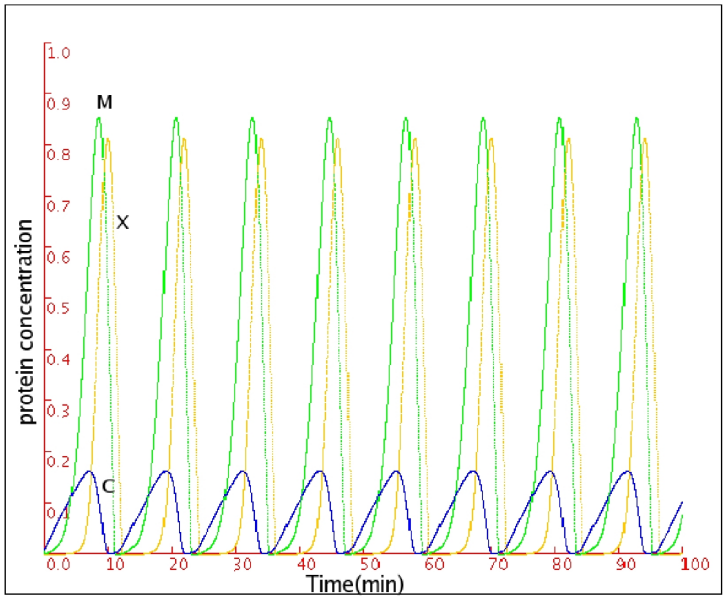

Fig. 7.3. The behaviour of the mitotic oscillator pathway with $V_2 = 0.5$ and simulation time = 100.

$$\alpha([\![PW]\!]^{1,100}_{|Time_C}(s_0)) = \alpha(\{7, 19, 31, 43, 55, 67, 79, 91\}) = 7 + 12\mathbb{Z},$$

This means that Cyclin concentration oscillates from time 7 with a period of 12. The oscillatory behaviour is automatically captured by our analysis. The abstraction function is used as a clever reader of long and complex series of numbers, and is able to extract the regularity in a set representing the collecting semantics of a variable. Additional information can be request about the oscillations, as time of the first oscillation, time of last oscillation, and number of oscillations. So the complete output of our Java-tool analysis is given by:

```
Time_C = 7 + 12Z
Oscillations number = 8 first oscillation = 7 last oscillation = 91
```

The power of this method is that now we can try all the simulations we want. We are no more dependent from a graphical output and we can see how the oscillation behavior changes if we change some starting parameters, for a very large range of values. For example we can answer to this question: "*how does the oscillation change if we change the values of the parameters V_{M1} and V_2 in the equations of the model?*" In the table 7.1 is reported the oscillation period of the cyclin, obtained by our simulation analysis changing V_{M1} from 0.1 to 3.0, with a step of 0.1 and changing V_2 from 0.1 to 3 with a step of 0.1. The starting time of analysis is 30 and the ending time of analysis is 100. The results for the value of V_2 from 2.3 to 3.0 is not reported in the table for problem of space since for this values of V_2 we have no oscillation for all the values of V_{M1}. The time requested for the analysis is

V_{M1} \ V_2	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2
0.1	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.2	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.3	36	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.4	28	48	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.5	24	35	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.6	22	28	42	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.7	21	24	34	43	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.8	20	22	29	38	42	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.9	19	20	25	34	38	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1.0	18	19	23	29	34	37	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1.1	18	18	21	26	31	34	35	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1.2	17	17	19	24	28	32	33	34	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1.3	17	16	18	21	26	29	31	32	N	N	N	N	N	N	N	N	N	N	N	N	N	N
1.4	17	16	17	20	24	28	30	31	31	N	N	N	N	N	N	N	N	N	N	N	N	N
1.5	16	15	17	19	22	26	28	29	30	29	N	N	N	N	N	N	N	N	N	N	N	N
1.6	16	15	16	18	21	24	26	28	29	29	29	N	N	N	N	N	N	N	N	N	N	N
1.7	16	14	15	17	20	23	25	26	27	28	28	N	N	N	N	N	N	N	N	N	N	N
1.8	16	14	15	17	19	21	24	25	26	27	27	27	N	N	N	N	N	N	N	N	N	N
1.9	16	14	14	16	18	20	23	24	25	26	26	26	27	N	N	N	N	N	N	N	N	N
2.0	15	14	14	15	17	19	21	23	24	25	25	25	26	26	N	N	N	N	N	N	N	N
2.1	15	14	14	15	16	18	20	22	23	24	24	25	24	25	N	N	N	N	N	N	N	N
2.2	15	13	13	14	16	18	20	21	23	23	24	24	24	24	24	N	N	N	N	N	N	N
2.3	15	13	13	14	15	17	19	21	22	22	23	23	24	24	23	24	N	N	N	N	N	N
2.4	15	13	13	14	15	16	18	20	21	22	22	23	23	23	23	24	24	N	N	N	N	N
2.5	15	13	13	13	14	16	17	19	20	21	22	22	22	23	23	22	22	N	N	N	N	N
2.6	15	13	12	13	14	15	17	18	20	21	21	21	22	22	22	22	22	23	N	N	N	N
2.7	15	12	12	13	13	15	16	18	19	20	20	21	21	21	21	22	21	20	23	N	N	N
2.8	15	12	12	12	13	14	16	17	19	20	20	20	21	21	21	21	21	21	21	23	N	N
2.9	15	12	12	12	13	14	15	17	18	19	19	20	20	20	21	21	21	21	20	21	N	N
3.0	15	12	12	12	13	14	15	16	17	18	19	19	20	20	20	20	20	20	20	20	21	N

Table 7.1. The oscillation period of cyclin concentration with respect to the values of the parameters V_{M1} and V_2. N means that no oscillation occurs. Time of analysis is from 30 to 100.

18885 millisecond, in a pentium III, 451Mhz with 322Mb of memory. In less than 20 seconds we have characterized the oscillator behaviour of 900 simulations with different starting parameters!

Example 2: Interval domain

The second example concerns the well-known intervals domain, introduced for the first time by Cousot&Cousot [18] in 1977. One of the most important informations we have to extract from a pathway, is the one concerning the values of concentration, particularly we want to know if a particular protein concentration remains in a certain range of values. Given a set A of rational the abstraction function α abstracts A in an interval of the form $[a\ b]$ where a and b are respectively the minimum and the maximum values in A. Table 7.2 represents an example of results of interval analysis where V_{M1} changes from 0.1 to 1.0, with a step of 0.1 and V_2 changes from 0.1 to 3 with a step of 0.1. The execution time is 3265 milliseconds (pentium III, 451Mhz 322Mb memory), the analysis starts at time 20 and stops at time 100. Looking at the table we are guaranteed that the protein concentrations remain in the intervals indicated for all the simulation time. Something better can also be done if we are wondering if the cyclin goes over a certain value. Suppose that our question is: *which are the values of* V_{M1} *and* V_2 *for which the cyclin concentration goes over the value of* 1.5*?* The results are reported in the table 7.3 obtained in 19786 milliseconds (pentium III, 451Mhz 322Mb memory). So we have characterized in few seconds the starting values of V_{M1} (from 0.1 to 3) and V_2 (from 0.1 to 1.7) for which cyclin concentration goes over 1.5, and a second important result is found.

$V_{M1} \setminus V_2$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.1	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.2	[0.0 0.81]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.3	[0.0 0.48]	[0.46 1.21]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.4	[0.0 0.36]	[0.01 0.71]	[0.46 1.52]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.5	[0.0 0.29]	[0.0 0.51]	[0.3 0.92]	[0.46 1.55]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.6	[0.0 0.25]	[0.0 0.41]	[0.05 0.66]	[0.46 1.13]	[0.46 1.56]	[0.46 1.57]	[0.46 1.57]	[0.46 1.57]
0.7	[0.0 0.22]	[0.0 0.34]	[0.01 0.52]	[0.26 0.81]	[0.46 1.33]	[0.46 1.56]	[0.46 1.57]	[0.46 1.57]
0.8	[0.0 0.2]	[0.0 0.3]	[0.0 0.44]	[0.1 0.63]	[0.46 0.95]	[0.46 1.5]	[0.46 1.56]	[0.46 1.57]
0.9	[0.0 0.18]	[0.0 0.27]	[0.0 0.38]	[0.03 0.53]	[0.26 0.75]	[0.46 1.09]	[0.46 1.54]	[0.46 1.56]
1.0	[0.0 0.16]	[0.0 0.24]	[0.0 0.33]	[0.01 0.45]	[0.14 0.62]	[0.44 0.86]	[0.46 1.23]	[0.46 1.55]

Table 7.2. The concentration intervals of cycline with respect to parameters V_{M1} and V_2. Analysis time is from 20 to 99.

$V_{M1} \setminus V_2$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
0.1	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.2	0.81	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.3	0.48	1.21	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.4	0.36	0.71	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.5	0.29	0.51	0.92	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.6	0.25	0.41	0.66	1.13	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.7	0.22	0.34	0.52	0.81	1.33	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.8	0.2	0.3	0.44	0.63	0.95	1.5	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
0.9	0.18	0.27	0.38	0.53	0.75	1.09	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
1.0	0.16	0.24	0.33	0.45	0.62	0.86	1.23	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
1.1	0.15	0.22	0.3	0.4	0.53	0.71	0.96	1.37	yes	yes	yes	yes	yes	yes	yes	yes	yes
1.2	0.14	0.2	0.27	0.36	0.46	0.6	0.8	1.07	1.49	yes	yes	yes	yes	yes	yes	yes	yes
1.3	0.13	0.19	0.25	0.32	0.42	0.53	0.68	0.88	1.18	yes	yes	yes	yes	yes	yes	yes	yes
1.4	0.13	0.18	0.23	0.3	0.38	0.47	0.59	0.75	0.97	1.28	yes	yes	yes	yes	yes	yes	yes
1.5	0.12	0.17	0.22	0.28	0.34	0.43	0.53	0.66	0.83	1.06	1.39	yes	yes	yes	yes	yes	yes
1.6	0.11	0.16	0.21	0.26	0.32	0.39	0.48	0.59	0.72	0.9	1.14	1.48	yes	yes	yes	yes	yes
1.7	0.11	0.15	0.19	0.24	0.3	0.36	0.44	0.53	0.64	0.79	0.97	1.23	yes	yes	yes	yes	yes
1.8	0.1	0.15	0.18	0.23	0.28	0.34	0.4	0.48	0.58	0.7	0.85	1.05	1.31	yes	yes	yes	yes
1.9	0.1	0.14	0.18	0.22	0.26	0.31	0.37	0.45	0.53	0.63	0.76	0.91	1.12	1.4	yes	yes	yes
2.0	0.1	0.14	0.17	0.21	0.25	0.29	0.35	0.41	0.49	0.58	0.68	0.81	0.98	1.19	1.48	yes	yes
2.1	0.09	0.13	0.16	0.2	0.23	0.28	0.33	0.39	0.45	0.53	0.62	0.73	0.87	1.04	1.26	yes	yes
2.2	0.09	0.13	0.16	0.19	0.22	0.26	0.31	0.36	0.42	0.49	0.57	0.67	0.78	0.92	1.1	1.33	yes
2.3	0.09	0.12	0.15	0.18	0.21	0.25	0.29	0.34	0.39	0.46	0.53	0.61	0.71	0.83	0.98	1.16	1.41
2.4	0.08	0.12	0.14	0.17	0.2	0.24	0.28	0.32	0.37	0.43	0.49	0.57	0.65	0.76	0.88	1.03	1.23
2.5	0.08	0.12	0.14	0.17	0.2	0.23	0.27	0.31	0.35	0.4	0.46	0.53	0.6	0.69	0.8	0.93	1.09
2.6	0.08	0.11	0.14	0.16	0.19	0.22	0.25	0.29	0.33	0.38	0.43	0.49	0.56	0.64	0.74	0.85	0.98
2.7	0.08	0.11	0.13	0.16	0.18	0.21	0.24	0.28	0.32	0.36	0.41	0.46	0.53	0.6	0.68	0.78	0.89
2.8	0.08	0.11	0.13	0.15	0.18	0.2	0.23	0.27	0.3	0.34	0.39	0.44	0.5	0.56	0.63	0.72	0.82
2.9	0.07	0.1	0.12	0.15	0.17	0.2	0.23	0.26	0.29	0.33	0.37	0.42	0.47	0.53	0.59	0.67	0.76
3.0	0.07	0.1	0.12	0.14	0.17	0.19	0.22	0.25	0.28	0.31	0.35	0.4	0.44	0.5	0.56	0.63	0.7

Table 7.3. The max values of the cyclin concentration with respect to the parameters V_{M1} and V_2. Time of analysis is from 20 to 99. 'yes' means that concentration is greater than 1.5.

Example 3: Constant domain

With the last analysis, we want now to characterize the constant behaviour of some proteins concentration, i.e. we want to know if a protein concentration is constant during a simulation. The constants domain, used in software analysis to find if a variable assume a constant value during the execution of a program is what we need. An abstraction analysis on the constants domain is able to find if a sequence of values is constant; a set of rational numbers A is abstracted in the value $x \in \mathbb{Q}$ if $\forall a \in A$, $a = x$, i.e. x is the only element of A, it is abstracted in the $\top$ element otherwise. It seems simple, but for thousands of values we can not do it without such an automatic tool. The values in A can also be rounded in order to verify if the values are constant within some tolerance limits. For example, the set {1.176158945472801, 1.176158969592738, 1.176158994270660, 1.1761590195060, 1.176159045298546} if rounded at the fourth decimal number is constant and results in the value 1.1761. Looking at the table 7.3, we notice that for some values of V_{M1} and V_2 the concentration of cyclin goes over the value 1.5. We may ask

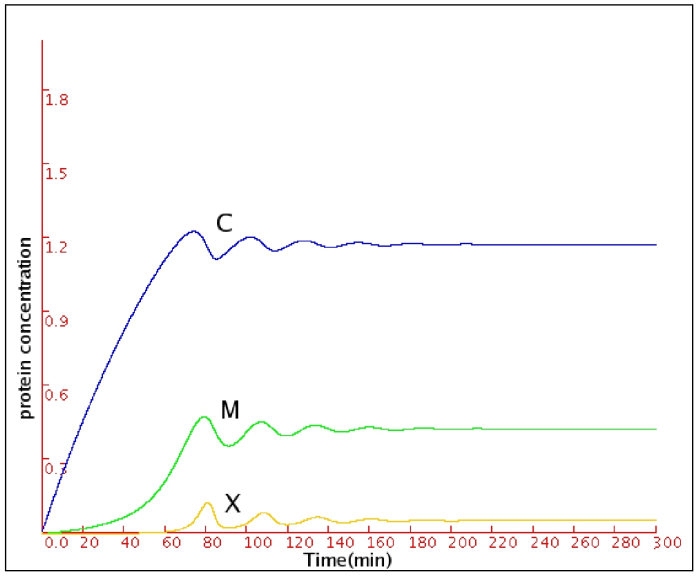

Fig. 7.4. A possible behaviour of the mitotic oscillator pathway: proteins concentration goes to constant values with $V_{M1} = 1.7$, $V_2 = 1.2$ and simulation time of 300.

if it takes some constant values in those cases. This consideration is supported by the results in figure 7.3, where after few time the reactants concentration become constant. So the question is: "*are there some values of* V_{M1} *and* V_2 *for which the concentration of cyclin remains constant from time 250 to 300?*" Table 7.4, obtained in 49034 milliseconds (pentium III, 451Mhz 322Mb memory) is the answer. Other interesting consideration can be done using the notion of abstract

$V_{M1} \setminus V_2$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
0.6	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.7	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.8	N	N	N	N	N	1.46	N	N	N	N	N	N	N	N	N	N	N	N	N	N
0.9	N	N	N	N	N	N	1.68	N	N	N	N	N	N	N	N	N	N	N	N	N
1.0	N	N	N	N	N	N	N	1.89	N	N	N	N	N	N	N	N	N	N	N	N
1.1	N	N	N	N	N	N	N	1.3	2.07	N	N	N	N	N	N	N	N	N	N	N
1.2	N	N	N	N	N	N	N	N	1.46	N	N	N	N	N	N	N	N	N	N	N
1.3	N	N	N	N	N	N	N	N	N	1.61	N	N	N	N	N	N	N	N	N	N
1.4	N	N	N	N	N	N	N	N	N	1.22	1.75	N	N	N	N	N	N	N	N	N
1.5	N	N	N	N	N	N	N	N	N	N	1.34	1.89	N	N	N	N	N	N	N	N
1.6	N	N	N	N	N	N	N	N	N	N	1.08	1.46	2.01	N	N	N	N	N	N	N
1.7	N	N	N	N	N	N	N	N	N	N	N	1.18	1.57	N	N	N	N	N	N	N
1.8	N	N	N	N	N	N	N	N	N	N	N	N	1.27	1.68	N	N	N	N	N	N
1.9	N	N	N	N	N	N	N	N	N	N	N	N	N	1.36	1.79	N	N	N	N	N
2.0	N	N	N	N	N	N	N	N	N	N	N	N	N	1.14	1.46	1.89	N	N	N	N
2.1	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1.22	1.55	1.98	N	N	N
2.2	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1.3	1.64	N	N	N
2.3	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1.12	1.38	1.72	N	N
2.4	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1.19	1.46	1.81	N
2.5	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	1.26	1.53	1.89

Table 7.4. The constant cyclin behaviour with respect to the parameters V_{M1} and V_2. Time of analysis is from 250 to 300. 'N' means that no constant concentration was found. The constant value of cyclin concentration is rounded to the second decimal.

interference for biological pathways introduced in the next section.

7.5 Abstract interference for biological pathways

In this section we introduce the notion of abstract interference for biological pathway. The aim is to characterize results of dynamic simulations as seen in the previous section in terms of interference as defined by Goguen [28]. Given h and l variables of a Program P, we can say that h interfere with l if:

$$\exists h_1, h_2 \in H, l_1 \in L, h_1 \neq h_2 :$$
$$[|P|]^l(h = h_1, l = l_1) \neq [|P|]^l(h = h_2, l = l_1),$$

where H is the domain of the variable h, L is the domain of l and where we mean with $[|P|]^l(h = h_1, l = l_1)$, the semantics of the program P restricted to the variable l starting from a generic starting state where h has h_1 as value and l has l_1 as value. So, if changes on starting values of variable h has effects on final value of variable l we have interference from h to l. In our pathway simulations the variable h can be a starting concentration value of a protein or a constant parameters of the Michaelis-Menten equation, and variable l can be the concentration value of another protein. As example we refer to the pathway with feedback of figure 7.1. Differential equation characterizing the pathway are those previously defined:

$$\frac{d[X_1]}{dt} = 10X_3^K X_5 - 5X_1^{0.5}, \frac{d[X_2]}{dt} = 5X_1^{0.5} - 10X_2^{0.5},$$
$$\frac{d[X_3]}{dt} = 5X_2^{0.5} - 1.25X_3^{0.5}, \frac{d[X_4]}{dt} = 5X_2^{0.5} - 1.25X_3^{0.5}, X_5 = 0.5$$

The values for the initial state of concentration are the following:

$$x1 = 1.1, X_2 = 0.5, X_3 = 0.9, X_4 = 0.75,$$

In figure 7.5 the pathway simulation is shown. Notice that changing the value of K from 0 to -16 the behaviour of the pathway completely changes. From an "interference" point of view we can say that variable K interfere with the variables X_1, X_2, X_3, X_4 of the simulation program. In this way we have characterize pathway simulation as variable interference. But we can go beyond applying the interval analysis on variable X_3. The result is that with $K = 0$ the variable X_3 is in the interval [0.648 0.912], and with $K = -16$ the variable X_3 is in the interval [0.899 1.220]. So using abstract interpretation we can say that variable K interferes with the property of variable X_3 of being in such an interval. The idea of characterizing variable interference in terms of abstract properties have been introduced in program security [26] where interference of properties of variables is introduced. Similarly, we introduce the notion of abstract interference for biological pathways.

Definition 7.8. ***Abstract interference 1*** *Given the pathway PW and the variable h and l, we say that h interfere with l with respect to the property α if*

$$\exists h_1, h_2 \in H, l_1 \in L, h_1 \neq h_2 :$$
$$\alpha([|PW|]^l(h = h_1, l = l_1)) \neq \alpha([|PW|]^l(h = h_2, l = l_1)),$$

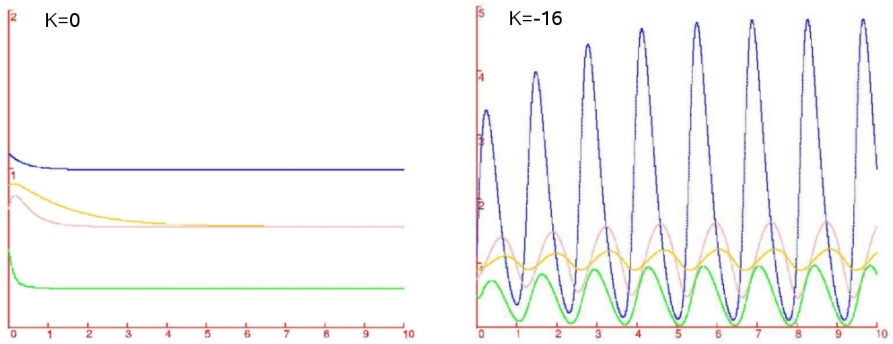

Fig. 7.5. On the left protein concentration with K=0, on the right with K=-16

So, in our example if α is the abstraction on the intervals domain, we say that K interferes with the variable X_3 with respect to the property α since we have:

$$\alpha([|FP|]^{X_3}(K = 0, X_3 = \bot)) = [0.648\ 0.912]$$

$$\neq$$

$$\alpha([|FP|]^{X_3}(K = -16, X_3 = \bot)) = [0.899\ 1.220]$$

So, changing the value of the variable K we interfere with the abstract property of the variable X_1 of being in a particular interval. But similarly to abstract interference used in program security, we can observe abstract properties of both the variables. So we introduce a second definition for abstract interference where we consider the abstract property ϕ and the abstract property α

Definition 7.9. ***Abstract interference 2***

Given the pathway PW and the variables h and l, the property ϕ of the variable h interferes with l with respect to the property α if:

$$\exists h_1^{\#}, h_2^{\#} \in \phi(H), l_1 \in L, h_1^{\#} \neq h_2^{\#} :$$

$$\alpha([|PW|]^l(h_1^{\#}, l_1)) \neq \alpha([|PW|]^l(h_2^{\#}, l_1))$$

In our example we consider α as the constant abstraction ϕ the interval abstraction. We abstract the starting concentration of X_1 and we evaluate its interference with starting concentration of X_2. We find that the property of X_1 of being in the interval [0.1 2.8] interfere with the property of the variable X_2 of being constant after 3 seconds with a constant value of 0.25. Indeed we have

$$\alpha([|FP|]^{X_2}(X_1 \in [0.1\ 2.8], X_2 = 0.5)) = 0.25$$

$$\neq$$

$$\alpha([|FP|]^{X_2}(X_1 \in [0.1\ 2.9], X_2 = 0.5)) = \top$$

Also the interference between variables properties have been characterized.

7.5.1 Abstract interference and mitotic oscillator

Thanks to abstract interference, deeper considerations can be done regarding the example of the mitotic oscillator.

- Looking to table 7.2, if $V_{M1} = 0.1$ the V_2 property of being in the interval [0.1 0.8] has no interference with the property of cyclin concentration of being in the interval [0.1 1.57].
- In table 7.3 if $V_{M1} \in [2.3\ 3.00]$ the property $V_2 \in [0.1\ 1.7]$ has no interference with the cyclin property of remaining in the interval [0.07 1.41]
- Looking at the same table, if $V_{M1} = 1.6$, the property $V_2 \in [1.1\ 1.7]$ has interference with cyclin property of being in [0.07 1.5]
- If $V_2 = 0.1$, the property $V_{M1} \in [2\ 3]$ has no interference with the property of cyclin of oscillating with a period of 15.

7.6 Conclusions

A possible use of the abstract interpretation theory on the simulation of biological pathways have been presented, particularly the abstraction on constants, intervals and congruences domains. We have shown how they can be used for extracting automatically properties about oscillatory behaviour and proteins concentration from thousands of simulations in few seconds. The advantages of this techniques are the following:

- The procedure is completely automated. So it is also simple to explore a large range of values to verify the desired property. For example, if you want to try one thousand of starting values for a variable, without abstract interpretation analysis you have to analyze one thousand of resulting graphs. With this kind of analysis the extraction of the properties of interest can be done automatically.
- We can set the precision we want, and we have obviously more precision than seeing graphical output. For example we have found the precise time from where a concentration value become constant that is not an information we can extract from a graphical output. Besides, we have automatically the output for all the proteins of the pathway: running one time our analysis we know immediately which proteins have the desired property.
- The method can be applied with no difficulties to pathways of thousand of proteins: abstract interpretation has been used to analyze programs of more than hundreds thousands of statements [12]. A pathway with thousands of proteins is only a program with thousands of variables to analyze. If the simulation model is well studied (differential equation or other) we can extract automatically the properties of all the proteins in one simulation.

The interactions between proteins in a pathway have been characterized with the notion of interference and abstract interference between variables [26], meaning that abstractions can also be applied to input parameters. Other abstract domains can be used to capture these kind of relations between input and output variables. For example in table 7.1 it seems to be a relation between V_{M1} and V_2 that cause the oscillation of cyclin, and such a relation should be captured by polyhedral analysis [21].

8

Conclusions

This thesis, treating both topological and dynamic points of view, concerns several aspects of biological networks analysis. Regarding the topological analysis of biological networks, the main contribution is the node-oriented point of view of the analysis. It means that instead of concentrating on global properties of the networks, we analyze them in order to extract properties of single nodes. An excellent method to face this problem is to use node centralities. Node centralities allow to identify nodes in a network having a relevant role in the network structure. This can not be enough if we are dealing with a biological network, since the role of a protein depends also on its biological activity that can be detected with lab experiments. Our approach is to integrate centralities analysis and data from biological experiments. A protocol of analysis have been produced, and the CentiScaPe tool for computing network centralities and integrating topological analysis with biological data have been designed and implemented. CentiScaPe have been applied to a human kino-phosphatome network and according to our protocol, kinases and phosphatases with highest centralities values have been extracted creating a new subnetwork of most central kinases and phosphatases. A lab experiment established which of this proteins presented high activation level and through CentiScaPe the proteins with both high centrality values and high activation level have been easily identified. The notion of node centralities interference have also been introduced to deal with central role of nodes in a biological network. It allow to identify which are the nodes that are more affected by the remotion of a particular node measuring the variation on their centralities values when such a node is removed from the network. The application of node centralities interference to the human kino-phosphatome revealed that different proteins affect centralities values of different nodes. Similarly to node centralities interference, the notion of centrality robustness of a node is introduced. This notion reveals if the central role of a node depends on other particular nodes in the network or if the node is "robust" in the sense that even if we remove or add other nodes the central role of the node remains almost unchanged. Further studies are needed to completely characterize the biological meaning of each centralities and to evaluate new analysis using node centralities interference and node centralities robustness. Lab experiments similar to the one of chapter 5 should be done to relate interference values and activation level of proteins.

The dynamic aspects of biological networks analysis have been treated from an abstract interpretation point of view. Abstract interpretation is a powerful framework for the analysis of software and is excellent in deriving numerical properties of programs. Dealing with pathways, abstract interpretation have been adapted to the analysis of pathways simulation. Intervals domain and constants domain have been succesfully used to automatically extract information about reactants concentration. The intervals domain allow to determine the range of concentration of the proteins, and the constants domain have been used to know if a protein concentration become constant after a certain time. The other domain of analysis used is the congruences domain, that if applied to pathways simulation can easily identify regular oscillating behaviour in reactants concentration. The use of abstract interpretation allows to execute thousands of simulation and to completely and automatically characterize the behaviour of the pathways. In such a way it can be used also to solve the problem of parameters estimation where missing parameters can be detected with a brute force algorithm combined with the abstract interpretation analysis. The abstract interpretation approach have been succesfully applied to the mitotic oscillator pathway, characterizing the behaviour of the pathway depending on some reactants. To help the analysis of relation between reactants in the network, the notions of variables interference and variables abstract interference have been introduced and adapted to biological pathways simulation. They allow to find relations between properties of different reactants of the pathway. Using the abstract interference techniques we can say, for instance, which range of concentration of a protein can induce an oscillating behaviour of the pathway. This part of the work can be enriched also with other domain and with further example of analysis.

References

1. Réka Albert and Albert L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan 2002.
2. Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.
3. Rajeev Alur, Calin Belta, Franjo Ivančić, Vijay Kumar, Max Mintz, George J. Pappas, Harvey Rubin, and Jonathan Schug. Hybrid modeling and simulation of biomolecular networks. *Lecture Notes in Computer Science*, 2034:19–32, 2001.
4. M. Antoniotti, B. Mishra, C. Piazza, A. Policriti, and M. Simeoni. Modelling cellular behavior with hybrid automata: Bisimulation and collapsing. In C. Priami, editor, *International workshop on Computational Methods in Systems Biology*, volume 2602 of *LNCS*, pages 57–74. Springer Verlag, February 2003.
5. M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics*, 25(1):25–29, May 2000.
6. Yassen Assenov, Fidel Ramirez, Sven-Eric Schelhorn, Thomas Lengauer, and Mario Albrecht. Computing topological parameters of biological networks. *Bioinformatics*, 24(2):282–284, January 2008.
7. G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(1), January 2003.
8. Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.
9. Albert-Laszlo Barabasi and Zoltan N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, February 2004.
10. Upinder S. Bhalla and Ravi Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283, january 1999.
11. Biondani, Viollet, Foretz, Laudanna, Devin-Leclerc, Scardoni, and De Franceschi. Identification of new functional targets of *ampk*α1 in mouse red cells. In *48th annual meeting of American society for cell biology*, 2008.
12. B. Blanchet, P. Cousot, R. Cousot, J. Feret, L. Mauborgne, A. Miné, D. Monniaux, and X. Rival. A static analyzer for large safety-critical software. In *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI'03)*, pages 196–207, San Diego, California, USA, June 7–14 2003. ACM Press.

13. Guido Caldarelli. *Scale-Free Networks: Complex Webs in Nature and Technology (Oxford Finance)*. Oxford University Press, USA, June 2007.
14. Nathalie Chabrier and Francois Fages. Symbolic model checking of biochemical networks. volume 2602 of *Lecture Notes in Computer Science*, pages 149 – 162. Jan 2003.
15. Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and José F. Quesada. The Maude system. In P. Narendran and M. Rusinowitch, editors, *rta10*, volume 1631 of *lncs*, pages 240–243. sv, 1999.
16. Melissa S. Cline, Michael Smoot, Ethan Cerami, Allan Kuchinsky, Nerius Landys, Chris Workman, Rowan Christmas, Iliana Avila-Campilo, Michael Creech, Benjamin Gross, Kristina Hanspers, Ruth Isserlin, Ryan Kelley, Sarah Killcoyne, Samad Lotia, Steven Maere, John Morris, Keiichiro Ono, Vuk Pavlovic, Alexander R. Pico, Aditya Vailaya, Peng-Liang L. Wang, Annette Adler, Bruce R. Conklin, Leroy Hood, Martin Kuiper, Chris Sander, Ilya Schmulevich, Benno Schwikowski, Guy J. Warner, Trey Ideker, and Gary D. Bader. Integration of biological networks and gene expression data using cytoscape. *Nature protocols*, 2(10):2366–2382, September 2007.
17. P. Cousot. Abstract interpretation based formal methods and future challenges, invited paper. In R. Wilhelm, editor, *Informatics — 10 Years Back, 10 Years Ahead*, volume 2000 of *Lecture Notes in Computer Science*, pages 138–156. Springer-Verlag, 2001.
18. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the 4th ACM Symp. on Principles of Programming Languages* (*POPL '77*), pages 238–252. ACM Press, New York, 1977.
19. P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Conference Record of the 6th ACM Symp. on Principles of Programming Languages* (*POPL '79*), pages 269–282. ACM Press, New York, 1979.
20. P. Cousot and R. Cousot. Higher-order abstract interpretation (and application to comportment analysis generalizing strictness, termination, projection and PER analysis of functional languages) (Invited Paper). In *Proc. of the 1994 IEEE Internat. Conf. on Computer Languages* (*ICCL '94*), pages 95–112. IEEE Computer Society Press, Los Alamitos, Calif., 1994.
21. P. Cousot and N. Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 84–97, Tucson, Arizona, 1978. ACM Press, New York, NY.
22. Paolo Crucitti, Vito Latora, Massimo Marchiori, and Andrea Rapisarda. Error and attack tolerance of complex networks. *Physica A: Statistical Mechanics and its Applications*, 340(1-3):388 – 394, 2004. News and Expectations in Thermostatistics.
23. Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
24. Steven Eker, Merrill Knapp, Keith Laderoute, Patrick Lincoln, and Carolyn Talcott. Pathway logic: Executable models of biological networks. In *Fourth International Workshop on Rewriting Logic and Its Applications (WRLA'2002*, volume 71 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002.
25. Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, March 1977.
26. R. Giacobazzi and I. Mastroeni. Abstract non-interference: Parameterizing non-interference by abstract interpretation. In *Proc. of the 31st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'04)*, pages 186–197. ACM-Press, NY, 2004. Venice, Italy, January 14-16,2004.
27. D. Gilbert. http://www.jfree.org/jfreechart/.

28. J. A. Goguen and J. Meseguer. Security policies and security models. In *1982 Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.
29. A. Goldbeter. A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. *Proc. Natl. Acad. Sci. USA*, 88(20):9107–11, 1991.
30. A. Goldbeter. *Biochemical Oscillations and Cellular Rhythms: The molecular bases of periodic and chaotic behaviour*. Cambridge University Press, Cambridge, 1996.
31. P. J. E. Goss and J. Peccoud. Quantitative modeling of stochastic systems in molecular biology using stochastic petri nets. In *Hybrid Systems: Computation and Control*, volume 95 of *Natl. Acad. Sci. USA*, pages 6750–6755, 1998.
32. Philippe Granger. Static analysis of arithmetical congruences. *International Journal of Computer Mathematics*, 30, 1989.
33. R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68(6):065103, Dec 2003.
34. P. Holme, M. Huss, and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4):532–538, March 2003.
35. Zhenjun Hu, Joe Mellor, Jie Wu, Takuji Yamada, Dustin Holloway, and Charles DeLisi. VisANT: data-integrating visual framework for biological networks and modules. *Nucl. Acids Res.*, 33(suppl_2):W352–357, 2005.
36. H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, May 2001.
37. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, October 2000.
38. M. P. Joy, A. Brock, D. E. Ingber, and S. Huang. High-betweenness proteins in the yeast protein interaction network. *J Biomed Biotechnol*, 2005(2):96–103, 2005.
39. Bjorn Junker, Dirk Koschutzki, and Falk Schreiber. Exploration of biological network centralities with centibin. *BMC Bioinformatics*, 7(1):219+, 2006.
40. Kinexus. http://www.kinexus.ca.
41. Kinexus. http://www.kinexus.ca/pdf/infopackage_kinex.pdf.
42. Dirk Koschützki, Katharina A. Lehmann, Leon Peeters, Stefan Richter, Dagmar T. Podehl, and Oliver Zlotowski. Centrality indices. In Ulrik Brandes and Thomas Erlebach, editors, *Network Analysis: Methodological Foundations*, pages 16–61. Springer, 2005.
43. P. Lecca, C. Priami, C. Laudanna, and G. Constantin. Predicting cell adhesion probability via the biochemical stochastic π-calculus. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 211–212. ACM Press, 2004.
44. Zsolt Lepp, Chunfei Huang, and Takashi Okada. Finding key members in compound libraries by analyzing networks of molecules assembled by structural similarity. *Journal of Chemical Information and Modeling*, 49(11):2429–2443, October 2009.
45. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.
46. M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June 2006.
47. Gheorghe Păun. Computing with membranes. *J. Comput. Syst. Sci.*, 61(1):108–143, 2000.
48. A. Regev. Representation and simulation of molecular pathways in the stochastic π-calculus. *Proceedings of the 2nd workshop on Computation of Biochemical Pathways and Genetic Networks*, 2001.

49. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In R.B. Altman et al., editor, *Pacific Symposium on Biocomputing*, pages 459–470. World Scientific, 2001.
50. Giovanni Scardoni. Interference analysis in systems biology. In *PLID 2005*, 2005.
51. Giovanni Scardoni. Analyzing biological pathways by abstract interpretation. In *EAAI 2006, Emerging Applications of Abstract Interpretation*, Wien, 2006.
52. Giovanni Scardoni, Michele Petterlini, and Carlo Laudanna. Analyzing biological network parameters with CentiScaPe. *Bioinformatics*, 25(21):2857–2859, 2009.
53. Urmi Sengupta, Sanchaita Ukil, Nevenka Dimitrova, and Shipra Agrawal. Expression-based network biology identifies alteration in key regulatory pathways of type 2 diabetes and associated risk/complications. *PloS one*, 4(12):e8100+, December 2009.
54. Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, November 2003.
55. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31, May 2002.
56. D.A. Smolen, P. Baxter and J.H. Byrne. Mathematical modeling and analysis of intracellular signaling pathways. In *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience*, pages 393–429. eds. Byrne, J.H. and Roberts, J.L. Elsevier, San Diego, 2004.
57. Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.
58. C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Proceedings of the Pacific Symposium on Biocomputing*, January 2004.
59. E. O. Voit. *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge University Press, 2000.
60. A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proceedings. Biological sciences / The Royal Society*, 268(1478):1803–1810, September 2001.
61. Duncan J. Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton University Press, Princeton, NJ, USA, 1999.
62. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.
63. S. Wuchty and P. F. Stadler. Centers of complex networks. *J Theor Biol*, 223(1):45–53, July 2003.
64. Takuji Yamada and Peer Bork. Evolution of biomolecular networks: lessons from metabolic and protein interactions. *Nature reviews. Molecular cell biology*, 10(11):791–803, November 2009.