UNIVERSITY OF VERONA

DEPARTMENT OF COMPUTER SCIENCE

GRADUATE SCHOOL OF NATURAL SCIENCES AND ENGINEERING

DOCTORAL PROGRAM IN COMPUTER SCIENCE

CYCLE XXXII

Computational Techniques for the Structural and Dynamic Analysis of Biological Networks

S.S.D. INF/01

Coordinator: ______________________
Prof. Massimo Merro

Tutor: ______________________
Prof. Rosalba Giugno

Doctoral Student: ______________________
Simone Caligola

Computational Techniques for the Structural and Dynamic Analysis of Biological Networks — SIMONE CALIGOLA
PhD Thesis

To Giulia,
for her extraordinary
support and patience

Abstract

The analysis of biological systems involves the study of networks from different *omics* such as genomics, transcriptomics, metabolomics and proteomics. In general, the computational techniques used in the analysis of biological networks can be divided into those that perform (i) structural analysis, (ii) dynamic analysis of structural properties and (iii) dynamic simulation. Structural analysis is related to the study of the topology or stoichiometry of the biological network such as important nodes of the network, network motifs and the analysis of the flux distribution within the network. Dynamic analysis of structural properties, generally, takes advantage from the availability of interaction and expression datasets in order to analyze the structural properties of a biological network in different conditions or time points. Dynamic simulation is useful to study those changes of the biological system in time that cannot be derived from a structural analysis because it is required to have additional information on the dynamics of the system. This thesis addresses each of these topics proposing three computational techniques useful to study different types of biological networks in which the structural and dynamic analysis is crucial to answer to specific biological questions. In particular, the thesis proposes computational techniques for the analysis of the network motifs of a biological network through the design of heuristics useful to efficiently solve the subgraph isomorphism problem, the construction of a new analysis workflow able to integrate interaction and expression datasets to extract information about the chromosomal connectivity of miRNA-mRNA interaction networks and, finally, the design of a methodology that applies techniques coming from the Electronic Design Automation (EDA) field that allows the dynamic simulation of biochemical interaction networks and the parameter estimation.

Abstract (Italian)

L'analisi dei sistemi biologici prevede lo studio di reti provenienti da diverse *omiche* come genomica, trascrittomica, metabolomica e proteomica. In generale, le tecniche computazionali usate nell'analisi delle reti biologiche possono essere suddivise in quelle che effettuano (i) un'analisi strutturale, (ii) un'analisi dinamica delle proprietà strutturali e (iii) una simulazione dinamica. L'analisi strutturale si basa sullo studio della topologia o della stechiometria della rete biologica come per esempio i nodi importanti della rete, motivi regolatori e la distribuzione dei flussi all'interno della rete. L'analisi dinamica delle proprietà strutturali si serve, generalmente, della disponibilità di datasets di interazioni e di dati di espressione per analizzare le proprietà strutturali della rete in diverse condizioni o punti temporali. La simulazione dinamica è utile per studiare quei cambiamenti del sistema biologico nel tempo che non possono essere dedotti da un'analisi strutturale perchè richiedono informazioni addizionali sulla dinamica del sistema. Questa tesi affronta ciascuno di questi argomenti proponendo tre tecniche computazionali utili per lo studio di diversi tipi di reti biologiche nelle quali l'analisi strutturale e dinamica è cruciale per rispondere a specifiche domande biologiche. In particolare, la tesi propone tecniche computazionali per l'analisi dei motivi regolatori di una rete biologica attraverso la progettazione di euristiche utili a risolvere in modo efficiente il problema dell'isomorfismo di sottografi, la costruzione di un nuovo workflow di analisi capace di integrare datasets di interazioni e di espressione al fine di estrarre informazioni riguardo alla connettività cromosomica nelle reti di interazione miRNA-mRNA e, infine, la progettazione di una metodologia che sfrutta tecniche provenienti dal campo dell'Electronic Design Automation (EDA) per la simulazione dinamica di reti di interazioni biochimiche e la stima dei parametri.

Contents

List of Figures VI

List of Tables VII

List of Listings IX

1 Introduction 1
1.1 Thesis contribution 2
1.2 Thesis overview 3

2 Biological Background 5
2.1 The Cell 5
2.2 High-throughput technologies 8
2.2.1 Next generation sequencing 8
2.2.2 Mass spectrometry 9
2.3 Biological networks 10
2.3.1 Cell signaling network 12
2.3.2 Protein-protein interaction network 12
2.3.3 Metabolic network 14
2.3.4 Gene regulatory network 14
2.3.5 Genetic interaction network 15
2.3.6 Other biological networks 15

3 Graph: the Basic Formalism to Represent Biological Networks 17
3.1 Basic notions on graphs 17
3.2 Graph properties and topology 18
3.3 Graph centrality 21

4 Mathematical and Computational Modeling of Biological Networks 25
4.1 Systems Biology 25
4.2 Mathematical models 25
4.3 Computational models 26
4.3.1 Ruled-based Systems 26
4.3.2 Process Algebra 26

4.3.3 Boolean Network . . . 27
4.3.4 Petri nets . . . 27
4.3.5 Structural properties of Petri nets . . . 28
4.3.6 Behavioural properties of Petri nets . . . 29
4.3.7 Extensions of Petri nets . . . 30

5 Techniques for the Structural and Dynamic Analysis of Biological Networks . . . 33
5.1 Analysis of biological networks . . . 33
5.2 Structural analysis . . . 34
5.2.1 Qualitative structural analysis . . . 34
5.2.2 Quantitative structural analysis . . . 36
5.3 Dynamic analysis of structural properties . . . 37
5.4 Dynamic simulation . . . 38

6 Search Strategies for the Subgraph Isomorphism Problem and Application to Biological Networks . . . 41
6.1 The variable ordering in subgraph isomorphism . . . 41
6.2 The RI ordering strategy . . . 42
6.3 Ordering strategies based on weighted centralities . . . 43
6.4 Experimental results . . . 43
6.5 Closing remarks . . . 47

7 Dynamic Analysis of Chromosome-wise miRNA-mRNA Connectivity . . . 49
7.1 Chromosome-wise connectivity . . . 49
7.2 A workflow to study the chromosome-wise miRNA-mRNA connectivity . . . 51
7.3 Experimental results . . . 54
7.3.1 Chromosome-wise miRNA-mRNA connectivity in cancer . . . 54
7.3.2 Lost miRNAs in cancer compared to healthy tissue . . . 58
7.4 Closing remarks . . . 62

8 Modeling and Dynamic Simulation of Metabolic Networks through Electronic Design Automation . . . 63
8.1 SystemC . . . 63
8.1.1 Time . . . 64
8.1.2 Modules . . . 64
8.1.3 Signals and ports . . . 65
8.1.4 Processes . . . 65
8.1.5 SystemC scheduler . . . 66
8.2 Overview of the methodology . . . 68
8.2.1 Stochastic Petri nets . . . 69
8.2.2 Modeling SPN through EFSM . . . 69
8.2.3 Implementation of metabolic networks through SystemC . . . 72
8.2.4 Assertion-based Verification for parameter estimation . . . 73
8.3 Experimental results . . . 75
8.3.1 Network construction from metabolomics data . . . 75
8.3.2 Integrating metabolomics observations with simulation analysis. . . . 76

8.4 Potential improvements of the methodology 79
8.5 Closing remarks 79

9 Conclusions 81

Appendices 83

A Publications 83
A.1 Published papers 83
A.2 Papers under review 84
A.3 Papers to submit 84
A.4 Book chapters 84

References 85

List of Figures

2.1 The structure of an eukaryotic cell (Figure source: *https://www.yourgenome.org/facts/what-is-a-cell*). 6

2.2 The central dogma of molecular biology. *DNA replication* is the basis of biological inheritance in all living organisms and it is the process by which a molecule of DNA is copied to produce two identical DNA molecules. *DNA transcription* is the process by which the DNA is transformed in RNA through the action of the enzyme RNA polymerase. In the *RNA translation* process, the ribosomes transform RNA into proteins. (Figure source: *https://www.yourgenome.org/facts/what-is-the-central-dogma*) 7

2.3 General NGS analysis workflow. Raw reads are preprocessed to assess their quality. Next, reads are aligned to a reference genome or transcriptome and, finally, the obtained alignment files are used for further analysis such as variant calling, peak calling and differential gene/transcripts expression analysis. 10

2.4 General MS analysis workflow. Raw MS data are preprocessed through filtering and normalization. Next, follows is a phase in which from the processed data the compounds are identified and quantified. Finally, the obtained quantified compounds are used, for example, for pathways analysis, network analysis and for modeling and simulation purposes. 11

2.5 Types of molecular interactions. Example of a molecular interaction network in which different types of physical and functional biological interactions are present. In blue, physical protein-protein interactions, they have not a specific direction. In black, gene regulation interactions, they can be physical or functional involving genes and proteins and they have specific directions. For example, a transcription factor (protein) can bind a specific DNA region of a gene or the down-regulation of a gene can affect the expression of another protein. In violet, metabolic interactions are biochemical reactions mediated by enzymes (protein) involving metabolites. In green, genetic interactions, they are functional interactions involving couples of mutated genes. In red, miRNA-mRNA interactions, they are physical directed interactions in which a microRNA binds specific regions of a messenger RNA. In yellow, cell signaling interactions, they usually involve proteins and they are physical and directed interactions. 13

3.1 Biological networks extracted from Figure 2.5 and their representation as graphs. (A) A protein-protein interaction network is modeled as an undirected weighted graph, (B) Signalling network are modeled as a directed graph, (C) miRNA-mRNA interaction networks are modeled as bipartite, directed and weighted graphs. Weights, usually, represent the strength or a score of confidence of the interaction. 18

3.2 Graph and subgraph isomorphism. (A) Q_1 and R_1 are two isomorphic graphs, (B) Q_2 is the query graph while R_2 is the reference graph. In this case, the graph Q_2 is a subgraph isomorphism of R_2 or, equivalently, R_2 contains a *match* of the subgraph Q_2. The different colors of the nodes represents the different labels. 20

3.3 Centrality measures commonly used to analyze biological networks: path-based measures, eigenvector-based measures and local measures. 22

4.1 Petri net formalism. (A) A simple example of Petri net model with seven places $P = \{p_1, p_2, \ldots, p_7\}$ and six transitions $T = \{t_1, t_2, \ldots, t_6\}$. This Petri net contains two tokens, the complete marking is $M = \{1, 0, 0, 0, 1, 1, 0, 0\}$. Examples of basic metabolic reaction of (B) synthesis, (C) decomposition and (D) reversible reaction modeled as Petri nets. 29

5.1 Structural and dynamic analysis of biological networks. Structural analysis (on the top of the figure) is based only on the complete topology of the network and it aims to discover properties such as the structural organization and the importance of the nodes of the network regardless which elements are active/inactive in a certain moment. Dynamic analysis of structural properties (on the middle of the figure) takes advantage of expression data in order to have an overview on how the topology of the network changes in different conditions or time points. Dynamic simulation (on the bottom of the figure) aims to describe how the biological network evolves in time. Usually, this analysis shows how the concentration (size of the nodes) of each element of the network changes in a time window. This change can be due to the activation of specific fluxes of biochemical reactions (red arrows). Dynamic simulation of biochemical reaction networks usually needs the knowledge of the topology of the network, the concentration of substances and the kinetic parameters of the biochemical reactions. 35

5.2 Network motifs. The following are motifs commonly found in biological networks. (A) *Feed-forward loop*. Type of networks: protein, neuron, electronic. (B) *Three chain*. Type of network: food webs. (C) *Four node feedback*. Type of network: gene regulatory, electronic. (D) *Three node feedback*. Type of network: gene regulatory, electronic. (E) *Bi-parallel*. Type of network: gene regulatory, biochemical. (F) *Bi-Fan*. Type of networks: protein, neuron, electronic (source: Pavlopoulos et al., 2011 [1]). 37

5.3 Dynamic simulation of an enzymatic reaction. E represents the enzyme concentration, S is the substrate, ES is the complex formed by E and S that releases the product P. In this simple example, the species S is entirely consumed after 60 seconds and converted into P. The purpose of the enzymes is to catalyze the reactions and they are not consumed (Figure source: *http://cbio.bmt.tue.nl/resolve/content/modelling_basics/modelTypes.php*). 39

6.1 The *variable ordering* μ_p of a pattern graph G_p determines how the nodes of the pattern are searched within the reference graph. In this case the red node is the first node to search while the blue is the last one. 42

6.2 The RI variable ordering creation. The violet node u_p^{i+1} is the node currently considered. The number of red nodes ($V_{u_p^{i+1},1}$) are evaluated at first because they are the nodes of the *core set* directly connected with the violet node. In case of parity between the violet node and others nodes, the number of green nodes are considered ($V_{u_p^{i+1},2}$). Finally the number of blues nodes ($V_{u_p^{i+1},3}$) are used for the final discrimination phase. The edges involved in the computation are highlighted in red. 43

6.3 Statistics regarding the properties of the pattern and target graphs used in the analysis: average number of nodes, edges, graph densities, labels and average frequency of the pattern graph labels in the corresponding target graphs 44

6.4 Performance of the strategies. The figure shows the ratio of SubGI instances in which each strategy has been the fastest solution. Instances are grouped by the type of graph: contact maps, PPIs and chemical structures. 44

6.5 Performance of the strategies based on the number of pattern edges. The figure shows for each methodology the ratio of SubGI instances for which it has been registered the fastest strategy (shown on top of the bars). Instances are grouped by graph typology and number of pattern edges (from 8 to 256). Patterns of 256 edges from chemical graphs were discarded because the sparsity of such graphs the extracted subgraphs tend to degenerate into simple paths. 45

6.6 Analysis of the running times of the strategies. (A) Heat map of the correlations between the investigated strategies applying the Mann-Whitney U test to the running times. (B) Similarities among strategies are computed by summing the absolute differences of their normalized running times. Normalization is performed by setting to 1 the slowest performance registered for each SubGI instance. U statistics and absolute time differences are used for the hierarchical clusterings of the strategies. 46

7.1 *Cis*- and *trans-chromosomal* interactions. In Blue, *cis* interactions involving two molecules (protein-protein or miRNA-mRNA) encoded in the same chromosome. In Red, *trans* interactions involving two molecules encoded in different chromosomes. Examples of *cis*- and *trans*-chromosomal interactions in (A) protein-protein interaction network (undirected graph) and (B) miRNA-mRNA interaction network (bipartite graph). 50

7.2 *Cis*- and *trans*-chromosomal interactions. Given a chromosome C, in a miRNA-mRNA interaction network, *trans* interactions can be of type *in*, involving mRNAs on C and miRNA elsewhere in the genome or *out*, involving miRNAs on C and mRNAs elsewhere in the genome. 51

7.3 Methodological workflow for studying the chromosome-wise connectivity of miRNA-mRNA interactions. (A) In the data integration step, miRNA and mRNA expression data from TCGA or other databases are correlated and filtered through TargetScan in order to obtain regulatory networks for all samples. (B) *Cis* and *trans* interactions are computed for each chromosome. *Cis* interactions involve miRNAs and genes of the same chromosome. *Trans* interactions can be either of type *in*, i.e., interactions between mRNAs of the chromosome of interest and miRNAs located in all other chromosomes, or *out*, i.e., interactions between miRNAs of the chromosome of interest with mRNAs of the other chromosomes. (C) The connectivity ratio (CR) is calculated for each chromosome of the network. 53

7.4 Chromosome-wise connectivity ratio (CR) for healthy and cancer samples. (A) Distribution of the chromosome-wise miRNA-mRNA CR for healthy (blue) and cancer (red) samples. The horizontal line represents CR = 1, i.e., the same amount of *cis* and *trans* interactions for a chromosome. The boxplots show a shift towards *trans* interactions in cancer since CR decreases in most of the chromosomes in cancer compared to healthy samples. The chromosomes 4, 14, 16, and 20 had the lowest CR together with 18 that had a CR = 0 in cancer because of the loss of the full interactome. Chromosome 13 had the highest CR. (B) Median log2 fold-change (log2FC) of the CR in cancer versus healthy samples. Chromosomes 4, 14, 16 and 20 show a strong change from healthy to cancer. Log2FC was calculated as following: $log2FC = log_2(CR_{cancer}) - log_2(CR_{healthy})$. In chromosome 18, it was not possible to calculate the log2 fold-change since CR in cancer was 0. For the complete list of CRs refer to tables 7.2 and 7.3. 55

7.5 Clustering of the chromosome-wise connectivity ratio (CR) for healthy and cancer samples. (A) Chromosome-wise miRNA-mRNA CR for healthy samples. The heatmap rows show the 10 considered TCGA cancer types while each column represents a chromosome. Hierarchical clustering of the CRs shows two main groups of chromosomes, the left one with a CR towards *trans* interactions and the right one with a CR towards *cis* interactions across cancer types. Chromosome 13 represents an individual cluster since it was the most *cis* chromosome. Overall, most chromosomes show about the same amount of *cis* and *trans* interactions in healthy samples. (B) The CR for each chromosome in the cancer tissues. The heatmap shows a shift to *trans* interactions for most of the chromosomes, particularly true for the cluster of chromosomes 4, 14, 16, 18 and 20. (C) Distribution of all CRs for normal and cancer tissues showing that the CR generally decreases in the cancer samples compared to the control samples. 57

7.6 Ratio (cancer VS healthy) of CR, number of miRNAs, mRNAs and miRNA-mRNA interactions. Each ratio was calculated taking the median value across cancer and healthy samples. 59

7.7 miRNAs physically lost and/or with lost centrality in the cluster of chromosomes with the most shift to *trans* interactions. A total of 31 miRNAs located on chromosomes 4, 14, 16, 18 and 20 were lost in the cancer miRNA-mRNA networks. Among these, 28 miRNAs were physically lost in cancer and nine miRNAs lost centrality in cancer. Six miRNAs were both lost physically and lost centrality in cancer. Their names are listed below the Venn diagram. 59
7.8 Closeness centrality in healthy and cancer miRNA-mRNA networks. Number of *most central* miRNAs in miRNA-mRNA interaction networks in healthy and cancer condition for each chromosome. A miRNA is considered among the *most centrals* if its centrality is in the upper quartile of the centralities of a miRNA-mRNA network. 60

8.1 SystemC scheduler phases: Initialization, evaluation, update, delta and timed notifications. 67
8.2 Overview of the SystemC methodology. (1) The SPN model is mapped into an extended finite state machine (EFSM) and (2) sythetisized into SystemC with the support for stochastic simulation. (3) A phase of parameterization is performed to retrieve parameters able to explain a biological property that can be defined through PSL language. A genetic-based input generator is used to generate parameters of the metabolic reactions that is guided by dynamic assertion-based verification (ABV). 68
8.3 Overview of the stochastic discrete simulation. 70
8.4 Examples of a basic reaction from the metabolite M_i to the metabolite M_j and their mapping in EFSM and SystemC models. (A) the Petri net model of a simple metabolic reaction, (B) the EFSM representation of the reaction related to M_i and M_j, (C) the SystemC template implementing the consumption and production processes of the metabolites M_i and M_j. Both `sc_module` have ports that manage the clock signal, the signal for exchange the metabolite concentration m_i and signals for the communication of the status of the reaction ($prod_R$ and $cons_R$). 71
8.5 Examples of assertion templates to verify a metabolite steady state. (A) Metabolite concentration stability, and (B) the oscillation of a metabolite concentration. σ and δ are user-defined tolerance constants. m_{TGT}, m_{MAX}, and m_{MIN} are user-defined concentration values of the observed metabolite. ta and ti are temporal counters initialized at the first oscillation, and that hold the time elapsed from the first state transition ($m_{MIN} \rightarrow m_{MAX}$ and $m_{MAX} \rightarrow m_{MIN}$, respectively). They are used to measure the positive edge and negative edge values. t is the counter, which is set, at each state transition from the second oscillation on, and it is used to measure the oscillation period. . . 74
8.6 Petri net model of the purine pathway. 77
8.7 Example of parameterization of the purine pathway in PLP-specific condition that leads the metabolites to stability within a simulation time of 10^5 clock cycles. 78

8.8 Results of the analysis of 10 selected parameterizations of the purine pathway in condition of stability for naive and PLP-specific cells. (A) Average difference of the delays obtained parameterizing the pathway in naive and PLP-specific condition. (B) Difference in the final concentration of the metabolites dAMP, dGMP and urate. 78

List of Tables

3.1 Summary of the centrality measure formulas adopted in this thesis. 23

7.1 TCGA cancer studies after preprocessing. Number of expressed miRNAs and mRNAs after correlation and TargetScan filtering (Figure 7.3 (A)) for each of the 10 TCGA cancer types included in this study. 54

7.2 Connectivity Ratio (CR) in healthy condition across TCGA studies. 56

7.3 Connectivity Ratio (CR) in cancer condition across TCGA studies. 56

7.4 miRNAs of chromosomes 4, 14, 16, 18 and 20 that were lost either physically and/or centrality in cancer compared to the corresponding healthy samples. References in which the considered miRNAs were found as tumour suppressors are reported. (*) Cancer types in bold highlights that miRNAs lost in our analysis were found as tumour suppressor in the same cancer tissue in literature. 61

8.1 **Time units in Systemc** 64

List of Listings

8.1 SystemC code in which the variable time represents 10 seconds. 64
8.2 SystemC code defining a module. 65
8.3 SystemC code representing read and write on ports. 65
8.4 Example of SystemC wait statement that suspends a SC_THREAD for 100 seconds. 66
8.5 Example of creation of a SystemC process as a SC_THREAD sensible to an event in a port start. 66

1

Introduction

The rapid growth of biological data is transforming biology into a data-driven science able to explain complex biological phenomena at molecular level and system level. This process was possible thanks to the advent of high-throughput technologies that allowed the development of different disciplines, called *omics*, that deal with the characterization and quantification of sets of bio-molecules that are responsible of the structure, function and dynamics of a living organism. The integration of these different types of data describing genes, transcripts, proteins, metabolites and so forth has allowed the creation of complex biological networks able to describe cell activities in physiological and pathological conditions of an organism. However, data itself is not knowledge and its interpretation would not have been possible without the theoretical and technological contribution made available by the Computer Science and other applied sciences such as Physics, Statistics and Mathematics. In fact, the synergistic action of different disciplines has created highly interdisciplinary fields such as Bioinformatics and Systems Biology that have unprecedentedly increased our knowledge of biological processes.

Several techniques have been proposed to analyze biological networks under different points of view: from the analysis of their structural properties to the analysis of their complex dynamic behavior. Techniques for the *structural analysis* aim to discover properties of the biological networks analyzing only their topology or stoichiometry. Many insights about the organization and functioning of different biological networks have been discovered studying the structure of the network through techniques coming from the graph theory. Most of these techniques have been applied to study the same biological network in different conditions or time points exploiting the possibility to integrate interaction datasets with expression data. This kind of analysis is named *dynamic analysis of structural properties*. However, some properties are not obviously derived from a structural analysis because they can depend on time and other biochemical parameters. This is the case, for example, of biochemical reaction networks in which an important aspect is the evolution in time of biochemical substances and for which *dynamic simulation* is of great importance and interest.

The design of techniques useful to discover static or dynamic properties implies the formulation of methods able to deal with the complexity and information lack of biological networks. For example, the resolution of certain computational problems defined on biological graphs can be very hard and it requires the design of smart heuristics able to reduce the time complexity of the problem. Theses methods are potentially applicable to any type of biological network because they are useful for their structural analysis. On the converse, specific biological networks such as regulatory networks and biochemical reaction networks can require the design of methods to study the dynamic change of their structural properties based on their condition and dynamic evolution in time. These methods need, respectively, the formulation of analysis workflows able to integrate data from different biological sources and intelligent computational procedures able to manage quantitative aspects such as concentrations and reaction parameters in order to explain the behavior of the network.

1.1 Thesis contribution

This work proposes three computational techniques that range from the structural analysis of biological networks to the dynamic simulation of metabolic networks passing from the dynamic analysis of structural properties in regulatory networks through the use of expression data in different conditions.

Structural analysis of biological networks makes extensive use of algorithms and measures coming from graph theory. Certain types of structural analysis on biological networks require efficient graph algorithms because of the complexity of the computational problem to solve. A common computational task involving graphs is the resolution of the subgraph isomorphism (SubGI) problem. This refers to the problem of searching a small pattern graph into a larger target graph. SubGI is a NP-complete computational problem that has important applications in Bioinfomatics, for example for searching protein complexes in large protein-protein interaction networks [2]. Several algorithms have been proposed to solve the SubGI problem in a very efficient way, one of these is the state-of-the-art RI algorithm [3]. The RI algorithm uses a strategy that is not dependent from information about the target graph because it exploits only the topological information about the pattern graph. This approach is both its strength and its limit because it can be important to have more global information about the target graph before start the search process. This thesis explores in Chapter 6 new SubGI search strategies, based on the RI algorithm, to speed-up the search into biological networks. The novelty of these techniques is that they exploit the information about the frequency of the target graph edge labels in combination with centrality measures in order to adapt the search strategy to the specific target graph. The performances of the proposed strategies have been tested on a well-known benchmark characterized by different types of biological networks. The contribution of the PhD candidate is the implementation of the proposed SubGI search strategies in C++ language and the collaboration in writing the article. The results of this work have been presented in the international conference on *Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB)* [4].

In general, graph measures can be used to retrieve interesting properties about the structural constraints and organization of a biological network. A recent work has shown that the analysis of the human protein-protein interaction network is crucial in order to understand how the division of labor between chromosomes has evolved and how it affects human diseases. In fact, it was shown that the *chromosome-wise connectivity* of protein-protein interaction networks differs between chromosomes and this may limit the impact of chromosomal aberrations as in the case of trisomy 21 [5]. An important aspect to better understand the human interactome is to explore the chromosome-wise connectivity at the level of non-coding RNA regulatory networks. In particular, the miRNA-mRNA chromosome-wise connectivity is largely unexplored and it is unknown how it changes in human diseases. Techniques based on the dynamic analysis of structural properties are a very effective tool because they can elucidate dynamic rewiring of the network in order to understand how the chromosome-wise connectivity of regulatory networks changes in the different disease conditions. This thesis proposes in Chapter 7 a new measure to calculate the ratio between cis- and trans-chromosomal miRNA-mRNA interactions together with an analysis workflow to study miRNA-mRNA regulatory networks in different conditions. The methodology was applied to cancer studies showing that the connectivity at the level of miRNA-mRNA chromosomal interactions strongly changes in cancer compared to healthy samples and this is especially true for a small cluster of chromosomes. The contribution of the PhD student is the collaboration in the design of the methodology, the implementation in the R/Bioconductor language and the analysis/interpretation of the results. The manuscript presenting these results is under preparation.

As discussed above, biochemical reaction networks such as metabolic networks require techniques that take into account quantitative aspects such as concentrations and reaction kinetics to perform a dynamic simulation of the system. The main problem of these methods is the lack of quantitative information in order to obtain simulation results matching the biological knowledge. Therefore, a parameterization step is often required and it needs the use of smart computational methods to manage the *state space explosion*. This thesis proposes in Chapter 8 a new methodology that takes inspiration from the application of electronic design automation (EDA) techniques in Systems Biology [6] to simulate and parameterize metabolic networks through the use of stochastic Petri nets. The methodology was applied to analyze the purine metabolic pathway in different conditions. The contribution of the PhD student is the design of the methodology and its implementation in the SystemC language, the analysis/interpretation of the results and the collaboration in writing the articles. The results of this methodology have been presented in the international conferences *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* (2019) and *Forum for Specification and Design Languages (FDL)* (2019) [7, 8]. Further results of the methodology have been presented in the international conference *Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB)* (2019) and submitted to the journal *Transactions on Embedded Computing Systems* (2020).

1.2 Thesis overview

The thesis starts with a description of the biological background, it continues with the description of several formalisms and techniques used for the analysis of biological networks and it ends with the description of the proposed techniques for the structural and dynamic analysis of biological networks.

Chapter 2 gives an overview of the basic concepts of molecular biology, a description of the modern technologies used to generate large-scale biological data and an introduction to the biological networks. Section 2.1 introduces the basic structure of the cell and its role in enclosing the DNA. Further, the central dogma of molecular biology and the concept of gene expression regulation are described. Section 2.2 introduces high-throughput technologies that allow to massively generate information about biological components such as genes, transcripts, proteins and metabolites. Section 2.3 introduces the general concept of biological networks along with the most important molecular interaction networks that govern the correct functioning of the cell: signaling networks, protein-protein interaction networks, metabolic networks, gene regulatory networks and genetic interaction networks. In addition, other biological networks are also briefly introduced.

Chapter 3 describes the formalism of graphs introducing basic concepts and techniques. Section 3.1 introduces the basic elements of graph theory and how these concepts are used to model biological networks. Section 3.2 describes graph properties and topology. Section 3.3 describes how to rank the nodes of a graph using graph centrality measures.

Chapter 4 introduces several formalisms used to model biological networks. In particular, after a short introduction, in Section 4.1, of the Systems Biology context and the formalisms for the modeling of biological networks, Section 4.2 introduces some of the commonly used mathematical models for the analysis of biological networks. Section 4.3 describes some of the computational models used in Systems Biology and, in particular, the formalism of Petri nets along with its related concepts of structural and behavioural analysis.

Chapter 5 gives an overall description of the analysis techniques that have been applied in the literature for the structural and dynamic analysis of the biological networks emphasizing how this thesis is placed in this context. After the introduction in Section 5.1, Sections 5.2 to 5.4 describe the methods for the structural analysis, dynamic analysis of structural properties and dynamic simulation of biological networks.

Chapter 6 describes the seven search strategies based on target-aware graph centrality measures that are proposed in this thesis to make the RI search strategy less dependent on local pattern graph properties. Section 6.1 introduces the concept of variable ordering that is crucial to solve the SubGI problem. Section 6.2 describes the search strategy of the RI algorithm. Section 6.3 describes the proposed strategies to speed the subgraph isomorphism up. Sections 6.4-6.5 describe, respectively, the experimental results and closing remarks.

Chapter 7 introduces the proposed new measure together with a workflow to study the chromosome-wise connectivity of miRNA-mRNA interactions in different conditions and the application to cancer studies. Section 7.1 describes the proposed formula to measure the connectivity of miRNA-mRNA interactions. Section 7.2 describes a general workflow for the construction of miRNA-mRNA interaction networks and the application of the proposed connectivity formula to them. Sections 7.3-7.4 show, respectively, the experimental results obtained with the proposed approach on cancer studies and the closing remarks.

Chapter 8 describes the proposed methodology to model, simulate and parameterize metabolic networks. Section 8.1 introduces the SystemC language and its basic building blocks such as modules, processes, signals and ports. Section 8.2 describes the methodology to model, simulate and parameterize metabolic networks using the SystemC language. Section 8.3 describes the experimental results obtained from the analysis of the purine metabolic pathway in two different experimental conditions.

Chapter 9 concludes the thesis discussing the obtained results.

2

Biological Background

This chapter gives an overview of the basic concepts of molecular biology, a description of the modern technologies used to generate large-scale biological data and an introduction to the biological networks.

Section 2.1 introduces the basic structure of the cell and its role in enclosing the DNA. Further, the central dogma of molecular biology and the concept of gene expression regulation are described.

Section 2.2 introduces high-throughput technologies that allows to massively generate information about biological components such as genes, transcripts, proteins and metabolites.

Section 2.3 introduces the general concept of biological networks along with the most important molecular interaction networks that govern the correct functioning of the cell: signaling networks, protein-protein interaction networks, metabolic networks, gene regulatory networks and genetic interaction networks. In addition, other biological networks are also briefly introduced.

2.1 The Cell

Cells are the basic building blocks of all organisms. They provide the structure and the fundamental functions to keep all living organisms alive. Cells are autonomous and self-sustaining but they exchange information with the environment and with the other cells, through a variety of chemical and mechanical signals, in order to accomplish various biological tasks. In general, cells can be divided into two main types: *prokaryotic* and *eukaryotic*. Both eukaryotic and prokaryotic cells share several features: a layer, called cell membrane, that separates the cell from the external environment, a medium, called cytoplasm, in which the biochemical reactions of the cell take place and the use of deoxyribonucleic acid (DNA) to store their genetic information. In both eucaryotic and procaryotic cells, DNA is transcribed into ribonucleic acid (RNA) and translated into proteins through the process of translation that is facilitated by specialized macromolecular machines called ribosomes. The main difference between prokaryotic and eukaryotic cells is that the latter have a membrane-bound nucleus. The nucleus is sorrounded by a membrane, called nuclear envelope, which contains and protect the DNA. The DNA determines the entire structure and function of the cell deciding if the cell has to to grow, mature, divide or die. In Figure 2.1 it is shown an example of the structure of an eukaryotic cell.

DNA is a double-stranded molecule characterized by subunits called nucleotides. Each nucleotide is composed by a phosphate group, a sugar group and a nitrogen base. The nitrogen

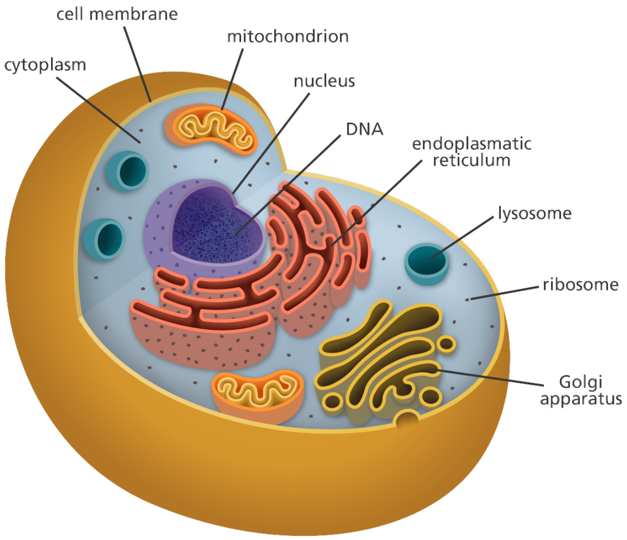


Fig. 2.1. The structure of an eukaryotic cell (Figure source: *https://www.yourgenome.org/facts/what-is-a-cell*).

bases that identify each nucleotide are *Adenine (A)*, *Thymine (T)*, *Guanine (G)* and *Cytosine (C)*. The order in which the bases appear in the DNA sequence encodes the instructions contained in the *genes*, that are the fundamental units used to make *proteins*, that are the molecules that effectively carry out the biological functions. The set of all genes is called *genome* and in eukaryotes it is organized in multiple *chromosomes*. The flow of information that involves DNA is described by the *central dogma of molecular biology*. This term was used for the first time by the biologist Francis Crick to express the idea that the processes that can involve DNA are the creation of new DNA from existing DNA (*duplication process*), the creation of RNA from DNA (*transcription process*) and the creation of proteins from RNA (*translation process*) (Figure 2.2). The central dogma states that the genes contain all the necessary information to make proteins and that the RNA, and in particular messenger RNA (mRNA), is the medium to transport this information to the ribosomes. DNA is converted into proteins through the process of *gene expression* that produces proteins depending on the needs of the cell.

However, the central dogma does not take into account an important process that strongly influences gene expression called *gene regulation*. This process includes a wide range of mechanisms used by the cell to control the production of specific gene products. Regulation can affect gene expression at each stage, for example at the transcriptional level and post-transcriptional level. Recently, non-coding RNAs (ncRNAs), that are RNA molecules not translated into proteins, have emerged as important regulators of gene expression. Among them, two important categories of functional ncRNAs include long non-coding RNAs (lncRNAs) and small non-coding RNAs such as microRNAs (miRNAs). In the last decade, extensive research on ncRNAs and in particular on miRNAs, has increased our understanding of post-transcriptional regulation of important cellular functions such as development, differentiation, growth and metabolism. In addition, miRNAs have been found as implicated in many human diseases such as cancer [9].

The cell can be considered as a dynamic system in which the different types of molecules such as DNA, mRNA, ncRNA, proteins and metabolites are linked together into a complex network of biochemical reactions and interactions; this allows the cell sustenance and its interaction with the environment. The study of the cell network as a whole is a very complex task, therefore typically the functions of the cell are studied based on the type of molecules. The study of each type of molecule involves the generation of different types of data and the use of specific technologies to study the so called *omics* such as genomics, transcriptomics, proteomics and metabolomics. These disciplines consist, respectively, in the study of the whole set of genes, transcripts, proteins and metabolites.

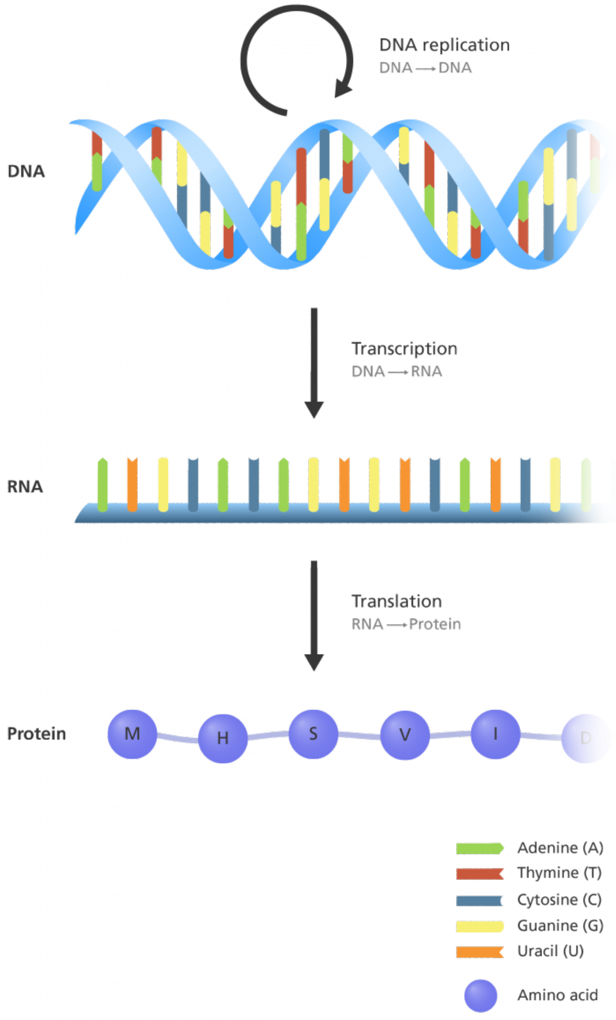


Fig. 2.2. The central dogma of molecular biology. *DNA replication* is the basis of biological inheritance in all living organisms and it is the process by which a molecule of DNA is copied to produce two identical DNA molecules. *DNA transcription* is the process by which the DNA is transformed in RNA through the action of the enzyme RNA polymerase. In the *RNA translation* process, the ribosomes transform RNA into proteins. (Figure source: *https://www.yourgenome.org/facts/what-is-the-central-dogma*)

2.2 High-throughput technologies

The advent of high-throughput technologies has increased our understanding of the molecular processes of the cell through the generation of a large amount of data coming from different omics. In particular, next-generation sequencing (NGS) technology has become a fundamental medium to study the genome and the transcriptome, while mass spectometry (MS) have been largely applied in proteomics and metabolomics. The analysis of data coming from different high-throughput technologies requires the use of specific tools and analysis pipelines in order to extract meaningful information. However, the integration of different types of biological data has become an important factor to understand the biology of the cell both from a qualitative and quantitative point of view.

2.2.1 Next generation sequencing

DNA sequencing is the process of determining the sequence of nucleotides of the DNA. Next generation sequencing (NGS) refers to DNA sequencing technologies that enabled the in-depth study of genomes. NGS has revolutionized the biological sciences because, unlike Sanger sequencing technology, it allowed (i) the generation of millions of short sequences, i.e. reads, in parallel, (ii) a faster process of sequencing, (iii) a low cost of sequencing and (iv) a detection of the output without the use of electrophoresis [10]. NGS is a very versatile technology. In fact, it can be used to sequence the whole genome of an organism (WGS), but also to study the transcriptome (RNA-seq), the set of protein-coding regions through the whole-exome sequencing (WES), specific gene regions through targeted (TS) or candidate gene sequencing (CGS), methylation (MeS) and the interaction between proteins and DNA (ChIP-seq) [11]. RNA-seq is useful, for example, to study transcriptional variations between different conditions of an individual and it is more sensitive in measuring gene expression compared to microarray [12]. WES is appropriate for the study of genetic variants that affect protein sequences because it provides coverage for more than 95% of human exons [13]. Conversely, WGS is well suited to study variants located outside the coding region and to study genomic rearrangements [14]. The study of the methylation is important because allows to gain information about the epigenetic DNA modifications involved in the control of gene expression [15]. Finally, ChIP-seq is useful because allows to study how proteins interact with DNA increasing our understanding of transcription factor regulatory action, chromatin modification and transcription.

NGS, unlike the Sanger method that can be considered a first-generation sequencing, generally can be divided into second and third generation sequencing.

The general workflow for a second-generation sequencing is characterized by a series of steps that is common between the different platforms: extraction and fragmentation of DNA/RNA, library construction through the binding of specific sequences, i.e. adapters, to the fragments, clonal amplification and sequencing [16]. After sequencing, reads are aligned to a reference sequence or assembled in order to perform different types of statistical analyses. However, second-generation NGS have some limitations. Due to the short length of the reads and problems related to the clonal amplification process, it is difficult to analyze certain regions of the genome such as areas with repeated DNA, high sequence homology or with extreme GC content [17]. Example of NGS second-generation platforms are Illumina, Roche, and Ion torrent.

Conversely, NGS of third generation are able to generate long reads with no amplification through a *single-molecule sequencing*. The available platforms that implement this technology are single-molecule real-time (SMRT) sequencing by Pacific Biosciences (PacBio) and nanopore sequencing by Oxford Nanopore Technologies (ONT). Each of these platforms uses a different technology, however the two important advantages of using single-molecule sequencing are the production of long reads and the precision in the detection of single base modification in a DNA molecule [17]. Independently from the specific technology, the basic data produced by NGS are the DNA raw reads in FASTQ format coming from a biological sample.

A typical NGS workflow starts with the preprocessing of the raw reads in order to assess their quality (Figure 2.3). In fact, reads could contain adapters or contaminants that it is possible to remove through the use of specific software. After the preprocessing step, the reads can be aligned to the reference genome or transcriptome in order to identify and annotate the DNA sequences. Typical software to align reads to a reference genome are STAR, Bowtie 2 and BWA that return alignment files in the SAM/BAM formats containing for each read the genome coordinates in which they were aligned. [18–20]. If the quality of the alignment is not good, further processing steps on reads can be performed in order to improve the accuracy of the next downstream analyses. The subsequent data analysis phase depends on the specific biological question that you want to answer and can include variant analysis, differential gene expression and peak calling. Each of these types of analysis can involve the use of different specialized tools and data format. For example, for differential gene expression analysis a phase of read quantification is needed in order to obtain a matrix of count representing the expression of each gene in the different samples.

2.2.2 Mass spectrometry

Mass spectrometry (MS) technologies have an extensive application in biology. Two of the common applications of MS are the study of small organic molecules and macromolecules [21]. Small organic molecules are compounds with a low molecular weight such as lipids, monosaccharides and metabolites while examples of macromolecules are nucleic acids and proteins. In the context of macromolecules, the study of proteins and their properties such as expression, interaction and activity was of particular interest in the last decades, making MS an important tool to analyze the *proteome* [22]. Conversely, the study of small molecules is becoming a new field of interest because it is now clear that small molecules play an important role in the disease etiology and treatment [23]. In this context, MS is a valuable tool because allows to obtain information regarding molecule mass, chemical formula, chemical structure and quantity of small molecules allowing the study of the *metabolome* [21].

Briefly, the aim of MS is the measure of the mass-to-charge (m/z) ratio of electrically charged molecules. However, MS cannot be applied in the same way for macromolecules and small molecules. In fact, despite each chemical species is formed by different elements and different quantities of them, there are several chemical species that cannot be discriminated simply by the mass. This is the case of macromolecules such as proteins that can have same molecular mass but different structural conformation. In that cases it is needed a phase before MS detection in order to differentiate the chemical species. For example, in proteomics proteolysis with specific enzymes is used for this purpose [24].

A typical MS data analysis workflow starts with raw MS data, that could be preprocessed through phases such as noise filtering, normalization, peak detection and matching [25, 26],

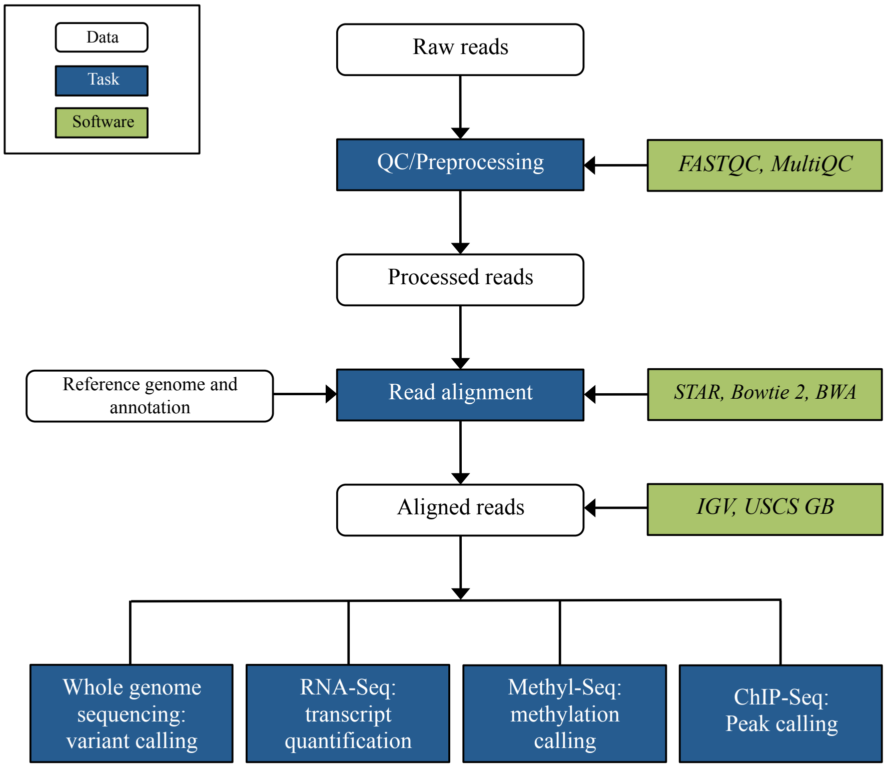


Fig. 2.3. General NGS analysis workflow. Raw reads are preprocessed to assess their quality. Next, reads are aligned to a reference genome or transcriptome and, finally, the obtained alignment files are used for further analysis such as variant calling, peak calling and differential gene/transcripts expression analysis.

and finishes with tables containing the expression values of the identified compounds (Figure 2.4). The identification and quantification phases involve the use of different software such as Mascot and MaxQuant [27, 28] for proteomics and MetFrag [29] for metabolomics, and specific databases useful to identify the compounds. The final expression values can be used for downstream analyses such as differential expression analysis, network and pathway analysis and modeling.

2.3 Biological networks

Data produced by NGS and MS are useful to characterize, for example, transcripts, proteins and metabolites that are involved in particular biological activities and/or conditions. However, usually the execution of a biological task by the cell is obtained through the non-linear interaction of many molecules. This type of biological behaviors can not be derived by a mere identification and/or quantification of the molecules involved. For this reason, the construction and analysis of biological networks has become crucial for the study of complex biological activities and it is the main topic of this thesis.

Intuitively, a biological network is a map that represents the connections between biological entities. When the biological entities are molecules such as genes, transcripts, proteins and metabolites the network can be called *molecular interaction network*. The interactions within a molecular interaction network can involve different types of molecules according to which

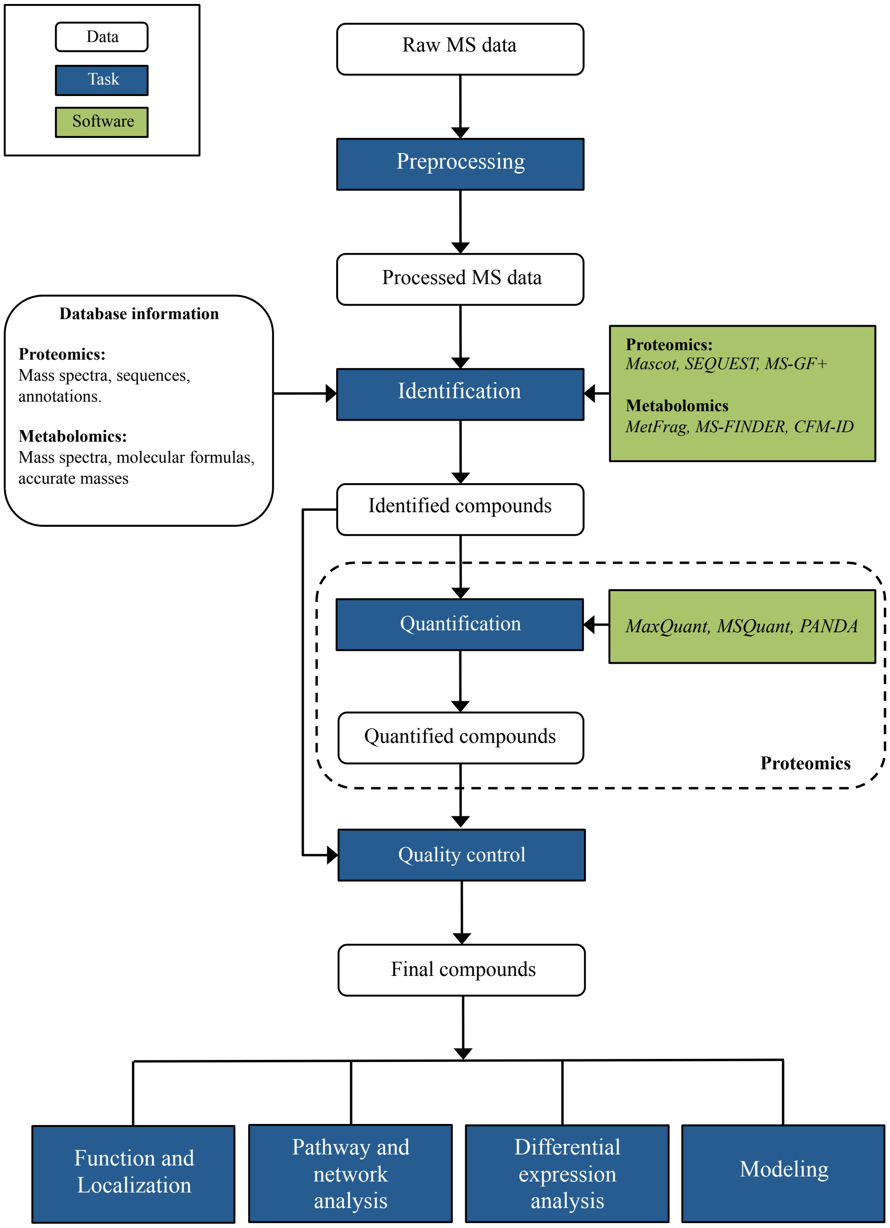


Fig. 2.4. General MS analysis workflow. Raw MS data are preprocessed through filtering and normalization. Next, follows is a phase in which from the processed data the compounds are identified and quantified. Finally, the obtained quantified compounds are used, for example, for pathways analysis, network analysis and for modeling and simulation purposes.

the interaction can be *physical* or *functional*. In a physical interaction there is a direct contact between the molecules while in a functional interaction the connection between the molecules can be indirect and virtually unknown. A network composed by biochemical reactions that lead to the transformation of a molecule to a different molecule is called *biochemical reaction network*. Usually this term refers to metabolic networks. The set of all molecular interactions forms a complex network called *interactome*. Figure 2.5 shows a small example of how different molecules interact in a molecular interaction network.

Until a few years ago biological networks were obtained through the study of few and specific cellular components. This made us understand in detail how some elements interact with each other to obtain a certain product or a change in a cell (i.e., *pathway*). This made possible to reconstruct some pathways like NF-kB and TGF-β signaling pathway [30,31]. However, our knowledge is still partial even in the most studied pathways. High-throughput technologies have shed light into many mechanisms of the cell through the large-scale identification of biological molecules, their expression patterns and their biochemical and genetic interactions [32]. The generation of this large-scale data has provided important information to construct different types of molecular networks that were crucial to understand better several biological processes. Based on the types of interactions, the main molecular interaction networks can be considered the following: *protein-protein interaction network (PPI)*, *genetic interaction network (GI)*, *gene regulatory network (GRN)*, *cell signaling network* and *metabolic network*. However, there are also other types of interesting biological networks that have been constructed and analyzed such as protein contact maps and chemical structures.

2.3.1 Cell signaling network

Cell signaling represents a very important biological process involved in the communication within a cell and between different cells. In fact, any cell can processes both intracellular and extracellular signals in order to control gene expression and as a consequence different activities such as differentiation, proliferation and cell death. Basically, the process of signaling starts through the binding of an extracellular signal by specific receptors located on the cell surface. Next, the signal is processed and propagated inside the cell through a series of biochemical reactions. This intracellular cascade of events is called *signaling pathway*. The set of molecules that interact in a specific signaling process constitutes a signaling network. Typically, the molecules involved in a signaling pathway are proteins in which phosphorylations catalyzed by protein kinases took place. It was shown that different signaling pathways can communicate each other in a synergistic or antagonistic way [33] and that results in a very complex signaling network. Examples of signaling pathways are the MAPK/ERK pathway that is an important regulator of cell cycle and proliferation, and Wnt signaling pathway that plays a critical role in embryonic development.

2.3.2 Protein-protein interaction network

Protein-protein interaction networks (PPI) represent how proteins cooperate to perform biological processes within the cell. The interaction between proteins is physical and it happens in specific binding regions. Protein interactions can be *stable* or *transient*. Stable interactions are, for example, the ones to create protein complexes such as ribosomes while transient interactions modify temporarily a protein to perform a specific action as in the case of protein kinases. The study of PPI networks is important because can be used, for example, to understand the role of some uncharacterized proteins and to acquire more knowledge on specific mechanisms of signaling pathways.

PPI networks knowledge took advantage from high-throughput technologies. In particular, in the last decades the rapid progress of MS technologies have increased our understanding of protein-protein interactions. One of the strength of MS is that can be used to obtain both a

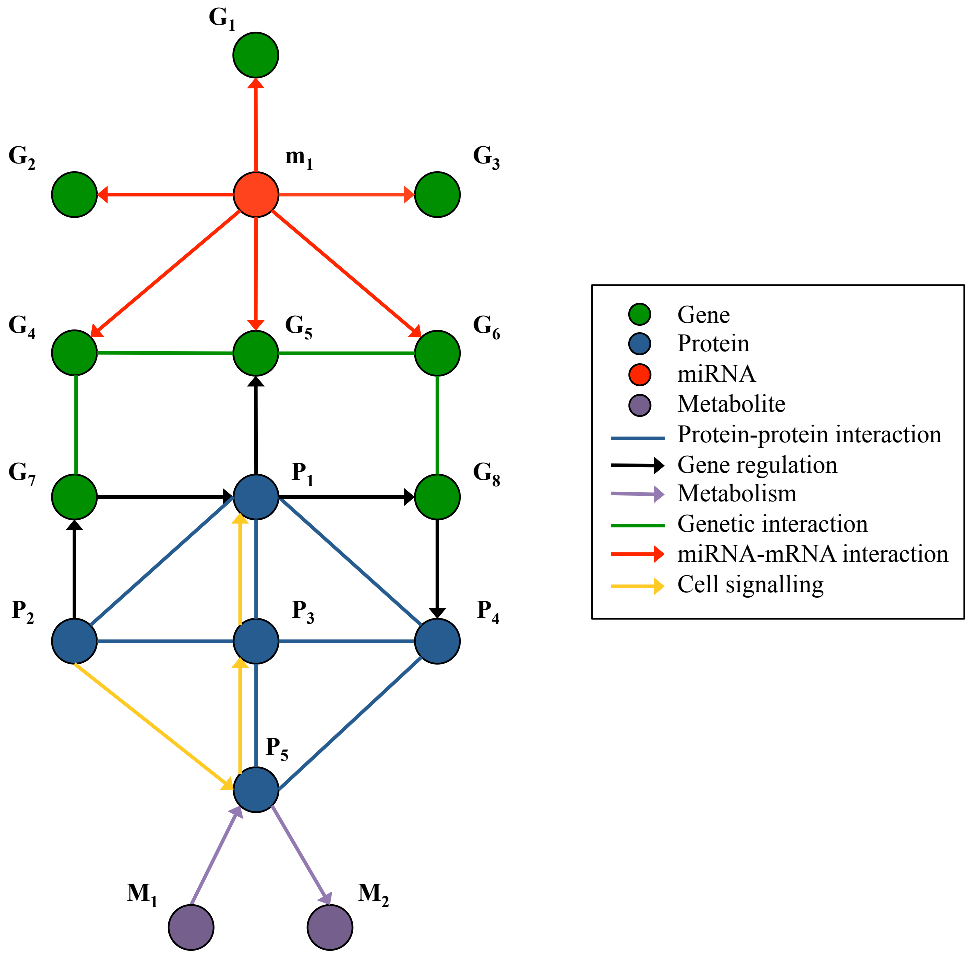


Fig. 2.5. Types of molecular interactions. Example of a molecular interaction network in which different types of physical and functional biological interactions are present. In blue, physical protein-protein interactions, they have not a specific direction. In black, gene regulation interactions, they can be physical or functional involving genes and proteins and they have specific directions. For example, a transcription factor (protein) can bind a specific DNA region of a gene or the down-regulation of a gene can affect the expression of another protein. In violet, metabolic interactions are biochemical reactions mediated by enzymes (protein) involving metabolites. In green, genetic interactions, they are functional interactions involving couples of mutated genes. In red, miRNA-mRNA interactions, they are physical directed interactions in which a microRNA binds specific regions of a messenger RNA. In yellow, cell signaling interactions, they usually involve proteins and they are physical and directed interactions.

qualitative and a quantitative overview of the protein-protein interactions, based on the technology and protocols used [34]. Qualitative techniques are useful to identify how proteins interact, while quantitative techniques also carry out information about the intensities of the identified interactions. Important applications of MS-based methods for PPIs discovery were, for example, the large-scale studies on the proteome of *budding yeast* and *Homo sapiens* that allowed to discover a large number of specific protein-protein interactions and its properties [35, 36].

2.3.3 Metabolic network

Metabolic networks represent the biochemical reactions between small molecules called metabolites. In such reactions the input metabolites, called *substrates*, are converted into *products* through the catalytic activity of enzymes. The interactions in a metabolic network have a specific direction that characterizes the *metabolic flow* or the *regulatory effect* of the reaction. The complete metabolic network of an organism represents the *Metabolism* and it is important because it generates essential components such as amino acids, sugars, and the energy required by the cell to perform its biological functions. The analysis of metabolic networks is useful because allows you, for example, to elucidate the role of specific metabolites or enzymes in controlling metabolic performance. In particular, several studies underlined the effectiveness to analyze topological properties of metabolic networks to discover the basic structures responsible for the robustness and error tolerance [37].

However, topological analysis does not take into account the quantitative relation between substrates and products (i.e., *stoichiometry*) and the information about the metabolic fluxes over the network. This simplification is not very suitable for the analysis of metabolic networks, in which the real performance depends most on the rate of the conversion of the substrates into products [38]. For this reason, methods for the quantitative structural and dynamic analysis have been proposed to analyze and predict the complex behaviour of metabolic networks. In particular, dynamic simulation comprises mathematical and computational techniques that take into account the concentration of metabolites and the reaction kinetics to accurately describe the progress of the metabolic networks in time.

The construction of accurate metabolic networks was boosted by high-throughput technologies. In fact, NGS gave the possibility to retrieve information about proteins and enzymes involved in the biochemical reactions while metabolomics MS data provided relevant biochemical and physiological information about metabolic processes.

2.3.4 Gene regulatory network

A gene regulatory network (GRN) represents how gene expression is controlled in a cell. In a GRN there are two actors that physically interact: genes and their regulators. Typical regulators are transcription factors (TF), i.e. proteins that are able to bind specific regions of DNA in order to turn *on* or *off* the transcription of a specific gene and, potentially, the corresponding protein. TFs act activating or inhibiting the recruitment of the RNA polymerase that is a protein responsible for the transcription of the genes. Further, TFs are themselves produced by the transcription and translation of a gene, therefore GRNs can be very complex. TFs are very important because their regulation activity is involved into crucial biological processes such as cell division, cell growth and cell death. Other important regulators that act in GRNs are miRNAs. Several studies showed that there is an interplay between transcriptional regulators such as TFs and post-transcriptional regulators such as miRNAs in order to decrease or potentiate signalling. In fact, it was shown that many predicted miRNAs targets are genes regulated at transcriptional level or transcribed themselves into TFs [39, 40]. However, regulatory networks formed by miRNAs and mRNAs targets have been also successfully used to study the onset and progression of several disease conditions where the miRNA regulation activity is relevant. Regulatory miRNA-mRNA networks were used, for example, to identify miRNAs and mRNA targets hat may affect the heart via NFAT hypertrophy and cardiac hypertrophy signaling and miRNA-mRNA connections that may be involved in the progression of H. pylori infection [41,42].

The analysis of GRNs is important because it is useful to elucidate the role of specific molecules (e.g., TFs, miRNAs) in the regulation of gene expression when their action is not obvious, for example, from a simple differential analysis. For this reason, very often GRNs are studied also through dynamic simulation to retrieve complex non-linear behaviors. Large-scale GRNs have been successfully constructed using several high-throughput techniques. A common used strategy was the combination of chromatin immunoprecipitation and microarray (ChIP-on-chip). This technique has been successfully used, for example, for the genome wide localization of TFs binding sites in yeast [43]. However, these techniques cannot be effectively applied to large and complex genomes such as the human genome; more robust strategies based on chromatin immunoprecipitation and DNA sequencing were proposed to obtain unbiased and precise global localization of transcription factors binding sites [44]. Another study showed the use of coupled chromatin immunoprecipitation and DNA sequencing to integrate TFs binding sites discovery with gene-expression and miRNA-target-site prediction to build a complex putative miRNA-transcription factor-target regulatory network [45]. miRNA-mRNA regulatory networks are usually constructed combining miRNA- and RNA-Seq techniques or microarray in order to correlate the expression of miRNAs and mRNAs and construct the network [46,47].

2.3.5 Genetic interaction network

A genetic interaction (GI) network represents the functional association between genes in which the simultaneous mutation induces a significantly different phenotype with respect to the single mutation of each. A GI does not imply either an interaction between the corresponding proteins or that the genes are expressed in the cell and, for this reason, the functional understanding of genetic interaction networks is not trivial. A GI can involve, for example, two genes that act in the same biological pathway or in different pathways that have a compensatory function that is not obvious. A genetic interaction is *positive* if the combination of the two mutants results in a greater fitness compared to the combination of the two corresponding single mutants. A genetic interaction is *negative* or *synthetic lethal* if the action of the two mutants results in a fitness defect that is more extreme than expected. The study of this type of interactions is of particular interest because they can be exploited to discover the gene function or new therapeutic targets [48].

For these reasons, in the last decade GI networks have been constructed and analyzed through large-scale studies because they gave us the opportunity to better understand gene function, genetic relationships, biological robustness and evolution [49]. The analysis of GIs allowed, for example, to discover synthetic lethal interactions in yeast [50], a significant overlap between genetic interactions and protein-protein interactions [51] and to discover the function of several genes in *Saccharomyces cerevisiae* [52].

2.3.6 Other biological networks

Other examples of biological networks are protein contact maps and protein chemical structures. A protein contact map represents the distance between pairs of amino acid residues of a protein three-dimensional structure. In particular, the pair of amino acid residues is connected if their distance is under a predetermined threshold. Protein chemical structures are networks that represent the chemical bonds between atoms.

3

Graph: the Basic Formalism to Represent Biological Networks

Biological networks represent how biological entities interact in a complex and non-linear way to perform biological tasks. A common mathematical formalism to represent biological networks is the *graph*. Graphs are a very intuitive and abstract way to represent the various types of biological networks interactions but also provide a solid mathematical framework to analyze them. This chapter describes the formalism of graphs introducing basic concepts and techniques.

Section 3.1 introduces the basic elements of graph theory and how these concepts are used to model biological networks.

Section 3.2 describes graph properties and topology.

Section 3.3 describes how to rank the nodes of a graph using graph centrality measures. The measures defined in this section are the building blocks of the subgraph isomorphism strategies proposed in Chapter 6.

3.1 Basic notions on graphs

Formally, a graph can be defined as a pair $G = (V, E)$, where V is the set of *nodes* and E is the set of *edges* connecting them. Given two nodes v_1, v_2, the pair (v_1, v_2) represents an edge. The edges of a graph can be *directed* if a direction of the connection is defined or *undirected* otherwise. A graph with undirected edges is called *undirected graph*, otherwise the graph is called *directed graph*. The *neighbourhood* $N(v) = \{v' \in V : (v, v') \in E\}$ of a node v is the set of nodes connected to v. Given a graph $G = (V, E)$ then $G' = (V', E')$ is called *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$ and $(v'_1, v'_2) \in E' \Rightarrow v'_1, v'_2 \in V'$. A graph $G = (V, E)$ is called *bipartite graph* if it is possible to partion the set V into two sets V_1 and V_2 such that $(v_1, v_2) \in E$ implies $(v_1) \in V_1$ and $v_2 \in V_2$ or $(v_2) \in V_1$ and $v_1 \in V_2$. It is possible to annotate the nodes and the edges of the graph with additional information. A function $\alpha : V \mapsto \Sigma_V$ is a injective function that maps nodes to a set of labels Σ_V. Labels can be, for example, the names of the nodes or specific IDs associated to them. A function $w : E \mapsto R$ associates a real number to each edge of the graph. Usually, a weight $w_{i,j}$ represents the strength of the connection between the two nodes, therefore a large weight corresponds to higher reliability of interaction. Graphs with labels and weights are called, respectively, *labeled graphs* and *weighted graphs*. A graph can be represented mathematically with a $|V| \times |V|$ *adjacency matrix* $A = \{a_{v_1,v_2}\}$, where $a_{v_1,v_2} = 1$ if the nodes v_1 and v_2 are connected and 0 otherwise. In undirected graphs A is symmetric.

In a biological network, nodes are the biological entities while edges are the physical or functional interactions in which they are involved. In this context, the terms graph and network can be used interchangeably. Examples of undirected biological networks are PPI networks and genetic interaction networks, while cell signalling networks, metabolic networks and GRNs are directed. In general, directed graphs are effective to model biological networks in which it is important to underline the sequential order of the interactions in order to obtain a specific output, as in the case of a signalling pathway. Most of the biological networks can be considered weighted because they carry out the information regarding the strength of the interactions or their *confidence score* to specify the probability that the interaction is true. Practical examples of weights assigned to the edges of a biological network are those used to assess the sequence or structural similarities between proteins, the co-expression of genes and the negative correlation of the pairs miRNA-target genes [46, 47, 53].

Figure 3.1 shows a summary of how the concepts on graphs explained in this paragraph can be applied to represent specific biological networks.

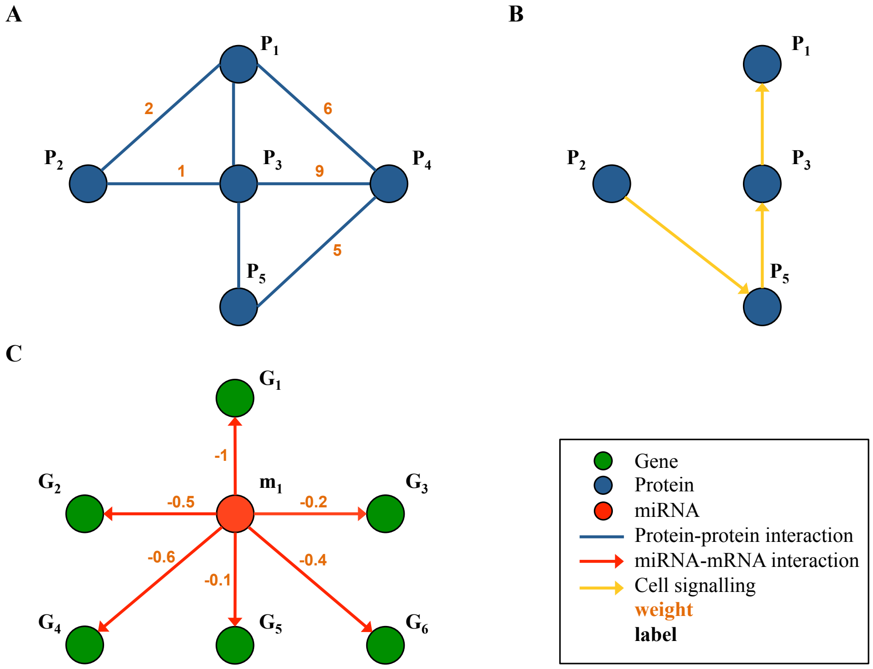


Fig. 3.1. Biological networks extracted from Figure 2.5 and their representation as graphs. (A) A protein-protein interaction network is modeled as an undirected weighted graph, (B) Signalling network are modeled as a directed graph, (C) miRNA-mRNA interaction networks are modeled as bipartite, directed and weighted graphs. Weights, usually, represent the strength or a score of confidence of the interaction.

3.2 Graph properties and topology

Graph properties and topology can provide important insights about a biological network such as the internal organization and the evolutionary constraints [1], allowing you to link the structure with the function.

An important property of a graph is the *graph density*: $D = \frac{2|E|}{|V|(|V|-1)}$. Density shows how many edges a graph has compared to the maximal number. A graph is *dense* if the number of edges is close to the maximum and *sparse* otherwise. The density varies based on the type of the biological network, though several studies show that the sparsity is feature of the biological networks and it is related to their robustness [54, 55]. A graph in which every pair of distinct nodes is connected by an edge is called *complete graph*.

Some of the properties of a graph and its nodes can be defined as a function of its *paths*. A path is a sequence of edges $S = \{(v_1, v_2), (v_3, v_4), \dots (v_{m-1}, v_m)\}$ that joins a sequence of distinct nodes. If there is a path between every pair of nodes the graph is called *connected* and *disconnected* otherwise. A *shortest path* between the nodes v_1 and v_2 is a path from v_1 to v_2 in which the sum of the weights of its constituents edges is minimized. In an unweighted graph all the weights are assumed to be 1. The distance $d = \delta(v_1, v_2)$ between the nodes v_1 and v_2 is defined as the length of the shortest path from v_1 to v_2. If two nodes v_1 and v_2 are not connected their distance can be defined as $d = \delta(v_1, v_2) = \infty$.

It was shown that complex real networks are made of building blocks, i.e. *motifs*, which recur significantly more often than random networks [56]. The task of searching network motifs in large biological networks is important and there is the necessity to design efficient algorithms able to manage the complexity of actual networks. Formally, the problem of searching small motifs into a large graph can be defined as a *subgraph isomorphism (SubGI)* problem. In general, given a *pattern graph* $Q = (V, E)$ and a *reference* (or *target*) *graph* $R = (V', E')$, a function $f : V \to V'$ is called *isomorphism* if M is an edge-preserving bijective function such that for $u, v \in V$, $(u, v) \in E$ if and only if $(f(u), f(v)) \in E'$. If the function f exists, the two graph Q and R are called *isomorphic*. The computational problem of verifying if two graphs are isomorphic is called *graph isomorphism (GI) problem*. However, it is often needed to take into account another constraint in the resolution of the GI problem: the compatibility of the nodes labels of the two graphs. The SubGI problem is obtained from GI assuming that the function f can be injective. More precisely, this type of SubGI is called *monomorphism*. Practically, the SubGI problem is the computational task to determine if a graph Q called *query graph* is present into a larger graph R called *reference graph*. Notably, SubGI is a NP-complete problem, therefore it is necessary to develop efficient heuristics to make the problem affordable. SubGI search strategies are subject of this thesis and they will be discussed in Chapter 6. Figure 3.2 shows two simple examples of isomorphic graphs and subgraphs.

The topology of a graph can reveal important properties of the modeled system. An important characteristics in network topology is the *degree distribution* $P(k)$. The degree distribution of a graph is defined as the fraction of the nodes with degree k. A related concept is the *cumulative degree distribution* that is the fraction of nodes of the graph that have degree smaller than k.

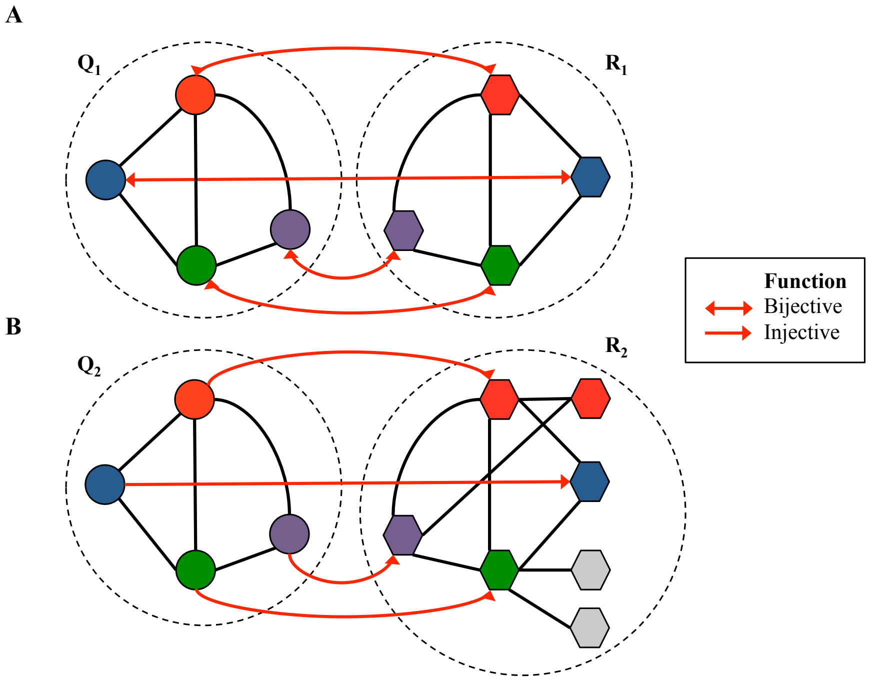


Fig. 3.2. Graph and subgraph isomorphism. (A) Q_1 and R_1 are two isomorphic graphs, (B) Q_2 is the query graph while R_2 is the reference graph. In this case, the graph Q_2 is a subgraph isomorphism of R_2 or, equivalently, R_2 contains a *match* of the subgraph Q_2. The different colors of the nodes represents the different labels.

3.3 Graph centrality

A centrality measure is a very useful indicator of the importance of a node within a graph. This concept can give, for example, important insights on which node is the most influential node over the network and which nodes are fundamental in order to maintain intact the communication within the network. The term *importance* of a node can assume different meanings based on what is considered important within the graph. This fact led to the different formulation of centrality measures over the years.

The three classical measures of graph centrality are *degree (DEG)*, *closeness (CL)* and *betweenness centrality (BET)* [57]. Among these, degree centrality is the most basic because it considers only how many nodes are adjacent to another. CL measures how close a node is from all other nodes in the graph taking into account shortest paths. BET centrality measures how much a node is in the middle of the shortest paths between pairs of nodes in the network. A node with high betweenness is a node that can potentially control the flow of information in the graph. CL and BET centrality are both based on shortest paths. However, the assumption that the communication between the nodes of the network take place over the shortest paths is not necessarily true. For these reasons, other centrality measures that do not take into account the shortest paths of the graph have been proposed. *Eigenvector centrality (EIG)* ranks the importance of a node taking the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of the graph. Practically, EIG assigns a score to a node based on the influence that the node has in the network. EIG assumes that a node is important if its neighbors are important too. A problem related to EIG centrality is that it neglects multiple shared paths between the nodes of a network [58]. *Information centrality (INFO)* was introduced as a measure that takes into account all possible paths giving them a weight that measures the information they contain [58].

Graph centralities were first developed for social network analysis but they had extensive application in biological networks. In fact, more recently, a number of centrality measures have been proposed to discover essential proteins in PPI networks. For example, *subgraph centrality (SUB)* [59] was introduced as a measure appropriate to characterizing network motifs in scale-free networks. The authors showed the effectiveness of this measure in analyzing essential proteins in PPI networks. Other examples of centrality measures introduced for studying the essentiality of the proteins in PPI networks are *local average connectivity-based centrality (LAC)* and *network centrality* [60,61]. Both measures rank the importance of a node evaluating the relationship between a node and its neighbors.

All the above centrality measures are commonly used to analyze biological networks. This thesis proposes, in Chapter 6, subgraph isomorphism strategies based on these centrality measures except DEG, because only non trivial centrality measures were considered. Here, they are divided into three main categories: path-based measures, eigenvector-based measures and local centrality measures (Figure 3.3). Path-based is a type of centrality measures that are considered *global* because they take into account the whole network using paths to explore it. On the contrary, local measures take into account only local information (e.g., neighborhood) of a node, excluding the rest of the network. Finally, eigenvector-based measures include both global and local measures in which the centrality of a node is based on the calculation of eigenvectors and eigenvalues of the adjacency matrix of the graph. The motivation behind the use of these measures is the exploration of a wide set of centralities in the perspective of a multi-approach environment in which the choice of the best strategy depends on the properties of the target graphs involved. All the formulas of the centrality measures adopted in this thesis are reported in Table 3.1.

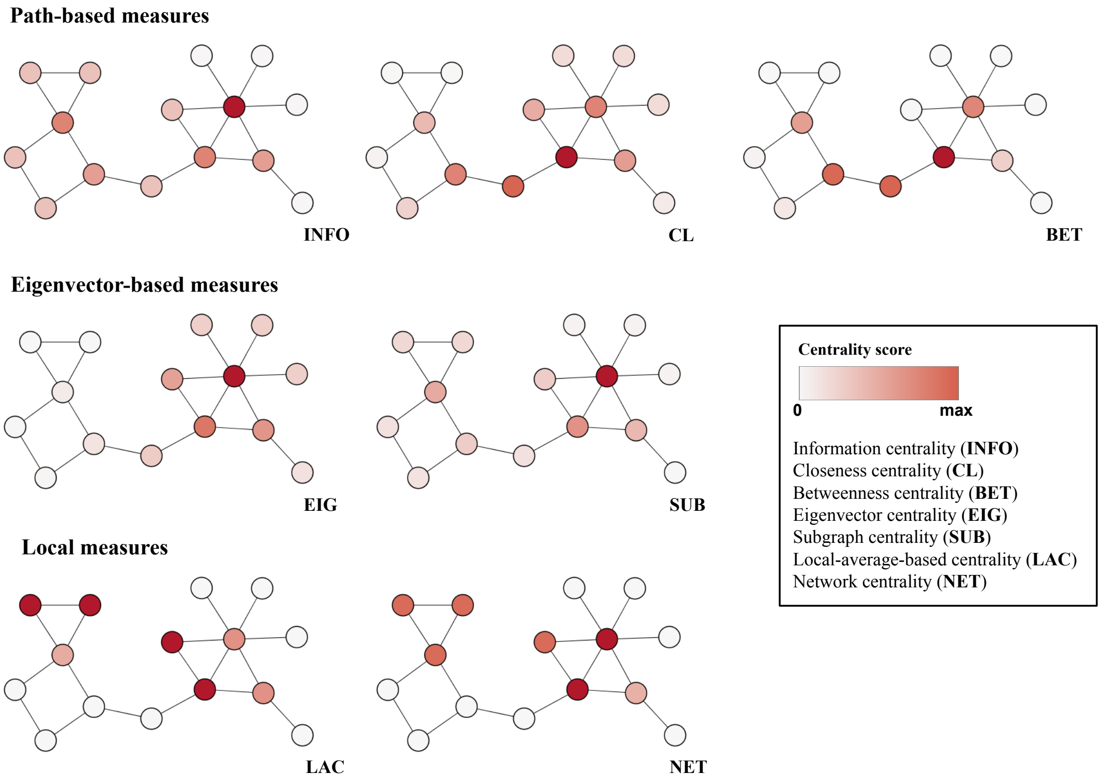

Path-based measures
INFO
CL
BET
Eigenvector-based measures
EIG
SUB
Local measures
LAC
NET
Centrality score
0
max
Information centrality (INFO)
Closeness centrality (CL)
Betweenness centrality (BET)
Eigenvector centrality (EIG)
Subgraph centrality (SUB)
Local-average-based centrality (LAC)
Network centrality (NET)

Fig. 3.3. Centrality measures commonly used to analyze biological networks: path-based measures, eigenvector-based measures and local measures.

Centrality measure	Description
Information centrality (INFO)	$INFO(u) = \left[\frac{1}{n}\sum_{v}\frac{1}{I_{u,v}}\right]^{-1}$ n is the number of nodes in the graph and $I_{u,v} = (c_{u,u} + c_{v,v} - c_{u,v})^{-1}$. Let D be the diagonal matrix containing the degree of each node and J with all elements equal to 1, $C = (c_{u,v}) = [D - A + J]^{-1}$. In addition, $I_{u,u} = \infty$, thus $\frac{1}{I_{u,u}} = 0$.
Closeness centrality (CL)	$CL(u) = \frac{n-1}{\sum\limits_{v \in V}\delta(u,v)}$ n is the number of nodes in the graph.
Betweenness centrality (BET)	$BET(u) = \sum\limits_{s \neq u \neq t}\frac{\rho(s,u,t)}{\rho(s,t)}$ $\rho(s,t)$ is the total number of shortest paths from s to t and $\rho(s,u,t)$ the number of those paths that pass through u.
Eigenvector centrality (EIG)	$EIG(u) = \alpha_{max}(u)$ α_{max} denotes the eigenvector corresponding to the largest eigenvalue of A, $\alpha_{max}(u)$ is the uth component of α_{max}.
Subgraph centrality (SUB)	$SUB(u) = \sum_{l=0}^{\infty}\frac{\mu_l(u)}{l!}$ $\mu_l(u)$ denotes the number of closed loops of length l which starts and ends at node u.
Local average connectivity-based centrality (LAC)	$LAC(u) = \frac{\sum\limits_{\omega \in N_u} deg_{C_u}(\omega)}{\lvert N_u \rvert}$ Let N_u be the set of neighbors of a node u, the subgraph induced by N_u is named C_u. The local connectivity of a node ω in C_u, $deg_{C_u}(\omega)$, is defined as the number of nodes in C_u it connects directly. The local average connectivity of a node u is defined as the average local connectivity of its neighbors.
Network centrality (NET)	$NET(u) = \sum\limits_{v \in N_u}\frac{\lvert N_u \cap N_v \rvert}{min(\lvert N_u \rvert - 1, \lvert N_v \rvert - 1)}$ N_u is the set containing all the neighbors of the node u.

Table 3.1. Summary of the centrality measure formulas adopted in this thesis.

4

Mathematical and Computational Modeling of Biological Networks

Chapter 3 introduced the graph as the basic formalism to represent biological networks. Graphs provide an effective toolbox of techniques useful to analyze the structure of the biological networks but also basic concepts to construct new methods. However, graphs by themselves are static representations and they don't manage quantitative aspects (e.g., stoichiometry) and dynamic simulation that are important to retrieve emergent behaviors of the biological networks and, in particular, for biochemical reaction networks. On the converse, the formalism of Petri nets has gained popularity over the years because of its effective mathematical properties for the structural and dynamic analysis of biochemical reaction networks.

After a short introduction, in Section 4.1, of the Systems Biology context and the formalisms for the modeling of biological networks, Section 4.2 introduces some of the commonly used mathematical models for the analysis of biological networks. Section 4.3 describes some of the computational models used in Systems Biology and, in particular, the formalism of Petri nets, that was adopted in the methodology proposed in Chapter 8 of this thesis for its interesting properties and extensions useful for the analysis of metabolic networks.

4.1 Systems Biology

Systems Biology is the discipline that aims to model and analyze complex biological systems through the use of a holistic approach. In fact, one of the most important goals of the Systems Biology is the discovery of emergent properties of the biological networks that are seen as systems of interacting components. For these reasons, modeling in Systems Biology very often refers to the dynamic simulation of the biological systems. Over the years, several modeling techniques have been proposed in Systems Biology that can be divided into two main categories: *mathematical models* and *computational models*.

In the following sections, mathematical and computational models will be described. This thesis is based on computational models and, in particular, on Petri nets that have been used to simulate metabolic networks.

4.2 Mathematical models

Mathematical formalisms are used to model and analyze the behavior of a biochemical reaction network such as its regulatory mechanisms and responses to stimuli. Mathematical formalisms for the modeling of biochemical reaction networks are usually divided into *stoichiometric modeling* and *kinetic modeling*.

Stoichiometric models try to retrieve useful properties about the fluxes of the network imposing the assumption that the metabolic system is at the steady-state, i.e. there is a balance between the rate of input and output biochemical reactions. Examples of stoichiometric models are Flux Balance Analysis (FBA), Elementary Flux Modes (EFMs) analysis and Extreme Pathways (ExPas) [62–64]. FBA tries to optimize specific objective functions to retrieve quantitative predictions about the metabolic network, while EFMs and ExPas use only the network stoichiometry, without imposing optimization principles, to retrieve important flux distribution over the network.

On the other side, kinetic modeling is based on ordinary differential equations (ODE) and integrate the kinetic parameters of the reactions to provide detailed quantitative description and dynamic behavior of the biochemical network. Kinetic modeling provides the most accurate description of biochemical reaction network.

4.3 Computational models

Computational models, unlike mathematical models, describe the evolution of a biological system using a sequence of states that changes based on the happening of internal and/or external events. Computational models encode this behavior through the use of *finite-state machines (FSMs)* that are *sensible* to specific conditions and that move their status from one to another. In this section will be described some of the computational models that have been used in Systems Biology to analyze the dynamic behavior of biological networks: Ruled-based Systems, Process Algebra, Boolean Networks and Petri nets [65, 66].

4.3.1 Ruled-based Systems

Ruled-based modeling are well suited to model biochemical networks in which there is a variation of the quantities among the molecular species. A simple representation of an enzymatic reaction where the enzyme E binds the substrate S to create the product P is the following:

$$E + S \leftrightarrows ES \tag{4.1}$$

$$ES \rightarrow P \tag{4.2}$$

Ruled-based Systems can be translated into models that take into account the quantities over time and those that are used only for a qualitative analysis of the system behavior.

4.3.2 Process Algebra

Biological systems can be seen as highly reactive and concurrent systems in which biological entities interact and synchronize each other. Process Algebra is well suited for the modeling of biological systems because it provides tools for the description of interactions, communications and synchronizations between independent agents. Furthermore, Process Algebra provides mathematical laws useful for the formal analysis of the system. In Systems Biology, Process Algebra is commonly used as an intermediate model which is subsequently transformed into another formalism such as ordinary differential equations.

4.3.3 Boolean Network

In Boolean Networks the biological entities can be in the status *active* or *inactive*. They move from a status to another according to specific boolean rules called *transfer functions*. A Boolean Network represents a set of elements that are linked each other through regulatory mechanisms. They can be used to analyze the evolution in time of the system but also to test the robustness/sensitivity of the system under perturbations.

A boolean rule TF is a function that maps a set of k binary values to one binary ouput as following:

$$TF : \{0,1\}^k \rightarrow \{0,1\} \tag{4.3}$$

Formally, a Boolean Network is charaterized by a set of boolean entities $\{\phi_1, \phi_2, \ldots, \phi_n\}$ and a set of boolean rules $\{TF_1, TF_2, \ldots, TF_m\}$. The values of the variables ϕ_i change based on the boolean rules TF_i that link them.

Thanks to their simplicity and effectiveness, Boolean Networks have been extensively applied in Systems Biology for the modeling of signaling networks and regulatory networks. This thesis proposes, in Chapter 8, a methodology that takes inspiration from a previous work that showed how to model Boolean Networks through the use of Electronic Design Automation (EDA) techniques [6]. In that work, the authors showed the results obtained from the study of the integrin activation signalling network. This thesis extends that methodology towards the study of metabolic networks dynamics taking into account the flux of metabolites in a quantitative way through the use of Petri nets.

The following sections introduce the formalism of Petri nets describing their properties and extensions that motivate the choice of PNs for the study of metabolic networks. However, the most relevant part is characterized by the PN extensions and, in particular, stochastic Petri nets because this thesis proposes a methodology based on them (see Chapter 8).

4.3.4 Petri nets

Petri nets (PNs) is a formalism used for the modeling of systems that are characterized as being concurrent, asynchronous, distributed, parallel, non-deterministic and stochastic. PN had a wide application in Computer Science because they have an intuitive graphical representation and well known mathematical properties. In the field of Systems Biology, the interest in PNs has grown because they have several useful characteristics for the structural and dynamic analysis of biochemical reaction networks (e.g., metabolic networks).

A PN is a directed bipartite graph in which the nodes are of two types: *places* and *transitions*. The places represent the actors of the system while the transitions are the events that may occur and can change the state of the places. The state of each place is characterized by a certain number of *tokens* that flow over the network based on the *firing* of the transitions. The directed arcs that link places and transitions determine which places are pre- and/or post-conditions for each transition.

Formally, a PN is a tuple $N = (P, T, W, M_0)$ where:

- $P = \{p_1, \ldots, p_n\}$ is the set of places;
- $T = \{t_1, \ldots, t_m\}$ is the set of transitions;
- $W : ((P \times T) \cup (T \times P)) \rightarrow \mathbb{N}$ is the weight function (or *multiplicity*);
  $W(x, y) = k$, with $k > 0$, represents the weight of an arc from x to y;

- M_0 is a vector of n non-negative integers representing the *initial marking* of the PN.

PNs have a simple graphical representation in which the places are represented as circles and the transitions are represented as squares. Circles and squares are linked through directed arcs that denote the corresponding flow directions of tokens along the places. Each place p is labeled with the number of tokens that it has. Each arc (p, t) or (t, p) is labeled with a natural number that is the weight or multiplicity of the arc. Usually, when the weight of an arc is 1 the label is omitted. A transition t has a *pre-condition*, denoted as t^-, and a *post-condition*, denoted as t^+. Pre- and post-conditions are n-dimensional vectors of non-negative numbers representing, respectively, the number of tokens needed to enable the transition and the number of tokens produced after the firing of the transition.

The total number of tokens in the network is denoted as *marking* and it is represented as an n-dimensional vector of non-negative numbers. A transition t with pre-condition t^- is enabled by the marking M if $t^- \leq M$. If the condition is true the transition t can fire and the result is a new marking M' as follows:

$$M' = M - t^- + t^+ \tag{4.4}$$

A given marking can enable several transitions and more than one transition can compete for a token, generating a *conflict*. The set of all the markings that it is possible to reach by a firing sequence starting from M_0 is called *reachability set* and it is denoted as $R(N, M_0)$.

Some of the properties of a PN are defined in terms of its *incidence matrix* A_N, that is a $n \times m$ matrix that has a row for each place and a column for each transition.

Figure 4.1 shows a simple Petri net model with the places $P = \{p_1, p_2, \ldots, p_7\}$ and the transitions $T = \{t_1, t_2, \ldots, t_6\}$. The weights of the arcs are all 1 and the initial marking is $M = \{1, 0, 0, 0, 1, 0, 0\}$. The transitions t_1 and t_2 are in conflict because must compete for the only one token that the place p_1 have.

4.3.5 Structural properties of Petri nets

In general, the structural analysis of a PN reveals those properties that depend only on its topology and not on the initial marking. Structural analysis of PNs is useful to determine the so-called *transition invariants* (T-invariants) and *place invariants* (P-invariants).

A T-invariant represents a possible loop in the PN, that is a sequence of transitions that bring the PN back to the marking it starts in. Formally, a T-invariant is an m-dimensional vector $X = (x_1, \ldots, x_m)^T$, for $i \in \mathbb{N}$, representing, for each transition, the number of firings required to reach again a previous marking M. The T-invariants of a PN are obtained solving the following equation:

$$A_N \cdot X = 0 \tag{4.5}$$

$X \neq 0$ is a T-invariant of the PN. In metabolic networks, the presence of T-invariants is an indicator of an important metabolic status (i.e., steady state) in which the metabolite concentrations remain stationary over the time.

A P-invariant underlines a sort of *conservation* in the number of tokens of the places during the evolution of the PN. Formally, a P-invariant is an n-dimensional vector $Y = (y_1, \ldots, y_n)^T$, with $y_i \in \mathbb{N}$ that can be obtained solving the following equation:

$$A_N^T \cdot Y = 0 \tag{4.6}$$

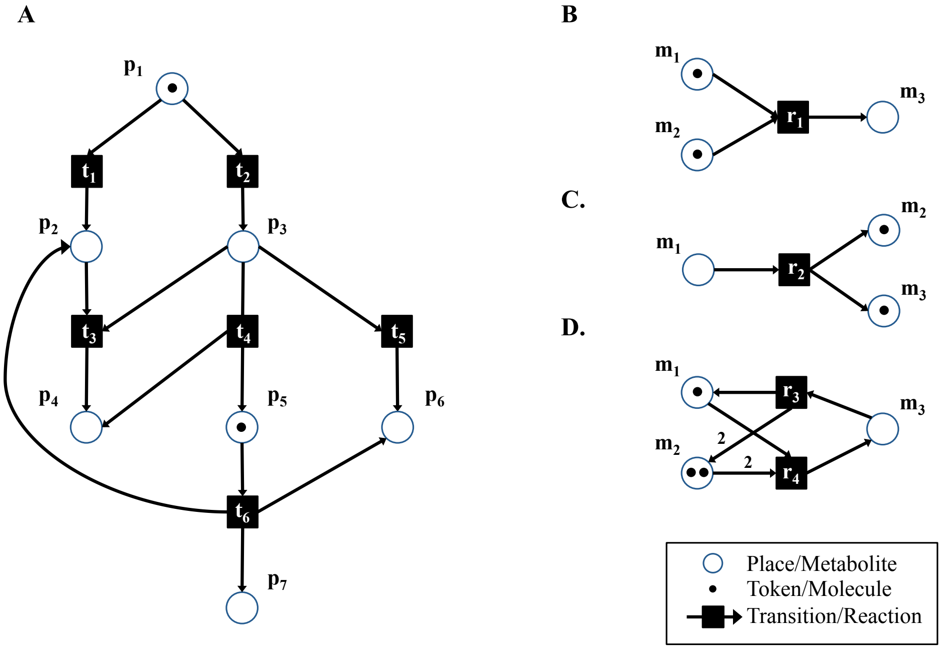


Fig. 4.1. Petri net formalism. (A) A simple example of Petri net model with seven places $P = \{p_1, p_2, \ldots, p_7\}$ and six transitions $T = \{t_1, t_2, \ldots, t_6\}$. This Petri net contains two tokens, the complete marking is $M = \{1, 0, 0, 0, 1, 1, 0, 0\}$. Examples of basic metabolic reaction of (B) synthesis, (C) decomposition and (D) reversible reaction modeled as Petri nets.

In a PN model of a metabolic network, P-invariant can be interpreted as the principle of mass conservation in chemistry.

4.3.6 Behavioural properties of Petri nets

Given a Petri net N with n places and m transitions, the properties that depend on the evolution of the network from an initial marking M_0 are called *behavioural properties* [67]. The three important behavioural properties of the PNs are: *reachability*, *boundedness* and *liveness*.

Given a marking M the problem to identify if M is reachable from an initial marking, i.e. $M \in R(N, M_0)$, is called *reachability problem*. In the analysis of metabolic networks, the solution of this problem is very important because, for example, it can underline the formation of a set of specific metabolites from another set of reactant metabolites [68]. However, the reachability problem is difficult to solve because it involves the traversing of the *reachability graph* that can lead to the so-called *state space explosion*.

A Petri net N is *bounded* if the number of tokens of each place never exceed a fixed number k during its evolution. Formally, N is called *k-bounded*, for $k \in \mathbb{N}$, if for each $M = (m_1, \ldots, m_n) \in R(N, M_0)$ and for each $i \in \{1, \ldots, n\}, m_i \leq k$. Boundedness is an important property of metabolic networks because it guarantees that there is no accumulation of specific metabolites in the network.

In a Petri net N, the liveness refers to the fact that it is always possible to make a transition starting from any reachable marking. Formally, for each $M \in R(N, M_0)$ and $t \in T$ there exists a marking $M' \in R(N, M)$ that enables t. When from a marking it is not possible to enable a transition, the PN is in a *deadlock* condition. In a PN model of a metabolic network, the liveness garantees that it is always possible to perform a biochemical reaction and it is not possible for the system to remain blocked.

4.3.7 Extensions of Petri nets

PNs is a powerful formalism to model biological networks but the basic formulation of PNs has different limitations. Several extensions have been proposed to increase the expressiveness of PNs, for example, other types of arcs and tokens. However, the main limitation is that basic PNs don't manage quantitative concepts such as *timing* and *randomness* that are important in the dynamic simulation of the biochemical reaction networks.

Some of the PN extensions that have been proposed to address these problems are described in the following.

- The concept of *test arcs* and *inhibitor arcs* was introduced, respectively, to verify the presence of tokens in a place without consuming them and to model the inhibitory action of a compound towards one or more transitions [69]. In metabolic networks, test arcs mimic the role of enzymes that are required but not consumed while inhibitor arcs represent the inhibition of a metabolite towards a metabolic reaction. This thesis proposes a methodology based on Petri nets in which this type of arcs are used to model the inhibition of specific reactions in metabolic networks.
- *Functional Petri nets (FPNs)* allow the simbolic definition of the weights in the way that they are function of the number of the places. This feature, in metabolic networks, models the fact that the concentration of a metabolite may influence the kinetic of the reaction [70].
- *Coloured Petri nets (CPNs)* assign a colour to each token and they allow to define in the arcs conditions about the colour assigned. This feature allows to describe different execution flows in the same PN. In the analysis of metabolic networks it is useful to differentiate the molecules of the same species based on how they partecipate in different reactions [71,72].
- In *Time Petri nets (TPNs)*, at each transition it is associated a time interval $[a_t, b_t]$. A transition can fire after a_t time units but not after b_t time units. The firing of a transition is instantaneous. TPNs have been used to analyze temporal aspects of biological networks [73,74].
- In *Continuous Petri nets (KPNs)*, the number of tokens of the places is not an integer number but it is a real number (called *marks*). Marks are used to model molecular concentrations and the transitions have an associated firing rate that expresses their speed [75,76]. In *Hybrid Petri nets (HPNs)*, the places and the transitions can be both discrete and continuous [77].
- *Stochastic Petri nets (SPNs)*. Similarly to the classical Petri nets, in SPNs the number of tokens is discrete. Contrary to the time-free PNs, a firing rate is associated to each transition of the net defined by probability distributions. In the case of biochemical reactions exponential distributions defined by a parameter λ are used for the timing of the transitions. Formally, a stochastic Petri net is a five-tuple $SPN = \{P, T, f, M_0, \Lambda\}$ where:
  - P is the set of the places, T the set of transitions.
  - $f : ((P \times T) \cup (T \times P)) \to N_0$ determines the set of relations between places and transitions (and between transitions and places)
  - M_0 is the initial marking of the net.
  - Λ is the set of exponentially distributed firing rates λ_k associated with the transitions.

The firing rates are defined through *hazard functions* $h_t = \lambda_t(m)$ that take in input the number of tokens needed (m) to enable a transition k and other parameters and they give the probability that the transition will occur in the next infinitesimal time interval. In the case of biochemical reaction networks *mass-action* and *Michaelis-Menten* propensity (hazard) functions are used. The activation of a transition is the same as for classic networks i.e. when all pre-places are sufficiently marked. However, there is a waiting time (or delay) τ that has to elapse before the transition can fire. The waiting time of a transition is an exponentially distributed random variable X_t with the following *probability density function*:

$$f_{X_t}(\tau) = \lambda_t(m) \cdot e^{(-\lambda_t(m) \cdot \tau)}, \tau \geq 0 \tag{4.7}$$

It can be proved that the semantics of a stochastic Petri net with exponentially distributed firing delays is described by a continuous time Markov chain (CTMC). This fact has important implications because the analysis of SPNs can be done by using the underlying Markov process with the instruments of the Markov chain theory.

Generally, the dynamic simulation of a SPN is performed through the Gillespie's algorithm, that is a well-established method to simulate biochemical reactions [78, 79].

The simulation methodology described in Chapter 8 of this thesis is based on SPNs and it uses techniques from the Electronic Design Automation (EDA) field well established for the modeling of electronic systems. In particular, the methodology is based on SystemC language that is very effective to model the concurrency of reactive and delayed processes. The aim of this work was the design of a general method that took into account the inherently randomness of the biochemical reactions and that it would allow to exploit the features of the SystemC language. SPN was chosen because it seemed to be the most suitable since it provides well-known algorithms (i.e., Gillespie methods) for the stochastic simulation using the concept of delays and discrete states.

5

Techniques for the Structural and Dynamic Analysis of Biological Networks

Chapters 3 and 4 introduced graphs and other common mathematical and computational formalisms used to model and analyze biological networks including Petri nets. Graphs can be considered the basic formalism to represent the biological networks. They provide useful techniques to analyze the structural properties of the biological networks. On the other side, Petri nets are more appropriate to study biochemical reaction networks, in which it is important to take into account the quantitative aspects of the system.

This chapter gives an overall summary of the analysis techniques, structural and dynamic, that have been applied in the literature for the analysis of the biological networks emphasizing how this thesis is placed in this context. After the introduction in Section 5.1, Sections 5.2 to 5.4 describe the methods for the structural analysis, dynamic analysis of structural properties and dynamic simulation of biological networks.

5.1 Analysis of biological networks

In Chapter 2, it has been emphasized that the advent of high-throughput technologies has allowed the construction of different types of biological networks. Over the years, it has been shown that biological networks are not random but they have a specific topology which is indicative of their internal organization. Thus, the structural analysis of biological networks is useful because provides novel information such as the motivation at the basis of certain topological constraints and the identification of nodes that control the information flow over the network.

In general, biological networks exhibit a complex dynamic behavior that enables the cells to react to various conditions and stimuli. However, the accurate study of the dynamics of a biological system has to deal with various problems that largely depends on the lack of molecular information and the computational cost to analyze it at high level of detail. The applicability of one computational technique rather than another is mostly related to the type of biological network under study, the property you want to observe, the information you have about the network and the level of accuracy according to which you want to analyze. Generally, the techniques used to analyze biological networks can be divided into those that perform, (i) *structural analysis*, (ii) *dynamic analysis of structural properties* and (iii) *dynamic simulation* of the system (Figure 5.1).

Structural analysis is, generally, considered a *static technique* because it uses only the information about the network structure. However, structural analysis can be used to retrieve both *qualitative* and *quantitative* information of the biological network. Qualitative structural analysis is able to elucidate topological and behavioural properties of the biological network while quantitative structural analysis uses stoichiometric methods to shed the light on quantitative aspects of the system.

However, with the generation of specialized interaction datasets and gene expression profiles in different conditions, it has become possible to perform a dynamic analysis of structural properties. This approach is useful because it allows to study the biological network in a more accurate way with respect to static techniques. In fact, in a biological network not all the elements must be active in the same time and this information can be obtained from expression data. Therefore, the dynamic analysis of structural properties uses the data integration to have more snapshots of the biological network to which apply the static techniques.

On the other side, dynamic simulation exploits the network structure but it needs additional information (e.g., molecular concentrations, kinetic parameters) that are necessary to describe in more detail the behavior in time of the biological networks. Dynamic simulation, usually, provides a more accurate description of biological and molecular processes and, often, also a quantitative overview of it. In addition, dynamic analysis can allow to discover emergent dynamic behaviors that a structural analysis cannot derive.

The next paragraphs show how structural and dynamic analysis have been applied to study the mechanisms of different types of biological networks.

5.2 Structural analysis

Structural analysis refers to all those methods used to study the properties of the biological networks analyzing its topology or stoichiometry. These methods can be divided into *qualitative* and *quantitative*. Qualitative analysis involves a wide use of techniques derived from graph theory but it also expoits Petri nets theory to retrieve topological or behavioural properties of the nodes of the whole network. Quantitative analysis uses stoichiometric methods in order to make quantitative predictions and they are, typically, applied to biochemical reaction networks. This thesis proposes techniques for the qualitative structural analysis. However, techniques for the quantitative structural analysis will be described briefly in Section 5.2.2.

5.2.1 Qualitative structural analysis

Qualitative structural analysis makes extensive use of methods derived from graph theory in order to analyze the topology of a biological network. Common strategies applied to study the topology of a biological network are the analysis of the importance of the nodes through centrality measures, the study of the average distance (AD) of the network, the analysis of network compactness and modularity.

In Chapter 3 graph centrality was introduced as a technique to rank the importance of a node of a graph. Centrality measures were extensively used to assess the *essentiality* of the nodes of the biological networks. With degree centrality, for example, the importance of a node is inferred from the number of its edges. It was shown that high connected nodes, i.e. *hubs*, have important properties. For example, several studies showed that hubs in yeast are more essential compared to other proteins that have a low number of connections and they seem to be evolutionarily conserved [80,81]. Closeness centrality measures how an information can spread

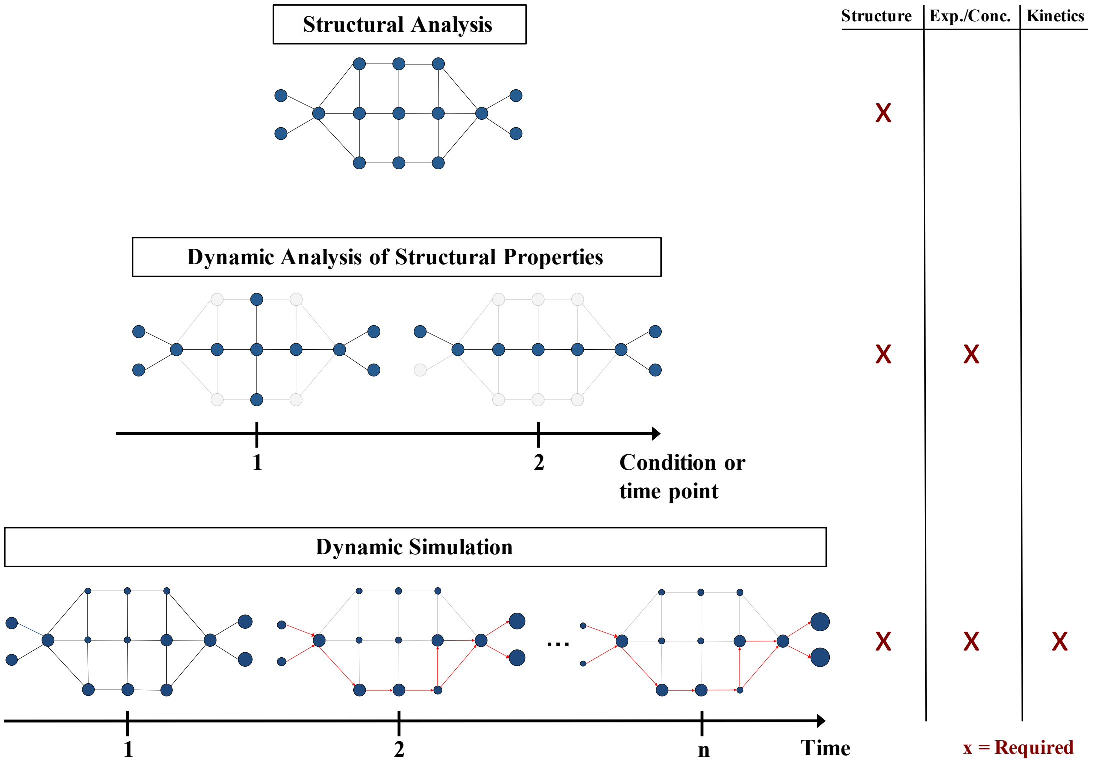


Fig. 5.1. Structural and dynamic analysis of biological networks. Structural analysis (on the top of the figure) is based only on the complete topology of the network and it aims to discover properties such as the structural organization and the importance of the nodes of the network regardless which elements are active/inactive in a certain moment. Dynamic analysis of structural properties (on the middle of the figure) takes advantage of expression data in order to have an overview on how the topology of the network changes in different conditions or time points. Dynamic simulation (on the bottom of the figure) aims to describe how the biological network evolves in time. Usually, this analysis shows how the concentration (size of the nodes) of each element of the network changes in a time window. This change can be due to the activation of specific fluxes of biochemical reactions (red arrows). Dynamic simulation of biochemical reaction networks usually needs the knowledge of the topology of the network, the concentration of substances and the kinetic parameters of the biochemical reactions.

quickly from a node to the others, thus it was useful to study the node of signaling networks and gene regulatory networks [82]. In addition, this measure has been applied to identify central metabolites and to study the core-structure in genome-based large-scale metabolic networks [83, 84]. Betweenness centrality ranks the nodes based on the shortest paths that pass through that node. This measure allowed to understand important properties in PPI networks such as the identification of nodes that have an intermediary role between proteins and nodes that are connectors (i.e., *bottlenecks*) between different processes that are essential [85, 86]. Centrality measures cover an important role in this thesis because they have been widely used both for the static and dynamic analysis of structural properties of the biological networks.

The study of the degree distribution of a biological network can give useful insights about the topology and the importance of its nodes. A particular type of networks are those that have a *scale-free* topology. In a scale-free network, the degree distribution follows a power law, i.e. the probability $P(k)$ of a node in the network to have k edges is $P(k) \sim k^{-\gamma}$, where the degree exponent γ usually is in the range $2 < \gamma < 3$. Generally, biological networks are considered scale-free. For example, several studies showed that the yeast protein interaction network, metabolic networks and prokaryotic and eukaryotic transcriptional networks have a scale-free topology [37,87]. In a scale-free network, there is a small number of hubs. In general, scale-free networks are robust against the random loss of a node but the removal of hubs can compromise their functionality [88]. Biologically speaking, robustness refers to the fact that a biological network can find alternative ways to carry out its function in the presence of a failure, for example in a node. In metabolic networks, robustness to the loss of a node indicates the presence of alternative pathways that bypass the missing reaction while in GRNs indicates the presence of alternative ways to control the information [89].

Other measure to analyze connectivity of biological network are *assortativity* and *dyadicity*. Assortativity measures the correlation between a node and its neighbors. In the PPI network of yeast, for example, it was discovered that hubs are well connected with non hubs, therefore they tend to be well separated [90]. Dyadicity measures how the nodes of the network are connected to nodes with similar functionalities. An example of application of dyadicity measure was to characterize genetic interaction network in the context of epistasis networks [91].

The compactness of a network can give useful insights on its structure. An indicator of compactness of a network is the *diameter*, that is the longest of all the calculated shortest paths in the network. It was shown that biological networks have large diameters and that this property could be associated with their modular structure. *Modularity* represents a very important property of the biological networks because was linked to robustness against perturbations of the network and adaptability to the environment. Further, modularity in biological networks was also related to the reuse of network motifs (Figure 5.2) in order to solve common evolutionary problems [92]. Motifs can give very useful information about the structure and functioning of the biological networks; for example motifs can reinforce specific activities, reduce noise and confer stability to the system [93]. Subgraph isomorphism (SubGI) algorithms are applied in motifs discovery, for example, for the prediction of the biological activity of a molecule and for searching protein complexes in protein-protein interaction networks [2, 94]. This thesis proposes SubGI strategies, based on centrality measures, useful to speed-up the process of searching network motifs into complex biological networks.

The above techniques can be used to analyze all biological networks in which a graph structure can be defined. However, for biochemical reaction networks in which stoichiometric constraints are present, other qualitative structural properties can be identified. In Chapter 4, the formalism of Petri nets was introduced to model biochemical networks due to their interesting mathematical properties. Structural analysis of a Petri net model of a biochemical reaction network can be used to qualitatively analyze it in order to prove its correctness and consistency. P-invariants and T-invariants were used, for example, to validate important biological processes such as apoptosis and to understand the processes at the basis of the iron homeostasis [95, 96]. However, Petri nets in this thesis have been exploited for their interesting features useful for the dynamic simulation of biochemical reaction networks.

5.2.2 Quantitative structural analysis

Quantitative structural analysis refers to those techniques that use network topology and additional information such as stoichiometry to make quantitative predictions. A typical technique for the quantitative structural analysis is *Flux Balance Analysis (FBA)*. The aim of FBA is the

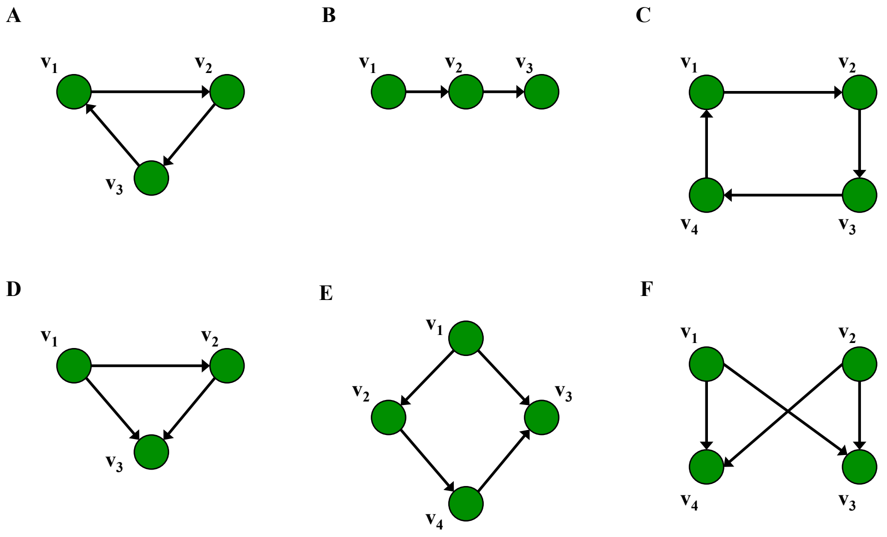


Fig. 5.2. Network motifs. The following are motifs commonly found in biological networks. (A) *Feed-forward loop*. Type of networks: protein, neuron, electronic. (B) *Three chain*. Type of network: food webs. (C) *Four node feedback*. Type of network: gene regulatory, electronic. (D) *Three node feedback*. Type of network: gene regulatory, electronic. (E) *Bi-parallel*. Type of network: gene regulatory, biochemical. (F) *Bi-Fan*. Type of networks: protein, neuron, electronic (source: Pavlopoulos et al., 2011 [1]).

analysis of flux distribution in a metabolic network using the assumption that the system is at the steady-state condition. FBA uses contraints to encode the stoichiometry of the network of reactions and other information such as the maximum speed of the enzymes to obtain optimal fluxes. The optimality of a flux is based on the chosen objective function and it is typically related to the maximization of the production of specific compounds such as ATP. Some successful applications of FBA have been the study of metabolic capacity of the *E. coli* system under knockouts and the study of robustness in the presence of changes of biomass composition in *Arabidopsis thaliana* [97, 98].

FBA searches for optimal flux distribution optimizing specific objective functions. Hower, it could be important to discovery other flux distributions that are not optimal but that can support the metabolism in the adaptation to certain environmental conditions [89]. Two related constraint-based modeling techniques are *Elementary flux modes (EFMs)* and *Extreme Pathways (ExPas)*. Both Elementary Modes and Extreme Pathways are non-decomposable pathways able to maintain the metabolic network at steady state. The analysis of EFMs and ExPas is a very effective method because, unlike qualitative structural techniques, it allows to simultaneously obtain information about key aspects of a biological network such as functionality, robustness and gene regulation, knowing only the structure of the network [99].

5.3 Dynamic analysis of structural properties

In general, structural analysis allows to perform a *static* analysis because the network is given in a specific condition or time point. However, a biological network represents a system that is inherently dynamic because its state depends on time and other variables. For example, the connections within a PPI network could be not necessarily always all active but the activation of one protein complex could depend on a particular signal received from a cell. As a result, a network property that can be verified in a specific condition it is not guaranteed to be valid in another condition.

The dynamic analysis of structural properties typically involves the integration between interaction datasets and expression data for the construction and analysis of the biological network. Examples of interaction datasets are those that contains predicted protein-protein interactions, predicted or validated miRNA-mRNA interactions and also comprehensive databases of annotated human non-coding RNAs interactions (e.g., *STRING* [100], *TargetScan* [39], *TarBase* [101], *Arena-Idb* [102]). Expression data (e.g., RNA-seq, microarray) are usually used to assess which biological entities can be considered expressed, i.e. *active*, and therefore if it should be maintained in the network. Expression data can be also used to build the network from scratch using, for example, correlation measures such as Pearson to determine if two biological elements are connected. Examples of use of these techniques are the creation of networks of co-expressed genes and the creation of miRNA-mRNA interaction networks correlating miRNA and gene expression data.

After the creation of the network, the techniques for the qualitative structural analysis can be applied to analyze the biological network. The most interesting thing is that the dynamic analysis of structural properties is useful to understand how the topology is affected by the physiological or pathological states of the biological network under study. Dynamic analysis of structural properties was used, for example, to assess temporal connectivity of hubs in the PPI network of yeast and to design a weighted centrality measure integrating gene expression data [103,104]. This thesis proposes a measure and a workflow to perform a dynamic analysis of miRNA-mRNA interaction networks in different conditions that are useful to study the *crosstalk* between chromosomes at the level of miRNA-mRNA regulation.

5.4 Dynamic simulation

As discussed in the previous section, dynamic analysis of structural properties takes advantage of expression data to study a biological network in a more dynamic way. However, expression data provide snapshots of single time points or conditions, therefore a complete evolution in time of the biological network is not provided. To obtain a more precise description of the behavior of the biological network it is needed to perform a dynamic simulation.

Very often dynamic simulation involves biochemical reaction networks (e.g., metabolic networks) that are systems composed by a set of interconnected biochemical reactions which rates depend on the activity of the enzymes (Figure 5.3). Enzymatic reaction rates are usually modeled using known chemical kinetics such as *mass-action* and *Michaelis-Menten* kinetics and integrated into the simulation system. The simulation methodology proposed in this thesis is based on mass-action that is the most basic law that governs the biochemical reactions. However, dynamic simulation can be performed also with biological networks in which the types of interactions can be more heterogeneous as in the case of signaling networks and gene regulatory networks. In general, based on the level of abstraction required to analyze a biological network, the methods to perform dynamic simulation can be qualitative or quantitative.

Qualitative methods usually require few or no parameters and they potentially handle large-scale systems. They model the biological network at a high level of abstraction to discover biological *attractors*. Several formalisms that have been proposed to perform a qualitative dynamic simulation such as Boolean Networks, Process Algebra and Rule-based Systems. Among

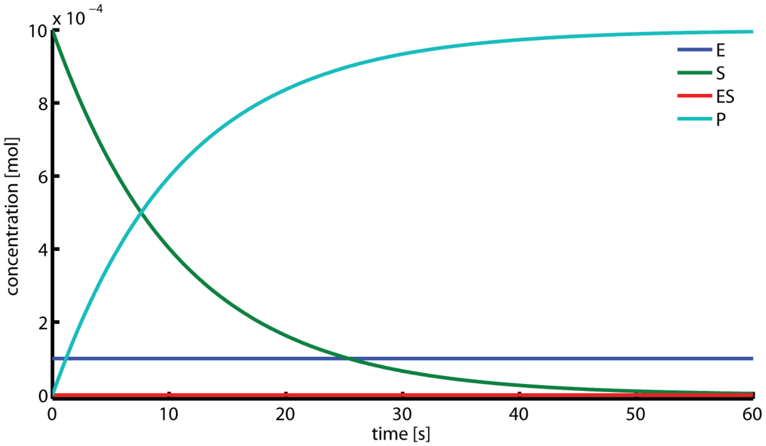


Fig. 5.3. Dynamic simulation of an enzymatic reaction. E represents the enzyme concentration, S is the substrate, ES is the complex formed by E and S that releases the product P. In this simple example, the species S is entirely consumed after 60 seconds and converted into P. The purpose of the enzymes is to catalyze the reactions and they are not consumed (Figure source: *http://cbio.bmt.tue.nl/resolve/content/modelling_basics/modelTypes.php*).

these, Boolean Networks represent an effective formalism because was largely applied for the qualitative simulation of biological networks, and in particular for gene regulatory networks and signaling networks. Boolean networks are discrete dynamical networks composed by a set of boolean variables that evolves in a discrete time. The state of each variable is described by boolean functions that take in input the state of a subset of variables in the network and return the corresponding output. This formalism was used, for example, to provide dynamical explanation of specific gene expression patterns in the gene regulatory network of *Arabidopsis thaliana* [105, 106].

On the other side, quantitative methods rely on mathematical formalisms and allow a more accurate modelling by describing the biological system with realistic concentrations and time scales [107]. A typical mathematical formalism for dynamic quantitative simulation of biological networks are ordinary differential equations (ODE). Quantitative methods can be deterministic or stochastic. Deterministic methods capture the global behaviour of the elements of the network, while stochastic models incorporate randomness to take into account possible fluctuations and noise due to a small number of interacting molecules in the environment. A problem related to quantitative methods is that they are more appropriate to analyse small and well studied systems due to the lack of kinetic parameters that are usually derived from *in vivo* or *in vitro* experiments [108, 109].

Another formalism commonly used for the simulation of biological networks are Petri nets. PNs have been widely applied in Systems Biology because, as seen before, they provide well-founded mathematical properties for the qualitative structural analysis of biological networks but they can be extended to perform also a quantitative dynamic simulation of the system. Coloured and Hybrid Functional Petri nets were used, for example, to study the mechanisms involved in the glycolytic pathway [74, 110]. Stochastic Petri nets have been successfully used to obtain new insights in the development of hepatic granuloma throughout the course of infection and to model signal transduction pathways in the process of angiogenesis [111, 111].

A recurrent issue in the dynamic simulation of biological networks is related to the concept of *parameterization*. Very often, due to the lack of quantitative information (e.g, kinetic parameters), it is necessary to explore the solution space of the parameters to identify which network configurations lead the model to satisfy certain biological properties. This process may require the tuning of unknown parameters to obtain results in agreement with biological knowledge or the reverse engineering of parameters from experimental data [112, 113]. This thesis proposes a simulation methodology based on stochastic Petri nets that tries to address the problem of parameterization of metabolic networks taking advantage of techniques well established in the field of Electronic Design Automation (EDA).

6

Search Strategies for the Subgraph Isomorphism Problem and Application to Biological Networks

The study of network motifs of biological networks can involve the application of subgraph isomorphism algorithms (SubGI) in order to search small pattern graphs into larger target graphs. Several strategies have been proposed in order to obtain better performance in the context of biological networks [114]. In general, the performance of SubGI strategies depend on the properties of the pattern and target graphs such as size, density and number of labels. However, the crucial aspect is the search strategy used by the algorithm.

A state-of-the-art algorithm for the SubGI problem is RI [3]. RI uses a very simple but effective search strategy that takes advantage of the local topological properties of the pattern graph. Two problems related to the RI algorithm are that (i) it doesn't perform an exploration of the global pattern graph structure and (ii) it doesn't exploit information about target graph before the search process.

In this chapter will be described seven search strategies based on target-aware graph centrality measures useful to make the search strategy less dependent on local pattern graph properties.

Section 6.1 introduces the concept of variable ordering that is crucial to solve the SubGI problem.

Section 6.2 describes the search strategy of the RI algorithm.

Section 6.3 describes several strategies to speed the subgraph isomorphism up.

Sections 6.4-6.5 describe, respectively, the experimental results and conclusions.

6.1 The variable ordering in subgraph isomorphism

The basic notions on graphs and SubGI problem formulation can be found in Section 3.1.

Given a pattern graph $G_p = (V_p, E_p, \alpha_p)$ and a target graph $G_t = (V_t, E_t, \alpha_t)$, where α_i is a mapping of the nodes of a graph into a set of labels. It is possible to define an ordering $\mu_p = (u_p^1, u_p^2, \ldots, u_p^{|V_p|})$ of the nodes of the pattern graph. The SubGI problem can be formulated as the combinatorial problem of finding all possible rearrangements of size $|V_p|$ of the target graph nodes $\mu_t = (u_t^1, u_t^2, \ldots, u_t^{|V_t|})$. A mapping between the nodes of the pattern and target graph can be represented as the ordered set of pairs $M = \{(u_p^1, u_t^1), (u_p^2, u_t^2), \ldots, (u_p^{|V_p|}, u_t^{|V_t|})\}$. The set of all possible mappings represents the *search space* of the problem and, usually, is represented as a tree that is visited in order to find a solution of the problem. The order in which the pattern nodes are mapped into the target graph is called *variable ordering* (Figure 6.1). In this context the term variable refers to the fact that the SubGI problem can be formulated as a constraint satisfaction problem where values (target vertices) are assigned to variables (pattern vertices) taking into account the constraints imposed by the SubGI rules [115]. The variable ordering of the pattern nodes has a strong impact in the SubGI search because it reduces the set of the possible forward assignments of the variables.

The strategy of choice of the variable ordering may depend on the properties of both pattern and target graph such as the topology and the frequency of the labels of the pattern nodes along the target graph [116]. An important concept adopted in the variable ordering is the *fail-first* principle according to which *high failure* and *most constrained* variables are queued early in the ordering.

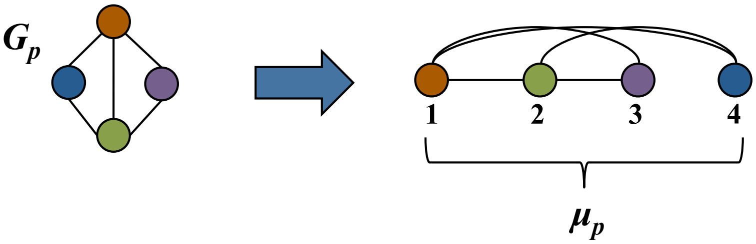


Fig. 6.1. The *variable ordering* μ_p of a pattern graph G_p determines how the nodes of the pattern are searched within the reference graph. In this case the red node is the first node to search while the blue is the last one.

6.2 The RI ordering strategy

RI uses a strategy that exploits the local structure of the pattern graph in order to maximize the number of constraints that can be verified after each partial mapping. Practically, RI builds the ordering preferring, at each step, nodes high connected with the current partial ordering (*core set*).

Formally, given a partial ordering $\mu_p = (u_p^1, u_p^2, \ldots, u_p^i)$ of the pattern nodes, the next vertex u_p^{i+1} is chosen such that it maximizes (in order) the cardinality of the following node sets:

- $V_{u_p^{i+1},1} = \{u' : (u_p^{i+1}, u') \in E$ and $u' \in \mu_p\}$
- $V_{u_p^{i+1},2} = \{u' : (u_p^{i+1}, u') \in E, u' \notin \mu_p$ and $\exists (u', u'')$ for at least one $u'' \in \mu_p\}$
- $V_{u_p^{i+1},3} = \{u' : (u_p^{i+1}, u') \in E, u' \notin V_{u_p^{i+1},1}$ and $u' \notin V_{u_p^{i+1},2}\}$

Figure 6.2 shows an example of how the sets of a node u_p^{i+1} (violet node) are calculated in order to choose the next node of the ordering. At first, the node that maximizes the connection towards nodes of μ_p (red nodes), namely the cardinality of the set $V_{u_p^{i+1},1}$, is preferred. If two or more nodes have the same number of connections, then their set of neighbors that are connected to the core set (green nodes) are taken into account ($V_{u_p^{i+1},2}$). In case of a further parity between nodes, the nodes not present in the previous node set are taken into account (blue nodes, $V_{u_p^{i+1},3}$).

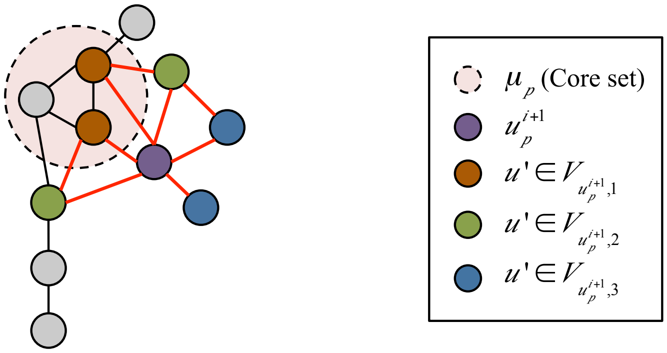


Fig. 6.2. The RI variable ordering creation. The violet node u_p^{i+1} is the node currently considered. The number of red nodes ($V_{u_p^{i+1},1}$) are evaluated at first because they are the nodes of the *core set* directly connected with the violet node. In case of parity between the violet node and others nodes, the number of green nodes are considered ($V_{u_p^{i+1},2}$). Finally the number of blues nodes ($V_{u_p^{i+1},3}$) are used for the final discrimination phase. The edges involved in the computation are highlighted in red.

6.3 Ordering strategies based on weighted centralities

The RI algorithm relies only on the topological information about the pattern graph. However, relevant information can be obtained exploiting the global structure of the pattern graph and the information about the target graph [117].

In this work, it was investigated the use of the seven centrality measures introduced in Section 3.3 that are commonly used in the analysis of biological networks. For each centrality measure was developed a modified version of the RI algorithm in which the value of the centrality of the nodes is used as a first criterion and, in case of equality, the original RI ordering strategy is applied.

Weighted centrality measures were used defining the weight of an edge of the pattern according to the frequency of the pair of the node labels in the target graph. Given an edge (v, v') of the pattern graph with labels l_1 and l_2, the frequency of the edge is computed as following:

$$freq_t(l_1, l_2) = |\{(u, u') \in E_t : \alpha_t(u) = l_1 \text{ and } \alpha_t(u') = l_2\}|/|E_t| \tag{6.1}$$

Given two edges, the idea is to give a higher score to the edges with lower frequency in the target graph. These new strategies, unlike the original formulation of RI, can take advantage of the properties of the target graph.

6.4 Experimental results

The performance of the proposed strategies was evaluated on a widely used benchmark of biological networks [114]. In particular, three types of biological networks were taken into account: protein contact maps, protein chemical structures and protein-protein interaction networks (see Section 2.3.6). They have different structural and labeling properties. Pattern graphs were extracted from the benchmark by varying the number of their edges from 8 to 256. For each edge amount, it was extracted an equal number of pattern graphs.

Figure 6.3 shows the properties of the pattern and target graphs used in the analysis. In particular, number of nodes, number of edges, graph density and number of distinct labels are

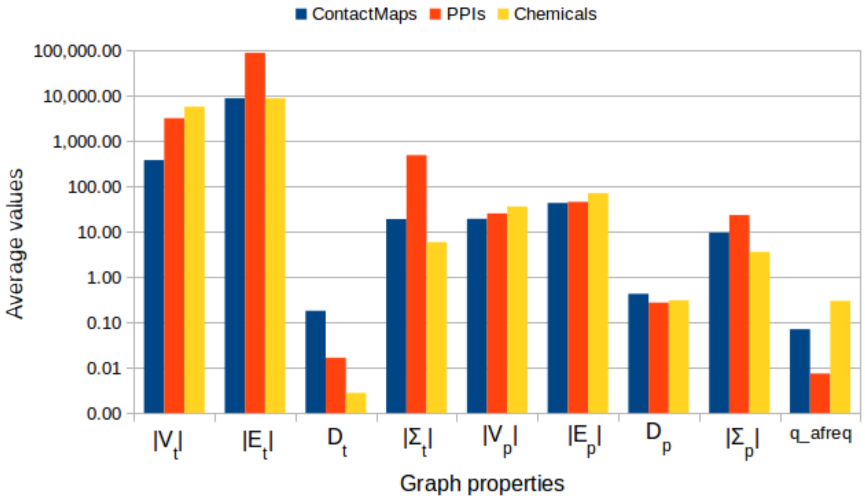


Fig. 6.3. Statistics regarding the properties of the pattern and target graphs used in the analysis: average number of nodes, edges, graph densities, labels and average frequency of the pattern graph labels in the corresponding target graphs

.

reported. In addition, it was reported the average frequency of the labels of the pattern graphs within the corresponding target graphs (*q_afreq*).

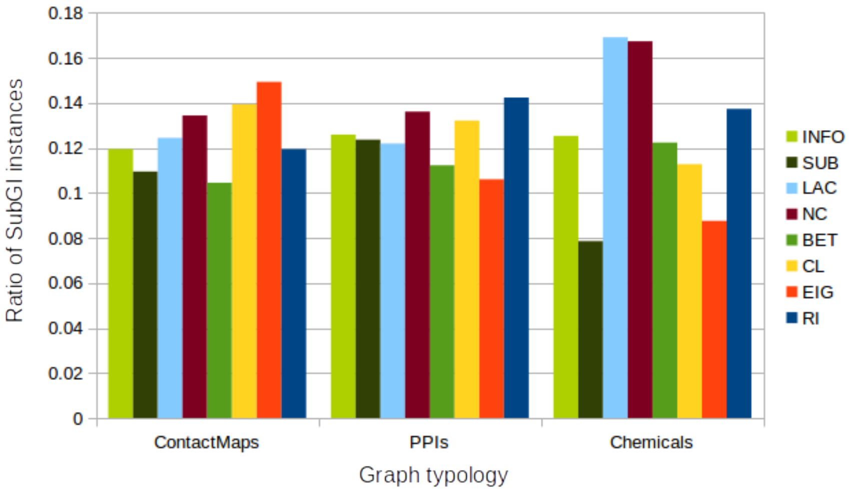


Fig. 6.4. Performance of the strategies. The figure shows the ratio of SubGI instances in which each strategy has been the fastest solution. Instances are grouped by the type of graph: contact maps, PPIs and chemical structures.

The strategies were tested searching a pattern into a corresponding target graph from which it was extracted. A timeout of 10 minutes was used to stop the execution of longer instances. The performance of the proposed strategies was evaluated counting how many times an approach was the fastest and by summarizing the execution time through clustering approaches. Figure 6.4 shows the ratio between the number for which a strategy outperformed all the others and the complete benchmark. Figure 6.5 reports the previous information with the details about the pattern graphs types. EIG is the best strategy in contact maps, followed by the closeness centrality strategy (CL). All the strategies show a stable trend on PPI and chemical graphs (Figure 6.5). In PPI networks the RI and NC ordering strategies are the most effective. In chemical graphs, local strategies such as LAC and NC are the predominant ones. From the results it emerges that local orderings are more suitable in sparse and semi-sparse graphs, while global measures are more effective for dense graphs. BET and CL strategies have opposite behavior because BET performance increases with more sparse graphs while CL tends to be better on dense graphs. The performance of the EIG strategy seems to be coorelated with the decrease of graph density while the others strategies do not seem to have a linear correlation with graph density.

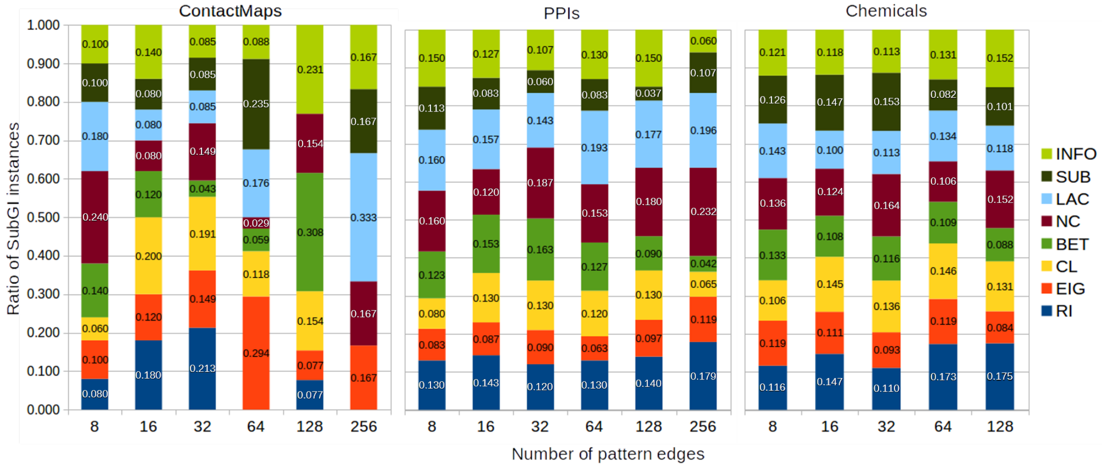


Fig. 6.5. Performance of the strategies based on the number of pattern edges. The figure shows for each methodology the ratio of SubGI instances for which it has been registered the fastest strategy (shown on top of the bars). Instances are grouped by graph typology and number of pattern edges (from 8 to 256). Patterns of 256 edges from chemical graphs were discarded because the sparsity of such graphs the extracted subgraphs tend to degenerate into simple paths.

Figure 6.6 (A) shows the performance similarities calculated through two-samples statistical tests. Strategies cluster together based on the type of centrality. Only BET strategy seems to be more similar in running time to the local orderings (Figure 6.6 (B)).

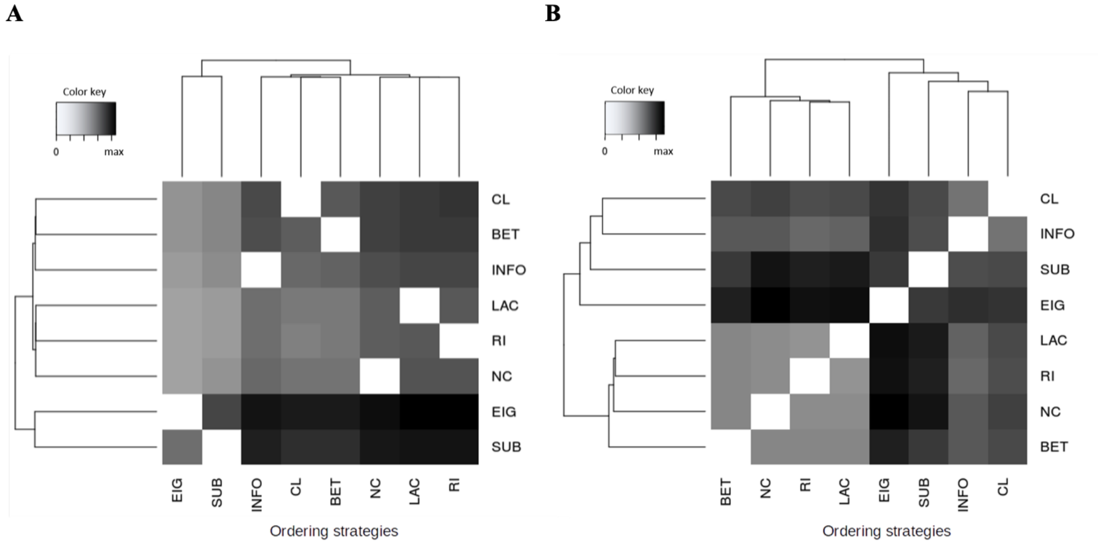


Fig. 6.6. Analysis of the running times of the strategies. (A) Heat map of the correlations between the investigated strategies applying the Mann-Whitney U test to the running times. (B) Similarities among strategies are computed by summing the absolute differences of their normalized running times. Normalization is performed by setting to 1 the slowest performance registered for each SubGI instance. U statistics and absolute time differences are used for the hierarchical clusterings of the strategies.

6.5 Closing remarks

This work showed the investigation of several alternative strategies to a well-established approach for solving the subgraph isomorphism problem. These strategies take into account the global topology of the pattern graph and the information about the frequency of the pattern graph labels within the target graph. The motivation of this study was to demonstrate that these information could be useful to address some of the limitations of the RI algorithm. In addition, this study provided insights for the design techniques that could exploit the properties of the graphs to choose the best strategy. Local strategies are better on the sparse target graphs while closeness- and eigenvector-based strategies are better on dense graphs.

The design of efficient heuristics for solving the SubGI problem is a very important task in Bioinformatics because they represent the basis for building more complete software and tools for the structural analysis of increasingly complex networks. In fact, static techniques such as SubGI algorithms are usually used within more sophisticated analysis workflows that integrate expression and interaction datasets to study dynamic topology changes of the biological networks based on condition or time. Therefore, the effectiveness of techniques for the structural analysis is also linked to the design of useful analysis workflows able to integrate them for the dynamic analysis of the structural properties.

7

Dynamic Analysis of Chromosome-wise miRNA-mRNA Connectivity

A recent work demonstrated that the chromosome-wise connectivity of the protein-protein interactome differs between human chromosomes and that this may influence the impact of chromosome abnormalities [5]. This result underlined the fact that the chromosomal organization has constraints that can be useful to study to better understand structural properties of the human interactome and which impact it can have on disease states. In the last decade, increasing evidence highlighted the role of non-coding RNAs in the regulation of crucial biological processes such as cell proliferation and cell death. Of particular interest, it was the study of microRNAs (miRNAs) because they were found implicated in many disease and, in particular, in cancer. An important aspect to understand the human interacome is the study of the miRNA-mRNA chromosome-wise connectivity, that is largely unexplored, to discover how it changes from the healthy to disease state.

This chapter introduces a new measure together with a workflow to study the chromosome-wise connectivity of miRNA-mRNA interactions in different conditions and the application to cancer studies.

Section 7.1 describes the proposed measure to analyze the connectivity of miRNA-mRNA interaction networks.

Section 7.2 describes the general workflow for the construction of miRNA-mRNA interaction networks and the application of the proposed connectivity measure to them.

Sections 7.3-7.4 show, respectively, the experimental results obtained with the proposed approach on cancer studies and the closing remarks.

7.1 Chromosome-wise connectivity

Kirk et al. [5] introduced the connectivity ratio (CR) as a measure of the protein-protein interactions ratio between intra-chromosomal, *cis*, interactions between proteins encoded on the same chromosome and inter-chromosomal, *trans*, interactions between proteins encoded on different chromosomes. Figure 7.1 (A) shows a simple example of PPI interaction network in which the protein interactions are considered at chromosomal level and divided in *cis* and *trans* in order to calculate the CR. Given a chromosome , the original formulation of the CR is the ratio between normalized *cis* and *trans* protein interactions. The normalization is performed dividing, respectively, by the number of *cis* and *trans* theoretical interactions.

In formula:

$$CR(C) = \frac{n_C/N_C}{m_C/M_C} \tag{7.1}$$

where n_C is the number of observed *cis*-chromosomal interactions of C, $N_C = \frac{\{C\}(\{C\}-1)}{2}$ is the normalization factor where $\{C\}$ represents the number of protein coding genes on chromosome C. m_C is the number of observed *trans*-chromosomal interactions of C, $M_C = \{C\} \cdot \sum_{D, D \neq C} \{D\}$ is the normalization factor where $\{D\}$ is the number of genes on any other chromosome than C. This CR measure was successfully used to analyze the human protein-protein interactions but it is not appropriate to study miRNA-mRNA interaction networks that have a bipartite graph structure.

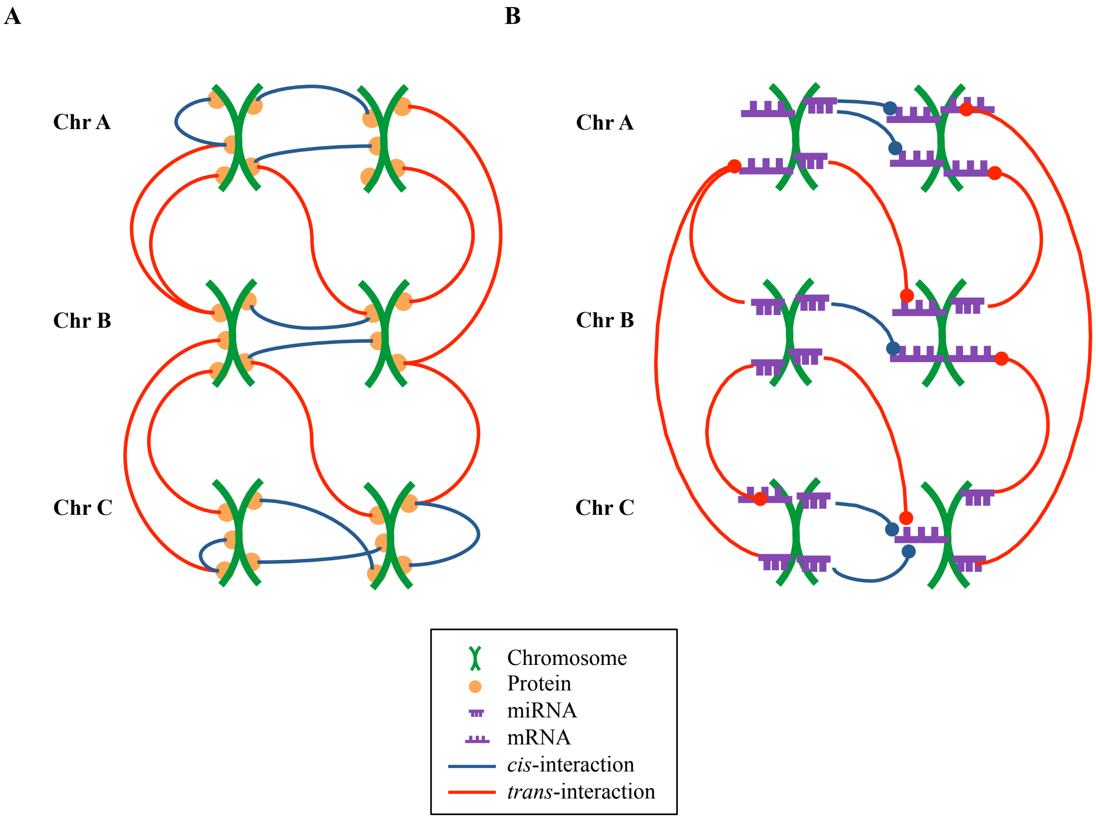


Fig. 7.1. *Cis-* and *trans-chromosomal* interactions. In Blue, *cis* interactions involving two molecules (protein-protein or miRNA-mRNA) encoded in the same chromosome. In Red, *trans* interactions involving two molecules encoded in different chromosomes. Examples of *cis-* and *trans*-chromosomal interactions in (A) protein-protein interaction network (undirected graph) and (B) miRNA-mRNA interaction network (bipartite graph).

The aim of this study was to design a new formula useful to measure the chromosome-wise CR of miRNA-mRNA interaction networks (Figure 7.1 (B)). To accomplish this task, *trans* interactions of a chromosome were divided into two groups: *trans-in*, which involves a chromosome's mRNA, and *trans-out*, which involves a chromosome's miRNA (Figure 7.2).

This distinction is important in order to correctly normalize for the theoretically number of possible *trans-in* and *trans-out* interactions, respectively.

In formula, given a chromosome C, the CR is defined as:

$$CR(C) = \frac{Cis}{Trans} \tag{7.2}$$

$$Cis = \frac{n_{Cis}}{n_{miRNA_C} \cdot n_{mRNA_C}} \tag{7.3}$$

$$Trans = \frac{n_{trans-in} + n_{trans-out}}{n_{mRNA_C} \cdot (n_{miRNA_{all}} - n_{miRNA_C}) + n_{miRNA_C} \cdot (n_{mRNA_{all}} - n_{mRNA_C})} \tag{7.4}$$

where n_{miRNA_C} and n_{mRNA_C} are, respectively, the number of miRNA and mRNA on the chromosome C; n_{miRNA_all} and n_{mRNA_all} are, respectively, the number of miRNA and mRNA on any other chromosome than C; n_{Cis} is the number of *cis*-chromosomal miRNA-mRNA interactions present on chromosome C normalized by the theoretically possible number of them; $n_{trans-in}$ and $n_{trans-out}$ are the number of *trans* interactions of type *in* and *out* normalized, respectively, by the number of theoretically possible interactions between the mRNAs of C and the miRNAs of all other chromosomes and vice versa.

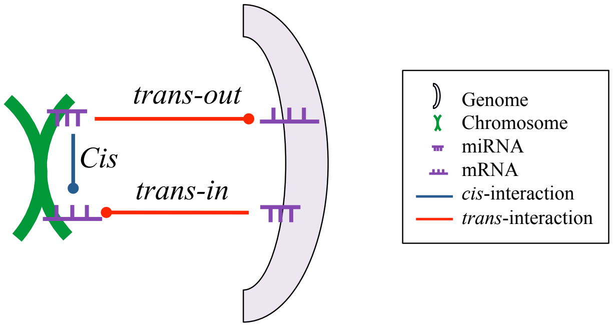


Fig. 7.2. *Cis*- and *trans*-chromosomal interactions. Given a chromosome C, in a miRNA-mRNA interaction network, *trans* interactions can be of type *in*, involving mRNAs on C and miRNA elsewhere in the genome or *out*, involving miRNAs on C and mRNAs elsewhere in the genome.

7.2 A workflow to study the chromosome-wise miRNA-mRNA connectivity

The study of miRNA-mRNA interactions involves the construction and analysis of complex regulatory networks. Such networks are built of nodes representing miRNAs and mRNAs, while edges between nodes represent their interactions. This graph representation offers insights into how miRNAs control complex biological phenomena and influence the progression and prognostication of disease [118, 119].

The methods used to analyse such networks can be divided into (i) the approaches that build regulatory networks using predicted or validated interactions and (ii) the methods that infer miRNA-mRNA interactions from expression data. Predicted interactions are obtained by applying several rules including the presence of conserved binding sites in the mRNAs and the sequence context [39]. Experimentally validated miRNA-mRNA interactions are collected into several databases such as mirTarBase and Tarbase [101, 120]. The approaches based on expression data allow the integration of miRNA and mRNA expression to infer their putative interactions assuming that miRNAs downregulate their mRNA targets [121, 122]. More specifically, negative correlations between miRNAs and mRNAs are identified by linear correlation measures.

Once a regulatory network has been built, structural analysis techniques can be used to relate the structure of the network to its function. As discussed in Section 5.2.1, a common type of structural analysis used to analyze biological networks involves characterization of the behaviour and importance of individual nodes through the use of centrality measures. Structural analysis methods for miRNA-mRNA regulatory networks have been used to detect alteration in cancer regulatory networks, discovering key miRNAs with roles as oncogenes or tumour suppressors among others [123]. The analysis of miRNA-mRNA regulatory networks is important because alterations on these networks affect the final protein abundance and consequently change the network connectivity of the protein interactome. This change could further alter the *crosstalk* between chromosomes in terms of interactions between protein-coding genes located in the same, *cis* interactions, or different chromosomes, *trans* interactions, by creating or deleting connections.

Figure 7.3 shows a workflow that integrates the proposed CR measure to analyze miRNA-mRNA connectivity at chromosomal level in order to understand how the CR dynamically changes in different conditions. The workflow was applied to cancer studies, however the approach is general and can be applied to any disease of interest in which expression data of miRNAs and mRNAs are provided.

The first step of the workflow is the data integration (Figure 7.3 (A)). Data can be obtained *in house* or from publicly available databases such as GEO and TCGA [124, 125]. miRNA and mRNA expression data are correlated using statistical methods such as Pearson in order to obtain a bipartite miRNA-mRNA interaction network. A threshold for the correlations must be chosen and, next, a p-value is calculated in order to maintain only the significant ones. To have more reliable interactions it is useful to use miRNA-mRNA prediction algorithms to filter those interactions that do not involve conserved binding sites in the mRNAs. To accomplish this task, the interactions predicted by the TargetScan algorithm are used for filtering [39]. The second step is the calculation of *cis*- and *trans*-chromosomal interactions of the above miRNA-mRNA interaction network (Figure 7.3 (B)). Finally, the last step is the actual calculation of the miRNA-mRNA CR using the formula proposed in Section 7.1 (Figure 7.3 (C)). The CR values obtained from the above workflow can be analyzed to obtain novel information on how the rewiring of chromosomal interactions changes in normal and pathological conditions and how this information can be relevant for the specific case study.

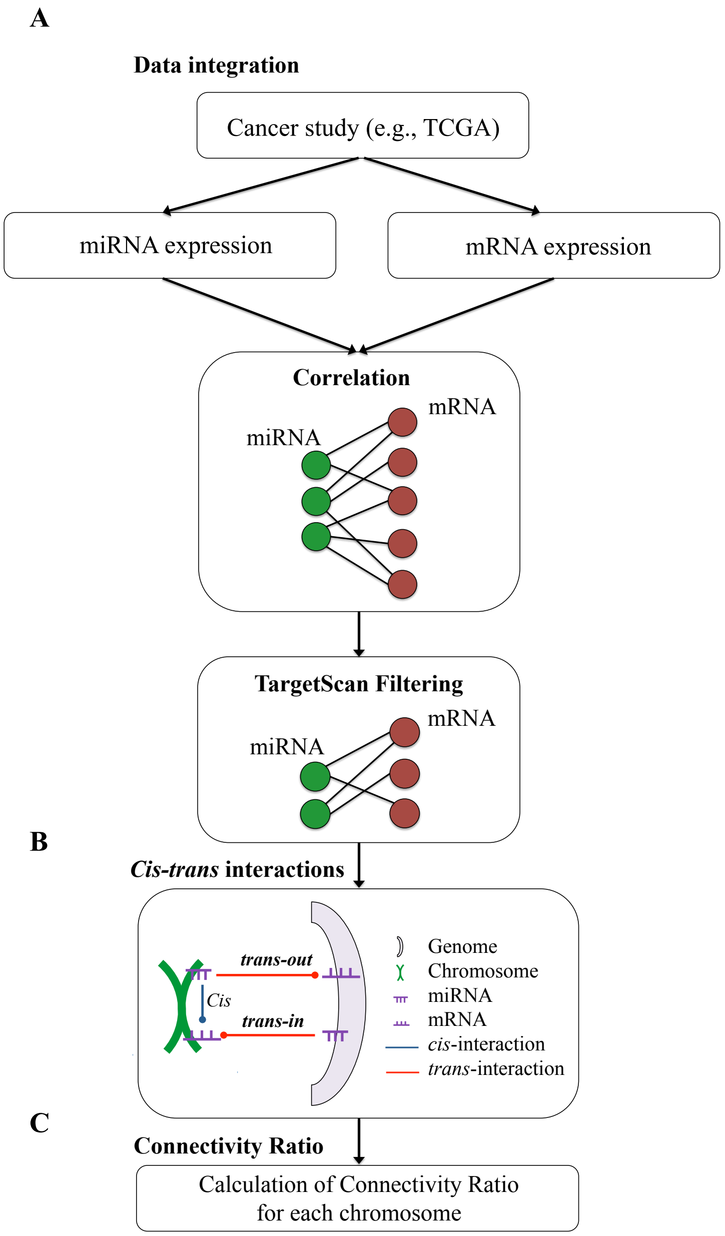


Fig. 7.3. Methodological workflow for studying the chromosome-wise connectivity of miRNA-mRNA interactions. (A) In the data integration step, miRNA and mRNA expression data from TCGA or other databases are correlated and filtered through TargetScan in order to obtain regulatory networks for all samples. (B) *Cis* and *trans* interactions are computed for each chromosome. *Cis* interactions involve miRNAs and genes of the same chromosome. *Trans* interactions can be either of type *in*, i.e., interactions between mRNAs of the chromosome of interest and miRNAs located in all other chromosomes, or *out*, i.e., interactions between miRNAs of the chromosome of interest with mRNAs of the other chromosomes. (C) The connectivity ratio (CR) is calculated for each chromosome of the network.

7.3 Experimental results

7.3.1 Chromosome-wise miRNA-mRNA connectivity in cancer

The above methodology was implemented in R language (version 3.5.0) and used to analyse how the chromosome-wise connectivity changes in cancer compared to normal tissue. To perform this study, TCGA pan-cancer study was used to retrieve miRNA and mRNA expression data [125]. TCGA data for 14 cancer studies were downloaded from FireBrowse (*http://firebrowse.org/*) in the form of reads per million (RPM) for miRNAs and RSEM estimated counts for mRNAs [126].

Before proceeding with the analysis workflow, several preprocessing steps have been performed. Firstly, only TCGA studies that had, at least 3 cancer samples and 3 healthy samples (miRNAs and mRNAs) coming from the same patient were maintained. Secondly, a filtering was performed to remove lowly expressed miRNAs and mRNAs. In particular, miRNAs and mRNAs were considered expressed if for each sample the number of counts was greater than 1. After removing low expressed miRNAs and mRNAs, data were log2 transformed. Principal Variance Component Analysis (PVCA) was performed for each study to observe the effect size of the covariates age and sex using the function pvcAnaly from the package ExpressionNormalizationWorkflow (Murugesan, 2018) [127]. A residual variance of 75% was used as threshold to perform batch correction of the samples. For batch correction the method Combat from the sva package was used [128]. After PVCA, only BRCA and LUAD cancer studies were batch corrected and only BRCA was maintained. At the end of these steps, 10 TCGA cancer studies remained for further analyses (Table 7.1).

Cancer abbreviation	Full name of cancer	No. of miRNAs in cancer/healthy	No. of mRNAs in cancer/healthy
BLCA	Bladder Urothelial Carcinoma	139/129	5,200/3,067
BRCA	Breast invasive carcinoma	132/155	5,639/5,200
CESC	Cervical squamous cell carcinoma and endocervical adenocarcinoma	142/170	4,684/2,199
HNSC	Head and Neck squamous cell carcinoma	142/167	5,213/4,371
LIHC	Liver hepatocellular carcinoma	137/180	4,456/3,610
LUSC	Lung squamous cell carcinoma	155/171	5,286/3,069
KIRC	Kidney renal clear cell carcinoma	129/159	4,930/5,419
KIRP	Kidney renal papillary cell carcinoma	128/173	5,135/4,667
STAD	Stomach adenocarcinoma	146/150	5,476/5,282
UCEC	Uterine Corpus Endometrial Carcinoma	132/165	4,757/3,732

Table 7.1. TCGA cancer studies after preprocessing. Number of expressed miRNAs and mRNAs after correlation and TargetScan filtering (Figure 7.3 (A)) for each of the 10 TCGA cancer types included in this study.

After the preprocessing, the CR workflow was executed. In particular, for each TCGA study and for each condition (cancer and healthy), a miRNA-mRNA interaction network was built by correlating expressed miRNAs and mRNAs maintaining only those interactions predicted by TargetScan v7.1 (Figure 7.3 (A) and Table 7.1). Correlation was performed using the biweight mid-correlation [129] through the function bicor from the WGCNA package and keeping negative miRNA-mRNA correlations with p-value < 0.05 [130]. P-values were calculated using the Student asymptotic method applying the function corPvalueStudent (package WGCNA).

miRNAs and mRNAs of each network were annotated with the chromosome information using ENSEMBL and miRBase v21 databases [131, 132].

Next, the CR for each miRNA-mRNA interaction network was calculated considering *cis*- and *trans*-chromosomal miRNA-mRNA interactions (Figure 7.3 (B-C) and Tables 7.2-7.3). The distribution analysis of the CRs showed that there is a strong shift towards *trans* interactions in cancer, i.e., CR decreased in most of the chromosomes (Figure 7.4 and Figures 7.5 (A-C)). In particular, chromosomes 4, 14, 16 and 20 formed a small cluster with the lowest CR and the strongest shift to *trans* interactions together with chromosome 18 that lost the full miRNA-mRNA interactome in cancer samples. The correlation, in terms of CR, between this cluster of chromosomes can be also seen from the hierarchical clustering in Figure 7.5 (B). An opposite trend was observed for the chromosome 13, which was the only chromosome having consistently more *cis* than *trans* interactions in both healthy and cancer samples.

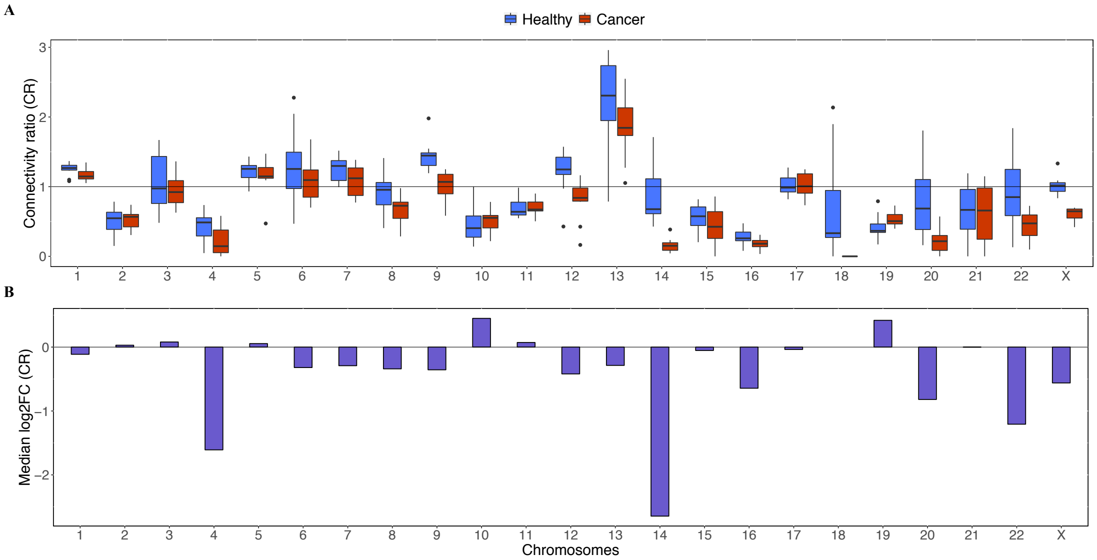


Fig. 7.4. Chromosome-wise connectivity ratio (CR) for healthy and cancer samples. (A) Distribution of the chromosome-wise miRNA-mRNA CR for healthy (blue) and cancer (red) samples. The horizontal line represents CR = 1, i.e., the same amount of *cis* and *trans* interactions for a chromosome. The boxplots show a shift towards *trans* interactions in cancer since CR decreases in most of the chromosomes in cancer compared to healthy samples. The chromosomes 4, 14, 16, and 20 had the lowest CR together with 18 that had a CR = 0 in cancer because of the loss of the full interactome. Chromosome 13 had the highest CR. (B) Median log2 fold-change (log2FC) of the CR in cancer versus healthy samples. Chromosomes 4, 14, 16 and 20 show a strong change from healthy to cancer. Log2FC was calculated as following: $log2FC = log_2(CR_{cancer}) - log_2(CR_{healthy})$. In chromosome 18, it was not possible to calculate the log2 fold-change since CR in cancer was 0. For the complete list of CRs refer to tables 7.2 and 7.3.

Chromosome	BLCA	BRCA	CESC	HNSC	KIRC	KIRP	LIHC	LUSC	STAD	UCEC
1	1.31	1.08	1.27	1.26	1.10	1.26	1.29	1.23	1.33	1.37
2	0.64	0.35	0.29	0.50	0.78	0.51	0.15	0.62	0.59	0.67
3	1.56	0.90	0.88	0.65	1.22	0.72	0.48	1.05	1.67	1.50
4	0.49	0.74	0.27	0.48	0.66	0.22	0.37	0.49	0.05	0.57
5	1.24	1.27	1.43	1.20	1.31	1.29	1.32	1.09	1.11	0.93
6	0.94	1.07	2.28	1.45	1.51	2.05	0.47	1.28	0.66	1.23
7	1.32	1.39	1.30	1.04	1.01	1.23	1.29	0.99	1.48	1.52
8	0.98	0.63	1.41	0.94	1.09	0.83	0.41	1.10	0.71	0.97
9	1.55	1.42	1.49	1.19	1.26	1.22	1.44	1.98	1.45	1.47
10	0.60	1.00	0.24	0.14	0.40	0.21	0.39	0.84	0.41	0.51
11	0.80	0.99	0.55	0.65	0.82	0.59	0.58	0.60	0.63	0.69
12	1.19	1.46	1.33	0.97	1.24	1.57	0.43	1.17	1.26	1.45
13	2.53	1.93	2.60	2.96	1.36	2.78	2.93	0.79	2.08	1.99
14	0.73	0.63	1.71	0.84	0.61	0.43	0.48	1.27	0.61	1.21
15	0.55	0.72	0.60	0.73	0.67	0.82	0.49	0.43	0.20	0.22
16	0.26	0.26	0.22	0.08	0.35	0.36	0.35	0.47	0.21	0.23
17	0.82	1.16	0.99	0.90	1.27	1.18	0.91	1.01	0.97	0.98
18	0.33	1.90	2.14	1.11	0.27	0.28	0	0	0.33	0.46
19	0.36	0.37	0.34	0.63	0.49	0.31	0.36	0.79	0.37	0.17
20	1.01	1.13	1.23	1.81	0.64	0.33	0.22	0.54	0.73	0.16
21	0.89	0.38	0	1.11	0.90	0.44	0.40	0	1.19	0.98
22	0.76	0.94	1.84	0.53	1.08	0.74	1.75	1.30	0.13	0.20
X	1.05	0.95	1.33	0.92	1.06	1.05	0.93	1.09	0.99	0.83

Table 7.2. Connectivity Ratio (CR) in healthy condition across TCGA studies.

Chromosome	BLCA	BRCA	CESC	HNSC	KIRC	KIRP	LIHC	LUSC	STAD	UCEC
1	1.35	1.11	1.15	1.24	1.12	1.14	1.20	1.05	1.22	1.06
2	0.56	0.58	0.31	0.64	0.44	0.32	0.42	0.61	0.59	0.74
3	0.95	1.09	0.63	0.77	1.32	0.78	0.70	1.09	1.36	0.89
4	0.47	0.11	0.06	0.58	0.22	0	0	0.05	0.43	0.18
5	1.11	1.09	1.47	1.29	1.14	1.38	0.47	1.15	1.22	1.15
6	1.27	1.45	1.12	1.14	1.07	1.68	0.70	0.83	0.71	0.92
7	1.38	1.08	0.81	1.28	0.77	0.78	1.11	1.24	1.29	1.13
8	0.87	0.77	0.51	0.66	0.74	0.98	0.47	0.71	0.78	0.29
9	1.25	1.17	1.18	1.07	0.58	0.94	0.88	1.07	1.25	0.65
10	0.47	0.78	0.58	0.22	0.58	0.53	0.39	0.32	0.59	0.66
11	0.74	0.57	0.65	0.68	0.82	0.90	0.66	0.50	0.67	0.79
12	1.06	1.16	0.85	0.78	0.83	0.16	0.43	0.82	1.02	0.85
13	2.55	1.87	1.27	1.88	1.05	1.77	2.22	1.72	1.82	2.48
14	0.38	0.09	0.20	0.18	0.04	0.13	0.08	0.08	0.23	0.17
15	0.28	0.86	0	0.68	0.48	0.79	0.37	0.53	0.25	0.25
16	0.19	0.18	0.14	0.31	0.22	0.23	0.14	0.03	0.27	0.12
17	1.06	1.03	0.89	0.73	1.22	0.84	1.25	1.24	0.96	0.99
18	0	0	0	0	0	0	0	0	0	0
19	0.63	0.73	0.61	0.51	0.49	0.46	0.44	0.50	0.40	0.56
20	0.19	0.57	0	0.08	0.53	0.31	0.25	0	0.27	0.10
21	1.15	0.19	0.94	0.19	0.99	0.40	0.89	0	0.43	1.10
22	0.68	0.36	0.44	0.50	0.53	0.19	0.10	0.28	0.62	0.72
X	0.50	0.65	0.42	0.68	0.57	0.70	0.68	0.54	0.69	0.64

Table 7.3. Connectivity Ratio (CR) in cancer condition across TCGA studies.

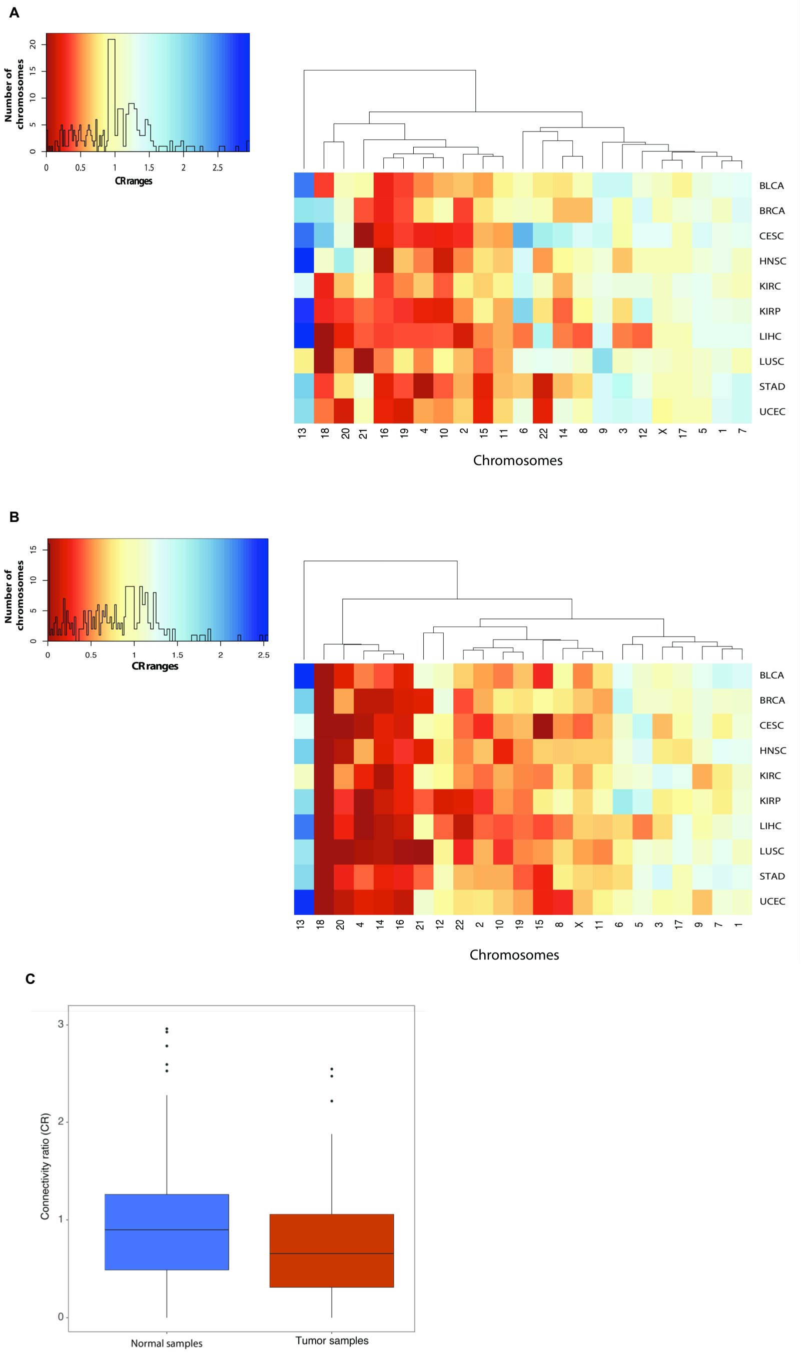


Fig. 7.5. Clustering of the chromosome-wise connectivity ratio (CR) for healthy and cancer samples. (A) Chromosome-wise miRNA-mRNA CR for healthy samples. The heatmap rows show the 10 considered TCGA cancer types while each column represents a chromosome. Hierarchical clustering of the CRs shows two main groups of chromosomes, the left one with a CR towards *trans* interactions and the right one with a CR towards *cis* interactions across cancer types. Chromosome 13 represents an individual cluster since it was the most *cis* chromosome. Overall, most chromosomes show about the same amount of *cis* and *trans* interactions in healthy samples. (B) The CR for each chromosome in the cancer tissues. The heatmap shows a shift to *trans* interactions for most of the chromosomes, particularly true for the cluster of chromosomes 4, 14, 16, 18 and 20. (C) Distribution of all CRs for normal and cancer tissues showing that the CR generally decreases in the cancer samples compared to the control samples.

7.3.2 Lost miRNAs in cancer compared to healthy tissue

It was shown that in cancer there was a general shift of the CR towards *trans* interactions. Subsequently, it was investigated how this shift was related to the change in the number of miRNAs, mRNAs or miRNA-mRNA interactions in cancer. To accomplish this task, we firstly analyzed for each chromosome the ratio between cancer and healthy of CR, number of miRNA, mRNA and miRNA-mRNA for each miRNA-mRNA interaction network (Figure 7.6). The ratios were summarized taking the median values of healthy and cancer along the different TCGA studies. We found that there was a general decrease in the number of miRNAs in cancer. In particular, chromosomes 14, 18 and 20 had the higher loss of miRNAs in cancer. Next, we calculated the Pearson correlation between the CR ratio and miRNA, mRNA and miRNA-mRNA interactions ratio between cancer and healthy samples. We discovered that the only significant linear correlation R was between the CR and the miRNA ratio (R = 0.55, p-value = 0.0061). This result underlines that the shift of the CR is more due to the loss of miRNAs. We focused on the study of which miRNAs were lost in the cluster of chromosomes with the highest shift towards *trans* interactions from healthy to cancer (i.e., chromosomes 4, 14, 16, 18 and 20) to understand how the loss of miRNAs was relevant to cancer.

At first, we analyzed those miRNAs that were present in the healthy network and that were lost in the cancer network (*physical loss*). We discovered that 28 miRNAs were physically lost in at least one of the ten cancers (Figure 7.7 and Table 7.4). Interestingly, tumour suppressors *miR-1-3p* and *miR-495-3p* located on chromosomes 18 and 14, respectively, were physically lost in most cancer studies.

Next, we explored network centrality for the miRNA-mRNA interaction networks to discover how many important miRNAs lost their centrality in cancer samples compared to the healthy regulatory networks. For each network, we called *most central* those miRNAs with the centrality value in the upper quartile. A *lost central* miRNA was a miRNA among the most central present in the healthy miRNA-mRNA network but absent in the corresponding cancer network. We used closeness centrality because it measures the possibility for a node to quickly communicate with other nodes, so in this case how miRNAs can cooperate efficiently in the regulation of common target genes. Since miRNA-mRNA interactions networks are bipartite graphs, we implemented through the package igraph [133], the notion of closeness centrality described in [134], setting the minimum distance between nodes of the same type as 2 and the minimum distance between nodes of different type as 1. We discovered that the chromosomes in the cluster of interest had few central miRNAs and lost a huge proportion of them (Figure 7.8). In particular, chromosomes 4, 14 and 18 lost all their most central miRNAs. Thus, the important miRNAs of these chromosomes were lost in cancer samples and their central role had not been replaced by other central miRNAs.

In summary, in the chromosome cluster of interest, we found three miRNAs not physically lost but with lost centrality in cancer compared to the healthy miRNA-mRNA network and a total of 28 physically lost miRNAs of which six miRNAs also lost their central role in the cancer network: *miR-381-3p*, *miR-409-3p*, *miR-218-5p*, *miR-138-5p*, *miR-1-3p* and *miR-495-3p* (Figure 7.7 and Table 7.4). These miRNAs are of particular interest because they could have an important role in the regulation of the functions of the healthy tissue and their loss can have a huge impact in the physiological activities. In addition, these miRNAs have been reported as tumour suppressor in several previous cancer studies. In particular, *miR-1-3p* was reported as tumour suppressor in bladder cancer by suppressing the proliferation, migration and invasion

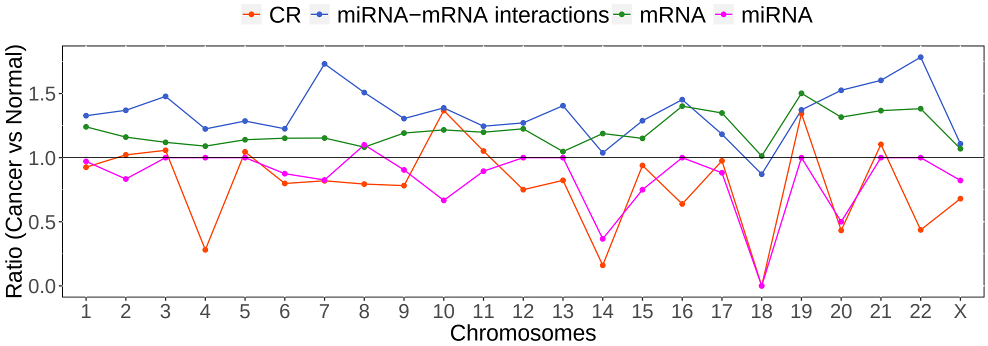


Fig. 7.6. Ratio (cancer VS healthy) of CR, number of miRNAs, mRNAs and miRNA-mRNA interactions. Each ratio was calculated taking the median value across cancer and healthy samples.

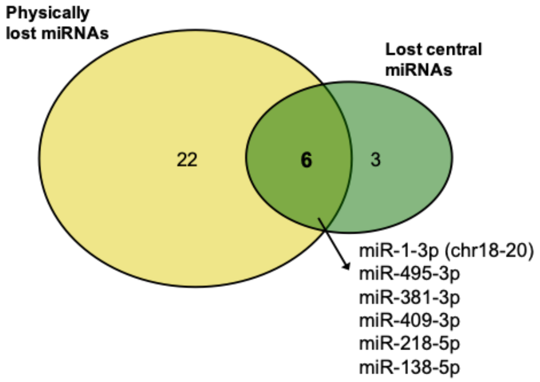


Fig. 7.7. miRNAs physically lost and/or with lost centrality in the cluster of chromosomes with the most shift to *trans* interactions. A total of 31 miRNAs located on chromosomes 4, 14, 16, 18 and 20 were lost in the cancer miRNA-mRNA networks. Among these, 28 miRNAs were physically lost in cancer and nine miRNAs lost centrality in cancer. Six miRNAs were both lost physically and lost centrality in cancer. Their names are listed below the Venn diagram.

through the upregulation of the gene *SFRP1* [135]. A previous study showed that *miR-495* was downregulated in malignant breast cancer and that its upregulation significantly suppressed proliferation and tumorigenesis inhibiting the G1-S phase transition through targeting of *Bmi-1* [136]. One study showed that *miR-381-3p* overexpression suppressed cell proliferation, cell cycle progression and migration by targeting *SETDB1* in breast cancer [137]. Another study showed that *miR-381-3p* induced apoptosis of renal cancer cells through the inhibition of *WEE1* [138]. *miR-409-3p* was involved in the suppression of cell growth and invasion of breast cancer cells by targeting *Akt1* [139] and in regulation of the expression of the oncogene E6 in cervical cancer [140]. Finally, the action of the miRNAs *miR-218-5p* and *miR-138-5p* in lung cancer were linked to a tumour suppressor activity negatively regulating *EGFR* in mouse and in the reversion of the gefitinib resistance in lung cancer cells via negatively regulating G protein-coupled receptor 124, respectively, [135, 141].

All together these results underline that miRNAs of the chromosomes 4, 14, 16, 18 and 20 have a central role in the regulation of the miRNA-mRNA network in healthy tissue and were dysregulated or completely lost in cancer. Their loss can have a critical impact on the

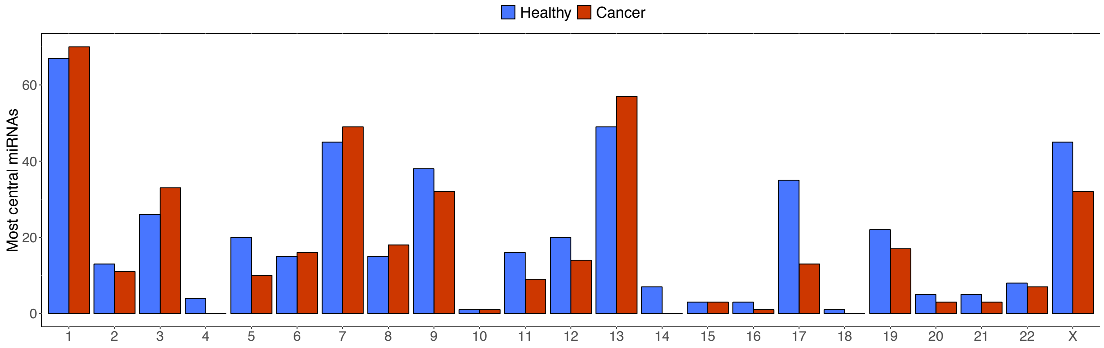


Fig. 7.8. Closeness centrality in healthy and cancer miRNA-mRNA networks. Number of *most central* miRNAs in miRNA-mRNA interaction networks in healthy and cancer condition for each chromosome. A miRNA is considered among the *most centrals* if its centrality is in the upper quartile of the centralities of a miRNA-mRNA network.

development and progression of different cancer types as their function have been linked to a tumour suppressor activity in several previous studies.

miRNA	Chr.	Loss type	Physical Loss (cancer study)	Central loss (cancer study)	(*) PubMedID
hsa-miR-410-3p	14	Physical	BLCA,CESC,HNSC, LIHC,LUSC,STAD, UCEC	-	-
hsa-miR-187-3p	18	Physical	BLCA,**KIRP**	-	(23916610)
hsa-miR-382-5p	14	Physical	BRCA,CESC,HNSC, **LIHC**,LUSC, UCEC	-	(29416925)
hsa-miR-758-3p	14	Physical	BRCA,CESC,HNSC, **LIHC**,**LUSC**,UCEC	-	(29032337), (30446524)
hsa-miR-136-5p	14	Physical	BRCA,CESC,HNSC, **KIRC**,**KIRP**,LIHC, UCEC	-	(29556316)
hsa-miR-370-3p	14	Physical	BRCA,CESC,LIHC	-	-
hsa-miR-487b-3p	14	Physical	BRCA,CESC,HNSC, LIHC,UCEC	-	-
hsa-miR-655-3p	14	Physical	CESC	-	-
hsa-miR-154-5p	14	Physical	CESC,LIHC,UCEC	-	-
hsa-miR-369-3p	14	Physical	CESC,HNSC,LIHC	-	-
hsa-miR-299-5p	14	Physical	CESC,HNSC,LIHC	-	-
hsa-miR-493-5p	14	Physical	CESC,**LIHC**,UCEC	-	(29328362), (22373578)
hsa-miR-409-5p	14	Physical	CESC,LIHC,UCEC	-	-
hsa-miR-376c-3p	14	Physical	CESC,HNSC,LIHC, UCEC	-	-
hsa-miR-411-3p	14	Physical	CESC	-	-
hsa-miR-493-3p	14	Physical	HNSC,**LIHC**	-	(22373578)
hsa-miR-134-5p	14	Physical	**KIRC**,**KIRP**	-	(28325280)
hsa-miR-379-5p	14	Physical	**KIRC**	-	(23618224)
hsa-miR-127-3p	14	Physical	KIRC,UCEC	-	-
hsa-miR-193b-3p	16	Physical	KIRP	-	-
hsa-miR-323a-3p	14	Physical	LIHC	-	-
hsa-miR-296-5p	20	Physical	**LUSC**	-	(26549165)
hsa-miR-342-3p	14	Centrality	-	CESC,KIRP	(25066298)
hsa-miR-140-5p	16	Centrality	-	CESC	(27588393)
hsa-miR-103a-3p	20	Centrality	-	HNSC,LUSC	-
hsa-miR-1-3p	18,20	Both	**BLCA**,**BRCA**,CESC, HNSC,KIRC,KIRP, LUSC,STAD,UCEC	HNSC	(28268231), (28268231), (28159933)
hsa-miR-495-3p	14	Both	**BRCA**,CESC,LIHC, LUSC,STAD,UCEC	LIHC,UCEC	(26020378)
hsa-miR-381-3p	14	Both	**BRCA**,**KIRC**,**KIRP**, LIHC,UCEC	KIRC	(30309377), (23778472)
hsa-miR-409-3p	14	Both	**BRCA**,**CESC**,KIRP, LIHC,UCEC	BRCA,LIHC	(26631969), (30711417)
hsa-miR-218-5p	4	Both	KIRP,LIHC	BRCA,KIRC,LIHC, LUSC	(27057632), (28192397)
hsa-miR-138-5p	16	Both	KIRP,**LUSC**	KIRP,LUSC	(24582749)

Table 7.4. miRNAs of chromosomes 4, 14, 16, 18 and 20 that were lost either physically and/or centrality in cancer compared to the corresponding healthy samples. References in which the considered miRNAs were found as tumour suppressors are reported. (*) Cancer types in bold highlights that miRNAs lost in our analysis were found as tumour suppressor in the same cancer tissue in literature.

7.4 Closing remarks

In this work, we have developed a general workflow for the dynamic analysis of structural properties. The methodology applies a new measure of connectivity ratio (CR) to analyse the chromosome-wise connectivity for miRNA-mRNA interactions. We applied the methodology to discover how the miRNA-mRNA chromosome-wise connectivity changes in cancer compared to healthy tissue. We discovered that the CR of most chromosomes decreased in the majority of cancers compared to healthy due to a strong shift towards *trans* interactions. This fact was particularly true for a small cluster composed by chromsomes 4, 14, 16, 18 and 20. Further, the decrease of the CR was linked to the loss of miRNAs with high centrality in the healthy network. Many of these miRNAs have previously been reported as tumour suppressors in several cancer studies.

Studying the cross-talk between chromosomes is becoming an emerging field of research, thanks to the development of novel experimental techniques such as chromosome conformation capture [142], which is providing fundamental discoveries on the spatial organisation of chromosomes in the cell and their interactions. These studies are crucial to increase our knowledge on modulation of gene expression regulated by regulatory elements that can be located in the same chromosome (in *cis*) or in another chromosome (in *trans*). A future analysis will consider to integrate this novel data with our results in order to understand whether *cis* and *trans* interactions at the PPI and miRNA-mRNA levels are somehow correlated with chromosome interactions. This analysis may allow us to discover that genes in close spatial proximity also physically interact at the protein level or miRNA-mRNA level.

This chapter showed the importance of dynamically studying the structural properties of a biological network in order to analyze its changes in different conditions. However, these types of analysis show how a biological network evolves in single time points or conditions. For this reason, an important task in Bioinformatics and Systems Biology is the design of methodologies that allow the dynamic simulation of the biological networks. They provide a more precise description of the evolution of the biological network in time allowing interesting predictions.

8

Modeling and Dynamic Simulation of Metabolic Networks through Electronic Design Automation

One of the aims of the Systems Biology is the modeling of biological systems in order to predict their complex emerging behaviors. As discussed in Section 5.4, a recurrent issue in the dynamic modeling of biological networks is related to the concept of *parameterization*. The aim of this work is to show that the use of languages, techniques, and tools well established in the context of Electronic Design Automation (EDA) can introduce automation and flexibility to model, simulate, and help the analysis of metabolic networks. We developed a methodology, implemented in the SystemC language based on the formalism of stochastic Petri nets (SPNs) for the simulation and the parameterization of metabolic networks in order to explain specific biological properties.

Section 8.1 introduces the SystemC language and its basic building blocks such as modules, processes, signals and ports.

Section 8.2 describes the methodology to model, simulate and parameterize metabolic networks using the SystemC language.

Section 8.3 describes the experimental results obtained from the analysis of the purine metabolic pathway in two different experimental conditions.

8.1 SystemC

SystemC is a language for the system-level modeling of hardware and software as an alternative to the common used languages VHDL and Verilog. Practically, SystemC [143] is a collection of C++ classes and macros that provides an *event-driven* simulation interface in C++. It allows designers to implement and simulate *concurrent processes*, each described using plain C++ syntax. SystemC processes communicate and synchronize in a simulated real-time environment, by using signals of all the data-types provided by C++, some additional ones provided by the SystemC library, as well as user-defined. The source code is compiled with the SystemC library, which includes a simulation kernel, to generate an executable. In general, SystemC allows designers to implement systems at different levels of abstraction and with different levels of details. It provides modeling features such as structural hierarchy and connectivity, communication abstraction, dynamic processes, timed event notifications, transaction-level modeling [144]. The key language features of SystemC are: *modules*, *ports*, *signals* and *processes*.

8.1.1 Time

The modeling of time is a very important feature of SystemC. There is a minimum and maximum amount of time representable that is given by a 64-bit integer value. The class used to represent time in SystemC is `sc_time`. It takes in input two parametrs: a `double` representing the amount of time and a time unit `sc_time_unit`. Table 8.1 shows the time units that can be represented in SystemC. Listing 8.1 shows an example of code to create a variable `time` that represents an amount of time of 10 seconds.

Type	Unit	Magnitude
`SC_FS`	femtosecond	10^{-15}
`SC_PS`	picosecond	10^{-12}
`SC_NS`	nanosecond	10^{-9}
`SC_US`	microsecond	10^{-6}
`SC_MS`	millisecond	10^{-3}
`SC_SEC`	second	10^{0}

Table 8.1. Time units in Systemc

Listing 8.1: SystemC code in which the variable `time` represents 10 seconds.

```
sc_time time = sc_time(10, SC_SEC);
```

8.1.2 Modules

Modules are the basic building blocks of a SystemC design hierarchy. A SystemC model usually consists of several modules that communicate via ports. Modules are C++ classes that extend the base class `sc_module`. The constructor of each module, that is created through the macro `SC_HAS_PROCESS`, requires the *name* of the module and can have several additional parameters. Listing 8.2 shows a simple example of a module `Place` that has the following integer parameters: `nIn`, `nOut`, `nInTr` and `nOutTr`.

Listing 8.2: SystemC code defining a module.

```
    template<typename t> class Place : public sc_core::sc_module{

    public:

        SC_HAS_PROCESS(Place);
        Place(sc_module_name name_, int nIn, int nOut, int nInTr, int nOutTr);
        virtual ~Place();

        void createProcesses();

        sc_in<bool> start;

        /*
        Other methods declaration ...
        */

    private:

        /*
        Variables declaration ...
        */

        // Input and output ports ...
        sc_in<t> inPort;
        sc_out<t> inAckPort;

};
```

8.1.3 Signals and ports

Signals are the medium (i.e., *channel*) to allow the communication of the SystemC modules. A signal can be defined using the keyword `sc_signal`. The value of a signal is available after a *delta cycle*. Delta cycle is used to simulate new updates and to wake up the processes in the phase of current time, therefore there is no time progress after a delta cycle. Signals link different SystemC modules through ports. Ports can be used as input (`sc_in<T>`), or output (`sc_out<T>`). Listing 8.2 shows the definition of the input port `inPort` and the output port `inAckPort` while Listing 8.3 shows an example of reading and writing on them.

Listing 8.3: SystemC code representing read and write on ports.

```
// Read of a port
this->inPort.read();

// Write of a port
this->inAckPort.write(1);
```

8.1.4 Processes

Processes are those components that perform functions and they are used to simulate the concurrency. In SytemC, processes are defined inside the module constructor and can be of two types: `SC_METHOD` and `SC_THREAD`. The main difference is that a process declared with the macro `SC_METHOD` can not be suspended through the `wait` statement while a process defined as `SC_THREAD` can be interrupted. Listing 8.4 shows an example of a `SC_THREAD` that is suspended for 100 seconds. SC_METHOD is useful to define, for example, activities that are periodically performed and that are atomic. SC_THREAD are used to simulate more complex behaviors because they are more versatile since they can be halted at any moment of the execution. By default a SC_THREAD is executed once, but usually they are maintained active in the whole duration of the simulation through the use of infinite loops.

Both SC_METHOD and SC_THREAD can be triggered by specific *events* that make these processes available to execution. Events allow for synchronization between processes and they are the key objects in SystemC models to provide event-driven simulation. A process can have a list of events, called *sensitivity list* that can trigger it. Listing 8.5 shows how the method createProcesses is defined as a SC_THREAD that can be triggered by an event in the port start. The SystemC component that deals with the execution of the processes is the scheduler, it decides which processes must be executed at each time step in order to simulate concurrency.

Listing 8.4: Example of SystemC wait statement that suspends a SC_THREAD for 100 seconds.

```
wait(100, SC_SEC);
```

Listing 8.5: Example of creation of a SystemC process as a SC_THREAD sensible to an event in a port start.

```
template<typename t> Place<t>::Place(sc_module_name name_,
        int nIn,
        int nOut,
        int nInTr,
        int nOutTr){

    /*
    Variables inizialitation ...
    */

    // Processes creation
    SC_THREAD(createProcesses);
    sensitive << this->start;
}
```

8.1.5 SystemC scheduler

SystemC language provides a simulation in which events rule the evolution of the system. In fact, events involve the execution of the processes and the updating of their output. SystemC is based on a simulation kernel that allows the modeling of concurrency choosing, at any time, what processes can be executed. The SystemC scheduler prepares the ready queue of the processes that can be run, based on the occurrence of events, and chooses the process that must run first. There are several types of events that can occurr in a SystemC environment:

- *Clock signal*, wakes up clocked processes and models the passing time.
- *Event notification*, wakes up the processes sensitive to that event.
- *Signal/port updating*, wakes up processes sensitive to that port or signal.

- *Delta and timed notification*, activates processes in the same simulation time or after the specified amount of time.

The SystemC scheduler works in 5 main steps (Figure 8.1): *initialization, evaluation, update, delta notification and timed notification* phase. In the initialization phase, the scheduler builds the first runnable queue and executes all processes until their end or their first `wait` statement. In the evaluation phase, the scheduler executes all processes in the runnable set using a co-operative multitasking. In fact, only one process can be executed at one time and can not preempt or interrupt other processes. This phase ends when there are no processes to execute. In the update and delta notification phase the scheduler updates ports and signals and puts in the runnable queue all processes that are sensitive to that events and delta notifications or timeout. In the timed notification phase, the simulation time is advanced until the earliest timed event (e.g. clock change) and the simulation ends if the are no timed events.

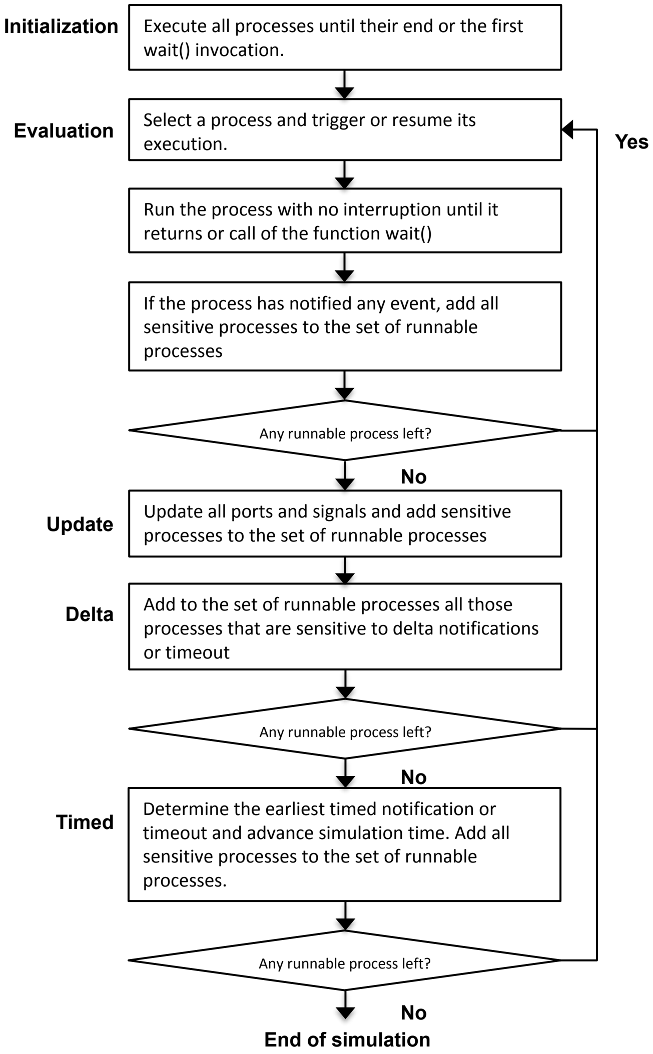

Initialization
Execute all processes until their end or the first wait() invocation.
Evaluation
Select a process and trigger or resume its execution.
Yes
Run the process with no interruption until it returns or call of the function wait()
If the process has notified any event, add all sensitive processes to the set of runnable processes
Any runnable process left?
No
Update
Update all ports and signals and add sensitive processes to the set of runnable processes
Delta
Add to the set of runnable processes all those processes that are sensitive to delta notifications or timeout
Any runnable process left?
No
Timed
Determine the earliest timed notification or timeout and advance simulation time. Add all sensitive processes to the set of runnable processes.
Any runnable process left?
No
End of simulation

Fig. 8.1. SystemC scheduler phases: Initialization, evaluation, update, delta and timed notifications.

8.2 Overview of the methodology

This thesis proposes a methodology that uses the SystemC statements to implement the formalism of Petri nets (PNs) to model and simulate metabolic networks. In particular, the methodology is based on stochastic Petri Nets (SPNs) because they are well suited to perform a dynamic simulation of biochemical reaction networks. SPNs have been largely used in the modeling of this kind of networks because allow to make quantitative simulations trough the Gillespie's algorithm, taking into account possible fluctuations and noise that are intrinsically present in the biological processes. Unlike the classical PNs, SPNs involve that the instantaneous firing of transitions are replaced by timed transitions, whereby each current firing rate depends on a function of the current concentration of the metabolites involved in the reaction and also on kinetic parameters. In addition to allowing the stochastic simulation of a metabolic network, the aim of this methodology is to find parameterizations of the system able to explain a defined biological property.

Figure 8.2 shows the main steps of the proposed methodology to model and simulate metabolic networks into Systemc language:

1. The SPN model is mapped into an extended finite state machine (EFSM) in order to be implemented into SystemC language.
2. The EFSM is sythetisized into SystemC with the support for stochastic simulation.
3. A phase of parameterization is performed to retrieve parameters able to explain a defined biological property. A genetic-based input generator is used to generate parameters of the metabolic reactions that is guided by dynamic assertion-based verification (ABV). In ABV, biological properties can be formally specified through the PSL language [145].

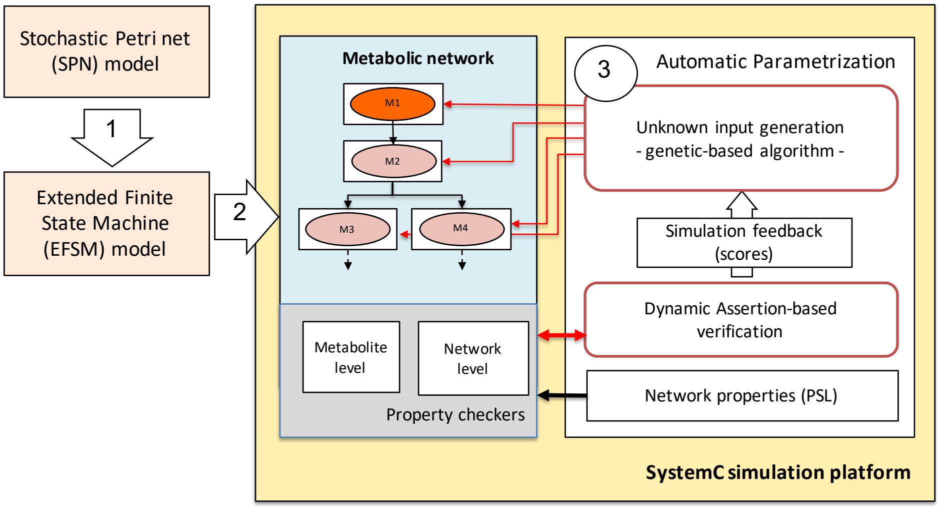

Stochastic Petri net (SPN) model
1
Extended Finite State Machine (EFSM) model
2
Metabolic network
M1
M2
M3
M4
Metabolite level
Network level
Property checkers
3
Automatic Parametrization
Unknown input generation - genetic-based algorithm -
Simulation feedback (scores)
Dynamic Assertion-based verification
Network properties (PSL)
SystemC simulation platform

Fig. 8.2. Overview of the SystemC methodology. (1) The SPN model is mapped into an extended finite state machine (EFSM) and (2) sythetisized into SystemC with the support for stochastic simulation. (3) A phase of parameterization is performed to retrieve parameters able to explain a biological property that can be defined through PSL language. A genetic-based input generator is used to generate parameters of the metabolic reactions that is guided by dynamic assertion-based verification (ABV).

8.2.1 Stochastic Petri nets

Stochastic Petri nets (SPNs) were introduced in Chapter 4 as an extension of the classic Petri nets to overcome some limitation such as the timing and randomness that are crucial for the simulation of biochemical reaction networks. In stochastic Petri nets, at each simulation step, it can be associated at every transition k a possible delay of firing τ_k that is determined by a random variable. Stochastic Petri nets are, usually, simulated using a well known technique called Gillespie's algorithm. This method, at each step chooses the time τ when executing a chemical reaction and it calculates a possible trajectory of the system. The classic method of the Gillespie algorithm is called Direct Method (DM) [78]. However, Gillespie proposed a variant of the direct method called First Reaction Method (FRM) [79] that is well suited for a concurrent and parallel implementation [146]. Our methodology takes inspiration from the FRM method to simulate the evolution of a metabolic networks through the SystemC language. Figure 8.3 shows a flowchart representing how stochastic simulation is performed in our methodology.

Given a metabolic network in which each reaction R is from a reactant M_i to a product M_j, the stochastic simulation starts with the generation of a possible time τ_R for every reaction R according to the following formula:

$$\tau_R = \left\lfloor \left(\frac{1}{a_R(m_i)}\right) \cdot ln\left(\frac{1}{rand_k}\right) \right\rfloor + 1 \quad (k = 1, \ldots, m) \tag{8.1}$$

where $rand_1, \ldots, rand_m$ are m random numbers from the uniform distribution $U(0, 1)$, $a_R(m_i) = c_i m_i$ is the mass-action propensity function where c_i is the reaction rate constant and m_i is the number of tokens of the reactant M_i. We consider a lower bound of each reaction delay (i.e., 1), which is associated to the minimum delay in the discrete simulation. At each step, the system executes the reaction with the minimum delay τ_R and updates the simulation time t. Further, the status of the reaction involved in a previous execution and the delays are updated. The simulation ends after a given number of simulation steps (N). The stochastic evolution of the system is obtained trough the use of EFSMs, that model the behaviour of each metabolite of the metabolic network.

8.2.2 Modeling SPN through EFSM

The EFSM model is widely used for modeling complex systems like reactive systems [147], communication protocols [148], buses [149] and controllers driving data-path [150] in the Electronic Design Automation (EDA) community.

Formally, an EFSM is defined as a 5-tuple $M = \langle S, I, O, D, T \rangle$ in which: S is a set of states, I is a set of input data, O is a set of output data, D is a n-dimensional linear space $D_1 \times \ldots \times D_n$, T is a transition relation so that T : $S \times D \times I \rightarrow S \times D \times\ O$. Any point in D is represented by a n-tuple $x = (x_1, ..., x_n)$ that models the state variables of the model. A pair $\langle s, x \rangle \in S \times D$ is called *configuration* of M, while a step in a EFSM $M = \langle S, I, O, D, T \rangle$ is defined as follows: If M is in a configuration $\langle s, x \rangle$ and it receives an input $i \in I$, it moves to the configuration $\langle t, y \rangle$ iff $((s, x, i), (t, y, o)) \in T$ for $o \in O$. In an EFSM, each transition is associated with two functions, *enabling function* and *update function*, which act on input, output, and state variables. The enabling function expresses a set of conditions on data, while the update function represents operation statements on data. A transition is fired if all conditions

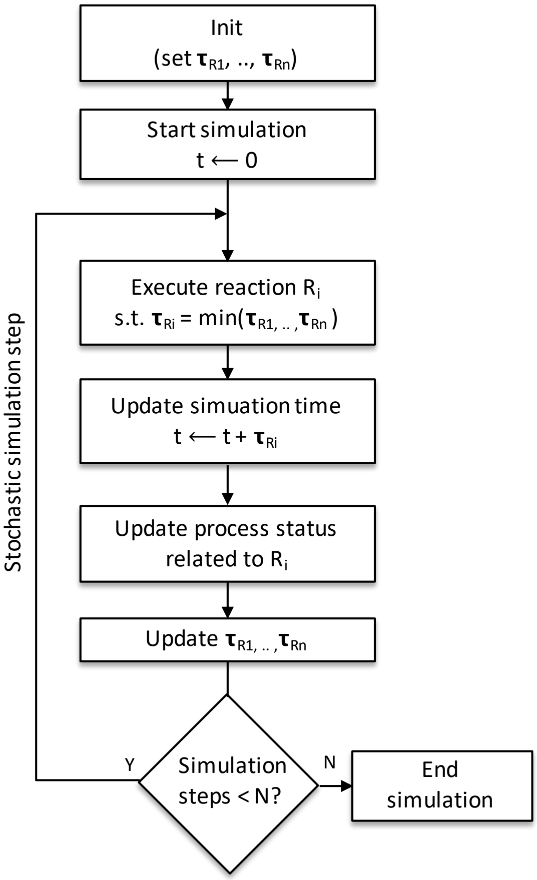


Fig. 8.3. Overview of the stochastic discrete simulation.

in the enabling function are satisfied, bringing the machine from the current to the destination state and performing the operations included in the update function.

EFSM has been adopted in this work to model the behaviour of each metabolite, which is characterized by production and consumption processes, providing a simple way to modularly synthesized the whole SPN model into SystemC. Figure 8.4 (A) shows a basic metabolic reaction R involving the metabolites M_i and M_j. Figure 8.4 (B) shows how the consumption (c_i) and production (p_j) processes are modeled through EFSMs to simulate the stochastic flow of tokens within a metabolic network. Each reaction involves the consumption process c_i of the reactant M_i, and the production process p_j of the reaction result M_j. We thus represent the production and consumption processes through two concurrent EFSMs, each one based on a two-state machine.

The consumption process of the metabolite M_i is initially in the `Ready` state which indicates the possibility to start the reaction. Next, the process moves to the `In consumption` state by updating the place M_j with the current reactant concentration. This operation enables the production activity of the concurrent process p_j. The consumption process waits in the `In consumption` state until the the reaction delay time expires. The reaction delay is computed by the production process that decides when notify that the reaction carried out. When the reaction is executed, the reactant concentration m_i is decreased by a given reaction constant m^0_R. If

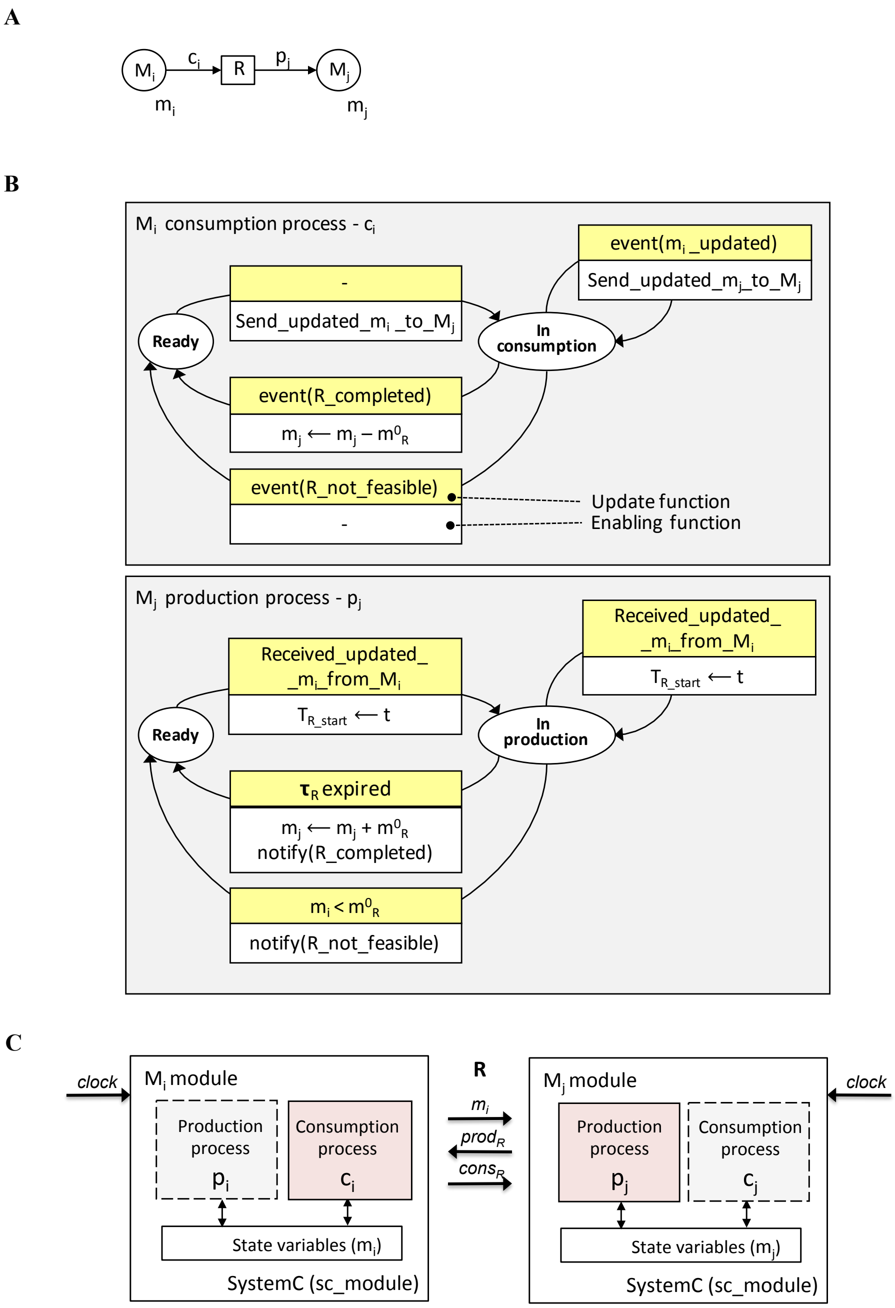


Fig. 8.4. Examples of a basic reaction from the metabolite M_i to the metabolite M_j and their mapping in EFSM and SystemC models. (A) the Petri net model of a simple metabolic reaction, (B) the EFSM representation of the reaction related to M_i and M_j, (C) the SystemC template implementing the consumption and production processes of the metabolites M_i and M_j. Both `sc_module` have ports that manage the clock signal, the signal for exchange the metabolite concentration m_i and signals for the communication of the status of the reaction ($prod_R$ and $cons_R$).

the reactant concentration is modified during the waiting time ($event(m_i\_updated)$ triggers), its new concentration is sent to p_j and the reaction delay time must be recomputed. This allows us to handle the multiple and concurrent evolutions of the SPN nodes and the corresponding effect on the rest of the network.

The production process of the metabolite M_j moves from the `Ready` state to the `In production` state as soon as it receives an updated reactant concentration. It computes the reaction delay time based on the reactant concentration and the mass-action parameter. Next, it saves the current simulation time t (which is updated by the SystemC scheduler at each simulation step) as the starting reaction time (T_{R_start}). When the reaction delay time (τ_R) expires, the reaction is completed and the M_j concentration is increased by the reaction constant m_R^0. In the `In production` state, the delay could be updated because of a change of the reactant concentration. A reaction is performed, automatically, when its delay is the minimum. The production process may become infeasible if the reactant concentration decreases under a threshold. This leads p_j to stop the process, to not increase the reaction result (i.e., M_j concentration) and to stop any M_i consumption process.

8.2.3 Implementation of metabolic networks through SystemC

In Section 8.1 the SystemC language constructs have been introduced. For modelling and simulating metabolic networks, they have been used in the proposed methodology as follows (Figure 8.4 (C)):

- *Modules*. Basically, a metabolic network is represented by a Systemc module (`sc_module`) for each metabolite. All metabolites are connected each other through signals (`sc_signal`).
- *Signals and ports*. Each metabolite is sensible to the clock signal in order to try to perform metabolic reactions at each time step. Therefore, the most basic port used by a metabolite module is the one used to read the clock value (`sc_in_clk`). All metabolites use the same signal (`sc_clock`) to allow the connection with the clock. Further, signals and ports have been used to allow communication and synchronization between the metabolites modules. For example, the reaction R the concentration m_i is sent from an output port (`sc_out<int>`) of the metabolite M_i to an input port (`sc_in<int>`) of the metabolite M_j through a signal (`sc_signal<int>`). The type of the ports and signals is integer because the quantities of tokens that flow within the network are integer values.
- *Processes*. They are the main computation elements and they are concurrent. Each metabolite behaviour has been modelled through two different processes (production and consumption), which react at each simulated instant of time to update the metabolite concentration. They are synchronized to update the metabolite concentration by considering both the consumption of the precursor metabolites and the production of the reaction products. Production and consumption processes are implemented as `SC_THREAD`. The use of `SC_THREAD` is important to flexibly simulate the simultaneous interactions between metabolites that could be activated or suspended at any time. Further, the processes update the state variables, which hold dynamic information of the metabolite such as its concentration.

The proposed template (Figure 8.4 (C)) distinguishes two sets of input data that can affect the model behaviour and one set of generated output:

- *Topological inputs (Input_Ti)*: They are inputs whose values are calculated at simulation time and depend on the topological interaction of the modelled metabolite with the reactants and the reaction product. Examples are concentration m_i and consumption status $cons_R$.

- *Parameters (Pi)*: They are inputs whose values depends on the environment characteristics and status, which are unknown at modelling time. Some examples are the parameters affecting the reaction rate (e.g., mass-action rate constants) and the initial concentration of metabolites. For each parameter, the platform generates different values with the aim of observing, via simulation, how such values affect the system dynamics.
- *Topological outputs (Output_Ti)*: They are outputs whose values are calculated at simulation time and depend on the metabolite status. Examples are concentration m_j and production status $prod_R$.

8.2.4 Assertion-based Verification for parameter estimation

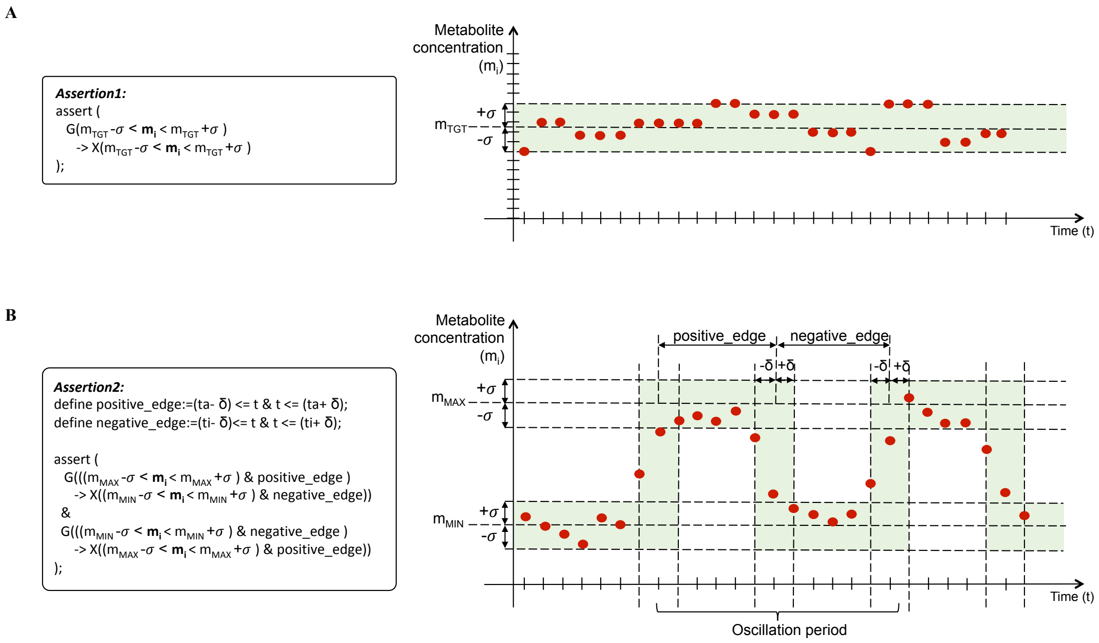
In the field of Electronic Design Automation (EDA), one of the most applied techniques is the functional verification of a model through the use of *assertions*. An assertion is a formal description of the behavior of system over time. The aim of the assertions is to detect errors in the model as well as driving the input pattern generation [151] during the model verification. In general, the verification of a model can be static or dynamic. This methodology uses simulation-based (i.e., dynamic) Assertion-based verification (ABV) through the use of assertions that are defined in PSL language and automatically synthesized into *checkers* [1] and integrated into the SystemC model. The checkers monitor observable variables of the model under validation during simulation and raise a failure signal when a contradicting behaviour is detected. In the context of metabolic networks, they aim at monitoring the concentration of the metabolites, whose temporal activity is a key to extrapolate and understand crucial biological properties such as stability of a biological entity (*simple attractors*) and oscillations (*complex attractors*).

The two examples in Figure 8.5 show how a biological behavior can be formalized through the use of assertions. In particular, the assertion in Figure 8.5 (A) checks whether the concentration of a metabolite remains stable (under a defined tolerance $\pm\sigma$) over time. The assertion in Figure 8.5 (B), more complex, checks the periodic oscillations of the metabolite concentration by considering defined tolerances ($\pm\sigma$, $\pm\delta$). For both the examples, if the value of the state variable m_i, during simulation, remains in the green zone, the assertion is satisfied. Otherwise, the assertion fails and the system raises an error signal.

In the proposed methodology, ABV is applied for the *automatic parameterization* (Figure 8.2), which aims at identifying the parameter settings that lead the network to satisfy the property described in the assertion. Furthermore, the ABV is responsible to give a *score* to the *unknown input generation* (Figure 8.2) for each simulation of the model. In fact, the unknown input generation is performed by a module taht implements a genetic algorithm that guide the system to find the mass-action parameters of the metabolic reactions. Genetic algorithms take advantage of the concept of the *fitness* of an individual to optimize an objective function. The definition of a proper fitness function (or scoring function) is crucial for the convergence of the algorithm. The aim of the fitness function, in our platform, is to assign a score to each simulated configuration of parameters. This score is needed to do an evolutionary step of the genetic algorithm used to generate new configurations.

The scoring methodology depends on the dynamic properties to check in the system and can be used to speed-up the convergence of the genetic algorithm as well as discard specific behaviours of the network. The definition of the fitness function depends on the property to check

[1] The platform relies on the IBM FoCs synthesizer [152] for the automatic synthesis of assertions.



```
Assertion1:
assert (
  G(m_TGT -σ < m_i < m_TGT +σ )
    -> X(m_TGT -σ < m_i < m_TGT +σ )
);
```

```
Assertion2:
define positive_edge:=(ta- δ) <= t & t <= (ta+ δ);
define negative_edge:=(ti- δ)<= t & t <= (ti+ δ);

assert (
  G(((m_MAX -σ < m_i < m_MAX +σ ) & positive_edge )
    -> X((m_MIN -σ < m_i < m_MIN +σ ) & negative_edge))
  &
  G(((m_MIN -σ < m_i < m_MIN +σ ) & negative_edge )
    -> X((m_MAX -σ < m_i < m_MAX +σ ) & positive_edge))
);
```

Fig. 8.5. Examples of assertion templates to verify a metabolite steady state. (A) Metabolite concentration stability, and (B) the oscillation of a metabolite concentration. σ and δ are user-defined tolerance constants. m_{TGT}, m_{MAX}, and m_{MIN} are user-defined concentration values of the observed metabolite. ta and ti are temporal counters initialized at the first oscillation, and that hold the time elapsed from the first state transition ($m_{MIN} \rightarrow m_{MAX}$ and $m_{MAX} \rightarrow m_{MIN}$, respectively). They are used to measure the positive edge and negative edge values. t is the counter, which is set, at each state transition from the second oscillation on, and it is used to measure the oscillation period.

but, in general, it is formulated as the distance between state vectors representing the *simulation trend* $s_i(t)$ in comparison to a defined *reference trend* $r_i(t)$ (i.e., desired behaviour of the system). The simulation trend is the actual behavior of the system under a certain configuration of parameters and it can be seen as a function that links the start and the ending point of that simulation. In $s_i(t)$, the end time represents a failure of the assertion. The reference trend is the target behavior of the system and it can be seen as a function that describe a behavior of the system that correctly verify a property (i.e., defined assertion). The ABV checks the simulation trend $s_i(t)$ and, eventually, interrupts the simulation and provide a score compared to $r_i(t)$. The score can be calculated at every step or only at the end of the simulation.

Given the state vectors $S = [s_1, s_2, \ldots, s_n]$ and $R = [r_1, r_2, \ldots, r_n]$ representing, respectively, the simulation and the reference trend the score is defined as follow:

$$score_c(t) = d(S, R)$$

where the function d is a distance function (e.g., Euclidean distance, maximum distance). The formulation of state vectors is very general and allows us to model biological behaviours such as the stability in concentration and the change of concentrations between time points. In the modeling of metabolism, an important assumption is that the components of the system are in a condition of equilibrium. To achieve the equilibrium of the concentrations of n metabolites, within a defined checking time, we defined the reference trend as follows:

$$r_i(t) = m_i, \forall t > 0, i = 1, 2, ..., n$$

where m_i is the starting concentration of a metabolite and t is the simulation time. The state vector of the simulation trend is defined as $S = [c_1, c_2, \ldots, c_n]$, where the terms c_i are the set of angular coefficients of the linear functions $s_i(t)$, linking the starting and the ending concentrations of the metabolites. The state vector of the reference trend is defined as $R = [0, 0, \ldots, 0]$ (i.e., the origin vector) because the aim is to mantain the starting concentrations of the metabolites as stable as possible to the end of the simulation. The ABV module uses the defined thresholds $\pm\sigma$ (Figure 8.5 (A)) to skip simulations that fail the properties during the checking time.

8.3 Experimental results

8.3.1 Network construction from metabolomics data

We analyzed the intracellular metabolic profile of resting naive CD4+ T cells isolated from lymph nodes and spleens of healthy SJL mice by magnetic cell sorting (all reagents from Miltenyi Biotech). Production of actively proliferating PLP139-151-specific encephalitogenic T cell lines was as previously described [153]. In brief, SJL mice were immunized subcutaneously with 300 mg of proteolypid protein (PLP)139-151 peptide. 10-12 days later, draining lymph nodes were removed and total cells were in vitro stimulated with PLP139-151 peptide for 4 days. T cell lines were obtained by re-stimulation of these cultures every 14 days for at least 3 times in the presence of irradiated splenocytes as antigen presenting cells at a ratio of 1:8 (T cell vs irradiated spleen cells). For the metabolomics analysis, actively proliferating PLP-specific encephalitogenic T cells were collected after two days of in vitro re-stimulation. The levels of purine metabolites were determined by performing metabolomics analysis in lysates from naive lymphocytes and PLP-specific T cells (*http://www.metabolon.com/*). Pathway and metabolism ontology analysis were performed using Cytoscape and MetScape plugin [154].

We built the Petri net model containing the most important features of *ex novo* purine synthesis and catabolism (Figure 8.6). The construction of the network reveled that several enzymes involved in nucleotide biosynthesis are regulated at enzymatic level by allosteric interactions and primarily by feedback inhibition. The purine synthesis pathway starts from the phosphoribosylpyrophosphate (PRPP) transformation to IMP by phosphoribosylpyrophosphate amidotransferase (PPAT), a regulatory allosteric enzyme that catalyzes the first step of the *de novo* purine nucleotide synthesis pathway. Purine biosynthesis is strongly inhibited by relevant pathway products such as AMP and GMP, which act directly on PPAT activity and thus on PRPP transformation [155]. The branch point at IMP node leading to AMP and GMP production is regulated by allosteric interactions, which are finely modulated in order to maintain equivalent levels of these two metabolites [156]. Moreover, AMP and GMP directly inhibit the production of their precursors, adenylosuccinate and xanthosine monophosphate (XMP) respectively [157, 158]. The enzyme activities, therefore, depend not only on the concentrations of the immediate substrates, but also on the end products. In our network, deoxy-AMP (dAMP) and deoxy-GMP (dGMP) represent the network final end products. These monomers are directly incorporated into DNA moiety and their basal concentration levels increase during proliferation [159]. The main players involved in the regulation of the purine metabolism include low variability in the intermediate and final products levels and integrated catabolic pathway ending in the production of urate as a waste compound. Previous studies have shown that urate is rapidly metabolized and has a central role in T cell proliferation, suggesting this metabolite may represent a key node in our network simulations [160].

Actively proliferating proteolipid protein (PLP)-specific T cells display a metabolic profile distinct from naive T lymphocytes. Proliferating cells show a shift to aerobic glycolysis, which is essential for the synthesis of protein and nucleic acid building blocks necessary for cell growth and division [161]. Our metabolomics analysis confirmed a general increase of purine pathway in proliferating PLP-specific T cells compared to naive T cells. In particular, we observed that deoxyadenosine monophosphate and deoxyguanosine were increased in proliferating PLP-specific cells. These metabolites are the end products of the purine pathway and correlated to an increase in DNA synthesis. Furthermore, we observed a 7-fold increment in deoxyinosine level, suggesting a hyper-activation of adenosine deaminase (ADA), which has an essential role in lymphocyte proliferation [162]. Substrates of ADA are both deoxyadenosine and adenosine, which are transformed into deoxyinosine and inosine respectively in the purine metabolic pathway. The adenosine value was 7-fold decreased in proliferating PLP-specific T cells compared to naive T cells, whereas the values of deoxyadenosine were not available in the metabolomics study. Our observations are in agreement with previous data showing that adenosine level must be low during proliferation due to its inhibitory effect on lymphocyte functions and provide evidence of increased ADA activity in proliferating cells [163]. Additionally, we observed an increase in xanthine and urate levels in PLP-specific cells, which are two important end compounds of the purine pathway. Interestingly, metabolomics analysis also showed changes in the level of other intermediate metabolites of the purine pathway, whose role and regulation is less characterized. These intermediates were considered for the construction of our metabolic network using KEGG database directly queried by Metscape plug-in. Due to the intrinsic complexity of metabolic networks and to guarantee a simplified but plausible and connected purine pathway, we added three intermediate metabolites in our network: xanthosine monophosphate (XMP), deoxyadenosine, and deoxyguanosine monophosphate (dGMP). Since their values were not provided in our metabolomics study, we assigned the same value of their immediate precursors for deoxyadenosine and dGMP, whereas for XMP we assigned the value of adenylosuccinate considering that both XMP and adenylosuccinate are products of IMP and their production rate is finely regulated in order to maintain equal intracellular amounts of AMP and GMP [156]. This assumption was clearly supported by our metabolomics analysis showing similar levels of AMP and GMP in the metabolic profiles of both naive T lymphocytes and actively-proliferating PLP-specific T cells.

8.3.2 Integrating metabolomics observations with simulation analysis.

We applied the proposed framework to understand how the dynamics of the purine pathway (Figure 8.6) changes between normal and autoreactive conditions.

Starting from metabolomics data obtained *in vitro*, we converted the relative concentrations into number of tokens for each metabolite multiplying by a factor of 10^3. With the term *concentration*, here, we refer to the number of tokens of a metabolite. Our model was simulated by generating reaction delays in the range $[1-1000]$ of all network reactions, which keep in equilibrium (i.e., steady state) the concentrations of each metabolite except dAMP, dGMP and urate that are the terminal nodes of the pathway. Several model assumptions were made to obtain the stability of all network metabolites over the simulation time. We assumed that the pathway is considered at steady state if the concentration of each element does not differ by more

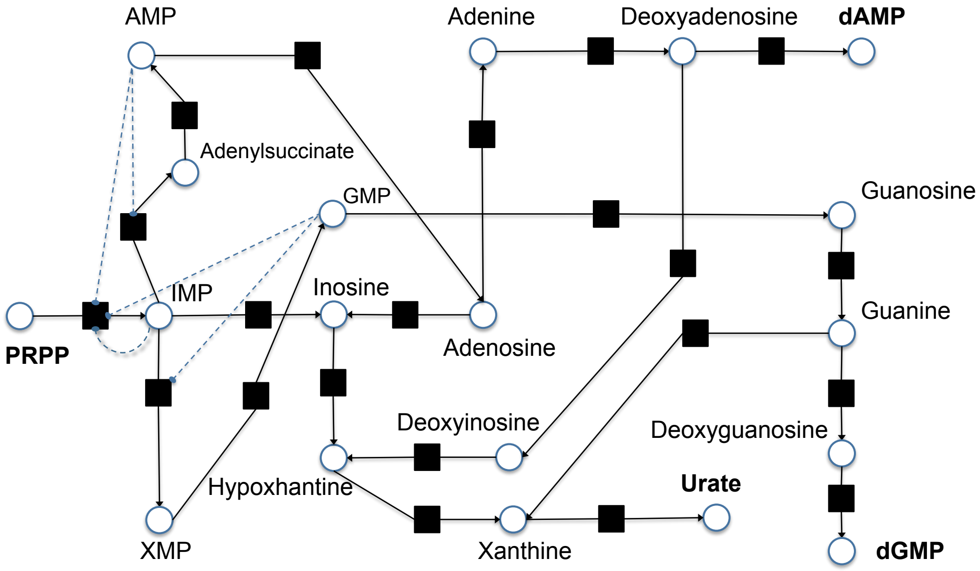


Fig. 8.6. Petri net model of the purine pathway.

than $\pm 50\%$ from the initial concentration and it is maintained stable throughout a simulation time of 10^5 simulation cycles. We formally specified this property through PSL assertions. In our model, the inhibition mechanisms are represented through the inhibition arcs, which is an extension of the classical Petri nets to represent the inhibition of a molecule when its concentration exceeds a certain threshold (Section 4.3.7). We assumed that a metabolite can inhibit a reaction when it grows up by 30% from its initial concentration. The genetic algorithm used for the unknown input generation has been configured with a population size of 250 individuals, a mutation probability of 0.05 and a crossover probability of 0.1. We defined the state vectors of the simulation and reference trends as described in Section 8.2.4. The fitness function was defined as the inverse of the Euclidean distance between the simulation and the reference trends. The selection method used to pick an individual from the population is *rank-based*, meaning that the reproduction is always done by taking individuals with better fitness.

We selected ten parameter configurations of the purine pathway for each condition. Fig. 8.7 shows the plot of the metabolite concentration obtained with one of such configurations. The complete parametrization phase required from 1 to 12 minutes for each network version. All the simulations were run on a machine equipped with an Intel(R) Xeon(R) CPU E5-2650 v4 clocked at 2200 Mhz and 16 GBs RAM, and the Ubuntu 16.04 operating system.

Our simulations led to interesting differences in the regulation of the purine pathway, suggesting that most of chemical reactions are highly favored in PLP-specific cells versus naive lymphocytes as shown in Fig. 8.8 (A). In fact, all metabolic reactions, with the exception of the reactions from Guanine to Xanthine (Guani:Xa) and from Adenosine to Inosine (Adeno:Ino), are speeded up in PLP-specific condition, having a lower average delay time generated by our framework (see Fig. 8.8 (A)). Further, the reaction from Deoxyadenosine to Deoxyinosine (DeoxA:DeoxI) had comparable delay times between naive and PLP-specific condition. Overall, the observed speed-up in the PLP-specific condition resulted in a greater production of the fundamental elements of the pathway dAMP, dGMP and urate (Fig. 8.8 (B)). Notably, the increased urate, dGMP and dAMP production in PLP-specific network reflects our metabolomics data and

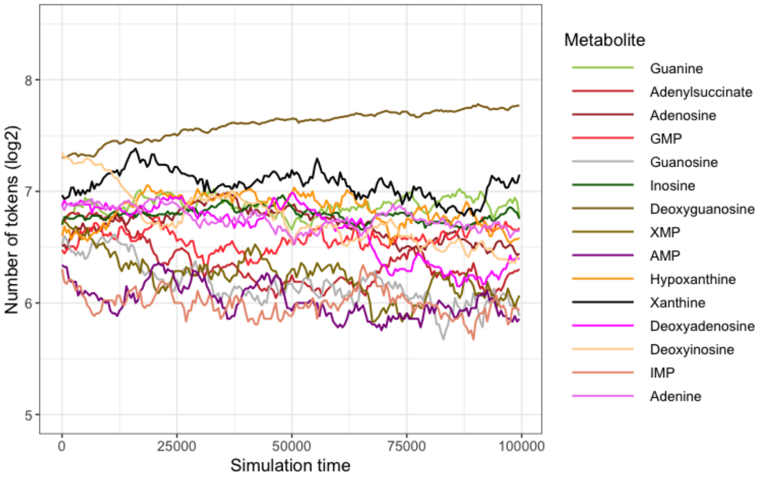


Fig. 8.7. Example of parameterization of the purine pathway in PLP-specific condition that leads the metabolites to stability within a simulation time of 10^5 clock cycles.

a well-known metabolic feature of proliferating lymphocytes [160], validating the potentiality of our methodology in simulating metabolic processes.

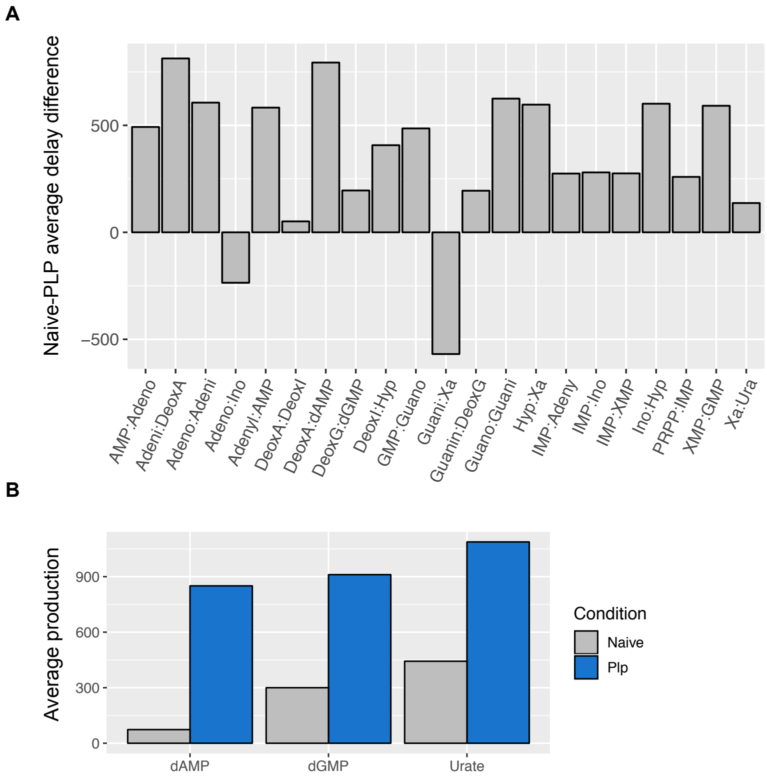


Fig. 8.8. Results of the analysis of 10 selected parameterizations of the purine pathway in condition of stability for naive and PLP-specific cells. (A) Average difference of the delays obtained parameterizing the pathway in naive and PLP-specific condition. (B) Difference in the final concentration of the metabolites dAMP, dGMP and urate.

8.4 Potential improvements of the methodology

In the context of hardware and software systems, functional verification is an important task for the verification of a model. It is a very complex task and it usually takes a lot of time and effort. In general, two main approaches to perform functional verification are model checking and simulation-based verification.

Model checking is a technique to formally verifying properties of a reactive system. In particular, techniques based on model checking verify the finite state machine of the system to check if a set of properties can be satisfied. The success of model checking is due to the fact that it helps the modeler to formally analyze all possible behaviours of a complex systems in a completely automatic way after a phase of modelling. However, the requirement to check all possible state to identify property violations makes this technique prone to the *state explosion problem*. On the other side, simulation-based verification is more efficient in terms of computational resources, but it is less sensitive in terms of error detection. Simulation-based verification can deal with very complex systems where model checking cannot be applied. However, in this type of verification the error detection is strictly dependent on the test sequence which may not detect some property. Both model checking and simulation-based verification are applied for the analysis of stochastic systems in which temporal properties must to be respected. In this context, a valuable approach is the simulation of the system for a certain number of iterations and the use of hypothesis testing to infer if there is a statistical evidence for the satisfaction of a defined property. This technique is called *statistical model checking (SMC)*. SMC has several strength compared to formal model checking methods. It requires only the execution of the system and it can be applied to very large systems because it can be easily parallelized. [164]. However, the main drawback related to SMC is that the guarantee that a defined property can be satisfied is probabilistic.

This thesis proposes a simulation-based ABV that aims to verify a defined property through simulations and genetic algorithms and to extract its parameters. The crucial point for the convergence of the genetic algorithm is the definition of the fitness function, as described previously. A potential improvement of the methodology could be the integration of a statistical model checking module that provides fitness value (i.e., score) for each simulated configuration of parameters [165]. This approach could speed-up the process of searching parameters guiding the genetic algorithm but it could be used in the perspective of the creation of a module for the global sensitivity analysis and calibration of parameters [166].

8.5 Closing remarks

This chapter described a methodology based on languages, techniques, and tools well established in the field of EDA to simulate and automatically parameterize the SPN model of metabolic networks. In particular, we applied the framework to study the purine metabolism pathway starting from metabolomics data obtained from naive lymphocytes and autoreactive T cells implicated in the induction of experimental autoimmune disorders. Thanks to the automatic parameterization of the model, we were able to reproduce the experimental results obtained in-vitro and to simulate the system under different conditions. From a biological point of view, the obtained simulation results suggest that the entire purine pathway is speeded-up in PLP-specific cells versus naive lymphocytes, according to our experimental data and literature.

Further studies are required to better exploit the prediction potential of our methodology by considering a higher number of metabolites and reactions and increasing the network complexity.

9

Conclusions

This thesis has proposed computational techniques that are useful for the structural and dynamic analysis of biological networks. The need of such computational techniques is due to the increasing biological data availability that raised a lot of biological questions that need to be resolved at different molecular levels and with the help of different approaches.

The structural analysis of biological networks requires, for example, the design of efficient computational strategies that deal with the solving of well-known graphs problems or the creation of new network analysis workflows able to integrate interaction and expression datasets in order to extract new knowledge. These techniques have been used to better understand the organization of several biological networks and how they change in physiological and pathological condition. On the other side, dynamic simulation requires smart and efficient methods because it has to deal also with the lack of knowledge that is still present, especially, in the study of biochemical reaction networks. A significant example is the study of metabolic processes in which the information about the kinetic parameters of the reactions is crucial to simulate the network dynamics and to discover important emerging behaviors.

The main contributions of this thesis can be summarized as follows:

- The design and exploration of new ordering strategies for solving the problem of subgraph isomorphism problem (SubGI) and, in particular, to make search strategy of the state-of-the-art RI less local through the use of centrality measures and more dependent on information about the target graph taking into account the distribution of the labels of the pattern nodes. It was shown that based on the type of the biological graph, an ordering strategy is better than another. This fact can be useful in the perspective of a multi-approach environment to chose the strategy depending on the properties of the target graphs involved.
- The formulation of a new measure to calculate the ratio between *cis*- and *trans-chromosomal* miRNA-mRNA interactions together with an analysis workflow to study miRNA-mRNA regulatory networks in different conditions. The study of the cross-talk between chromosomes is becoming an emerging research field of interest through the novel experimental techniques such as chromosome conformation capture. The proposed work can be useful for the integration of *cis* and *trans* regulatory interactions with PPI and chromosome interaction data.
- The design of a methodology implemented in SystemC language that applies techniques of the Electronic Design Automation (EDA) to provide modeling, simulation and parameterization of metabolic networks. Thanks to this methodology, the purine pathway was analyzed

in two different conditions obtaining interesting results that emphasize the potential of this methodology in simulating metabolic processes.

A

Publications

A.1 Published papers

1. Vincenzo Bonnici, Simone Caligola, Antonino Aparo, Rosalba Giugno. "Centrality speeds the subgraph isomorphism search up in target aware contexts". In *Computational Intelligence Methods for Bioinformatics and Biostatistics (CIBB)*, 2018. *Lecture Notes in Computer Science*. Springer, pages 19-26, 2020.
2. Annalisa Adamo, Jessica Brandi, Simone Caligola, Pietro Delfino, Riccardo Bazzoni, Roberta Carusone, Daniela Cecconi, Rosalba Giugno, Marcello Manfredi, Elisa Robotti, et al. "Extracellular Vesicles Mediate Mesenchymal Stromal Cell-Dependent Regulation of B Cell PI3K-AKT Signaling Pathway and Actin Cytoskeleton". *Frontiers in Immunology*, 10:446, 2019.
3. Henna Konttinen, Sohvi Ohtonen, Sara Wojciechowski, Anastasia Shakirzyanova, Simone Caligola, Rosalba Giugno, Yevheniia Ishchenko, Damián Hernández, Mohammad Feroze Fazaludeen, Shaila Eamen, et al. "PSEN1ΔE9, APPswe, and APOE4 confer disparate phenotypes in humanipsc-derived microglia". *Stem Cell Reports*, 13(4):669-683, 2019.
4. Simone Caligola, Tommaso Carlucci, Franco Fummi, Carlo Laudanna, Gabriela Constantin, Nicola Bombieri, and Rosalba Giugno. "Automatic Parameterization of the Purine Metabolism Pathway through Discrete Event-based Simulation". In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1-4, 2019.
5. Vincenzo Bonnici, Simone Caligola, Giulia Fiorini, Luca Giudice, and Rosalba Giugno. "LErNet: characterization of lncRNAs via context-aware network expansion and enrichment analysis". In *IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1-8, 2019.
6. Simone Caligola, Tommaso Carlucci, Franco Fummi, Carlo Laudanna, Gabriela Constantin, Nicola Bombieri, and Rosalba Giugno. "Efficient Simulation and Parametrization of Stochastic Petri Nets in Systemc: A Case Study from Systems Biology". In *Forum for Specification and Design Languages (FDL)*, pages 1-7, 2019. (Candidate as best paper).
7. Nicola Bombieri, Simone Caligola, Antonio Mastrandrea, Silvia Scaffeo, Tommaso Carlucci, Franco Fummi, Carlo Laudanna, Gabriela Constantin, and Rosalba Giugno. "Modelling, simulation, and tuning of metabolic networks through electronic design automation". In *Computational Intelligence methods for Bioinformatics and Biostatistics (CIBB)*, 2019.

A.2 Papers under review

1. Sanna Loppi, Paula Korhonen, Maria Bouvy-Liivrand, Simone Caligola, Ana Hernández de Sande, Natalia Kolosowska, Flavia Scoyni, Anna Rosell, Teresa Garcia Berrocoso, Sighild Lemarchant, Dhungana Hiramani, Joan Montaner, Jari Koistinaho, Katja Kanninen, Minna Kaikkonen, Rosalba Giugno, Heinäniemi Merja, Tarja Malm. "Peripheral inflammation preceeding ischemia impairs neuronal survival through mechanisms involving miR-127 in aged animals". *Aging Cell*, 2019.
2. Nicola Bombieri, Silvia Scaffeo, Antonio Mastrandrea, Simone Caligola, Tommaso Carlucci, Franco Fummi, Carlo Laudanna, Gabriela Constantin, Rosalba Giugno. "SystemC implementation of Stochastic Petri Nets for Simulation and Parametrization of Biological Networks". *Transactions on Embedded Computer Systems*, 2020.

A.3 Papers to submit

1. Simone Caligola, Francesco Russo, Vincenzo Bonnici, Søren Brunak, Rosalba Giugno, Kirstine Belling. "Chromosome-wise miRNA-mRNA interaction variation influences impact of cancer". *Scientific Reports*.

A.4 Book chapters

1. Antonella Mensi, Vincenzo Bonnici, Simone Caligola, Rosalba Giugno. (2019) "Construction and Analysis of miRNA Regulatory Networks". In: Laganà A. (eds) *MicroRNA Target Identification. Methods in Molecular Biology*, vol 1970. Humana Press, New York, NY.

References

1. G. A. Pavlopoulos, M. Secrier, C. N. Moschopoulos, T. G. Soldatos, S. Kossida, J. Aerts, R. Schneider, and P. G. Bagos, “Using graph theory to analyze biological networks,” *BioData Mining*, vol. 4, no. 1, p. 10, 2011.
2. Y. Tian, R. C. Mceachin, C. Santos, D. J. States, and J. M. Patel, “SAGA: a subgraph matching tool for biological graphs,” *Bioinformatics*, vol. 23, no. 2, pp. 232–239, 2006.
3. V. Bonnici, R. Giugno, A. Pulvirenti, D. Shasha, and A. Ferro, “A subgraph isomorphism algorithm and its application to biochemical data,” *BMC Bioinformatics*, vol. 14, no. 7, p. S13, 2013.
4. V. Bonnici, S. Caligola, A. Aparo, and R. Giugno, “Centrality speeds the subgraph isomorphism search up in target aware contexts,” in *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer, 2018, pp. 19–26.
5. I. K. Kirk, N. Weinhold, K. Belling, N. E. Skakkebæk, T. S. Jensen, H. Leffers, A. Juul, and S. Brunak, “Chromosome-wise protein interaction patterns and their impact on functional implications of large-scale genomic aberrations,” *Cell Systems*, vol. 4, no. 3, pp. 357–364, 2017.
6. N. Bombieri, R. Distefano, G. Scardoni, F. Fummi, C. Laudanna, and R. Giugno, “Dynamic modeling and simulation of leukocyte integrin activation through an electronic design automation framework,” in *International Conference on Computational Methods in Systems Biology*. Springer, 2014, pp. 143–154.
7. S. Caligola, T. Carlucci, F. Fummi, C. Laudanna, G. Constantin, N. Bombieri, and R. Giugno, “Automatic parameterization of the purine metabolism pathway through discrete event-based simulation,” in *2019 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2019, pp. 1–4.
8. ——, “Efficient simulation and parametrization of stochastic petri nets in systemc: A case study from systems biology,” in *2019 Forum for Specification and Design Languages (FDL)*. IEEE, 2019, pp. 1–7.
9. R. Garzon, G. Marcucci, and C. M. Croce, “Targeting microRNAs in cancer: rationale, strategies and challenges,” *Nature reviews Drug discovery*, vol. 9, no. 10, p. 775, 2010.
10. Kchouk, Mehdi and Gibrat, Jean-François and Elloumi, Mourad, “Generations of sequencing technologies: From first to next generation,” *Biology and Medicine*, vol. 9, no. 3, 2017.
11. J. K. Kulski, “Next-generation sequencing – An overview of the history, tools, and "Omic" applications,” *Next Generation Sequencing–Advances, Applications and Challenges*, pp. 3–60, 2016.
12. F. Ozsolak and P. M. Milos, “RNA sequencing: advances, challenges and opportunities,” *Nature reviews genetics*, vol. 12, no. 2, p. 87, 2011.
13. B. Rabbani, M. Tekin, and N. Mahdieh, “The promise of whole-exome sequencing in medical genetics,” *Journal of human genetics*, vol. 59, no. 1, p. 5, 2014.
14. K. Wang, C. Kim, J. Bradfield, Y. Guo, E. Toskala, F. G. Otieno, C. Hou, K. Thomas, C. Cardinale, G. J. Lyon *et al.*, “Whole-genome DNA/RNA sequencing identifies truncating mutations in RBCK1 in a novel Mendelian disease with neuromuscular and cardiac involvement,” *Genome medicine*, vol. 5, no. 7, p. 67, 2013.
15. G. Chang, S. Gao, X. Hou, Z. Xu, Y. Liu, L. Kang, Y. Tao, W. Liu, B. Huang, X. Kou *et al.*, “High-throughput sequencing reveals the disruption of methylation of imprinted gene in induced pluripotent stem cells,” *Cell research*, vol. 24, no. 3, p. 293, 2014.
16. S. R. Head, H. K. Komori, S. A. LaMere, T. Whisenant, F. Van Nieuwerburgh, D. R. Salomon, and P. Ordoukhanian, “Library construction for next-generation sequencing: overviews and challenges,” *Biotechniques*, vol. 56, no. 2, pp. 61–77, 2014.
17. A. Ameur, W. P. Kloosterman, and M. S. Hestand, “Single-molecule sequencing: towards clinical applications,” *Trends in biotechnology*, vol. 37, no. 1, pp. 72–85, 2019.

18. A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras, "STAR: ultrafast universal RNA-seq aligner," *Bioinformatics*, vol. 29, no. 1, pp. 15–21, 2013.
19. B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2," *Nature methods*, vol. 9, no. 4, p. 357, 2012.
20. H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows–Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.
21. E. J. Finehout and K. H. Lee, "An introduction to mass spectrometry applications in biological research," *Biochemistry and molecular biology Education*, vol. 32, no. 2, pp. 93–100, 2004.
22. R. Aebersold and M. Mann, "Mass spectrometry-based proteomics," *Nature*, vol. 422, no. 6928, p. 198, 2003.
23. D. S. Wishart, "Small molecules and disease," *PLoS computational biology*, vol. 8, no. 12, p. e1002805, 2012.
24. W. B. Dunn, "Mass spectrometry in systems biology: An introduction," in *Methods in enzymology*. Elsevier, 2011, vol. 500, pp. 15–35.
25. I. Karaman, "Preprocessing and pretreatment of metabolomics data for statistical analysis," in *Metabolomics: From Fundamentals to Clinical Applications*. Springer, 2017, pp. 145–161.
26. T.-H. Tsai, M. Wang, and H. W. Ressom, "Preprocessing and analysis of LC-MS-based proteomic data," in *Statistical Analysis in Proteomics*. Springer, 2016, pp. 63–76.
27. M. Brosch, L. Yu, T. Hubbard, and J. Choudhary, "Accurate and sensitive peptide identification with Mascot Percolator," *Journal of Proteome Research*, vol. 8, no. 6, pp. 3176–3181, 2009.
28. S. Tyanova, T. Temu, and J. Cox, "The MaxQuant computational platform for mass spectrometry-based shotgun proteomics," *Nature Protocols*, vol. 11, no. 12, p. 2301, 2016.
29. C. Ruttkies, E. L. Schymanski, S. Wolf, J. Hollender, and S. Neumann, "MetFrag relaunched: incorporating strategies beyond in silico fragmentation," *Journal of Cheminformatics*, vol. 8, no. 1, p. 3, 2016.
30. M. Karin, "Nuclear factor-κB in cancer development and progression," *Nature*, vol. 441, no. 7092, p. 431, 2006.
31. L. Mishra, R. Derynck, and B. Mishra, "Transforming growth factor-ß signaling in stem cells and cancer," *Science*, vol. 310, no. 5745, pp. 68–71, 2005.
32. X. Zhu, M. Gerstein, and M. Snyder, "Getting connected: analysis and principles of biological networks," *Genes & Development*, vol. 21, no. 9, pp. 1010–1024, 2007.
33. X. Guo and X.-F. Wang, "Signaling cross-talk between TGF-β/BMP and other pathways," *Cell Research*, vol. 19, no. 1, p. 71, 2009.
34. K. Yugandhar, S. Gupta, and H. Yu, "Inferring protein-protein interaction networks from mass spectrometry-based proteomic approaches: A mini-review," *Computational and Structural Biotechnology Journal*, 2019.
35. A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld *et al.*, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, p. 631, 2006.
36. M. Y. Hein, N. C. Hubner, I. Poser, J. Cox, N. Nagaraj, Y. Toyoda, I. A. Gak, I. Weisswange, J. Mansfeld, F. Buchholz *et al.*, "A human interactome in three quantitative dimensions organized by stoichiometries and abundances," *Cell*, vol. 163, no. 3, pp. 712–723, 2015.
37. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, p. 651, 2000.
38. D. Koschützki, B. H. Junker, J. Schwender, and F. Schreiber, "Structural analysis of metabolic networks based on flux centrality," *Journal of Theoretical Biology*, vol. 265, no. 3, pp. 261–269, 2010.
39. B. P. Lewis, I.-h. Shih, M. W. Jones-Rhoades, D. P. Bartel, and C. B. Burge, "Prediction of mammalian microRNA targets," *Cell*, vol. 115, no. 7, pp. 787–798, 2003.
40. Q. Cui, Z. Yu, E. O. Purisima, and E. Wang, "Principles of microRNA regulation of a human cellular signaling network," *Molecular Systems Biology*, vol. 2, no. 1, 2006.
41. Xue, Jin and Zhou, Dan and Poulsen, Orit and Hartley, Iain and Imamura, Toshihiro and Xie, Edward X and Haddad, Gabriel G, "Exploring miRNA-mRNA regulatory network in cardiac pathology in Na+/H+ exchanger isoform 1 transgenic mice," *Physiological Genomics*, vol. 50, no. 10, pp. 846–861, 2018.
42. Yang, Jue and Song, Hui and Cao, Kun and Song, Jialei and Zhou, Jianjiang, "Comprehensive analysis of Helicobacter pylori infection-associated diseases based on miRNA-mRNA interaction network," *Briefings in Bioinformatics*, 2018.
43. B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin *et al.*, "Genome-wide location and function of DNA binding proteins," *Science*, vol. 290, no. 5500, pp. 2306–2309, 2000.
44. C.-L. Wei, Q. Wu, V. B. Vega, K. P. Chiu, P. Ng, T. Zhang, A. Shahab, H. C. Yong, Y. Fu, Z. Weng *et al.*, "A global map of p53 transcription-factor binding sites in the human genome," *Cell*, vol. 124, no. 1, pp. 207–219, 2006.

45. M. B. Gerstein, A. Kundaje, M. Hariharan, S. G. Landt, K.-K. Yan, C. Cheng, X. J. Mu, E. Khurana, J. Rozowsky, R. Alexander *et al.*, “Architecture of the human regulatory network derived from ENCODE data,” *Nature*, vol. 489, no. 7414, p. 91, 2012.
46. H. K. Lee, A. K. Hsu, J. Sajdak, J. Qin, and P. Pavlidis, “Coexpression analysis of human genes across many microarray data sets,” *Genome Research*, vol. 14, no. 6, pp. 1085–1094, 2004.
47. X. Zhou, X. Xu, J. Wang, J. Lin, and W. Chen, “Identifying miRNA/mRNA negative regulation pairs in colorectal cancer,” *Scientific Reports*, vol. 5, p. 12995, 2015.
48. Nijman, Sebastian MB, “Synthetic lethality: general principles, utility and detection using genetic screens in human cells,” *FEBS Letters*, vol. 585, no. 1, pp. 1–6, 2011.
49. Boucher, Benjamin and Jenna, Sarah, “Genetic interaction networks: better understand to better predict,” *Frontiers in Genetics*, vol. 4, p. 290, 2013.
50. A. H. Y. Tong, M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Pagé, M. Robinson, S. Raghibizadeh, C. W. Hogue, H. Bussey *et al.*, “Systematic genetic analysis with ordered arrays of yeast deletion mutants,” *Science*, vol. 294, no. 5550, pp. 2364–2368, 2001.
51. T. Reguly, A. Breitkreutz, L. Boucher, B.-J. Breitkreutz, G. C. Hon, C. L. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong *et al.*, “Comprehensive curation and analysis of global interaction networks in Saccharomyces cerevisiae,” *Journal of Biology*, vol. 5, no. 4, p. 11, 2006.
52. A. Bender and J. R. Pringle, “Use of a screen for synthetic lethal and multicopy suppressee mutants to identify two new genes involved in morphogenesis in Saccharomyces cerevisiae.” *Molecular and Cellular Biology*, vol. 11, no. 3, pp. 1295–1305, 1991.
53. L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic *et al.*, “STRING 8–a global view on proteins and their functional interactions in 630 organisms,” *Nucleic acids research*, vol. 37, no. suppl_1, pp. D412–D416, 2008.
54. R. D. Leclerc, “Survival of the sparsest: robust gene networks are parsimonious,” *Molecular Systems Biology*, vol. 4, no. 1, 2008.
55. C. I. Del Genio, T. Gross, and K. E. Bassler, “All scale-free networks are sparse,” *Physical Review Letters*, vol. 107, no. 17, p. 178701, 2011.
56. R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, “Network motifs: simple building blocks of complex networks,” *Science*, vol. 298, no. 5594, pp. 824–827, 2002.
57. L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.
58. K. Stephenson and M. Zelen, “Rethinking centrality: Methods and examples,” *Social networks*, vol. 11, no. 1, pp. 1–37, 1989.
59. E. Estrada and J. A. Rodriguez-Velazquez, “Subgraph centrality in complex networks,” *Physical Review E*, vol. 71, no. 5, p. 056103, 2005.
60. M. Li, J. Wang, X. Chen, H. Wang, and Y. Pan, “A local average connectivity-based method for identifying essential proteins from the network level,” *Computational biology and chemistry*, vol. 35, no. 3, pp. 143–150, 2011.
61. J. Wang, M. Li, H. Wang, and Y. Pan, “Identification of essential proteins based on edge clustering coefficient,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1070–1080, 2011.
62. J. D. Orth, I. Thiele, and B. Ø. Palsson, “What is flux balance analysis?” *Nature Biotechnology*, vol. 28, no. 3, p. 245, 2010.
63. S. Schuster, T. Dandekar, and D. A. Fell, “Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering,” *Trends in Biotechnology*, vol. 17, no. 2, pp. 53–60, 1999.
64. C. H. Schilling, D. Letscher, and B. Ø. Palsson, “Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective,” *Journal of Theoretical Biology*, vol. 203, no. 3, pp. 229–248, 2000.
65. V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine, “Rule-based modelling of cellular signalling,” in *International conference on concurrency theory*. Springer, 2007, pp. 17–41.
66. F. Ciocchetta and J. Hillston, “Process algebras in systems biology,” in *International school on formal methods for the design of computer, communication and software systems*. Springer, 2008, pp. 265–312.
67. T. Murata, “Petri nets: Properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
68. V. N. Reddy, M. L. Mavrovouniotis, M. N. Liebman *et al.*, “Petri net representations in metabolic pathways.” in *ISMB*, vol. 93, 1993, pp. 328–336.

69. T. Agerwala, “Complete model for representing the coordination of asynchronous processes,” Johns Hopkins Univ., Baltimore, Md.(USA), Tech. Rep., 1974.
70. R. Hofestädt and S. Thelen, “Quantitative modeling of biochemical networks,” *In Silico Biology*, vol. 1, no. 1, pp. 39–53, 1998.
71. M. Heiner, I. Koch, and K. Voss, “Analysis and simulation of steady states in metabolic pathways with Petri nets,” in *Workshop and tutorial on practical use of coloured Petri nets and the CPN tools (CPN’01)*, 2001, pp. 15–34.
72. K. Voss, M. Heiner, and I. Koch, “Steady state analysis of metabolic pathways using Petri nets,” *In Silico Biology*, vol. 3, no. 3, pp. 367–387, 2003.
73. M. Heiner, I. Koch, and S. Schuster, “Using time-dependent Petri nets for the analysis of metabolic networks,” 2000.
74. H. Genrich, R. Küffner, and K. Voss, “Executable Petri net models for the analysis of metabolic pathways,” *International Journal on Software Tools for Technology Transfer*, vol. 3, no. 4, pp. 394–404, 2001.
75. D. Gilbert and M. Heiner, “From Petri nets to differential equations–an integrative approach for biochemical network analysis,” in *International Conference on Application and Theory of Petri Nets*. Springer, 2006, pp. 181–200.
76. M. Heiner, D. Gilbert, and R. Donaldson, “Petri nets for systems and synthetic biology,” in *International school on formal methods for the design of computer, communication and software systems*. Springer, 2008, pp. 215–264.
77. R. David and H. Alla, *Discrete, continuous, and hybrid Petri nets*. Springer, 2005, vol. 1.
78. D. T. Gillespie, “Exact stochastic simulation of coupled chemical reactions,” *The Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
79. ——, “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions,” *Journal of Computational Physics*, vol. 22, no. 4, pp. 403–434, 1976.
80. H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, “Lethality and centrality in protein networks,” *Nature*, vol. 411, no. 6833, p. 41, 2001.
81. X. He and J. Zhang, “Why do hubs tend to be essential in protein networks?” *PLoS Genetics*, vol. 2, no. 6, p. e88, 2006.
82. X.-L. Li, *Biological data mining in protein interaction networks*. Igi Global, 2009.
83. H.-W. Ma and A.-P. Zeng, “The connectivity structure, giant strong component and centrality of metabolic networks,” *Bioinformatics*, vol. 19, no. 11, pp. 1423–1430, 2003.
84. M. R. Da Silva, H. Ma, and A.-P. Zeng, “Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks,” *Proceedings of the IEEE*, vol. 96, no. 8, pp. 1411–1420, 2008.
85. G. Scardoni and C. Laudanna, “Centralities based analysis of complex networks,” *New Frontiers in Graph Theory*, pp. 323–348, 2012.
86. H. Yu, P. M. Kim, E. Sprecher, V. Trifonov, and M. Gerstein, “The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics,” *PLoS Computational Biology*, vol. 3, no. 4, p. e59, 2007.
87. R. Albert, “Scale-free networks in cell biology,” *Journal of cell science*, vol. 118, no. 21, pp. 4947–4957, 2005.
88. A.-L. Barabasi and Z. N. Oltvai, “Network biology: understanding the cell’s functional organization,” *Nature Reviews Genetics*, vol. 5, no. 2, p. 101, 2004.
89. C. Klein, A. Marino, M.-F. Sagot, P. Vieira Milreu, and M. Brilli, “Structural and dynamical analysis of biological networks,” *Briefings in functional genomics*, vol. 11, no. 6, pp. 420–433, 2012.
90. S. Maslov and K. Sneppen, “Specificity and stability in topology of protein networks,” *Science*, vol. 296, no. 5569, pp. 910–913, 2002.
91. T. Hu, A. S. Andrew, M. R. Karagas, and J. H. Moore, “Functional dyadicity and heterophilicity of gene-gene interactions in statistical epistasis networks,” *BioData Mining*, vol. 8, no. 1, p. 43, 2015.
92. N. Kashtan and U. Alon, “Spontaneous evolution of modularity and network motifs,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 39, pp. 13 773–13 778, 2005.
93. C. P. Bracken, H. S. Scott, and G. J. Goodall, “A network-biology perspective of microRNA function and dysfunction in cancer,” *Nature Reviews Genetics*, vol. 17, no. 12, p. 719, 2016.
94. E. Gifford, M. Johnson, D. Smith, and C. Tsai, “Structure-reactivity maps as a tool for visualizing xenobiotic structure-reactivity,” *Network Science*, vol. 2, pp. 1–33, 1996.
95. M. Heiner and I. Koch, “Petri net based model validation in systems biology,” in *International Conference on Application and Theory of Petri Nets*. Springer, 2004, pp. 216–237.

96. A. Sackmann, D. Formanowicz, P. Formanowicz, I. Koch, and J. Blazewicz, "An analysis of the Petri net based model of the human body iron homeostasis process," *Computational Biology and Chemistry*, vol. 31, no. 1, pp. 1–10, 2007.

97. J. Edwards and B. Palsson, "The Escherichia coli MG1655 in silico metabolic genotype: its definition, characteristics, and capabilities," *Proceedings of the National Academy of Sciences*, vol. 97, no. 10, pp. 5528–5533, 2000.

98. H. Yuan, C. Cheung, P. A. Hilbers, and N. A. van Riel, "Flux balance analysis of plant metabolism: the effect of biomass composition and model structure on model predictions," *Frontiers in Plant Science*, vol. 7, p. 537, 2016.

99. J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles, "Metabolic network structure determines key aspects of functionality and regulation," *Nature*, vol. 420, no. 6912, p. 190, 2002.

100. D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork *et al.*, "The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible," *Nucleic Acids Research*, p. gkw937, 2016.

101. I. S. Vlachos, M. D. Paraskevopoulou, D. Karagkouni, G. Georgakilas, T. Vergoulis, I. Kanellos, I.-L. Anastasopoulos, S. Maniou, K. Karathanou, D. Kalfakakou *et al.*, "DIANA-TarBase v7. 0: indexing more than half a million experimentally supported miRNA: mRNA interactions," *Nucleic Acids Research*, vol. 43, no. D1, pp. D153–D159, 2014.

102. V. Bonnici, G. De Caro, G. Constantino, S. Liuni, D. D'Elia, N. Bombieri, F. Licciulli, and R. Giugno, "Arena-Idb: a platform to build human non-coding RNA interaction networks," *BMC Bioinformatics*, vol. 19, no. 10, p. 231, 2018.

103. J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth *et al.*, "Evidence for dynamically organized modularity in the yeast protein–protein interaction network," *Nature*, vol. 430, no. 6995, p. 88, 2004.

104. M. Li, H. Zhang, J.-x. Wang, and Y. Pan, "A new essential protein discovery method based on the integration of protein-protein interaction and gene expression data," *BMC Systems Biology*, vol. 6, no. 1, p. 15, 2012.

105. C. Espinosa-Soto, P. Padilla-Longoria, and E. R. Alvarez-Buylla, "A gene regulatory network model for cell-fate determination during Arabidopsis thaliana flower development that is robust and recovers experimental gene expression profiles," *The Plant Cell*, vol. 16, no. 11, pp. 2923–2939, 2004.

106. L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla, "Genetic control of flower morphogenesis in Arabidopsis thaliana: a logical analysis." *Bioinformatics (Oxford, England)*, vol. 15, no. 7, pp. 593–606, 1999.

107. R. Steuer and B. H. Junker, "Computational models of metabolism: stability and regulation in metabolic networks," *Advances in Chemical Physics*, vol. 142, p. 105, 2009.

108. A. Hoffmann, A. Levchenko, M. L. Scott, and D. Baltimore, "The IκB-NF-κB signaling module: temporal control and selective gene activation," *Science*, vol. 298, no. 5596, pp. 1241–1245, 2002.

109. J. J. Tyson, "Modeling the cell division cycle: cdc2 and cyclin interactions." *Proceedings of the National Academy of Sciences*, vol. 88, no. 16, pp. 7328–7332, 1991.

110. H. Matsuno, Y. Tanaka, H. Aoshima, M. Matsui, S. Miyano *et al.*, "Biopathways representation and simulation on hybrid functional Petri net," *In silico biology*, vol. 3, no. 3, pp. 389–404, 2003.

111. L. Albergante, J. Timmis, L. Beattie, and P. M. Kaye, "A Petri net model of granulomatous inflammation: implications for IL-10 mediated control of Leishmania donovani infection," *PLoS Computational Biology*, vol. 9, no. 11, p. e1003334, 2013.

112. C. Zhan and L. F. Yeung, "Parameter estimation in systems biology models using spline approximation," *BMC Systems Biology*, vol. 5, no. 1, p. 14, 2011.

113. G. Goel, I.-C. Chou, and E. O. Voit, "System estimation from metabolic time-series data," *Bioinformatics*, vol. 24, no. 21, pp. 2505–2511, 2008.

114. V. Bonnici and R. Giugno, "On the variable ordering in subgraph isomorphism algorithms," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 14, no. 1, pp. 193–203, 2017.

115. C. Solnon, "Alldifferent-based filtering for subgraph isomorphism," *Artificial Intelligence*, vol. 174, no. 12-13, pp. 850–864, 2010.

116. V. Carletti, P. Foggia, A. Saggese, and M. Vento, "Introducing VF3: A new algorithm for subgraph isomorphism," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2017, pp. 128–139.

117. R. E. Tarjan and M. Yannakakis, "Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs," *SIAM Journal on Computing*, vol. 13, no. 3, pp. 566–579, 1984.

118. A. S. Flynt and E. C. Lai, “Biological principles of microRNA-mediated regulation: shared themes amid diversity,” *Nature Reviews Genetics*, vol. 9, no. 11, p. 831, 2008.

119. A. Jacobsen, J. Silber, G. Harinath, J. T. Huse, N. Schultz, and C. Sander, “Analysis of microRNA-target interactions across diverse cancer types,” *Nature Structural & Molecular Biology*, vol. 20, no. 11, p. 1325, 2013.

120. S.-D. Hsu, F.-M. Lin, W.-Y. Wu, C. Liang, W.-C. Huang, W.-L. Chan, W.-T. Tsai, G.-Z. Chen, C.-J. Lee, C.-M. Chiu *et al.*, “miRTarBase: a database curates experimentally validated microRNA–target interactions,” *Nucleic Acids Research*, vol. 39, no. suppl_1, pp. D163–D169, 2010.

121. A. Bisognin, G. Sales, A. Coppe, S. Bortoluzzi, and C. Romualdi, “MAGIA2: from miRNA and genes expression data integrative analysis to microRNA–transcription factor mixed regulatory circuits (2012 update),” *Nucleic Acids Research*, vol. 40, no. W1, pp. W13–W21, 2012.

122. G. T. Huang, C. Athanassiou, and P. V. Benos, “mirConnX: condition-specific mRNA-microRNA network integrator,” *Nucleic Acids Research*, vol. 39, no. suppl_2, pp. W416–W423, 2011.

123. Y.-y. Kang, Y. Liu, M.-L. Wang, M. Guo, Y. Wang, and Z.-F. Cheng, “Construction and analyses of the microRNA-target gene differential regulatory network in thyroid carcinoma,” *PloS One*, vol. 12, no. 6, p. e0178331, 2017.

124. T. Barrett, D. B. Troup, S. E. Wilhite, P. Ledoux, D. Rudnev, C. Evangelista, I. F. Kim, A. Soboleva, M. Tomashevsky, K. A. Marshall *et al.*, “NCBI GEO: archive for high-throughput functional genomic data,” *Nucleic Acids Research*, vol. 37, no. suppl_1, pp. D885–D890, 2008.

125. J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network *et al.*, “The cancer genome atlas pan-cancer analysis project,” *Nature Genetics*, vol. 45, no. 10, p. 1113, 2013.

126. B. Li and C. N. Dewey, “RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome,” *BMC Bioinformatics*, vol. 12, no. 1, p. 323, 2011.

127. J. Li, P. R. Bushel, T.-M. Chu, and R. D. Wolfinger, “Principal variance components analysis: Estimating batch effects in microarray gene expression data,” *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*, pp. 141–154, 2009.

128. J. T. Leek, W. E. Johnson, H. S. Parker, A. E. Jaffe, and J. D. Storey, “The sva package for removing batch effects and other unwanted variation in high-throughput experiments,” *Bioinformatics*, vol. 28, no. 6, pp. 882–883, 2012.

129. R. R. Wilcox, *Introduction to robust estimation and hypothesis testing*. Academic press, 2011.

130. P. Langfelder and S. Horvath, “WGCNA: an R package for weighted correlation network analysis,” *BMC Bioinformatics*, vol. 9, no. 1, p. 559, 2008.

131. T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down *et al.*, “The Ensembl genome database project,” *Nucleic Acids Research*, vol. 30, no. 1, pp. 38–41, 2002.

132. S. Griffiths-Jones, R. J. Grocock, S. Van Dongen, A. Bateman, and A. J. Enright, “miRBase: microRNA sequences, targets and gene nomenclature,” *Nucleic Acids Research*, vol. 34, no. suppl_1, pp. D140–D144, 2006.

133. G. Csardi, T. Nepusz *et al.*, “The igraph software package for complex network research,” *InterJournal, Complex Systems*, vol. 1695, no. 5, pp. 1–9, 2006.

134. K. Faust, “Centrality in affiliation networks,” *Social networks*, vol. 19, no. 2, pp. 157–191, 1997.

135. Z. Shi, L. Wang, H. Shen, C. Jiang, X. Ge, D. Li, Y. Wen, H. Sun, M. Pan, W. Li *et al.*, “Downregulation of miR-218 contributes to epithelial–mesenchymal transition and tumor metastasis in lung cancer by targeting Slug/ZEB2 signaling,” *Oncogene*, vol. 36, no. 18, p. 2577, 2017.

136. L. Wang, J.-L. Liu, L. Yu, X.-X. Liu, H.-M. Wu, F.-Y. Lei, S. Wu, and X. Wang, “Downregulated miR-45 Inhibits the G1-S Phase Transition by Targeting Bmi-1 in Breast Cancer,” *Medicine*, vol. 94, no. 21, 2015.

137. M. Wu, B. Fan, Q. Guo, Y. Li, R. Chen, N. Lv, Y. Diao, and Y. Luo, “Knockdown of SETDB1 inhibits breast cancer progression by miR-381-3p-related regulation,” *Biological Research*, vol. 51, no. 1, p. 39, 2018.

138. B. Chen, L. Duan, G. Yin, J. Tan, and X. Jiang, “Simultaneously expressed miR-424 and miR-381 synergistically suppress the proliferation and survival of renal cancer cells—Cdc2 activity is up-regulated by targeting WEE1,” *Clinics*, vol. 68, no. 6, pp. 825–833, 2013.

139. G. Zhang, Z. Liu, H. Xu, and Q. Yang, “miR-409-3p suppresses breast cancer cell growth and invasion by targeting Akt1,” *Biochemical and Biophysical Research Communications*, vol. 469, no. 2, pp. 189–195, 2016.

140. L. Sommerova, M. Anton, P. Bouchalova, H. Jasickova, V. Rak, E. Jandakova, I. Selingerova, M. Bartosik, B. Vojtesek, and R. Hrstka, “The role of miR-409-3p in regulation of HPV16/18-E6 mRNA in human cervical high-grade squamous intraepithelial lesions,” *Antiviral Research*, vol. 163, pp. 185–192, 2019.

141. Y. Gao, X. Fan, W. Li, W. Ping, Y. Deng, and X. Fu, “miR-138-5p reverses gefitinib resistance in non-small cell lung cancer cells via negatively regulating G protein-coupled receptor 124,” *Biochemical and Biophysical Research Communications*, vol. 446, no. 1, pp. 179–186, 2014.
142. E. Lieberman-Aiden, N. L. Van Berkum, L. Williams, M. Imakaev, T. Ragoczy, A. Telling, I. Amit, B. R. Lajoie, P. J. Sabo, M. O. Dorschner *et al.*, “Comprehensive mapping of long-range interactions reveals folding principles of the human genome,” *Science*, vol. 326, no. 5950, pp. 289–293, 2009.
143. “SystemC - Accellera Systems Initiative,” http://www.systemc.org.
144. L. Cai and D. Gajski, “Transaction Level Modeling: An Overview,” in *Proceedings of the 1st IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, ser. CODES+ISSS, 2003, pp. 19–24.
145. IEEE, “Property specification language - psl,” 2017, https://standards.ieee.org/findstds/standard/1850-2010.html.
146. C. Dittamo and D. Cangelosi, “Optimized parallel implementation of gillespie’s first reaction method on graphics processing units,” in *2009 International Conference on Computer Modeling and Simulation*. IEEE, 2009, pp. 156–161.
147. T. J. Koo, B. Sinopoli, A. Sangiovanni-Vincentelli, and S. Sastry, “A formal approach to reactive system design: unmanned aerial vehicle flight management system design example,” in *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404)*. IEEE, 1999, pp. 522–527.
148. H. Katagiri, K. Yasumoto, A. Kitajima, T. Higashino, and K. Taniguchi, “Hardware implementation of communication protocols modeled by concurrent EFSMs with multi-way synchronization,” in *Proceedings of the 37th Annual Design Automation Conference*. ACM, 2000, pp. 762–767.
149. A. Zitouni, S. Badrouchi, and R. Tourki, “Communication architecture synthesis for multi-bus SoC,” *Journal of Computer Science*, vol. 2, no. 1, pp. 63–71, 2006.
150. A. Guerrouat and H. Richter, “A component-based specification approach for embedded systems using FDTs,” in *ACM SIGSOFT Software Engineering Notes*, vol. 31, no. 2. ACM, 2005, p. 14.
151. M. Boulé and Z. Zilic, *Generating hardware assertion checkers*. Springer, 2008.
152. Y. Abarbanel, I. Beer, L. Gluhovsky, S. Keidar, and Y. Wolfsthal, “Focs–automatic generation of simulation checkers from formal specifications,” in *International Conference on Computer Aided Verification*. Springer, 2000, pp. 538–542.
153. L. Piccio, B. Rossi, E. Scarpini, C. Laudanna, C. Giagulli, A. C. Issekutz, D. Vestweber, E. C. Butcher, and G. Constantin, “Molecular mechanisms involved in lymphocyte recruitment in inflamed brain microvessels: critical roles for P-selectin glycoprotein ligand-1 and heterotrimeric Gi-linked receptors,” *The Journal of Immunology*, vol. 168, no. 4, pp. 1940–1949, 2002.
154. A. Karnovsky, T. Weymouth, T. Hull, V. G. Tarcea, G. Scardoni, C. Laudanna, M. A. Sartor, K. A. Stringer, H. Jagadish, C. Burant *et al.*, “Metscape 2 bioinformatics tool for the analysis and visualization of metabolomics and gene expression data,” *Bioinformatics*, vol. 28, no. 3, pp. 373–380, 2011.
155. A. Fridman, A. Saha, A. Chan, D. E. Casteel, R. B. Pilz, and G. R. Boss, “Cell cycle regulation of purine synthesis by phosphoribosyl pyrophosphate and inorganic phosphate,” *Biochemical Journal*, vol. 454, no. 1, pp. 91–99, 2013.
156. A. N. Lane and T. W.-M. Fan, “Regulation of mammalian nucleotide metabolism and biosynthesis,” *Nucleic acids research*, vol. 43, no. 4, pp. 2466–2485, 2015.
157. L. Hedstrom, “IMP dehydrogenase: structure, mechanism, and inhibition,” *Chemical Reviews*, vol. 109, no. 7, pp. 2903–2928, 2009.
158. M. B. Van Der Weyden and W. N. Kelly, “Human Adenylosuccinate Synthetase Partial Purification, Kinetic and Regulatory properties of the enzyme from placenta,” *Journal of Biological Chemistry*, vol. 249, no. 22, pp. 7282–7289, 1974.
159. A. A. Fernández-Ramos, V. Poindessous, C. Marchetti-Laurent, N. Pallet, and M.-A. Loriot, “The effect of immunosuppressive molecules on T-cell metabolic reprogramming,” *Biochimie*, vol. 127, pp. 23–36, 2016.
160. T. Eleftheriadis, G. Pissas, A. Karioti, G. Antoniadi, S. Golfinopoulos, V. Liakopoulos, A. Mamara, M. Speletas, G. Koukoulis, and I. Stefanidis, “Uric acid induces caspase-1 activation, IL-1β secretion and P2X7 receptor dependent proliferation in primary human lymphocytes,” *Hippokratia*, vol. 17, no. 2, p. 141, 2013.
161. M. G. Vander Heiden, L. C. Cantley, and C. B. Thompson, “Understanding the Warburg effect: the metabolic requirements of cell proliferation,” *Science*, vol. 324, no. 5930, pp. 1029–1033, 2009.
162. T. Hovi, J. Smyth, A. Allison, and S. Williams, “Role of adenosine deaminase in lymphocyte proliferation.” *Clinical and Experimental Immunology*, vol. 23, no. 3, p. 395, 1976.

163. J. Linden and C. Cekic, “Regulation of lymphocyte function by adenosine,” *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 32, no. 9, pp. 2097–2103, 2012.
164. A. Legay, B. Delahaye, and S. Bensalem, “Statistical model checking: An overview,” in *International conference on runtime verification*. Springer, 2010, pp. 122–135.
165. L. Bu, D. Peled, D. Shen, and Y. Zhuang, “Genetic synthesis of concurrent code using model checking and statistical model checking,” in *International Symposium on Model Checking Software*. Springer, 2018, pp. 275–291.
166. S. K. Palaniappan, B. M. Gyori, B. Liu, D. Hsu, and P. Thiagarajan, “Statistical model checking based calibration and analysis of bio-pathway models,” in *International Conference on Computational Methods in Systems Biology*. Springer, 2013, pp. 120–134.