Semantic approaches for the meta-optimization of complex biomolecular networks

Ali Ayadi

ÉCOLE DOCTORALE "MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR"

Laboratoire ICube - UMR7357

ÉCOLE DOCTORALE "SCIENCES DE GESTION"

Laboratoire LARODEC - LR01ES02

Thèse en cotutelle internationale présentée par :

Ali AYADI

soutenue le : **28 septembre 2018**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg et l'Université de Tunis**

Discipline/Spécialité : **Informatique**

Semantic approaches for the meta-optimization of complex biomolecular networks

Approches sémantiques pour la méta-optimisation des réseaux biomoléculaires complexes

THÈSE dirigée par :

Mme Cecilia ZANNI-MERK	Professeur, INSA Rouen, Université de Normandie, LITIS
Mme Saoussen KRICHEN	Professeur, ISG de Tunis, Université de Tunis, LARODEC

RAPPORTEURS :

Mme Claudia FRYDMAN	Professeur, Université d'Aix-Marseille, LIS
Mme Lina SOUALMIA	Maîtres de conférences HDR, Université de Rouen, LITIS

Autres membres du jury :

M. Edward SZCZERBICKI	Professeur, University of Newcastle Australia
M. François de BERTRAND de BEUVRON	Maîtres de conférences, INSA Strasbourg, ICube
M. Olivier POCH	Directeur de recherches, Université de Strasbourg, ICube

Acknowledgements

First of all, I would like to express my gratitude to my PhD advisers, Cecilia ZANNI-MERK, François de BERTRAND de BEUVRON, Saoussen KRICHEN and Julie THOMPSON who have been actively interested in my work. I would like to thank them as well for encouraging me and for allowing me to grow as a research scientist.

I deeply acknowledge the extraordinary and meticulous support of Cecilia ZANNI-MERK for the free exchange of ideas, constructive criticism, guidance, encouragement and moral support throughout the work. I am truly thankful for helping me achieve personal and professional goals.

Special thanks to François de BERTRAND de BEUVRON who graciously supported me with thesis comments and for his critical knowledge feedback that enriched my thesis with appropriate context, as well as for his encouragement and moral support.

I am also thankful to Saoussen KRICHEN for allowing me the opportunity to pursue a career in research, and for her encouragements.

In addition, I am very thankful to Julie THOMPSON for providing biological data and her help in validating the experimental results presented in this thesis.

Besides my advisers, I am very thankful to Claudia FRYDMAN and Lina SOUALMIA for accepting to read and review my thesis manuscript.

I gratefully acknowledge Olivier POCH and Edward SZCZERBICKI for accepting to be members of my thesis committee.

I would also like to thank my mid-thesis committee, Claudia FRYDMAN and Olivier POCH, for their insightful comments and encouragement.

I am thankful to thank the University of Strasbourg and the University of Tunis for funding my PhD studies. In addition, I would like to thank ICube administrative staff.

Lastly, I would like to thank my parents, my sister Zouhayra AYADI, my fiancee Eya MERSNI, and my close friends for their unfailing support and endless inspiration throughout these past three years of my PhD studies. In particular, Abdoul-Djawadou SALAOU for his help in validating my experimental results.

List of publications

The work of this thesis is based on the following publications:
International peer-reviewed conferences

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. A multi-objective method for optimizing the transittability of complex biomolecular networks, 22th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Belgrade, Serbia, Procedia Computer Science, septembre 2018.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. A multi-objective mathematical model for the optimization of the transittability of complex biomolecular network, 22th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Belgrade, Serbia, Procedia Computer Science, septembre 2018.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. Ontological reasoning for understanding the behaviour of complex biomolecular networks. In : Computer Systems and Applications (AICCSA), 2017 IEEE/ACS 14th International Conference on. IEEE, 2017. p. 1486-1493.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. BNO: An ontology for describing the behaviour of complex biomolecular networks, 21th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Marseille, France, Procedia Computer Science, septembre 2017, doi:10.1016/j.procs.2017.08.159.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. CBNSimulator: a simulator tool for understanding the behaviour of complex biomolecular networks using discrete time simulation, 21th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Marseille, France, page 8, Procedia Computer Science, avril 2017, doi:10.1016/j.procs.2017.08.157.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron. Understanding the Behaviour of Complex Biomolecular Networks by Combining Logical and Semantic Modeling, 9th International Conference Semantic Web Applications and Tools for Life Sciences, Amsterdam, Netherlands, page 12, Volume 1795, décembre 2016.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron. Qualitative Reasoning for Understanding the Behaviour of Complex Biomolecular Networks, the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KEOD 2016, Porto, Portugal, pages 144-149, Volume 2, n° 978-989-758-203-5, octobre 2016, doi:10.5220/0006065901440149.*

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen. Logical Semantic Modeling of Complex Biomolecular Networks, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016, York, United Kingdom, pages 475 - 484, Procedia Computer Science, Volume 96, septembre 2016, doi:http://dx.doi.org/10.1016/j.procs.2016.08.108.*

- *A. Ayadi, F. de Bertrand de Beuvron, C. Zanni-Merk, J. Thompson. Formalisation des réseaux biomoléculaires complexes, EGC 2016 – 16èmes Journées Francophones "Extraction et Gestion des Connaissances", Reims, France, Revue des Nouvelles Technologies de l'Information, Volume RNTI-E-30, janvier 2016.*

International peer-reviewed journals

- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen and Julie Thompson. A multi-objective method for optimizing the transittability of complex biomolecular networks, IEEE Transactions on Biomedical Engineering (submitted July 2018).*
- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen and Julie Thompson. A novel semantic approach for understanding the dynamic behaviour of biological networks, International Journal of Kinesiology and Sport Science (submitted June 2018).*
- *A. Ayadi, C. Zanni-Merk, F. de Bertrand de Beuvron, S. Krichen and Julie Thompson. BNO - an ontology for understanding the transittability of complex biomolecular networks, Journal of Web Semantics (submitted November 2017).*

Contents

List of Figures

List of Tables

General introduction

We offer here a general introduction of this thesis, realized at the Engineering Science, Computer Science and Imaging (ICube) laboratory within the context of a Joint PhD program ('thèse en cotutelle') with the Operational Research, Decision and Process Control (LARODEC) laboratory.

The research work began on 15 May 2015 and was conducted simultaneously between two teams (Figure 1):

- In France, within the SDC (Sciences, Données et Connaissances) team in collaboration with the CSTB (Complex Systems and Translational Bioinformatics) team of the ICube laboratory in Strasbourg (Laboratoire des sciences de l'Ingénieur, de l'Informatique et de l'Imagerie - UMR7357) under the direction of professor **Cecilia ZANNI-MERK** and the co-supervision of **François de BERTRAND de BEUVRON** and **Julie THOMPSON**.
  The SDC team, in particular, Cecilia and François fundamentally work in designing and producing formal models for the development of Knowledge-Based Systems. These software reproduce the behaviour of a human expert performing an intellectual task in a specific field. They are based on the explicit nature of knowledge, which is formalized in different ways. Among these formal models, ontologies which are generally used with a set of rules that are chained together to simulate the reasoning of a human expert.
  With respect to the CSTB team, they are working on developing validated high throughput computational biology to study the behaviour of biological systems ranging from protein families to relational systems such as "hyperstructures" (macromolecular complex, organelles, viruses) or biological networks (metabolosome, transcriptional, developmental or disease-related networks).

- In Tunisia, within the Decision Support and Game Theory team of the LARODEC laboratory (Laboratoire de Recherche Opérationnelle, de Décision et de Contrôle de processus LR01ES02) under the direction of professor **Saoussen KRICHEN**.
  This team focuses on the theoretical aspects (formal models, axiomatic analyses, complex studies), for the representation of complex problems and the production of algorithms for their exact or approached resolution, the design of intelligent systems (knowledge-based systems, decision support systems, etc.) and their implementation in real applications.

The interdisciplinary nature of this project is enriched by the contribution of the three teams mentioned above. The complementarity of their skills promoted the elaboration and management of the project.

Biological and scientific context

Cells do not live in stable conditions but are subject to intra and extra-cellular stimuli from their environments that vary over time [9]. With the recent development of high-throughput technologies, huge amounts of data have been generated to describe the complex processes and molecular mechanisms at work in the cell, through the study of cellular components on several levels (genes, proteins, metabolites, etc.) [10, 11, 12]. A major challenge is how to extract important knowledge from all these data in order to understand and infer cellular functions and behaviour in different conditions. Systems biology involves a comprehensive quantitative analysis of the manner in which all the components of a biological system interact functionally over time [13]. This integrative discipline aims to combine all information (from different levels) in order to understand the processes and behaviours of all cellular components while studying the interactions that take place among them.

Figure 1 – Research laboratories and institutions in which this thesis has been conducted.

The cell can be considered as a complex system consisting of thousands of different molecular entities (genes, proteins and metabolites), which interact with each other physically, functionally and logically to create a molecular network [6, 14]. To reduce the complexity, most traditional studies have focused only on a particular level of the cellular system, such as gene regulatory networks, protein-protein interaction networks, or metabolic networks. Various approaches have been developed to model, analyse and understand such networks, including ordinary differential equations [15], stochastic methods [16, 17], Boolean networks [18], Bayesian networks [19], Petri nets [20], etc., and comparative studies of these techniques have been performed (e.g. [21], [22], [23]). Nevertheless, few approaches have been developed to study the cellular system as a whole, and in particular the interactions among the different types of molecular networks. Furthermore, most of the existing modelling techniques do not take into account the dynamics of the network [24, 25, 26, 27].

Recently, some authors have started to address the dynamic aspects, and have introduced concepts such as the 'controllability' [28] of a network, where the ability to steer a complex directed network from any initial state toward any other desired state is measured by the minimum number of required driver nodes (nodes with the ability to steer the entire network). They showed that in order to have complete controllability, the minimum number of driver nodes is 80% of nodes in a regulatory biomolecular network. This result led other groups to develop a theoretic framework for studying transitions between two specific states of directed complex networks, a concept they call the 'transittability' [6] of the network.

Our thesis belongs within this context by designing and developing a new platform to simulate the state changes of complex molecular networks to understand and steer their behaviour over time.

Aims and objectives

As discussed in the previous section, the overall goal of this study is to propose an intelligent system that enables biologists to simulate the state changes of biomolecular networks with the goal of steering their behaviours.

To achieve this aim, our objectives are:

- characterise the molecular components of a cell;
- understand the dynamic interactions between molecular components and environmental stimuli;
- provide a tool for biologists to reproduce the behaviour of complex networks;

- infer an optimal set of external stimuli to be applied during a predetermined time interval to steer the network from its current state to the desired state.

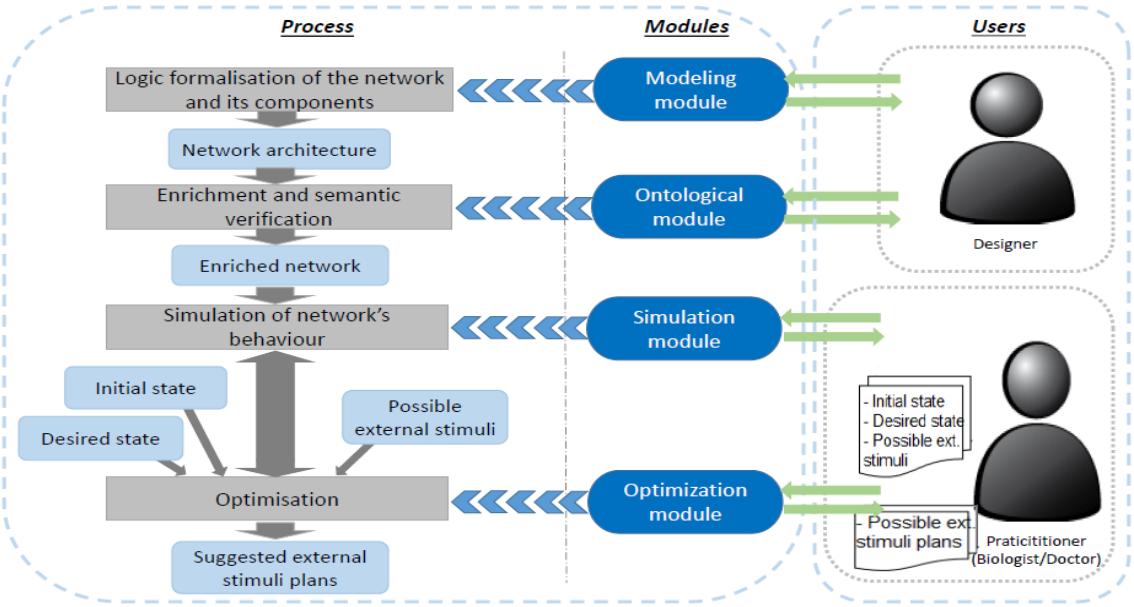


Figure 2 – General architecture of our proposed platform.

Figure 2 illustrates, the general architecture of our platform which combines four modules:

- The first module must provide a comprehensive approach to model a complex biomolecular network considering all its levels and their molecular components. This logical formalization must take into account the complexity and heterogeneity of these molecular components and their multilevel structure.

- The second module has an essential role because it ensures the management, modelling and sharing of expert knowledge. This ontological module is based on a formal model providing a better integration and interoperability of diverse information assets and can easily accommodate changes without requiring to re-define the platform's design. This module uses semantic technologies which offer new knowledge or new relationships in order to enable machines to understand and respond to complex human requests based on their semantic and contextual meaning [29]. This module takes as input all the native information introduced by the expert (state of the network, its structure, etc) through the logic-based formalization provided by the first module. Then, the ontological module provides output inferred network that is composed of native and inferred knowledge about its transition states.

- The simulation module consists of a simulator of qualitative models of complex biomolecular networks based on a discrete, logical formalism. It also allows users to simulate and/or analyse its qualitative dynamical behaviour. Indeed, this simulator integrates all the information given by the expert (the enriched network with native and inferred knowledge) with other parameters in order to better reproduce the conditions of the evaluated biomolecular network and its components over time. The results generated during the simulation are graphically displayed to the users to facilitate their interpretation and are then transmitted to the optimization module.

- The optimization module firstly parses the input file in order to extract the initial and desired states of the networks, and all the possible external stimuli defined by the practitioner. Then, based on the evaluation criteria values, this module will offer the best transition sequences for driving the biomolecular network from the initial state to the desired state.

Contributions and fields of research concerned

As Figure 2 illustrates, the architecture of this platform combines four module. Each module corresponds to an independent discipline. Consequently, our contributions have been classified into four disciplines as follows:

- **Mathematical systems modelling**: In this domain, we propose a logic-based modelling approach to addresses the problem of modelling complex biomolecular networks considering the diversity and heterogeneity of their molecular components, and adopting a global vision which considers their multi-level aspects. This formalization focuses on the structure of the network (model the diverse components and their interactions), network control (identify the function and role of each component) and network dynamics (observe its behaviour over time).
- **Knowledge engineering**: The main goal of this work is to provide a semantic approach which provides the necessary knowledge for modelling and understanding the behaviour of complex biomolecular networks and their state changes. Moreover, we develop the Biomolecular Network Ontology to formalize the domain knowledge of complex biomolecular networks making it visible and accessible to all biologists working on this topic.
- **Computer simulation**: In this field we propose two approaches to simulate biomolecular networks: qualitative and quantitative. These approaches are both based on logic-based modelling and reproduce the behaviour of complex biomolecular networks and their components over time.
- **Combinatorial optimization**: In this discipline, our works consist of adapting existing optimization technologies such as the multi-objective genetic algorithm, with the goal of optimizing the transittability of complex biomolecular networks. This approach provides the best set of external stimuli for driving the network.

Thesis outline

As shown in Figure 3, the structure of this dissertation is organized as follows:

- The **Introduction** gives a general introduction for the research background, research issues, research scope and contributions.
- The first part I **State-of-the-art** presents the theoretical foundations of this thesis, including a detailed literature review on all previous research done on each topic.
  - *Chapter 1* presents the biological environment we are working in. Our main focus lies in complex biomolecular networks and their transittability.
  - *Chapter 2* provides an overview of the background information about mathematical models in systems biology, reviews the most popular among them, and presents the main problem addressed by our thesis in this field: a logical modelling of complex biomolecular networks.
  - *Chapter 3* provides an overview of the background information about ontologies, describes the major bio-ontologies in systems biology, and presents the main problem addressed by our thesis: a domain ontology for describing the complex biomolecular networks domain.
  - *Chapter 4* provides an overview of the background information about the simulation in systems biology, details the major simulation tools and platforms in literature, and presents the main problem addressed by our thesis in this topic: a qualitative and discrete-event simulator for understanding the behaviour of complex biomolecular networks.
  - *Chapter 5* provides an overview of the background information about optimization tools, including a synthesis of the works conducted in systems biology optimization problems, and presents the main problem addressed by our thesis in this field: a genetic algorithm for solving and optimizing the transittability of complex biomolecular networks.
- The second part II **Contributions** describes our theoretical contributions. It is divided into four chapters 6, 7, 8 and 9. Each chapter present our contributions within a specific research area.

- *Chapter 6* introduces the proposed logic-based approach for describing and modelling complex biomolecular networks following systems theory: the structural, functional and behavioural aspects. This efficient formalism aims to represent the dynamic behaviour of biomolecular networks. Then, we explain this proposed approach with a concrete case study clarifying how this technique can be used in practice.
- *Chapter 7* details a proposed semantic approach based on four ontologies to provide a rich description for modelling biomolecular networks and their state changes. Moreover, we detail our development of the Biomolecular Network Ontology (BNO) which formalizes the domain knowledge of complex biomolecular networks and then applies the method to an application case study. This semantic approach provides the necessary concepts for modelling the dynamic behaviour and the transition states of complex biomolecular networks.
- *Chapter 8* presents two approaches for simulating complex biomolecular networks. The first one consists of a method of qualitative simulation based on the formal logical modelling (presented in Chapter 6) that qualitatively simulate the biomolecular network and interpret it behaviour over time. The second proposed approach is inspired by the Discrete Event System Specification formalism (DEVS) to easily reproduce, analyse and understand the behaviour of complex biomolecular networks. The proposed simulation approaches have been applied to the same case study.
- *Chapter 9* introduces a multi-objective genetic algorithm-based method for optimizing the transittability of complex biomolecular networks considering various criteria such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of input signals, the minimization of the cost of these signals, the minimization of the number of target nodes, the minimization of patient discomfort.

- The third part III **Validation** presents, details and discusses our results.
  - *Chapter 10* presents a prototype that we have developed to validate our proposals as well as the experiments we have conducted to determine the performance of our prototype. We propose a simulation tool, so-called 'CBNSimulator', based on the logical model of the biomolecular network and taking advantage of the performance of a discrete-event simulation model for understanding the evolution and the behaviour of complex biomolecular networks.
  - *Chapter 11* discusses the results of various experiments that we have conducted in order to evaluate our contributions and compare the performance of our approaches with the literature researches.

- The **Conclusion** summarises the results of this thesis and proposes some future directions.

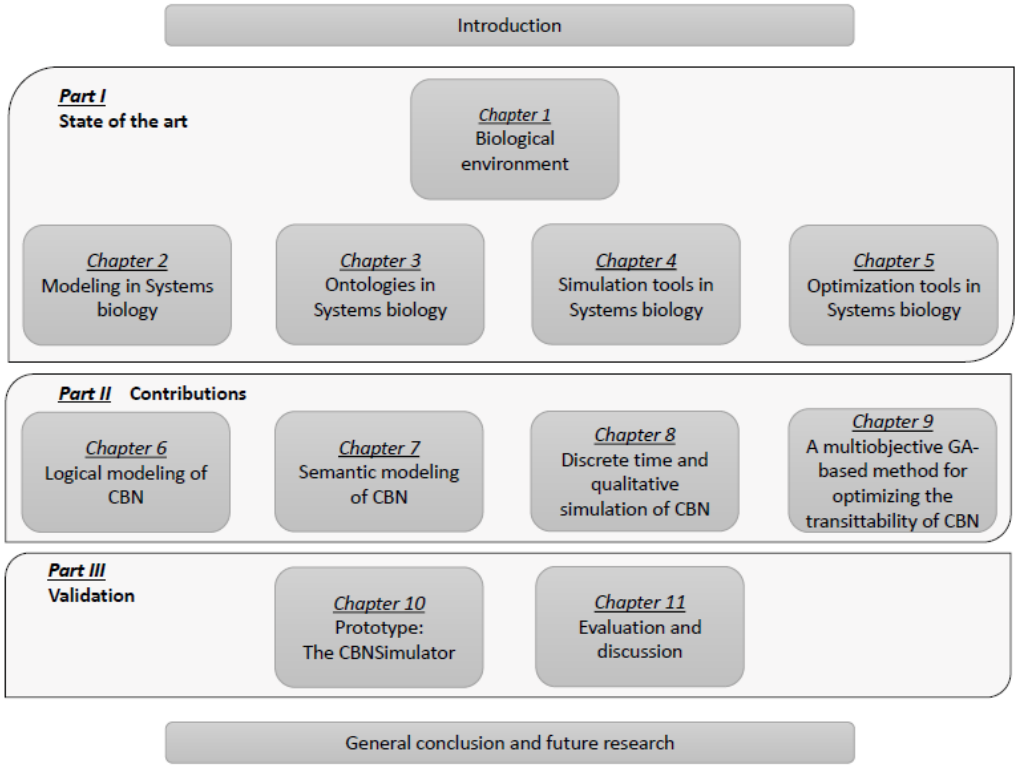


Figure 3 – The main structure of this thesis.

Part I

State-of-the-Art

This first part aims firstly to present the biological environment we are working in by exploring the conceptual history of systems biology and defining its main concepts. Then, secondly, to give an overview of the various tools and approaches that have been proposed in the different research fields covered by this thesis. At the end of each chapter, a section will be devoted to define the problem statement related to each particular field of research.

This part is divided into five chapters:

Chapter 1

Biological environment: from molecular biology to systems biology

Contents

1.1 Introduction

The objective of this chapter is first to describe the cellular system in which we are interested, its characteristics and its functions. Then, we will address the specific context that interests us, the 'transittability' of complex biomolecular networks which concerns the ability to steer this network from a specific state to another desired state [6].

This background chapter will start by providing a brief introduction to biology, such as the definition of the DNA, RNA, genes, chromosomes, proteins, metabolites and the gene expression from DNA to proteins. Then we highlight the rapid accumulation of biological data in recent decades and how they gave rise to systems biology. We then outline the goal of systems biology and the requirement for other methods for studying the behaviour of complex biomolecular networks and their components. We present also various types of molecular networks, in particular: the gene regulatory network (GRN), protein-protein interaction (PPI) network, and metabolic network (MN) in order to focus on their global properties and characteristics. And finally, we discuss some concepts related to the controllability of complex biological networks allowing to steer the dynamic network behaviour from a state to another one.

1.2 Biological background

1.2.1 Deoxyribonucleic acid (DNA)

The Deoxyribonucleic acid (DNA) was discovered by Frederich Miescher in 1869, then in 1953, James Watson determines its structure [30]. The DNA contains all the genetic information, called the *genome*, which enables the development, functioning and reproduction of living organisms [31]. This genetic information determines the role of different cells (in multicellular organisms), and all the mechanisms to survive and reproduce (in single-cell organisms). The DNA is composed of two polynucleotide chains composed of four units called nucleotides[1]. Each nucleotide includes phosphate, sugar and one of the four bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). As described in Figure 1.1, nucleotides are attached together to form two long strands creating a structure called a double helix [32]. Each helix is a polymer of nucleotides attached together by phosphodiester bonds and the two helices are connected together through hydrogen bonds. These bonds are formed by pairs of bases, considering that each base pair is composed of one purine base (A or G) and one pyrimidine base (C or T), matched according to these rules: G pairs with C, and A pairs with T. The total length of the human DNA is around 3 billion bases[2].

Physically, DNA is stored as a component of the sub-cellular structures called *chromosomes*, located in the nucleus in the Eukaryotic cell. The number of chromosomes varies among species. Humans have 22 pairs of chromosomes, plus the sex chromosomes.

A *gene* is a specific sequence of nucleotide bases along a chromosome containing information for the construction of proteins. A gene is divided into non-coding regions (introns) and coding regions (exons) [31]. All the DNA of a cell constitutes the **genome**.

1.2.2 Ribonucleic acid (RNA)

The Ribonucleic acid (RNA) is another type of nucleic acid that can also contain or transport genetic information. RNA can be found on both nucleus and cytoplasm in contrast to DNA which is only located in the nucleus of the cell [33]. As showed in Figure 1.1 and similarly to DNA, RNA is also built from purine and pyrimidine nucleotides (Uracil take the place of Thymine), but forms a single helices (unlike the DNA's double helix)[3]. Biologically, RNA molecules are produced as a result of gene transcription from one of the two helix of the DNA molecule and can be in one of three different types: messenger RNA (mRNA), transfer RNA (tRNA) or ribosomal RNA (rRNA) [33]. The messenger RNA is a chemically unstable molecule that is synthesized in the nucleus based on a single DNA strand using the RNA polymerase enzyme to carry the sequence information.

[1] https://www.ncbi.nlm.nih.gov/books/NBK26821/
[2] http://www.livescience.com/37247-dna.html
[3] https://www.nature.com/scitable/definition/ribonucleic-acid-rna-45

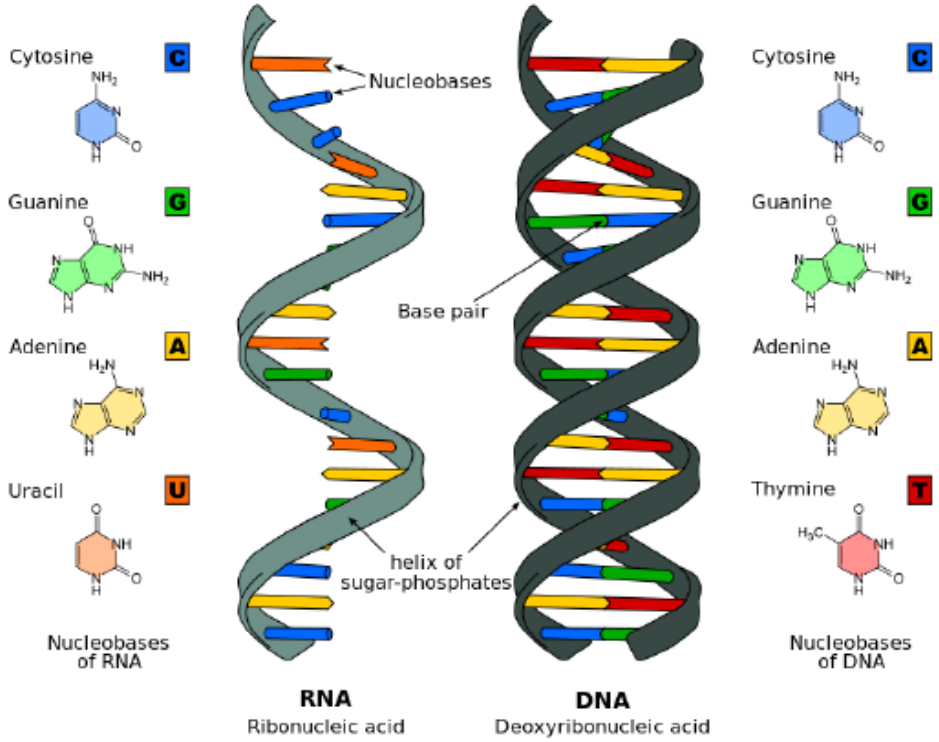


Figure 1.1 – DNA and RNA structure (Image credit: Wikimedia).

During the translation process, mRNA is edited by the elimination of introns via RNA splicing, and only exons are transported to the ribosome. The information in the mRNA is organized as a series of codons, each one containing three bases, then it is translated into a specific protein by the transfer RNA which is an RNA molecule acting as an adapter among amino acids and mRNA [34].

All the RNAs of a cell constitutes the **transcriptome**, and it is specific to each cell according to its identity, as well as its immediate needs.

1.2.3 Proteins

Proteins[4] perform a vast array of functions within organisms. As illustrated in Figure 1.2, a protein is a polypeptide or a macromolecule consisting of building blocks called *amino acids* attached together in a linear chain. Proteins have a complex structure and their functional diversity is largely due to the diversity of their three-dimensional structure[5]. We distinguish three main protein's roles [33]: (*i*) Cell structure and cell mobility, for example the muscles, are almost entirely composed of proteins, which allow its contraction. (*ii*) Recognition, signal detection and transmission of information, such as haemoglobin that is used to transport oxygen in the blood. And (*iii*) Chemical metabolism, indeed a specific class of proteins called enzymes increases these chemical reactions (catalysis, etc.).

The set of proteins in a cell constitutes the **proteome** which is the functional product of gene expression.

1.2.4 Metabolites

Metabolites[6] are the intermediate products of metabolic reactions catalyzed by diverse enzymes that naturally occur within cells. This term is used to describe small molecules. We distinguish two kinds

[4] https://www.khanacademy.org/science/biology/macromolecules/proteins-and-amino-acids/a/introduction-to-proteins-and-amino-acids

[5] https://www.nature.com/scitable/topicpage/protein-structure-14122136

[6] https://www.news-medical.net/life-sciences/What-are-Metabolites.aspx

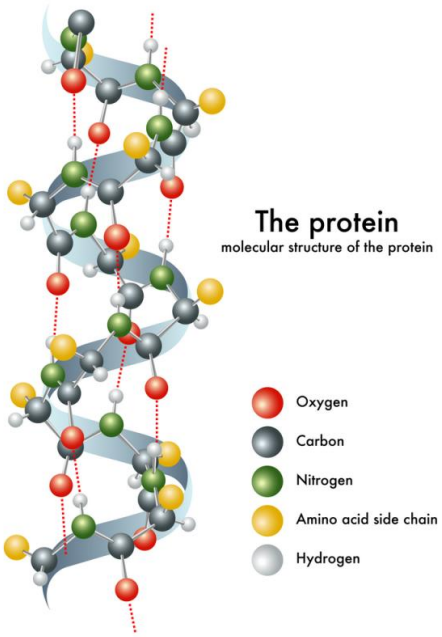


Figure 1.2 – Protein structure (Image credit: Wikimedia).

of metabolites: (*i*) The primary metabolites which are synthesized by the cell because they are indispensable for its growth, such as amino acids, alcohols, vitamins, organic acids, nucleotides (inosine-5'-monophosphate and guanosine-5'-monophosphate) [33]. And (*ii*) the secondary metabolites which are compounds produced by an organism that is not required for primary metabolic processes, although they can have other functions. As illustrated in Figure 1.3 displays the structure of some metabolites.

The set of primary and secondary metabolites constitutes the **metabolome**. Unlike the genome, the transcriptome and the proteome, the metabolome is not encoded.

We must also note that metabolites can be the reactants, products of the metabolic pathway which is a linked series of chemical reactions occurring within a cell.

1.2.5 Gene expression

As discussed previously, the DNA stores all the genomic information required for a cell to operate. Gene expression focuses on the study of the process by which the instructions in DNA are converted into a functional product which is called '*the central dogma of molecular biology*' [33, 35, 36, 37]. This clarifies the flow of genetic information from DNA to RNA to produce a protein. The process by which the DNA instructions are converted into proteins is called *gene expression*, which has two main steps: (*i*) the *transcription* and (*ii*) the *translation* [37].

In the transcription step, the information in the DNA of every cell is converted into small, portable mRNA. And during the translation step, these mRNA travel from the cell nucleus to the ribosomes where they are ready to make specific proteins.

As described in Figure 1.4, we distinguish three states of the central dogma:

- From existing DNA to make new DNA: **DNA replication phase**,
- From DNA to make new RNA: **transcription phase**,
- From RNA to make new proteins:**translation phase**.

The gene expression process starts with the *DNA replication* in which there is a production of identical DNA helices from a single double-stranded DNA molecule in order to ensure that each new cell receives the correct number of chromosomes. The gene expression has *transcription* as a second step [36]. Here, one of the DNA double helices serves as a template for the production of the RNA. In post-transcriptional

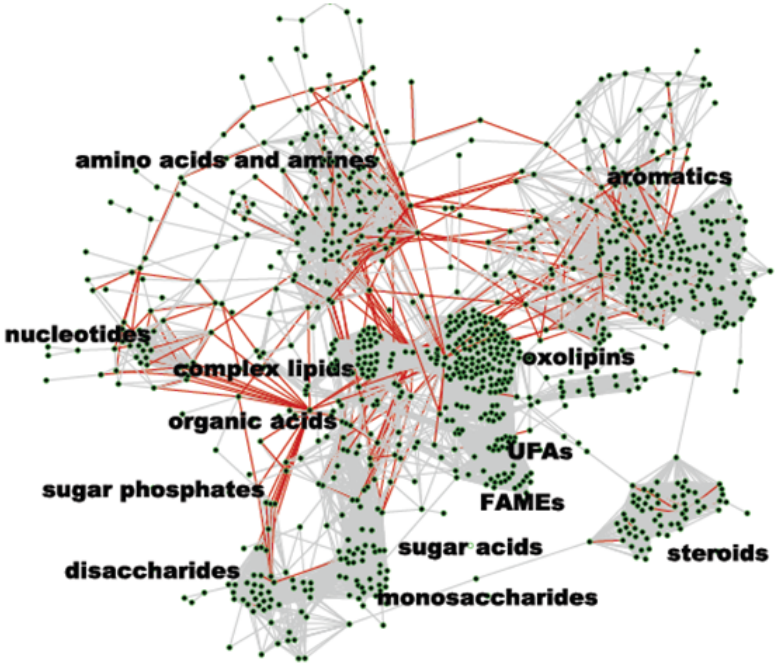


Figure 1.3 – Examples of metabolites (Image credit: the West Coast Metabolomics Center).

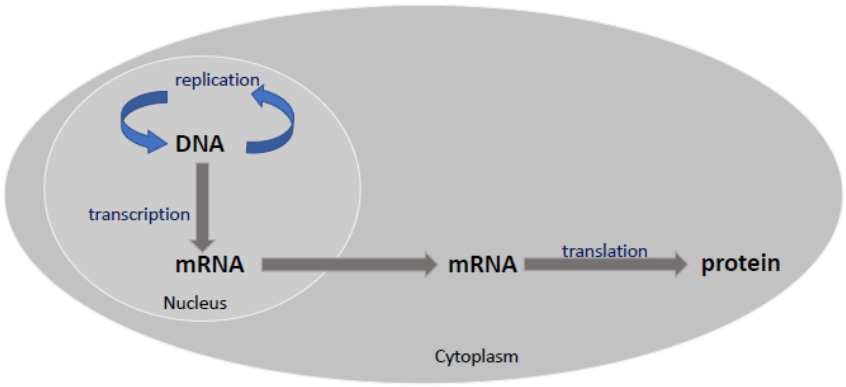


Figure 1.4 – The central drogma of life.

processing, a pre-mRNA is edited to contain only coding sections (exons) to form a mature mRNA script. The mRNA is then transmitted to the cytoplasm where it is matched to ribosomes for protein synthesis [33]. Finally, the tRNA attaches specific amino acids to the mRNA to form complete polypeptide chains: the proteins. Thus this translation phase ensures the conversion of the genetic information in DNA into proteins.

Therefore, **genome**, **transcriptome**, **proteome** and **metabolome** are the major molecular complexes of cells [38, 39]. A correlation among these four large sets can be explained by the expression of the genome defines a transcriptome, then a specific proteome determines the metabolome. In contrast, the metabolome regulates gene expression so that the cell permanently adapts its proteome to its metabolic state. However, it should not be forgotten that the control of the metabolome is only part of one of the proteome's functions, which also realize many other roles such as the communication of the cell with its environment or the structuring of the cell [33].

Merging proteomic, transcriptomic, and metabolomic information to facilitate the study of cellular behaviour, is among the major aims of systems biology.

1.3 From molecular biology to systems biology

In the 20^{th} century, there has been an important revolution of molecular biology. This advance is due to the explosion of high-throughput technologies which generate huge amounts of molecular-level data. These so-called 'omics' (genomic, transcriptomic, proteomic and metabolomic) techniques aimed primarily at the detection and the study of genes (total gene expression analysis [40]), RNAs (RNA-Seq [41]), proteins (mass spectrometry [42]) and metabolites (liquid chromatography [43]) in a specific biological sample [44]. However, despite these advances, the molecular biology of the 20th century has remained fragmented and incomplete. Indeed, each laboratory focus only on a particular phenomenon concerning a cellular type of a specific organ in an environment. This limitation is due to the inability of classical biological approaches to address biological systems as wholes and thus to confront their complexity. This complexity results from the heterogeneity and diversity of the components involved, the dynamic nature of the interactions among these components, and the non-linear nature of the behaviour resulting from these interactions.

According to Sauer et al. in [45]: '*The reductionist approach has successfully identified most of the components and many of the interactions but, unfortunately, offers no convincing concepts or methods to understand how system properties emerge ... the pluralism of causes and effects in biological networks is better addressed by observing, through quantitative measures, multiple components simultaneously and by rigorous data integration with mathematical models*'.

To fill these gaps, the area of systems biology was introduced to complete classical biological approaches. Systems biology is an approach that addresses the complexity of biological systems and their dynamic behaviour at all relevant organizational levels (from molecules, cells and organs to organisms). It combines reducing and integrative methods to emphasise both the components of the system and the interactions among them which generate emergence phenomena at higher organizational levels.

In contrast to classical biology, this field is based on the understanding that the whole is greater than the sum of the parts[7]. It approaches the complexity of biological systems with the integration of many scientific disciplines such: biology, computer science, engineering, bioinformatics, physics and others, to predict how these systems change over time and under diverse conditions.

It is now clear in the minds of all biologists that understanding cellular behaviour requires analysis of its dynamic interactions (its evolution), its multi-variate data (measurements of millions of molecules and multiple parameters) and its multi-level data (from genome to metabolome).

1.4 Complex biomolecular networks

As discussed in the previous section, with the rapid accumulation of omics-data from high-throughput technologies, the study of biomolecular networks has become one of the keys focuses in systems biology. Indeed, high-throughput technologies pave the way for the reconstructions of different biomolecular networks according to molecular-level defined by omics data. Figure 1.5 depicts the different types of biological networks according to molecular components using high-throughput omics technologies. In this section, we review these different types of molecular networks and define the complex biomolecular network.

Therefore, depending on the type of its cellular elements and their interactions, we can distinguish the three basic types of networks, the Gene Regulatory networks (GRNs), the Protein-Protein Interaction networks (PPINs), and the Metabolic networks (MNs).

- The Gene Regulatory networks (GRNs) describe the interactions among approximately 21,000 genes (DNAs and RNAs). They are represented as directed graphs where the nodes represent genes and edges model the type of regulation (activation or inhibition) if one gene regulates the transcription of the other gene [46].
- The Protein-Protein-Interaction networks (PPINs) model the interactions among proteins within an organism (about 80,000 proteins in the human organism). These networks are represented as undirected graphs where the nodes are the proteins and the undirected edges model the connection between them. These types of interactions depend on the physical or biochemical interaction that exists between the pair of proteins [47]. This network mainly contains details on how proteins perform together to ensure the biological processes.

[7] `https://www.systemsbiology.org/about/what-is-systems-biology/`

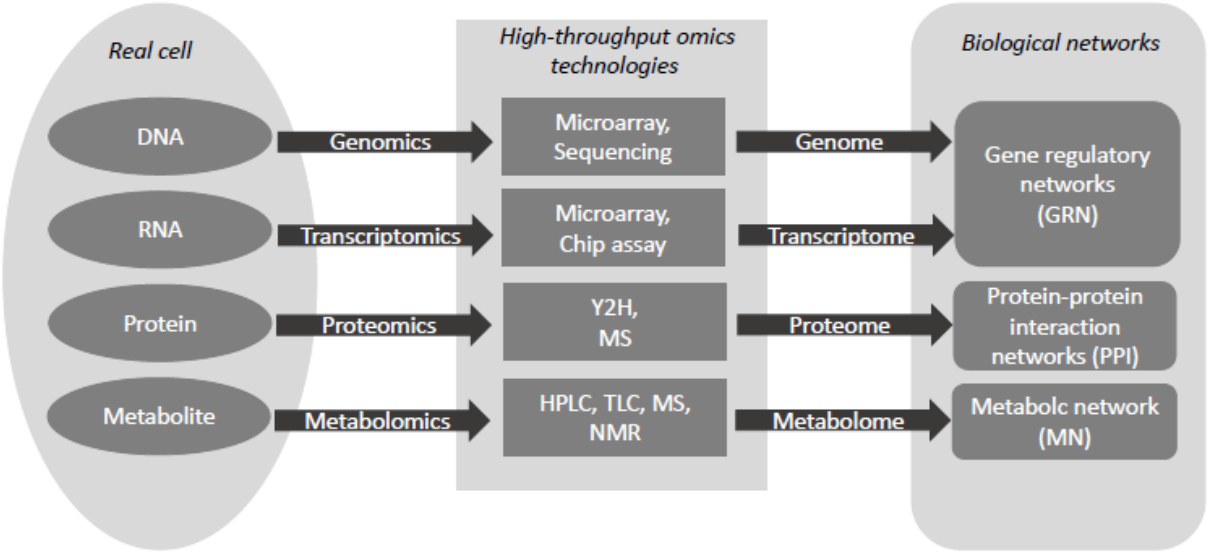


Figure 1.5 – Types of biological networks according to molecular components using high-throughput omics technologies.

- The metabolic process consists of a series of chemical reactions that begins with a particular metabolite called 'substrate' and converts it into some other metabolites called 'products' [48]. Thus, the Metabolic networks (MNs) describe the biochemical reactions among approximately 42,000 metabolites. They are represented as directed graphs whose nodes are the metabolites and the edges represent the type of the biochemical reaction which transforms the substrates into products by the help of enzymes. They are labelled by the stoichiometric coefficient of the metabolites in the reaction.

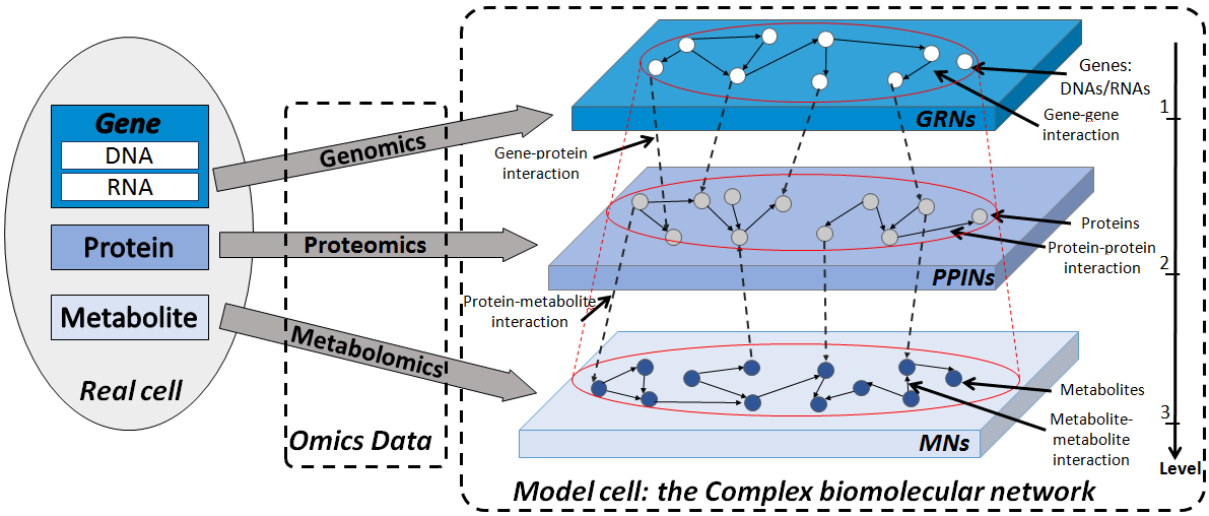


Figure 1.6 – Multi-level modelling of a biomolecular network from a real cell.

As discussed above, the cell is a complex system consisting of thousands of diverse molecular entities (genes, proteins and metabolites) which interact with each other physically, functionally and logically creating a biomolecular network [6, 14]. Indeed, omics technologies describe the cell networks and processes through the study of cellular entities. These technologies operate at various levels such as in genomics (the qualitative study of genes), in proteomics (the quantitative study of proteins) and in metabolomics (the quantitative study of metabolites) [49, 50]. Thus, the complexity of this complex biomolecular network

appears by its decomposition into the three levels presented above: the genome, transcriptome, proteome and metabolome which are the major molecular complexes composing the cell [38, 39]. As illustrated in Figure 1.6, complex biomolecular networks typically include gene regulatory networks, protein-protein interaction networks and metabolic networks. It consists of different types of nodes, denoting cellular entities, and various nature of edges, representing interactions among cellular components.

This complex network facilitates the understanding of biological mechanisms of a cell and its transittability.

1.5 Transittability of complex biomolecular networks

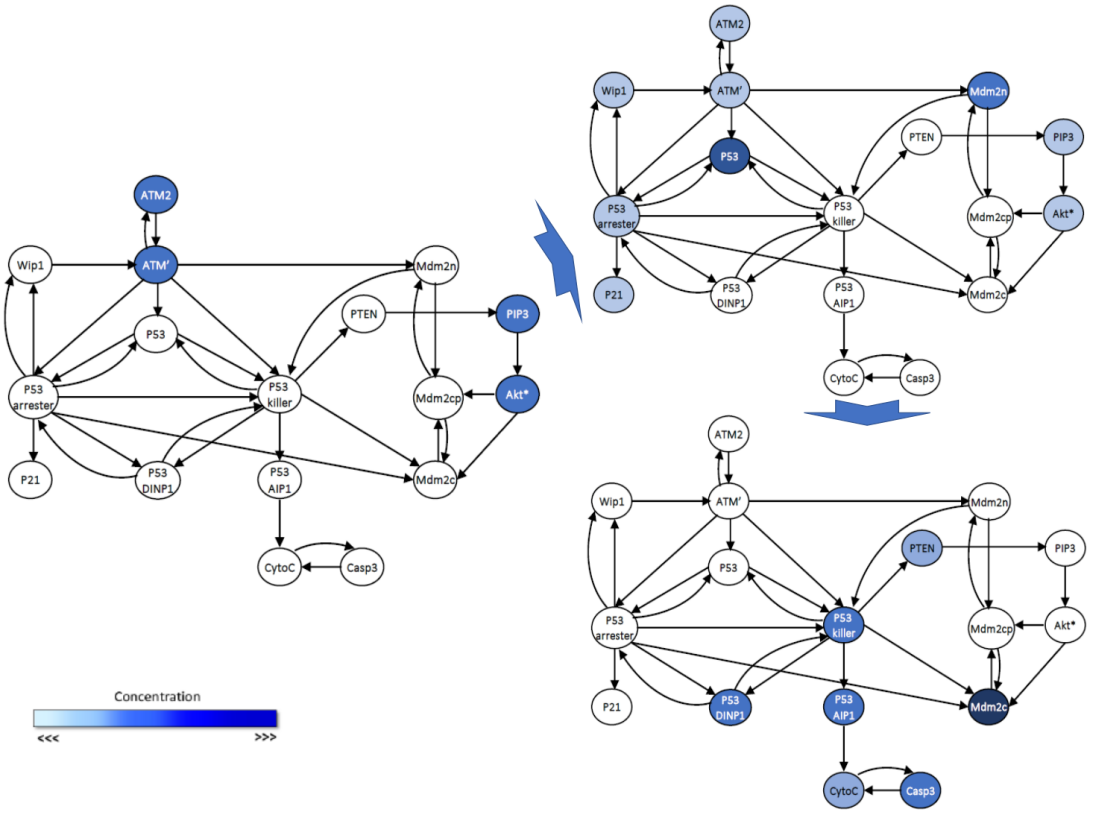


Figure 1.7 – The transittability of the P53-mediated cell damage response network: colour changes in the nodes indicate changes in the concentration of the associated molecules.

Cells do not live in stable conditions, but in environments that vary over time [9]. In fact, they are always subjected to intra and extra-cellular stimuli, such as changes in their physical and chemical properties or in their environment. In order to survive, the cell reacts more or less rapidly by adapting its behaviour in accordance with the new features of its environment. As discussed in the previous section, biomolecular components interact with each other to form so-called biomolecular networks, which determine the cellular behaviours of living organisms. Indeed, controlling the cellular behaviours by regulating some biomolecular components in the network is one of the most outstanding problems in systems biology.

Recently, some authors have started to address the dynamic aspects of biological systems, and have introduced concepts such as the 'controllability' [28, 51] of a network, where the ability to steer a complex directed network from any initial state toward any other desired state is measured by the minimum number of required driver nodes (nodes with the ability to steer the entire network). It has been shown that in order to achieve complete controllability, the minimum number of driver nodes is 80% of the nodes in a regulatory biomolecular network. This result led other groups to develop a theoretic framework for studying transitions between two specific states of directed complex networks, a concept they call

'transittability' [6]. In general, this concept expresses the idea of steering the complex biomolecular network from an unexpected state to a desired state. The theorems were developed with continuous-time linear time-invariant systems (a theory investigating the response of a linear and time-invariant system to an arbitrary input signal) and applied to 4 different biological systems consisting of up to 17 molecules and 40 interactions. Figure 1.7 shows an example of a biological system. At each stage of this network, the blue-filled node indicates that the level of this node concentration at that state, a blue-filled node means high level and an empty node means low level. For more details about the components of the P53-mediated cell damage response network, please refer to the TP53 website at `http://p53.fr/` which contains extensive information on various aspects on p53 and also links to other p53 sites. Moreover, this example is described in detail in [6].

With the idea of making the biomolecular network evolve, our research works aim to design and develop a platform that provides an optimal set of external stimuli to be applied during a predetermined time interval to steer the biomolecular network from its current state to a desired state. Our original approach is based on the cooperation of semantic technologies [52], combinatorial optimization and simulation.

1.6 Summary

In this chapter, we presented the biological environment we are working in by defining its main concepts such as DNA, ARN, proteins, metabolites and gene expression. Then we give a brief history of biology area describing its revolution from the classical molecular biology to the appearance of systems biology. We discussed also how omics technologies operate at various levels (genomics, proteomics, metabolomics) creating complex biomolecular networks. Finally, we discussed some concepts which are introduced to study the dynamic behaviour of complex biomolecular networks, such as the 'controllability' and 'transittability', some of which have inspired the work presented in this thesis.

In the next chapters, we will present the state-of-the-art of the different disciplines covered by this thesis.

Chapter 2

Modelling in systems biology

Contents

2.1 Introduction

Numerous formalisms have been proposed in the literature for describing and studying biological networks. Indeed, modelling these complex systems is the first step towards understanding, designing, simulating and controlling the behaviour of these biological networks [53]. Therefore we can use different kinds of models: discrete, continuous or hybrid models for the same biological network. This classification is done according to the nature of the mathematical formalisms used (Boolean, logic, differential equations, etc.) [54]. Thus, each model focuses on a specific problem and treats a well-defined level of intracellular abstraction.

This chapter aims to classify and detail these mathematical tools for modelling biological networks. We firstly introduce a classification of modelling tools based on their major properties and then highlight some of the most popular and important mathematical models in systems biology. Finally, we provide a comparison among these modelling formalisms and discuss their main characteristics.

2.2 Major properties and dimensions of modelling

With the explosion of high-throughput technologies which generated huge amounts of molecular-level omic data, it becomes necessary to provide models that can be used to understand and formalize knowledge about them. Therefore several mathematical models have been proposed in the literature. These models are very varied from the point of view of abstraction levels (from the gene level to the metabolic level through proteins), details of analysis (discrete, continuous), etc. A number of bibliographic reviews focused on these modelling tools. Most of them classify these models according to different criteria. In the following section, we will focus on these categories and detail some of the most popular models.

2.2.1 Discrete vs Continuous vs Hybrid models

2.2.1.1 Discrete models

These methods consider a molecular component as an object that can steer from a state to another and represent the biological interactions among these entities as Boolean functions [55]. Discrete models simplify the calculation and offer the possibility of realizing formal checks. However, despite their simplicity to model and simulate specific biological questions, they are too complex when the user focuses on an important number of states. Moreover, modelling a molecular component by a Boolean value do not take into account the reality because it can not have intermediate states [56].

2.2.1.2 Continuous models

In order to understand in detail the functioning of a biological system (especially from a temporal point of view), this approach focuses on its continuous evolution. The variables represent the concentrations of molecular components comprising the system, and the global evolution is guided by a system of differential equations. This approach can be close to the real-life but is difficult to implement when the size of the system is high. This complexity is due to the non-linear character of the differential equations. To address this limits, an intermediate approach called hybrid model was proposed.

2.2.1.3 Hybrid models

They are dynamical models that combine the advantages of both discrete and continuous models. The principle is to mix between the classic differential equations representing the evolution of chemical concentrations, with cellular automata or agent-based models representing the individual behaviour of some molecular components of interest. Thus, a hybrid model corresponds to any interaction or coupling between two or more models that are not based on the same formalism [57]. More details about hybrid modelling in biology are presented in a review *Hybrid Modelling in Biology: a Classification Review* [57] written by Stéphanou and Volpert.

2.2.2 Quantitative vs Qualitative models

2.2.2.1 Quantitative models:

They are usually represented as a set of differential equations given that they are a suitable approach for modelling real systems. These models require all the parameters to be defined, but given the large size of biological systems, it is not possible to list all of them, moreover, their values are not yet known.

2.2.2.2 Qualitative models:

Unlike quantitative models, these models provide general descriptions of biological systems because they require few or no parameters. These models are usually representing as discrete or hybrid models.

A detailed comparison study between quantitative and qualitative models can be found in [58].

2.3 Overview of the existing mathematical models in systems biology

Biological networks are extremely complex. To understand their operation, we not only need to identify their molecular components (genes, proteins and metabolites) and their interactions but also we need to know how their dynamics evolve over time [59].

This section highlights some of the most popular and important modelling tools.

2.3.1 Boolean models

Boolean network models were initially proposed as a special case of discrete dynamic models. A Boolean network model consists of a set of nodes whose state is binary (*0* or *1*) and is determined by other nodes in the network through Boolean functions. These Boolean functions are expressed together with the logical operators: *AND*, *OR*, and *NOT*. In terms of complexity, Boolean networks lie between static network models and continuous dynamic models, making them attractable and powerful approaches to model large-level biological systems. Boolean models can be used to describe the qualitative temporal behaviour of the system and to understand how perturbations change its behaviours. They also provide a coherent network representation.

This Boolean model has been applied to the modelling of several genetic regulation networks of diverse organisms such as: the differentiation of floral organs in Arabidopsis thaliana plants [60], the embryonic development of Drosophila [61], the mammalian cerebral cortex [62], the process of apoptosis (cell death) [63], and signalling networks in a variety of biological systems such: the ABA Signal Transduction network [64], the Embryonic stem cells (ES cells) [65].

2.3.2 Logical models

The logical approach was initially developed by René Thomas and his collaborators [66], then it was introduced in biology by Kauffman [18, 56]. This model represents a compromise between the static model (structural analysis) and differential equations in terms of complexity and precision [67]. In a logical graph, the nodes represent genes which are associated with discrete levels of expression, and edges represent the interactions among genes. Each interaction is associated with an expression level threshold from which the regulator (starting node of the interaction) has an effect on the target gene. For each gene, a discrete logical function indicates to which qualitative level the gene tends when he is submitted to a given combination of interactions. In their simplest form, logic models associate to each molecular component one of this two discrete states: *ON* or *OFF*. Therefore, logic models with only two binary states are generally considered as Boolean models.

Logic models have been applied to the modelling of several signal transduction pathways involved in diverse processes such as: proliferation [68], cell cycle regulation [69], apoptosis [70], or differentiation [71]. A detailed survey of logic models can be found in [72].

2.3.3 Petri nets models

Petri net is a graphical and mathematical approach for modelling systems in which the notion of events and evolution is important. Initially, this model was developed by Petri in 1962 [73] to represent discrete-event systems. Originally, only systems that changed discreetly could be represented with Petri networks. Then, many extensions were produced (Wim Bos list the main extensions in [74]). Some of them are really dedicated to biological system modelling, such as the hybrid Petri net networks which are used to model the evolution of systems in which some variables evolve discreetly, others continuously, and whose rates of change depending on the system's variables. In general, a Petri net is a directed, weighted bipartite graph consisting of two types of nodes: *places* and *transitions*. In this network, circles represent the places and boxes represent the transitions. We usually consider a token to be a unit of weight of a molecule. A place-transition or transition-place connection is made by a weighted edge representing how much of the input places (reactants) are required to produce tokens for the output places (products) in a reaction. A transition can only fire when it is enabled, meaning that each of its input places has at least one token in the current marking. More formally, a Petri network consists of five elements:

- $P = \{p_1, p_2, \ldots, p_m\}$ a set of places,
- $T = \{t_1, t_2, \ldots, t_n\}$ a set of transitions,
- $F \subseteq (P * T) \cup (T * P)$ a set of edges,
- $W : F \to \{1, 2, 3, \ldots\}$ a weight function,
- $M_0 : P \to \{0, 1, 2, 3, \ldots\}$ the initial marking.

Petri net was mainly used to model signalling pathway networks [75]. This signaling Petri net aims to predict signal flow through a cell-specific network in experimental conditions [75]. Each place is devoted to a signaling protein, and each transition is associated with a phosphorylation interaction.

A number of research studies have been focused on Petri net to model biological systems such as for modelling and validation of the sucrose breakdown pathway in the potato tuber [76], the enzymatic reaction chains [77], and the biochemical reaction systems [78]. More details on approaches using Petri net for modelling biological systems can be found in dedicated survey [79, 80, 81].

2.3.4 Bayesian network models

The Bayesian models were initially introduced into biology issues in order to infer regulatory networks from DNA chips by Friedman et al. in [82]. Then, Microsoft's research contributed to the development of approaches based on Bayesian models [83, 84]. In a Bayesian model, the variables can be discrete or continuous. Bayesian models are static in nature, but they are easily extensible to dynamic modelling problems [85].

This graphical formalism defines and simplifies a joint law of probabilities of a model. It is, therefore, a graphical and probabilistic model. The variables have a probability distribution conditioned by the state of other variables in the model. These variables represented by nodes are associated with conditional probability distributions. The edges indicate a statistical dependence between two variables [86]. Bayesian network model consists of three major steps: (*i*) the structure must be proposed, (*ii*) the parameters or probabilities associated with edges and nodes must be set, and (*iii*) the final network must be evaluated.

Bayesian networks have been applied for inferring the structure of many biological networks from experimental data. Among these applications, we can cite the Characterizing Loss Of Cell Cycle Synchrony (CLOCCS) [87], the comparison of Acute Lymphoblastic Leukemia (ALL) [88].

2.3.5 Graphical Gaussian models

Graphical Gaussian models also called covariance selection or concentration graph models, have been used to model gene association networks. The main objective of Graphical Gaussian models is to use partial correlations as a measure of independence of any two genes. This model relies on assessing the conditional dependencies among genes in terms of partial correlation coefficients among the gene expressions and results are displayed in an undirected network [89]. A detailed introduction to Graphical Gaussian models can be found in [90] and [91].

Graphical Gaussian models have been applied for inferring the genetic network of many biological systems such as the genetic network of S. cerevisiae [92], the Arabidopsis thaliana transcriptome [93], the isoprenoid gene network in Arabidopsis thaliana [94].

2.3.6 Differential equation models

Among dynamic modelling methods, the most commonly used in biology is the differential description. Concentrations or activities of molecular components are usually represented by positive real quantities called also variables which can vary continuously over time. The variation of these quantities is formalized by writing a system of coupled differential equations. Thus, molecular components and their interactions constitute a system of ordinary differential equations. In most biological systems, the interactions considered are non-linear which leads to differential models that are almost impossible to solve and analyse. Consequently, this model requires the use of 'numerical simulations'. Starting from initial constraints and conditions, this model tries to have a solution similar to the exact solution by calculating the values of the concentrations of molecular components involved over time, using small time intervals. Moreover, this approach requires to define the values for all parameters. Since these values are not always established experimentally, simulations are carried out with very approximate or arbitrary values. Therefore, it is difficult to predict the dynamic behaviour of the biological system. Thus, this method can be used to model small systems where it is possible to know in advance the exact values of their parameters, but for more complex biological models, it is difficult to specify all the corresponding dynamic properties and associated conditions.

During the last decades, several kinds of researches have been proposed to describe the regulation of gene expression using a differential equation model. Among them, we can cite [95, 96, 97, 98, 99, 100]. Diláo et al. [101] developed the 'GeneticNetworks' which is a software tool, for modelling genetic networks using linear differential equations. Other authors have proposed and simulated differential models for molecular networks, for example in the gene expression regulation in bacteria [102], cell-cycle control [103], Circadian rhythms [104], specific gene expression profiles during embryonic development [105, 106, 107] and response of Escherichia coli cells to carbon deprivation [108]. Moreover, differential equations models have been used to model metabolic networks, for example for analysing the central metabolism of an environmental bacterium: the Methylobacterium extorquens in [109] or in Corynebacterium glutamicum [110]. Differential equation models are also applied to protein-protein interaction networks (only from the perspective of studying signal transduction networks) as studying feedback effects on signal dynamics in a Mitogen-Activated Protein Kinase (MAPK) [111], controlling signal transduction cycles [112], and understanding the CovR/S signal transduction system in [113].

2.3.7 Cellular automata models

Cellular Automata model is a discrete dynamical formalism. Within this model the notions of space, time, and states of the system are discrete. Each point in the spatial network is called a cell which may have any one of the finite numbers of states. The states of the cells in the network are updated according to a local rule. Consequently, the state of a cell at a given time t depends only on its state at time $t-1$ and the states of its neighbours at time $t-1$, and all the cells on the network are updated synchronously. Using cellular automata it is possible to model diverse systems. The first works on cellular automata were made by Von-Neumann in [114]. Subsequently, this model was detailed by Stephen Wolfram in the 1980s [115].

Application of cellular automata models can be seen in the modelling of the immune system [116], the development of an artificial brain [117], the enzymatic reaction [118] and some other biological systems [119].

2.3.8 Agent-based models

Agent-based model is a rule-based, discrete-event and discrete-time computational modelling methodology that employs computational objects focusing on the rules and interactions among the individual components (agents) of the system [120, 121, 122].

An agent is an interactive computer system situated in an environment and able to autonomous action in this environment in order to meet its design objectives. A multi-agent system consists of a set of agents

interacting in a dynamic environment as defined by Michael Wooldridge in [123].

Application of agent-based models can be seen in the modelling of the control pathways affecting the transcription factor nuclear factor kappa B (NF-kB) [124], the behaviour of the toll-like receptor 4 (TLR-4) signalling pathway [125] and the Pseudomonas aeruginosa Biofilm Formation [126].

2.4 Comparison among these modelling formalisms

As detailed above, biological systems and in particular intracellular components have been modelled by diverse modelling approaches. This diversity is essentially due to the complexity and heterogeneity of the molecular components and their interactions [54]. After analysing each modelling approach, we should note that each model has its own benefits and drawbacks, and can be used according to the user's objective.

Indeed, there is no perfect approach to model a complex biomolecular network considering all its levels and their molecular components. Moreover, we note that there is no agreement on the classification of these models because we cannot compare them on the basis of a particular criterion. The same model can be used in different ways with different interpretations of biology. But, we note that these models differ in many other criteria including how to interpret biological facts. In order to highlight their properties and their utility, we compare them according to various criteria. The classification proposed here aims to illustrate the major characteristics of each approach for modelling biological systems. Table 2.1 summarises the main features of these models.

For example, Boolean models have the capacity to simplify the dynamics of gene networks which enable an efficient analysis of large-scale networks. However, it ignores the intermediate states of gene expression and it can update the gene in an asynchronous way which may miss many important dynamic behaviours.

The power of Petri net models is their capacity to provide an intuitive representation of a biological system due to their graph-based structure which is suitable for analysing the global behaviour of the system. However, the graphical representation becomes too complex for analysis when the biological network is large because it requires to define a lot of transition rules.

Bayesian networks provide a graphical formalism defining a joint law of probabilities of a model which is easily extensible. However, they are computationally expensive because it needs to analyse all potential network topologies corresponding to all possible sets of directed acyclic graphs linking the molecular components. This causes a combinatorial explosion of the number of possible structures and parameters creating an NP problem.

Graphical Gaussian models provide a powerful model to represent statistical dependencies among random variables. However, they suffer from poor scalability and interpretability.

Differential equation models have the capacity to model and analyse the dynamics of the system. This model provides a result similar to the exact and real solution. However, the power of this model can only be applied to a small-scale system. In fact, despite the evolution of high-throughput technologies, there is always a lack of data. Moreover, if the network scale is large, parameter estimation will generate a high computational cost. Indeed, the simulation of large-scale networks using differential equations cannot be performed because of the complexity of the problem and the high number of network components. Also, the quantitative parameters of biological systems cannot be experimentally measured. This leads to creating more difficulties in setting the inial conditions of the network simulation. We note that differential equation models are more precise and specific about a system, but require a large effort in model construction and a complete set of quantitative data.

Agent-based models can be constructed in the lack of complete knowledge about the system using simple rules. Also, these models easily incorporate space because they have their origins in two-dimensional cellular automata. Agent-based models incorporate stochastic aspects. However, they do not allow to find patterns in an existing dataset. Moreover, they require a lot of interactions among each agent and they need frequent communication.

Table 2.1 classifies these modelling methods according to their properties.

Table 2.1 – Table of the main approaches applied to modelling biological networks.

Model	*Differential equation*	*Boolean, Logic models*	*Petri nets network*	*Graphical Gaussian*	*Bayesian network*	*Agent based model*	*Cellular automata*
Analysis type	quantitative	qualitative	qualitative	quantitative	qualitative	qualitative	qualitative
Category	continuous	discrete, stochastic	discrete, continuous	stochastic, continuous	discrete, continuous	discrete, continuous	discrete, continuous
Advantages	quantitative dynamics, suitable for modelling small-scale networks, analytical models	simplify the dynamics of gene network, efficient analysis of large network	representation of the concentration flow, intuitive and hierarchically modelling	computational efficiency, represent rich statistical dependencies among random variables	intuitive representation of biochemical networks due to their graph-based structure	representation of cells having a specific behaviour, incorporate stochastic aspects	space representation, easy to understand results
Limits	need for quantitative data, high computational cost	ignore the intermediate states of gene expression, miss many important dynamic behaviours	number of unbounded tokens in basic semantics, inability to test for exactly a specific marking in an unbounded place and to take action on the outcome of the test	poor scalability and interpretability, difficult to identify high-level interactions between functional variables	high computational cost, an NP-Complete problem	lack of analytical methods, requires a lot of interactions among each agent	lack of analytical methods

2.5 Thesis contribution in this field

Several studies have been conducted to model, analyse and understand the behaviour and processes of cells. However, most of them do not examine the interactions among all the intervening molecules types and do not consider the different abstraction level within the cell. As a result, these modelling approaches are impractical to understand the transittability of complex biomolecular networks. Therefore, to accomplish this task, we must take into account the analysis of the structure and dynamics of the whole cell rather than just focusing on isolated parts [24, 25, 26, 27].

With this idea in mind, we think that a formal logic-based formalism is needed for modelling and understanding complex biomolecular networks as a whole considering its structure, function and behaviour. Indeed, we aim to design and develop a logic formalism for biologist that allows (*i*) the formal expression of various types of biological knowledge, (*ii*) the translation of this knowledge into logical notions for analysis, simulation and optimization, (iii) the integration of the different levels of intracellular abstraction (genomic, proteomic and metabolic). Therefore, this model will be considered as a well-organized knowledge base that contains all the available information for modelling a biological network in a logic, clear and consistent manner.

It is important to note that this logic-based model can be considered as a logical model. Logical models represent one of the youngest approaches to model and formalize the multi-level aspect of biomolecular networks underlying their different level properties. Indeed, this model is capable of modelling the biomolecular network processes and functions on different temporal and spatial levels in an explicit way. Moreover, this multi-level model can be a good framework for the integration of additional computational module, in our case it serves to integrate the logic modelling with the semantic modelling module, the simulation module and the optimization module. All of these modules are based and joined together using this logic-based model. Using this type of models, we can also integrate discrete with continuous modelling, thereby opening the possibility to simulate modules that can be best represented with discrete values and modules best described using continuous equations. Furthermore, this model offers a higher expressive power due to its formal language. Indeed, in these logical models, there is a class of well defined mathematical structure in which the domain knowledge (all the elements required for modelling biomolecular networks) is interpreted.

2.6 Summary

In this chapter, we focused only on some popular mathematical modelling tools. Many literature reviews detailed these modelling approaches and are available in [26, 127, 128]. In fact, different modelling approaches are used in systems biology and we cannot say that a model is better than other. As detailed above, some models are too small systems, others to large systems, and some of them are qualitative, others are quantitative. Each of these models focuses on a specific problem and treats a well-defined level of intracellular abstraction. These models can also be applied to the same biological network to study different problems with different point of views. However, all these models are not adequate and do not allow to understand the transittability of complex biomolecular networks. In fact, they are more specific and oriented to a particular problem in systems biology. Therefore, with the goal of understanding and analysing the transittability of biomolecular networks, we need to implement a more specific model which can provide all the elements required for studying this problem. That is why we hope to contribute to this discipline by proposing a logical-based approach for modelling the dynamic behaviour of biomolecular networks. This formalism is based on the three levels of analysis defined by the systems theory: structural, functional and behavioural modelling. Indeed, it aims at describing and analysing all the properties and mechanisms of complex biomolecular networks. This logic-based modelling will form the basic element for modelling and understanding the transittability of these complex networks.

Despite existing mathematical modelling tools for formalizing biological networks, there are also semantic technologies which are useful to provide a powerful integration and management of available biological data and knowledge, thus making them more understandable to biologists, enabling an efficient analysis of biological networks and reasoning. The next chapter is devoted to exploring the role of semantic technologies, especially ontologies in systems biology.

Chapter 3

Ontologies in systems biology

Contents

3.1 Introduction

Modelling knowledge requires semantic structures and representation formalisms that translate the complexity of human thought and describe characteristics of the real word. Actually, it is the field of semantic technologies and more particularly ontologies that address these issues. An ontology is considered as a cognitive artefact allowing the conceptualization and shared exploitation of knowledge. It consists of a vocabulary of the field in which the meaning of the terms and the relationships between the different notions are specified. Beyond its descriptive ability, ontologies provide reasoning capabilities.

In this chapter, we present a complete state-of-the-art on ontologies. First, we give some background information about ontology by presenting its different definitions, components and typologies. Next, we describe the methodologies, formalisms, languages and editor tools of ontology construction. We then define the different languages and software for ontology reasoning. Moreover, we review the major applications of ontologies in systems biology through a comparison study. Finally, we discuss the contribution of our thesis to both ontology and systems biology field.

3.2 Concept of Ontology

An ontology has two different definitions depending on the field of interest, philosophy or computer science:

In philosophy In philosophy, the term ontology '*is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations*' [1].

In computer science In the field of Artificial Intelligence (AI) the most cited definition of ontology is the definition proposed by Gruber [129]: '*An ontology is an explicit specification of a conceptualization*'. Other definitions of ontology for ontological engineering have been proposed and are sometimes improvement of already proposed definitions or are complementary to them. For example, Studer et al. [130] defines an ontology as '*a formal, explicit specification of shared conceptualization*'. Moreover, Pierra et al. in [131] define an ontology as '*a collection of explicit, formal and consensual descriptions of all the concepts of a domain in the context where these concepts have a precise meaning, without any restriction or rule corresponding to a particular use*'.

These proposed definitions gave emphasis to the notions of *specification* and *conceptualization*. The concept of specification is a formal description of how an object should be defined to satisfy a specific criterion. Therefore, a specification must be explicit because all the concepts of an ontology must be clearly defined. The second concept of conceptualization can be defined as an intentional semantic structure that encodes implicit knowledge constraining the structure of a piece of a domain. It is usually a logical theory that conceptualization explicitly in some language.

The following section presents the basic components of an ontology.

3.3 Ontology components

Ontologies represent the semantics of a domain's concepts in terms of *classes* and *properties*. A class also called concept regroups and abstracts the domain objects having common characteristics. A property called also *attribute* characterizes the objects of the domain by one or more values. The property can be defined on a domain indicating the class of objects it describes and associated with a codomain indicating the type of data in which it can take its values. Classes have an extension consisting of a set of *instances* called also *individuals* that indicate objects in the domain. An instance belongs to one or more classes and a set of property values.

Pierra et al. [131] defines formally an ontology by this quadruplet:

$$O =< C, P, Sub, Applic >$$

where:

[1] `https://www.merriam-webster.com/dictionary/ontology`

- C: represents all the *classes* used to describe the concepts of a given domain,
- P: represents the set of *properties* used to describe the instances of all the classes C,
- *Sub*: $C \rightarrow 2^C$ is the *subsumption* relation which, for each class c_i of the ontology, associates its direct subsumed classes. These classes check the property $\forall c_1, c_2 \in C$, c_1 subsume c_2 if and only if $\forall x \in c_2, x \in c_1$. Sub defines a partial order on C and 2^C denotes the power set of C,
- *Applic*: $C \rightarrow 2^P$ associates to each ontology class the properties that are applicable for each instance of this class.

3.4 Typologies of ontologies

Ontologies can be classified according to several domains [132],[133], [134] depending on:

- the object of generality,
- the level of detail,
- the level of completeness,
- the level of representation formalism.

Each domain contains different kinds of ontologies.

3.4.1 According to the object of generality

The ontology types are classified into four groups according to the level of generality used in the description of a domain [135].

- *Top-level ontologies*: These ontologies aim to define the largest possible knowledge because they contain general concepts. They are reusable among different domains and reduce the ambiguity of the basis of the ontology. In these ontologies, it is rare to find individuals because they aim only to propose a hierarchy of knowledge. An example of this type of ontologies in biology is the GENIA ontology [136] which propose a general description of the basic ontological entities in the life sciences domain.
- *Domain ontologies*: They are a specification of a high-level ontology. These ontologies specify particular domains and are linked to high-level ontologies by one or more high-level ontology concepts.
- *Task ontologies*: They are used to describe concepts for solving specific activity problems. They contain terms and properties that describe problem solutions. These ontologies are independent of the domain.
- *Application ontologies*: A more precise level is specified in the application ontologies. They describe precisely the specific activities of an application domain. It is possible to consider these ontologies as an association between task and domain ontologies.

3.4.2 According to the level of detail

According to the level of detail and the precision of the ontology, we distinguish two kinds of ontologies [137]:

- *Large granularity*: This type consists of high-level ontologies having a broad granularity because the concepts they contain will be refined later in other domain or application ontologies.
- *Fine granularity*: In contrast to the previous type, these ontologies are very detailed and have a richer vocabulary providing a detailed description of the relevant concepts of a domain or task.

3.4.3 According to the level of formality

According to the used language or formalisms of representation, we distinguish four kinds of ontologies [138]:

- *Informal ontologies*: they are expressed in natural language and are easily understandable by the users.
- *Semi-informal ontologies*: they are expressed in a more structured and limited.langage.
- *Semi-formal ontologies*: they are expressed in an artificial language.
- *Formal ontologies*: they are expressed in an artificial language with a formal semantics that allows verification.

These different types of ontologies are used according to the field of application and the purpose of the ontology.

3.5 Ontology building: methodologies, formalisms, languages and tools

Over the years, the use of ontologies in knowledge engineering has known different methodologies, formalisms, languages and tools [139]. In the following sections, we will detail each of them.

3.5.1 Ontology engineering methodologies

There are several detailed reviews for ontology engineering methodologies such as the works of Fernández-López Mariano [140]. Here, we briefly review the literature by citing only some of the major methods such as the Uschold and King's method, SENSUS method, METHONDOLOGY method and Stanford's method.

3.5.1.1 Uschold and King's method

This method proposed by Mike Uschold and Martin King in [141] consists of four steps:

- *Identifying the objectives and context of ontology*: this step aims to define why this ontology will be developed, for what purpose and who will use this ontology.
- *Ontology development*: this step is divided into three activities: (*i*) identifying the concepts and relationships among them, (*ii*) ontology coding using a language such OWL, RDF, etc., and (*iii*) integration and reuse of existing ontologies.
- *Ontology evaluation*: this step consists in checking and evaluating the ontology.
- *Documentation of the ontology*: this step comments and documents the code of the ontology in order to facilitate its maintenance.

This method is one of the first proposed in the ontology engineering literature. Its general steps are considered as the basis of any ontology construction.

3.5.1.2 SENSUS method

The SENSUS method [142] proposes to develop a domain ontology from a top-level ontology. It proposes to define the relationships among the specific and general terms of the domain, then to delete the terms that are not specific to the domain of the ontology. This method aims to re-use the terms of existing ontologies. The Sensus ontology was developed based on the SENSUS method and contains more than 50000 concepts hierarchically organized.

3.5.1.3 METHONTOLOGY method

This method [143] follows several steps:

- Identification of the ontology development process,
- Lifecycle based on the evolution of prototypes,
- Project management, development and support activities.

The METHONTOLOGY follows the project management techniques in which the ontology is considered as a finality of the method. In this method, five development steps are defined:

- *Specification*: What is the purpose of the ontology, who are its users and what is its size?
- *Conceptualization*: This is the most important step in the construction. This step treats the organisation of knowledge from defining the candidate terms to define the instances.
- *Formalization*: This step translates knowledge into an ontology.
- *Implementation*: This step translates the ontology into a specific ontology language such as OWL, RDF, etc.
- *Maintenance*: This step corrects and updates the ontology.

3.5.1.4 The Stanford's method

The Standford's method [144] has been developed by Stanford University. It is divided into seven steps:

- Determine the domain and the scope of the ontology,
- Reuse existing ontologies,
- List important ontology terms,
- Define classes and hierarchy,
- Define the class properties and their attributes,
- Define the facets (restrictions or constraints) of attributes,
- Create instances of classes in the hierarchy.

These different steps are implemented through a set of questions, for example concerning the first step, 'what will ontology cover?', 'What is the purpose of the ontology?', 'What is the kind of questions that the ontology has to answer?', 'Who will use and manage the maintenance of the ontology?'.

This is the method used in our research because it appeared clear and rigorous. Moreover, this method is linked to the Protégé editor.

3.5.2 Types of formalisms

There are diverse types of formalisms to model an ontology and different languages for each type.

- Graphs:
  - Topic Maps[2], Semantic Networks[3]
  - Web-oriented: RDF and RDF Diagram
- Logic:
  - First order: KIF
  - Description: KL-One, OIL, DAML+OIL, OWL
- Object orientation: UML + OCL

In order to understand the structure of ontologies and the power they can provide to the world of knowledge representation, we present in the following sections some of these languages.

[2] `http://www.topicmaps.org/`
[3] `http://intelligence.worldofcomputing.net/knowledge-representation/semantic-nets.html`

3.5.3 Languages

An ontology language is a formal language used to encode ontologies. A number of ontology languages have been developed during the past few years by the research community [145], such as common Algebraic specification language, common logic, CycL, DOGMA, Gellish, IDEF5, KIF, RIF, and OWL. Ontology languages can be classified into three categories (*i*) logical languages, (*ii*) frame based languages and (*iii*) graph-based languages. The following sections detail some major languages.

3.5.3.1 KIF

Knowledge Interchange Format (KIF) [146] is a language based on first-order predicates with extensions to model definitions and meta-knowledge. As described in its website[4], KIF has declarative semantics and is logically comprehensive. The Ontolingua tool[5] allows users to build KIF ontologies at a higher level of description by importing definitions of predefined ontologies.

3.5.3.2 KL-ONE

KL-ONE [147] is a language based on description logic [148]. It is a formalization of knowledge representation based on frames [146]. This formalism divides its descriptions into two basic classes of concepts: primitive and defined. Primitives are domain concepts that are not fully defined[6].In fact, new terms can be defined using operations of concept conjunction, for example, the '*and*' operator is used to specify that a new concept is a common specialization of several other concepts. Moreover, new roles can be introduced to represent the properties that exist among individuals in the modelled domain. Definitions of concepts include restrictions on possible values, the number of values, or the type of values a role can have for a concept.

3.5.3.3 RDF and RDF Schema

Resource Description Framework (RDF) RDF[7] is a graphical formalism for modelling and describing meta-data. RDF is based on the notion of the triplet (subject, predicate and object).

- *Subject (resource)*: it is an information entity that can be referenced by an identifier. This identifier must be a URI (Universal Resource Identifiers).
- *Predicate (property)*: it is a specific aspect, an attribute, a characteristic or a relationship used to describe a resource.
- *Object (value)*: it is a literal (single string) or a resource.

The subject and the object are resources linked together by the predicate. RDF uses XML syntax, but it gives no specific meaning for vocabulary as 'a subclass of' or 'type'. The modelling primitives provided by RDF are basic and limited.

Resource Description Framework Schema (RDFS) RDF Schema[8] is a language that complete RDF with a vocabulary of terms and relationships such as: rdfs:Class, rdfs:Property, rdfs:type, rdfs:subClassOf, rdfs:subPropertyOf, rdfs:range and rdfs:domain. RDFS is recognized as an ontology language since it allows to organize resources hierarchically using subsumption links (rdfs:subClassOf, rdfs:subPropertyOf), to specify constraints on property values (rdfs:domain, rdfs:range). Thus, the specific classes of a domain will be defined as instances of the Resource Class and its properties as instances of the Resource Property. Then, the notion of hierarchy (of classes or properties) will be realized using the properties subClassOf or subPropertyOf.

[4] www-ksl.stanford.edu/knowldge-sharing/kif/
[5] http://ksl-web.stanford.edu/kst/ontolingua.html
[6] https://en.wikipedia.org/wiki/KL-ONE
[7] https://www.w3.org/RDF/
[8] https://www.w3.org/TR/rdf-schema/

RDF and RDFS limits While RDF and RDFS have been designed to be as generic as possible, this simplicity of language is not enough to describe complex situations. For example, it is impossible to define that two classes are disjoint or to define cardinality restrictions [149]. In order to address these issues, the W3C proposed a new and more expressive language: the Web Ontology Language.

3.5.3.4 DAML-ONT

DARPA Agent Markup Language (DAML-ONT) [150] is based on XML and RDF. DAML-ONT has been developed in October 2000 by the Defense Advanced Research Projects Agency (DARPA) to propose a more advanced expression of RDF classes.

3.5.3.5 DAML + OIL

A number of work has been done in the field of knowledge representation, among them we can cite the most important such as: Simple HTML Ontology Extensions (SHOE)[9], OntoBroker[10], Ontology Inference Layer (OIL)[11], and DAML + OIL[12] which replaced DAML-ONT[13]. DAML + OIL is a language based on previous Semantic Web Working Group (W3C) standards such as RDF and RDF Schema. It completes these languages with richer modelling primitives. DAML+OIL was developed using the ontology language DAML-ONT in order to combine several components of the OIL language. OIL is a web-based representation and an inference layer for ontologies. It combines the primitives of frame-based language with the formal semantics and reasoning provided by description logic.

3.5.3.6 OWL

Web Ontology Language (OWL)[14] was developed in 2004 by the Semantic Web Working Group (W3C) in order to explicitly represent the meanings of vocabulary terms and the relationships among them. OWL also aims to make the web-based resources easily accessible to automated processes [151] by structuring them in an understandable and standardized way, and by adding them with meta-information. To do this, OWL has more powerful functionalities to express meaning and semantics than XML, RDF, and RDF Schema [152]. In addition, OWL takes into account the diffused nature of knowledge sources and allows information to be collected from distributed sources [153].

- *Why OWL?*

XML[15] provides a syntax for structured documents but does not impose semantic constraints on the meaning of documents. RDF is a data model to represent objects (resources) and relationships among them. This model provides a simple semantics that can be represented in XML syntax. RDFS is a vocabulary definition language for describing properties and classes represented by RDF resources. RDFS defines graphs of RDF triplets with a semantics of generalization or hierarchization of these properties and classes. OWL adds vocabularies for the description of properties and classes, relationships among classes, cardinalities, characteristics of properties, and enumerated classes. OWL is developed as an extension of the RDF vocabulary and is derived from the ontology language DAML + OIL. Therefore, OWL shares several features in common with RDF, RDFS and XML.

- *Sub-languages of OWL*

OWL consists of three expressive sub-languages: OWL Lite, OWL DL, and OWL Full.

- *OWL Lite*: supports users who mainly need a classification hierarchy and simple constraints. It is a restriction of OWL DL. OWL Lite only supports a subset of OWL language constructions and is easy to implement.

[9] https://www.cs.umd.edu/projects/plus/SHOE/onts/
[10] http://www.semafora-systems.com/en/products/ontobroker/
[11] http://xml.coverpages.org/oil.html
[12] https://www.w3.org/TR/daml+oil-reference
[13] http://www.daml.org/2000/10/daml-ont.html
[14] https://www.w3.org/OWL/
[15] https://www.w3.org/XML/

- *OWL DL*: uses the description logic (DL) [154]. It supports users who need maximum expressiveness while retaining computer completeness and the possibility of decision making. It includes all the constructions of the OWL language which can only be used under certain restrictions.
- *OWL Full*: is the entire language and uses all the above OWL primitives. It allows free mixing of the OWL with RDF Schema and does not enforce a strict separation of classes, properties, individuals, and data values. OWL Full is designed for maximal RDF compatibility and is, therefore, the natural place to start for RDF users.

3.5.3.7 OCL

Object Constraint Language (OCL)[16] is a language that allows users to write expressions and constraints on object-oriented models. OCL allows expressing two types of constraints on the state of an object or a set of objects.

3.5.4 Editing tools

Many ontology building tools (editing and visualization) use various formalisms and offer different functionalities. All of these tools offer support for the ontology construction process but few of them offer conceptualization assistance. Among these tools we can cite:

- The *Protégé* editor[17] has been developed at Stanford University [136] and it was currently the most used editor for ontology. Initially based on the frame model [155], the current version of Protégé allows the development of ontologies according to the OWL ontology model. Protégé offers all the necessary functionalities for editing the diverse elements of an OWL ontology (concepts, properties, instances, etc.). Moreover, it offers the ability to specify constraints and use external reasoners and rule engines such as Pellet[18], Fact++[19], Hermit[20], etc., to check the consistency of the ontology and infer new knowledge. Protégé is enriched by the contributions of the users and developers community thanks to its architecture based on plugins that allow extending its functionalities. Actually, this editor is able to integrate several ontologies and manage different versions of the same ontology.
- *OntoEdit*[21] was developed by the Knowledge Management Group of the University of Karlsruhe. This editor provides a graphical environment for inspection, navigation, coding and modification of an ontology. OntoEdit provides functional subsets to export an ontology according to diverse representation languages (XML, Flogic, RDFS, DAML + OIL). In its commercial version, OntoEdit is part of the software suite proposed by Ontoprise.
- *PLibEditor*[22] allows the development of ontologies according to the PLIB ontology model. This editor is based on an ontological database architecture (OntoDB) [156] which allows storing the ontology model, the ontology and its instances. PLIBEditor can also manipulate OWL ontologies. In fact, in its last version presented in [157], OntoDB allows to store both PLIB and OWL ontology models and ensures a transformation between these two models. PLIBEditor is also able to check the content of an ontology or a set of ontological-based instances and reason on them. This editor can also import and export ontologies and their instances in the standardized PLIB exchange format.
- *Ontolingua*[23] is a server situated at Stanford University and allows a user, or a group of users, to visualize existing ontologies and develop new ontologies through a standard web browser. This tool offers different functionalities such as the reuse (by merging or extension) of existing ontologies in different domains, the real-time collaboration of a geographically distributed group to develop an ontology, export of ontologies in different formats in order to use them in diverse applications.

[16] http://www.omg.org/spec/OCL/2.0/About-OCL/
[17] https://protege.stanford.edu/
[18] https://www.w3.org/2001/sw/wiki/Pellet
[19] http://owl.man.ac.uk/factplusplus/
[20] http://www.hermit-reasoner.com/
[21] http://www.semafora-systems.com/
[22] http://www.plib.ensma.fr/
[23] http://www.ksl.stanford.edu/software/ontolingua/

- Unlike Protégé, OntoEdit and PLibEditor, which are more interested in the formal representation of the ontology concepts, the *Differential Ontology Editor DOE* editor[24] privileges informal description to describe concepts more precisely. This editor uses a 'differential semantics' to annotate the generalization or specialization hierarchies by applying a number of rules detailed in [158].

Here, we have detailed only the most popular ontology editors but there are others in the literature such as KAD-Office[25], SWOOP[26], etc.

3.6 Ontology reasoning

3.6.1 Semantic Web Rule Language

Semantic Web Rule Language (SWRL)[27] is a standard language proposed by the W3C. This language combines OWL DL and the Rule Markup Language (RuleML) [159]. SWRL retains the expressivity of OWL DL and rules from RuleML.

3.6.2 SWRL sytax

Rules in the SWRL language are implication rules. Hence, the syntax of SWRL is in this form [160]:

$$antecedent \rightarrow consequent$$

This syntax implies that the consequent must be true when the antecedent is satisfied. OWL expressions can occur in both antecedent and consequent [161]. Both the antecedent and consequent consist of zero or more atoms. An empty antecedent is treated as true (satisfied by every interpretation), so the consequent must also be satisfied by every interpretation, however, an empty consequent is treated as false (not satisfied by any interpretation), so the antecedent must also not satisfied by any interpretation. Multiple atoms are treated as a conjunction written: $atom_1 \wedge atom_2 \wedge \ldots \wedge atom_n$.

3.6.3 Reasoning systems for description logic

A number of software systems have been implemented to reason on various description logics. Among these reasoners we can cite CEL[28], Fact++[29], fuzzyDL[30], KAON[31], SPASS/MSPASS[32], Pellet[33], QuOnto[34], RacerPro[35], etc.

3.7 Overview of existing ontology applications in systems biology

Bioinformatics is the study of information content and information flow in biological systems and processes [162]. With the explosion of biological data generated by high-throughput technologies, bioinformatics has grown rapidly in the past two decades to deliver software and applications for assisting expert biologists in their works [163].

As detailed in Chapter 2, over the last decades' new omics technologies have emerged and revolutionized biological researches producing an accumulation of data and knowledge about molecular mechanisms in cells. All these data were stored in heterogeneous and various sources of data. In this way, diverse data sources have been developed to allow researchers to share and reuse these data in the life sciences [164]. However, the diversity of these data sources induce the propagation of misinformation. These data

[24] http://www.eurecom.fr/~troncy/DOE/
[25] https://www.topincs.com/SemanticPLM/1435
[26] http://semanticweb.org/wiki/Swoop.html
[27] https://www.w3.org/Submission/SWRL/
[28] http://lat.inf.tu-dresden.de/systems/cel/
[29] http://owl.man.ac.uk/factplusplus/
[30] http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html
[31] http://kaon2.semanticweb.org/
[32] http://www.cs.man.ac.uk/~schmidt/mspass/
[33] http://pellet.owldl.com/
[34] http://www.dis.uniroma1.it/~quonto/
[35] http://www.racer-systems.com/

integration problems gave rise to semantic web technologies, especially ontologies which may be used as a unifying framework to solve these problems. In particular, ontologies are used in a wide range of systems biology. Moreover, with the creation of the National Center for Biomedical Ontology (NCBO) in 2006 [165, 166, 167], an incredible amount of ontologies has emerged in the Open Biological and Biomedical Ontologies (OBO) Foundry[36] providing a large variety of bio-ontologies. By the exploration of these bio-ontologies via browsers such the Ontology Lookup Service[37] and the BioPortal[38], it is remarked that these ontologies treat different parts of systems biology such as cell types [168, 169], molecular functions [170], experimental data analysis [171], etc.

Among these bio-ontologies, we can count the popular Gene Ontology (GO) [170] which aims to formalize knowledge about biological processes, molecular functions and cell components. The Cell Ontology (CO) [168] which provides a rich vocabulary for cell types. The Protein Ontology (PO) [172] which provides an ontological representation of protein-related entities by explicitly defining them and showing the relationships among them. The Systems Biology Ontology (SBO) [173] which is a set of controlled vocabularies of terms commonly used in Systems Biology, and in particular in computational modelling. The Biological Pathway Exchange (BioPAX) [174] which is an ontology that defines biological pathway data, such as metabolic pathways or molecular interactions. It works based on the mathematical formalism CellML[39]. The Anatomical Entity Ontology (AEO) [175] which provides a detailed classification for tissues and organs. The Mammalian Phenotype Ontology (MPO) [176] which provides a classification of phenotypic information related to the mouse and other mammalian species. The Phenotype and Trait Ontology (PATO) [177] which defines composite phenotypes and phenotype annotation. The Human Phenotype Ontology (HPO) [178] which provides a standardized vocabulary of phenotypic abnormalities encountered in human disease. These ontologies (MP, PATO, and HPO) were originally designed for the reporting of phenotypes.

3.8 Comparison among these bio-ontologies

As presented in Table 3.1 these bio-ontologies differ in the type of knowledge they describe, their intended purpose and their level of abstraction. Although there are several promising bio-ontologies in the systems biology domain, until now and to the best of our knowledge, there is no ontology for modelling the behaviour of complex biomolecular networks. In fact, very few researches use ontologies for defining the possible biological functions, like signal transducer activity in the case of the GO [170], or the Cell Behaviour Ontology (CBO) [179] which describes and focuses on cell and tissue biology.

3.9 Thesis contribution in this field

As was discussed, current ontologies for systems biology domain do not focus on the description of the biomolecular network's transittability. In fact, there is a lack of standard representation of complex biomolecular network's components and their interactions which are the basics of the transittability notion. As was shown in the previous chapters, these entities are complex and have several kinds of interactions among them. So, developing an ontology to formally define this concrete domain is more than necessary. This will be the topic of one of our contributions (Chapter 7), in which we propose a new ontology for the representation of this domain.

3.10 Summary

This chapter has reviewed ontology by presenting the definition of the basic ontology's concepts (components, types, ontologies development, etc.). It also describes the important literature and applications of ontologies in systems biology highlighting their important role in cell classification, phenotype descriptions, etc. This chapter discusses also issues and limits of these bio-ontologies, especially for modelling

[36] http://www.obofoundry.org/
[37] http://www.ebi.ac.uk/ols/index
[38] http://bioportal.bioontology.org/
[39] https://www.cellml.org/

Table 3.1 – Description of some popular biological ontologies.

Biological ontology name	*Content*	*Reference*	*Link*
Cell Ontology (OBO)	Cellular types	[168]	`http://obofoundry.org/ontology/cl.html`
Gene Ontology (GO)	Biological process	[170]	`http://www.geneontology.org/`
Protein Ontology (PO)	Protein entities	[172]	`http://pir.georgetown.edu/pro/`
Systems Biology Ontology (SBO)	Systems biology nomenclature	[173]	`http://www.ebi.ac.uk/sbo/main/`
Biological Pathway Exchange (BioPAX)	Molecular entities	[174]	`http://www.biopax.org/`
Anatomical Entity Ontology (AEO)	Cell classification	[175]	`http://bioportal.bioontology.org/ontologies/AEO`
Mammalian Phenotype Ontology (MPO)	Phenotypic qualities	[176]	`https://www.ebi.ac.uk/ols/ontologies/mp`
Phenotype and Trait Ontology (PATO)	Phenotypic qualities	[177]	`http://agroportal.lirmm.fr/ontologies/PATO`
Human Phenotype Ontology (HPO)	Phenotypic qualities	[178]	`http://human-phenotype-ontology.github.io/`
Cell Behavior Ontology (CBO)	Cellular behaviour	[179]	`http://cbo.biocomplexity.indiana.edu/cbo/`

the behaviour of complex biomolecular networks, and suggests some approaches that might be useful in the future.

Before detailing the proposed semantic modelling of biomolecular networks, the following chapter will review background on simulation and optimization methods in systems biology.

Chapter 4

Simulation tools in systems biology

Contents

4.1 Introduction

Simulation of biomolecular networks is essential for studying biological systems from small reaction networks to large sets of cells [180]. Models are useful for many different purposes, among them testing theoretical hypotheses, guiding experiments and looking at experimentally unreachable scenarios, as well as the ability to predict new behaviours of these complex networks.

This chapter focuses on the notion of simulation with emphasis on its use in biological systems. The first section presents the principles and characteristics of the simulation theory. The second section reviews the most popular simulation tools and platforms in systems biology through a comparison study. We will also discuss the issues and limits of these simulations and define the contribution of our thesis in this field.

4.2 Principles of simulation

4.2.1 Definition

The simulation is one of the most effective decision support tools available to designers and managers of complex systems. It consists in constructing a model of a real system and performing experiments on this model in order to understand the behaviour of this system and improve its performance.

Formally, the simulation is the imitation of a real-world process or the functioning of a system over time [181]. Thus, a simulation can be defined as a representation of the functioning of a system or process. Through simulation, a model is implanted with unlimited variations producing complex scenarios. These capabilities allow analysis and understanding of how individual elements interact and affect the simulated environment.

Advances in computer science have significantly increased the power of simulations. Computer simulations are often based on iterative methods for calculating the detailed state of an entire system at the instant $t+1$ depending on its state at the instant t. They are holistic because they simultaneously consider many properties of a system.

4.2.2 Relation between modelling and simulation concepts

The concept '*modelling*' has different meanings: (*i*) modelling refers to all the activities of creation, development and execution of virtual models of the system to be studied. (*ii*) Modelling merges with numerical simulation. (*iii*) Modelling is the development of relationships between the characteristic variables of a given system, able to simulate the behaviour of that system in a given context.

In our research, we follow the third definition. In order to study the behaviour of a complex system, it is essential to distinguish between two different phases: (*i*) the development of models and (*ii*) their use in a concrete study. Only the first phase is called *modelling* (which we have detailed in Chapter 2) and the second phase is called *simulation* (that is the main topic of this chapter). The use of models is not limited to the simulation. Thus, the simulation is one among several objectives of the modelling. The separation and distinction between the two phases contribute to the exchange of models and their application.

As discussed in Chapter 2, the modelling aims to conceptualize the real world to find a relevant representation of reality. This concept is developed through examples and generalizations based on logical reasoning. In fact, every object in the real world has not a universal definition and does not depend totally on a specific model. Everyone is able to provide a personal definition based on its ideas. Communication between humans is only possible through common and shared parts of a concept. Thus, the main objective of modelling is the formal description of this concept (usually expressed in mathematical formalism). In chapter 2, we have detailed these different categories of mathematical models.

On the other hand, the simulation phase consists of one possible exploitation of this modelling. This simulation is guided by the objectives of its study by applying simplifications on the real properties of the studied object. For a concrete system, the models developed during the modelling phase are used and applied to its problem. It is at this stage that the simulation of the modelled system is used to obtain results that are otherwise too difficult to obtain.

4.2.3 Uses of simulation

Simulation is used in research to study the behaviour of complex systems, or systems composed of multiple interdependent processes [182], and it can be used for a variety of research purposes [183] such as:

- *Prediction*: the simulation provides hypotheses about the future behaviour of a system or phenomena.
- *Proof*: the simulation confirms the theoretical knowledge.
- *Discovery*: the simulation is used to predict and discover new unexpected consequences or knowledge.
- *Explanation*: the simulation is used to clarify the behaviours which are observed but their reasons are unknown.
- *Critique*: the simulation can be used to explore, modify and correct the theoretical explanations for phenomena proposed by researchers.
- *Prescription*: the simulation can propose a better mode of operation or method of analysis.

4.2.4 Levels of abstraction

According to the aim of the simulation and the degree of detail of the behaviour analysis that the user wishes to simulate, diverse levels of abstraction are possible: *macro*, *meso* and *micro* levels.

- The *macro-simulations* are simulations in which the system is studied as a whole without considering local actions and interactions [184]. These simulations analyse global characteristics of the system. They are important when the real system is too complex and need to be divided into small elements.
- The *meso-simulations* do not study a system at a global or local level but use the aggregation of local elements and focus on the interactions among these aggregations [185]. These simulations constitute an intermediate level where local and global components are combined in the analysis of the system. They are used in the case of a system that is difficult to qualify and quantify at local levels or when the user does not require fine decomposition (when it is necessary to study the interactions among the components of the system).
- The *micro-simulations* have gradually emerged in diverse fields where scientists want to make progress in the detail, realism and understanding of system [186]. These simulations consider the system at its most local level.

In the case of meso-simulations and micro-simulations, the global behaviour is the implication of the most local levels and results from interactions among local components. Consequently, it is possible to combine these different levels of abstraction within the same simulation.

4.3 Overview of existing simulation tools in systems biology

There are two kinds of simulation tools based on the mathematical modelling approaches detailed in Chapter 2: *population-based* models and *individual-based* models (called also agent models).

Population-based models describe the general behaviour of the whole system by the aggregation of all the individual states and behaviours. These models are represented by a mathematical formalism (detailed in Chapter 2). They can not describe the behaviour of individual components independently of others. However, in order to simulate individual behaviours, it is necessary to use individual-based models. These models are described by rules applying to each individual in order to manage their behaviour. These rules are simple or complex.

4.3.1 Mathematical and population-based simulation

Based on different approaches for modelling biological systems, diverse platforms for the simulation of their properties have been developed.

These simulators started with the development of the BioProcess simulator (BPS) by Aspen Technology [187] in 1985. This simulator includes a number of operation modules specific for diverse biochemical processes such as the ultrafiltration, the chromatography, etc. Inspired by the BioProcess, other simulators were developed such as the BioPro designer developed by the Biotechnology Process Engineering Center at the Massachusetts Institute of Technology (MIT) and completed by INTELLIGENT [188], the Environmental Simulation Program (ESP) developed by OLI systems, and the GPS-X [189] developed by Hydromantis. All these simulators share the same characteristics of the BioProcess simulator and are mostly developed to simulate macroscopic behaviour such as cell development and are based on unstructured models.

Other simulation tools to understand biochemical pathways in the cell, using a structured model of the system have been published by diverse academic groups. We can cite: KINSIM [190], MetaModel [191], Jarnac/SCAMP [192], MIST [193], Gepasi [194], DBsolve [195], ProMot/Diva [196] and BioSPICE [197]. All these simulator tools aim to simulate metabolic networks based on differential equations. These simulations tools are basically designed to simulate structured, dynamic, and deterministic systems.

On the other hand, MetaFluxNet [198] has recently been developed to execute static simulations used for metabolic analysis. This simulator interacts with users through standard SQL commands for querying three databases: ENZYME [199], LIGAND [199] and EcoCyc [200]. Based on these three databases, MetaFluxNet allows users to develop metabolic models corresponding to their objectives.

Another type of mathematical modelling based on stochastic aspects is used for simulating gene expression. In this category, we can cite the STOCKS simulator which is a software for stochastic simulation based on the Gillespie algorithm [201]. This simulator was used for simulating the protein synthesis and mRNA levels of the LacZ gene of E. coli. Another simulator the StochSim [202] which is also based on a stochastic simulation algorithm and a simple two-dimensional spatial structure. StochSim was used to simulate the pathway controlling chemotaxis in E. coli [203].

4.3.2 Individual-based simulation

Among the popular individual-based models used for simulating biological systems, we have selected the Potts model, Lattice gas automata, Cellular automata and Multi-agent systems. There are several reviews that cover the uses of these models for simulating biological systems, three of them were used in the development of this section [204], [205].

4.3.2.1 Cellular Automata

This model was detailed in Chapter 2. Cellular automata were adopted in 1966 with the publication of Von Neumann's book 'Theory of self-reproducing automata' [206]. Then they were popularized by John Conway through the Game of Life which was described in an article by Martin Gardner [207]. Cell automata have been used to model diverse systems such as the developmental processes in biology [208, 209], the entire developmental lifecycle of the cellular slime mould Dictyostelium discoideum [210, 211], the cell growth and cell death [212], and the cell polarity [213]. A detailed survey containing more examples simulated with cellular automata can be found in [205].

4.3.2.2 Multi-Agent Systems

As already detailed in Chapter 2, multi-agent systems are based on the modelling of interacting agents. It is not necessary to model an environment because it is always modelled implicitly in the form of an agent. Each modelled agent have its own rules and there are several types of agents as detailed in Chapter 2. According to Grimm, multi-agent systems have two advantages. The first is characterised as 'pragmatic' and comes from the difficulty of studying complex behaviour using mathematical models. The second reason is called 'paradigmatic' and it occurs when mathematical models show their limits to simulate and explain the regulation, emergence, resilience, persistence, etc. in ecosystems [214]. However, there is no precise definition of a Multi-agent system and the evaluation of this model is delicate for lack of tools and methodology [215, 216]. There are many multi-agent model used to simulate systems in diverse

areas, such as ecology [217, 218], solving complex problems like the Travelling salesman problem [219]. In systems biology, works focus on protein relocations [220], the immune system [221] and cellular functions [222].

4.3.2.3 Potts model

The Potts model was developed to model and simulate the spin interactions in the crystalline networks [223]. This model is sometimes described as a modified cellular automaton. It consists of a regular matrix of cells and each cell have a value describing its state. The difference with the cellular automata is that the individual can cover several boxes (meshes) where an individual in the cellular automata can cover only one box (mesh). In the Potts model, an individual is represented by a region of the matrix and the boxes have a label that allows them to know to which individual they belong. The Potts model is related to and generalized by several other models including the XY model, the Heisenberg model and the N-vector model. This model was used to analyse the tissue reorganization by self-organization of cells according to the relationships among them [210, 224]. The Potts model is sometimes described as the Cellular Potts Model or CPM [204].

4.3.2.4 Lattice gas automata

The Lattice-Gas system is called also interacting Particle System. This model was proposed by Hardy, Pazzis and Pomeau who were interested in fluid mechanics [225]. The Lattice-Gas model became known in 1986 with the work of Frish, Hasslacher and Pomeau on the Navier-Stokes equation [226]. Lattice-Gas system consists of a regular matrix in which each box (mesh) has several sites, each site corresponds to one direction called a channel. In the description of this system, the commonly used term is not a box (mesh) but we talk about a node. A particle placed in this network moves from a node to another through the channels corresponding to its direction. Each particle takes the information concerning its velocity to know how many nodes it must move in steps of time. This allows the management of collisions among the particles. This notion of the node to node moving makes the Lattice-Gas system closer to cellular automata that's why many studies called it LGCA for Lattice-Gas Cellular Automata [226]. Collision management can be compared with the transition rules of a cellular automaton [205, 227]. A recent extension of the LGCA model considers continuous models instead of discrete entities and it is called the LBCA for Lattice Boltzmann Cellular Automata. This model was mainly used for the simulation of physical systems, such as fluid mechanics. Nevertheless, there are some interesting implementations of biological systems simulation in hemodynamics [228, 229] and the metastasis formation in cancer development [230].

4.3.3 Computational simulation platforms

4.3.3.1 Simulation standard

SBML

System Biology Markup Language (SBML)[1] is a modelling language project introduced by Hiroaki Kitano [48] in 2001 and derived from the XML standard. The SBML aims to standardize the biochemical process models. The first version was produced in 2001 [231] and published in 2003 [231]. Different versions of the platform appeared like SBML Level 3 Version 1 Core in 2010 [232]. Some software are able to edit models in SBML standards such as CellDesigner, V-Cell and Snoopy.

CellML

Similarly to the SBML, CellML[2] is also a language project deriving from the XML standard. This project is supported by the Bioengineering Institute of Auckland University. It was initially introduced by Waren Hedley and Melanie Nelson [233] in 2000 on the Physiome project designed to model organ functioning from the cellular level. Some software are able to edit models in the CellML formats such as OpenCell and V-Cell.

[1] `http://sbml.org/Main_Page`
[2] `https://www.cellml.org/`

4.3.3.2 Simulation tools

E-Cell

E-Cell[3] is a project started in 1996 at the Laboratory for Bioinformatics at Shonan-Fujisawa University in Keio in Japan, under the name of ECL for Electronic Cell Laboratory. Actually, it is in its third version. This platform was developed to model the biochemical processes of the cell. The objective of the platform's authors is to provide a model as precise as possible of a real cell and to describe the structural characteristics of the cell taking into account its physic and chemistry reactions. E-Cell requires models written in SBML or EML (E-Cell Model) languages. It has been used to model the mitochondrial metabolism [234], the red blood cell metabolism [235] and the circadian rhythm in Drosophila under the influence of light [236].

CompuCell3D

CompuCell3D[4] is available online in its version 3.7.2. This platform appeared in 2003 before being officially introduced by Lzaguirre et al. in 2004 [237]. It derives from the work of François Graner and James Glazier (CPM and Q-Potts) in 1992 [224, 238]. CompuCell3D is presented by its authors as a multi-model support because it is based on the parallel use of three components: a Potts model, a diffusion management module, and a module that manage the continuous mathematical models. CompuCell3D is developed for the study of morphogenesis and is based on the authors' expertise to model the cellular rearrangements directed by the forces acting on their surface, complemented by the consideration of cell genetics and the presence of chemical substances in the environment. This platform requires models written in CCML or CC3DML which are derived from the XML standard. CC3DML is an extension of the CCML language that allows the description of phenomena in 3 dimensions and CCML is limited to 2 dimensions. CompuCell3D has provided convincing results in morphogenesis by reproducing patterns characteristic of microbial evolution in [239], in embryogenesis by allowing simulation of the gastrulation process in the chicken embryo [240], and in the segmentation of the vertebrate embryo [241, 242]. Several examples of simulation done by this platform have been referenced by Maciej Swat [243].

SimCell

SimCell[5] is a model based on a variant of cellular automata called dynamic cellular automata proposed by Wishart et al. [244]. This dynamic cellular automaton consists of cellular automata in which a box of the cellular automata represents at most one macromolecule or a multitude of small molecules. The rest of cellular automata characteristics describes a regular two-dimensional cellular automaton with heterogeneous, probabilistic and synchronous functioning. This model is intended to be a general structure for simulating most cellular processes. SimCell has three steps: (*i*) create the network using the graphical user interface, (*ii*) simulate the behaviour of this network using the cellular automata, and (*iii*) generate at any time during the simulation, graphs and tables to present the evolution of the different entities involved in the network. SimCell uses colours to model species, however, it contains 70 colours, therefore it is limited to 70 species. In addition, when the number of molecules is large, the clarity of the cellular automata becomes complicated. As well as the analysis of the graphs. SimCell is suitable for simulating simple biochemical networks in a reduced environment. It gives good results to analyse the Brownian movement and highlight the role of luck in the proper functioning and regulation of cellular processes [244].

V-Cell

V-Cell[6] has been developed by the CCAM (Center for Cell Analysis and Modeling) in 1997 and is based on the work of Schaff et al. [245]. It is designed as a modelling tool for cellular processes for research in cell biology. For this reason, it is able to model the different nature of complex models (mathematical or empirical). V-Cell is comparable to SimCell in many characteristics. The difference between them is that SimCell is constrained to 70 kinds of entities, however, there is no limit of entities for V-Cell. Moreover,

[3] http://www.e-cell.org/
[4] http://www.compucell3d.org/
[5] http://wishart.biology.ualberta.ca/SimCell/
[6] http://vcell.org/

the managed space is not two-dimensional but three-dimensional. V-cell is less suitable for simulating the details of elementary processes such as the diffusion but is more suitable for understanding the general functioning of the cell. It has been used to model cytoskeleton dynamics [246] and the biochemical origin of electrophysiological phenomena in neurons [247].

4.3.4 Discrete Event System Specification

Based on systems theory, the Discrete Event System Specification (DEVS) is a formalism introduced by Zeigler in 1976 to describe discrete-event system in a hierarchical and modular manner. It is theoretically a well-defined system formalism [248]. The DEVS models are seen as black boxes with input and output ports used to describe system structure and behaviour. Therefore, DEVS offers a platform for modelling and simulating complex systems in different domains. DEVS defines two kinds of models: *atomic* models and *coupled* models representing respectively the behaviour and the internal structure of a part of a model.

4.3.4.1 Basic Models

Formally, an atomic model is defined as a seven-tuple $AM = \langle X, Y, S, \delta_{int}, \delta_{ext}, \lambda, ta \rangle$ where X is the set of input events, Y is the set of output events, S is the set of state variables, $\delta_{int} : S \to S$ is the internal transition function, and $ta : S \to R^{+}_{0,+\infty}$ is the time advance function. $\delta_{ext} : Q \times X \to S$ is the external transition function, where $Q = \{(s, e) | s \in S, 0 \leq e \leq ta(s)\}$ is the set of total states and e is the time elapsed since the last transition and $\lambda : S \to Y$ is the output function.

A DEVS model is always in a state $s \in S$ at a given time. The model can transit from a state to another using the transition functions δ_{int} and δ_{ext}. In the absence of external events, it remains in the state s for a lifetime of $ta(s)$. When $ta(s)$ is reached, the model outputs value $y \in Y$ through its ports using the output function $\lambda(s)$, then it changes to a new state defined by $\delta_{int}(s)$. In the case of an external event triggered by external inputs, the external transition function determines the new state given by $\delta_{ext}(s, e, x)$, where s is the current state, e is the time elapsed since the last transition, and $x \in X$ is the external event received.

4.3.4.2 Coupled models

A coupled model defines how a set of models (atomic or coupled) are matched together to create a new model. Formally, a coupled DEVS model can be defined as $CM = \langle X, Y, D, M_d \in D, EIC, EOC, IC, select \rangle$ where X is the set of input ports for the reception of external events, Y is the set of output ports for the emission of external events, D is the set of all DEVS components (atomic and coupled), M_d is the DEVS model of the component $d \in D$, EIC is the set of input links that connect the inputs of the components that it contains, EOC is the set of output links that connect components to the output of coupled model, IC the set of internal links that connects the output ports of the components to the input ports of the components in the coupled models, and $select(D) \to m_d$ the selection function to resolve the activation process of models.

Through the select function, the coupled model can organize the modelling. This determines the order in which conflicting models (external or internal) are to be scheduled.

4.3.4.3 Benefits of DEVS

DEVS has been extended in order to be able to model and simulate continuous and complex systems. Several works [249, 250, 251, 252] have proved that discrete-event methods and in particular the DEVS formalism present several advantages:

- Separation of the modelling and simulation phases.
- Ability to represent a system in its functional and structural form.
- Hierarchical modular modelling, which improves verification and validation, also enhancing reusability.
- Computational time reduction: for a given precision, the number of calculations can decrease.

- Hybrid systems modelling: the discrete-event paradigm provides a unified theory to model and simulate systems with continuous and discrete components.
- Ability to be extended to new fields of study or to be integrated into other modelling approaches (Petri nets, cellular automata, etc.).

The DEVS formalism has been used to simulate enzymes [253] and the tryptophan synthetase metabolic pathway [254].

4.4 Comparison among these simulation tools and platforms

Most of these simulation tools propose the use of complicated mathematical models that are composed of equations for simulating the behaviour of biomolecular systems. However, the mathematical models for simulating the behaviour of biomolecular systems are sometimes complicated to solve considering the high number of molecular components and their heterogeneity. In fact, one of the major problems in simulating complex biological systems using mathematical models is the lack of quantitative data. Indeed, most of the biological knowledge available is qualitative but for quantitative simulation, a large amount of numerical data (such as concentrations of metabolites and enzymes, flux rates, kinetic parameters and dissociation constants) is required.

Moreover, these methods do not follow individuals over time, instead, they track only total populations. They also consider that the interactions among molecular components are homogeneous and assume that the entire system is just the sum of its components which is not necessarily true.

Others simulation tools such as Boolean and logic networks are suited to simulate the small network and particularly Gene Regulatory Networks and signalling pathways. However, it becomes impractical to simulate large biomolecular network sizes (for n nodes, we have 2^n possible states).

Despite mathematical methods, we have also focused on individual-based models such as cellular automata, Potts models, Lattice gas models, multi-agent models, etc. Both can be considered as cellular automata and they have the same characteristics. These models respond to the simulation of complex systems and are a suitable tool for studying the behaviour of such systems. The advantages of this kind of simulation tools are that they are easy to implement and simple in design. But the interactions and relationships among individuals are complex. Multi-agent models provide a detailed description of complex systems. However, if they provide a high level of detail, their design generates a high cost. As well as, the disadvantages of such systems consist of the inability of agents to the more complex organization and the excess parallelism of task's execution. This creates difficult and obscure models.

Others simulation tools such as SimCell, are respectively based on the cellular automata and a discrete model was totally designed for analysing the response of each individual in a population (respectively, each component in the biological system) and how this individual contributes within the population. Thus, these models focus only on the equilibrium and regulation of populations through interactions that exist among their individuals under the constraints of the environment. These models focus only on reproducing the modification of individuals over generations under the constraints of environmental change or competition among individuals. Consequently, they aim only on the evolution of systems at the population scale. Thus, we can conclude that all these simulation tools are dedicated to very specific problems.

On the other hand, some platforms are designed to treat different problems, such as V-Cell. These generic platforms require that the system must be formalized under the required standard formalism to be simulated. Among these specific standards, we have discussed the SBML and CellML corresponding respectively to the V-Cell and Snoopy platforms. The limit of these generic platforms is that they can not simulate multi-level models. They are designed to address only a specific level of the system that typically takes place at a given time and place. Therefore, they can not focus on the interactions among the different omics levels.

We also discussed another formalism the discrete-event system specification for coupling different heterogeneous data models, different levels of a system, specification of sub-models of the system in a single formalism, etc. Under the DEVS formalism, the behaviour of the whole system is easier to simulate if it is represented by a set of states that characterize the individual's activity. In this formalism, a definite type of response of the individual to external stimulus corresponds to a set of states. Moreover, these stimuli are generally not represented by continuous functions in the environment but by occasional events.

In its original version, DEVS formalism allows the specification of event status changes and provides a modular and hierarchical view of dynamic systems. So, we think that it is the most suitable for simulating complex biological systems.

Table 4.1 provides a comparison of all these methods according to their characteristics. Indeed, we compare these different simulation methods based on various criteria, such as

- *The abstraction level* focuses on the level of detail (macro-, meso- and micro-levels).
- *The topic interest* focuses on the interested subject (specific to a subject or general to diverse topics).
- *The network size* focuses on the size of the biomolecular network (small- or large-scale).
- *The category* focuses on the type of the simulation (discrete, continuous or stochastic).
- *The data type* focuses on the type of data to be simulated (qualitative or quantitative data).
- *The dynamic simulation* focuses on the level of details of the simulation (local or global vision)
- *The difficulty level* focuses on the complexity with which the model can be understood and manipulated (easy or hard).
- *Cost*: includes all the charges for design, implementation, operation and maintenance (low or high cost).

After discussing all the properties of these simulation methods, we note that no method is better than the rest, only more suitable for a certain problem. Each of these tools has its own uses and is best suited for solving problems of certain scale and complexity. However, most of these methods consider the biomolecular network as a simple network usually taking account only one level by focusing only on modelling isolated parts of this network, such as metabolic networks, gene regulation networks, and protein-protein interaction networks. However, the interconnection among these different network levels reflects the importance of a general approach that focuses on the multi-level properties of biomolecular networks to replace these traditional methods. In practice, qualitative models and quantitative models complement each other. The choice between qualitative models and quantitative models depends on the availability of kinetic information, the size of the systems and the types of questions to be addressed.

So, we conclude that the choice of a simulation method is determined by the characteristics of the system in which we are interested, how realistic an estimation of it we want, and our mathematical and computational resources. Thus, all of these simulation methods can be used but each method incorporates different levels of detail and requires different computational effort.

4.5 Thesis contribution in this field

The whole behaviour of the cell is an emergent behaviour of many component's interactions (such as DNA, proteins, and metabolites) belonging to different levels (genome, proteome and metabolome). That is why, after the detailed comparison presented above, we conclude that it is not enough to simply describe the system's components or to only simulate a specific level such as genetic or metabolic level, etc. Indeed, there are still many difficulties that inhibit the construction of a complete simulation method. Therefore, in order to understand the global behaviour of biological systems, it is also necessary to detail the local behaviour of each component at the molecular level and to understand what happens when certain external stimuli or intern malfunctions occur. So, rather than the previous approaches which focus on traditional reductionist methods, we think that the behaviour of the complex biomolecular network emerges from the network-level interactions and requires an integrative simulation tool. That is why we will propose in Chapter 8 a simulation approach based on the DEVS formalism and combines logical-based modelling and qualitative simulation. This simulation approach will consider all the omic-levels and will be able to predict and reproduce the behaviour of the biomolecular network and its components under a wide variety of stimuli or stresses.

Table 4.1 – Comparison table of the main approaches applied to simulate biological networks according to their characteristics.

v		*Mathematical and population-based simulation*	*Individual-based simulation*	*Platform simulation*		*DEVS*
				specific platforms	*generic platforms*	
Abstraction level	*macro*	✗	✗	✗		✗
	meso					
	micro	✗	✗		✗	
Topic interest	*specific to a problem*	✗		✗	✗	✗
	general (different problems)		✗			✗
Network size	*small-scale*	✗	✗	✗	✗	✗
	large-scale		✗	✗		✗
Category	*discrete*				✗	✗
	stochastic	✗	✗	✗	✗	✗
	continuous	✗	✗	✗		✗
Data type	*quantitative*	✗	✗	✗	✗	✗
	qualitative		✗			✗
Dynamic simulation	*global vision*	✗	✗	✗		✗
	local vision				✗	
Difficulty level	*easy*		✗	✗	✗	✗
	hard	✗	✗			
Cost	*low*	✗				✗
	high	✗(depends on the model)	✗	✗	✗	

4.6 Summary

This chapter presented the principles (definitions, uses, levels of abstraction, etc.) that characterize the concept of simulation in general. It also detailed the different simulation tools and platforms in different categories and for different purposes. In fact, each simulation uses a model that makes it possible to study a particular phenomenon and explain the performance of the simulation results in relation to the real phenomenon and the user's desired objective. This state-of-the-art reveals the limitations of some simulation tools based on mathematical modelling (due to their enormous size it becomes complicated to define exactly their parameters).

Chapter 5

Optimization tools in systems biology

Contents

5.1 Introduction

Optimization is a rapidly growing discipline that affects various research fields, such as vehicle routing problems, scheduling problems, assignment problems, air traffic problems, etc. but also satisfy the needs of system biology. The optimization of these systems allows finding an ideal configuration for saving effort, time, money, energy or improving satisfaction. Some real-world problems require the simultaneous optimization of several criteria which can sometimes be contradictory. This is the case of the transittability of complex biomolecular networks that can be considered as multi-objective optimization problems.

In this chapter, we present the state of the art of optimization methods in general and multi-objective optimization methods in particular. The first section presents some background definitions about the optimization concept and its basic components. Next, we give the different classification of optimization problems according to their characteristics by focusing on mono and multi-objective optimization problems. Then we present the popular optimization methods for solving multi-objective problems. Moreover, we review the major applications of optimization problems in systems biology through a comparison study. Finally, we discuss the contribution of our thesis to understand the transittability of complex biomolecular networks.

5.2 Optimization problem: definition and basic concepts

In this section, we will define all the basic components of the optimization problem. To do this, we refer to many literature reviews detailing these notions and available in [1, 255, 256].

5.2.1 Definition

An *optimization problem* is defined as a problem that aims to search the minimum or maximum (the optimum) of a given function f, so-called *objective function* [257].

According to M. Ejday [257] the optimization concept consists of two steps. These steps are illustrated in Figure 5.1.

- A first modelling step in which the user defines the *objective function*, the main optimization variables so-called *decision variables*, and the equality and inequality constraints. In the next sections, we will detail these concepts.
- The second step of resolution consists in the search for the best values by optimizing the objective function defined in the first modelling step. This resolution is done through an *optimization algorithm* [258].

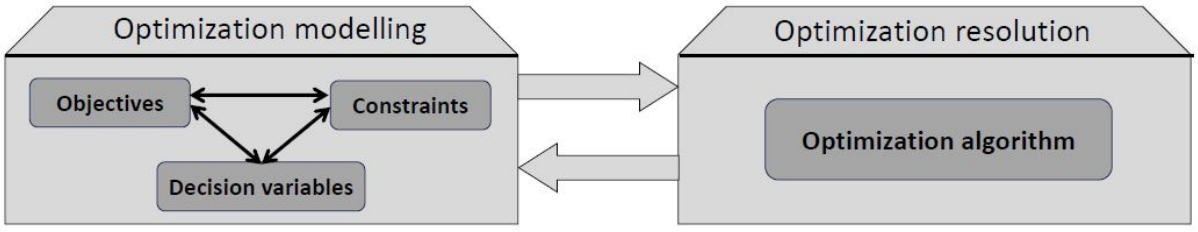


Figure 5.1 – Modelling and resolution steps of an optimization problem.

Optimization has been introduced in order to improve the services provided regardless of their application areas. Indeed, the need for optimization comes from the engineer's need to provide the user with a system that best meets its requirements. This system must be designed to (*i*) use the minimum necessary for its functioning (a minimum cost of resources), (*ii*) consume less energy (minimum operating cost), and (*iii*) respond to the user's request (satisfy all requirements).

This section defines some basic concepts of optimization theory. To do this, we have used these works [1, 255, 257, 256] to give some definitions of these notions.

5.2.2 The objective function

The *objective function* called also *cost function* or *optimization criterion* denoted by f is used in mathematical optimization to describe a function that the optimization algorithm will have to *optimize* (find an optimum). This function represents the goal that the decision-maker wishes to achieve, and it is also used for measuring the goodness of values for the decision variable.

5.2.3 The vector of decision variables

The decision vector $\mathcal{X}$ is composed of the different decision variables of the problem. These variables express qualitative or quantitative data that need to be determined in order to solve the given problem. A decision variable is denoted by $x \in \mathcal{X}$ and can be a number, vector, function, etc. It is by changing the values in this vector that an optimum of the function f is obtained.

5.2.4 Constraints and delimitation of the research space

There are two types of constraints: inequality constraints denoted by the vector $g(x)$ and inequality constraints denoted by the vector $h(x)$. These sets of constraints define a restricted space for finding the optimal solution.

We distinguish two kinds of inequality constraints:

- Constraints of type $Bi_{inf} \leq x \leq Bi_{sup}$: the values of the decision variables that validate these constraints define the *research space* or *state space*. This space is the space of all possible and complete solutions for the optimization problem. This space is finite and defined by the domains of definition of the different decision variables of the problem (Figure 5.2a).
- Constraints of type $c(x) \leq 0$ or $c(x) \geq 0$: the values of the decision variables that check these constraints define the *space of the achievable values*. This space is shown in Figure 5.2b.

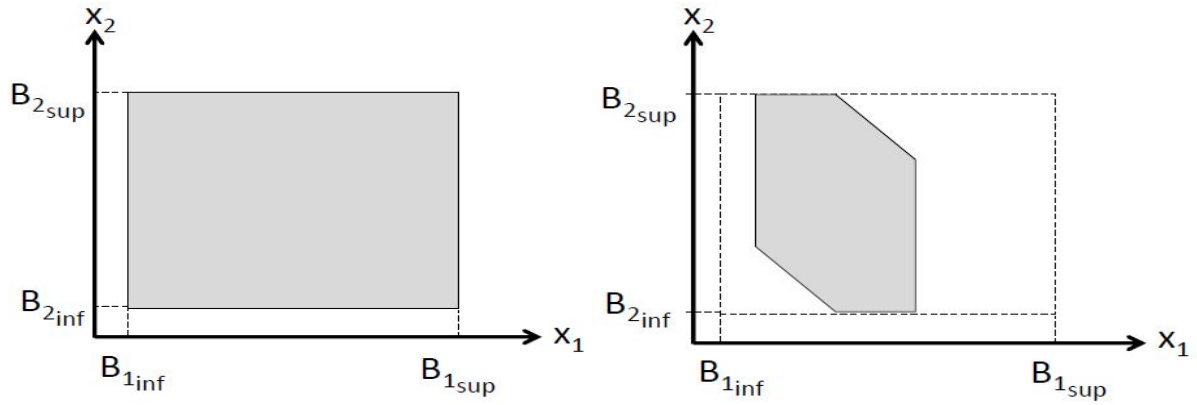


Figure 5.2 – Example of merging: 5.2a The research space. 5.2b The achievable space.

5.2.5 The different types of optimum points

For a given function f, there may be few points at which the function reaches larger (or smaller) values which may be higher (or lower) within some given neighbourhood. The higher points are called *relative maxima* (or *local maxima*), while the lower points are called as *relative minima* (or *local minima)*. Within a given interval, a function must have one (or more) highest and lowest point(s)[1]. As well as, the highest point is called *global maximum* and the lowest point is called *global minimum* [1, 259].

[1] `https://math.tutorvista.com/statistics/global-minimum.html`

In the following sections, we will only consider the problem of minimization. In fact, the problem of maximization follow the same rules. Here, we treat the case of the minimization without losing generality because maximizing f_i means minimizing $-f_i$. Since, for example if f is a numerical objective function, maximizing (f) is same as minimizing $(-f)$.

5.2.5.1 Local maximum and minimum

Optimization functions can have '*hills* and *valleys*': places where they reach a *minimum* or *maximum* value. It may not be the minimum or maximum for the whole function, but only for a local interval.

Local minimum

First, we need to choose an interval. Then we can say that a *local minimum* is the point where the height of the function at a is smaller than (or equal to) the height anywhere else in that interval. Formally, if and only if $f(a) \leq f(x)$ for all x in the interval.

Formally, a point x^* is a *local minimum* of the function f only if $f(x^*) \leq f(x)$ for all $x \in V(x^*)$, where $V(x^*)$ defines the interval of a *neighborhood* of the point x^*. This definition corresponds to points M_1 M_2 and M_4 in Figure 5.3.

Local maximum

Likewise, a *local maximum* is the point where the height of the function at a is greater than (or equal to) the height anywhere else in that interval. Formally, if and only if $f(a) \geq f(x)$ for all x in the interval.

5.2.5.2 Global maximum and minimum

The maximum or minimum over the entire function f is called a *Global* maximum or minimum. Formally, a point x^* is a global minima only if $f(x^*) \leq f(x)$, for all x in the domain of f. This definition corresponds to the point M_3 in Figure 5.3.

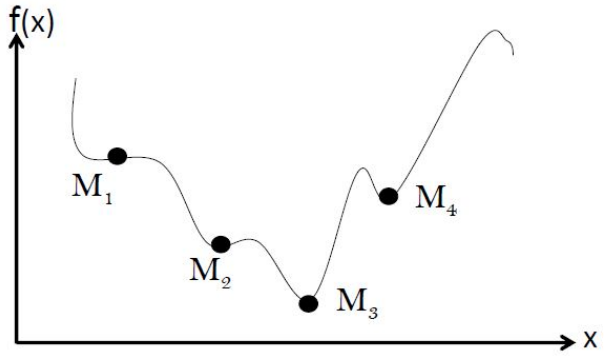


Figure 5.3 – Global minimum and local minima [1].

5.3 Classification of optimization problems

According to [1], the different optimization problems are classified according to their characteristics as follows:

- Number of decision variables:
  - One $\Longrightarrow$ *single-variable* or *mono-objective* optimization problem.
  - Multiple $\Longrightarrow$ *multi-variable* or *multi-objective* optimization problem.
- Type of decision variables:
  - Continuous real number $\Longrightarrow$ *continuous* optimization problem.

  - Integer number $\Longrightarrow$ *discrete* optimization problem.
  - Permutation on a finite set of numbers $\Longrightarrow$ *combinatorial* optimization problem.

- Type of the objective function:
  - Linear function of the decision variables $\Longrightarrow$ *linear* optimization problem.
  - Quadratic function of the decision variables $\Longrightarrow$ *quadratic* optimization problem.
  - Non-linear function of decision variables $\Longrightarrow$ *non-linear* optimization problem.

- Formulation of the problem:
  - With constraints $\Longrightarrow$ *constrained* optimization problem.
  - Without constraints $\Longrightarrow$ *unconstrained* optimization problem.

Indeed, there are different types of optimization problems that depend on the number and type of the decision variables, the type of the objective function (linear, quadratic or non-linear) and the formulation of the problem (with or without constraints). In addition to these characteristics, the number of optimization objectives divides optimization problems into two major kinds. In this chapter, we focus on these two categories of optimization problems: *mono-objective optimization* and *multi-objective optimization*. In the following sections, we present the main concepts that are related to them. Moreover, we define some optimization methods for solving these problems.

5.4 Mono-objective optimization problem

Mono-objective optimization consists of **one and only one** objective or criterion to be optimized. This is a category of optimization problems that are generally 'easy' to solve. The term easy here is used not to denote the level of difficulty of this type of problems but their difficulty in comparison with multi-objective problems [1]. Indeed, mono-objective problems do not present a conflict of interest in focusing on one of several optimization criteria in contrast to multi-objective problems that have to consider several criteria.

Mono-objective optimization seems easy, but they have also difficulties such as the non-linear objective function that cannot be expressed analytically in terms of parameters. In fact, modelling the problem in the form of a single equation can be a very difficult task. Moreover, reducing the mathematical formulation of the problem to a single objective function can lead to errors and mistakes in the modelling. This problem does not exist in the multi-objective optimization where a certain degree of flexibility is allowed [1].

A mono-objective optimization problem is mathematically defined as:

Minimize/Maximize	$f(x)$	(function to be optimized)
with	$g(x) \leq 0$	(inequality constraints)

With $x \in \mathbb{R}^m$, $g(x) \in \mathbb{R}^p$ and $h(x) \in \mathbb{R}^q$.

5.5 Multi-objective optimization problem

Most real optimization problems are described by **several criteria** frequently contradictory and which must be optimized simultaneously [255]. Multi-objective optimization aims to optimize these multitudes of objectives at the same time considering that they can be contradictory and generate conflicts of interest. These objectives can be explicitly defined as objective optimization criteria or formulated as constraints. Formally, a multi-objective optimization problem is defined by the triplet (X, F, g) that consists in *minimizing* $F(X)$ for $X \in \mathcal{X}$ with $g(X) \leq 0$, and mathematically defined as follows:

Minimize	$F(X) = (f_1(X), f_2(X), \ldots, f_n(X))$	(functions to be optimized)
for	$X \in \mathcal{X}$	
with	$g(X) \leq 0$	(inequality constraints)

With $x \in \mathbb{R}^m$,$F(X) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^p$ and $h(x) \in \mathbb{R}^q$.

The m decision variables are the values to be chosen in an optimization problem. These values are denoted by x_i with $i \in \{1, \ldots, m\}$. The vector X of the m decision variables is denoted by $X = (x_1, x_2, ..., x_m)$. The set of decision variable vectors form the search space $\mathcal{X}$. The n objective functions

to be optimized are denoted by f_i with $i \in \{1, \ldots, n\}$. The vector F of the n objectives (for $X \in \mathcal{X}$) is represented by $F(X) = (f_1(X), f_2(X), \ldots, f_n(X))$. The p inequality constraints and the q equality constraints limit the values that the decision variables can take. They are respectively denoted by $g_i(X)$ with $i \in \{1, \ldots, p\}$ and $h_i(X)$ with $i \in \{1, \ldots, q\}$.

As discussed above, a multi-objective optimization problem has the advantage of providing a level of flexibility which is absent in mono-objective optimization. In addition, this flexibility affects the space of solutions by changing it from a single solution to several solutions. As well as, this space depends on the adopted optimization approach. Indeed, the objectives are sometimes contradictory, therefore, optimizing one objective can negatively affect one or more others. As a result, an optimal solution does not exist but we are talking about *optimized solutions* because multi-objective optimization cannot optimize all the objectives at the same time and should give the priority to only some of them. It is in this case that the concept of compromise is introduced because certain objectives will be preferred (optimized) to the exclusion of others having a bad quality performance. These solutions are considered useful and appropriate according to the user requirements.

Thus, in contrast to the mono-objective optimization problem, where the optimum is not a simple point but a set of points so-called the *set of the best compromises* or *Pareto Front* which is an area of solutions that offer a good compromise between the different objectives [1]. Therefore, the optimality in a multi-objective context is based on the notion of *dominance relation* and *Pareto-optimal* (efficient) solution. We will recall their definitions in the following sections.

5.5.1 Dominance relation

In multi-objective optimization problem, the solutions extracted are called *Pareto solutions* and constitute the *surface of compromise*. The goodness of a solution is determined by the *dominance relation*. A solution x_1 dominates another solution x_2 (noted $x_1 \prec x_2$) if both the following conditions are true:

- Solution x_1 is no worse than x_2 in all objectives.
- Solution x_1 is strictly better than x_2 in at least one objective.

$$x_1 \prec x_2 \begin{cases} f_i(x_1) \leq f_i(x_2) & \forall i \in 1, \ldots, n \\ \exists j \in 1, \ldots, n & f_j(x_1) < f_j(x_2) \end{cases}$$

Solutions that dominate the others but are not dominate themselves are called *non-dominated solutions*. Through this definition, the dominance relation is considered as a filter of the bad elements to keep only solutions that cannot be compared with each other.

5.5.2 Pareto-optimal solutions

A number of criteria distinguish the selected solutions called *Pareto-optimal solutions* (or *non-dominated solutions*) when they are not dominated by any other solutions in the feasible space. The non-dominated set of the entire feasible decision space is called the *Pareto-optimal set*. The boundary defined by the set of all point mapped from the Pareto-optimal set is called the *Pareto-optimal front*.

Like there are global and local optimum in the case of mono-objective optimization, we can also define global and local Pareto-optimal sets in multi-objective optimization. According to [256] we have:

- *Local optimality in the Pareto sense*: A solution x_1 is locally optimal in the Pareto sense, if there exists a positive real $\epsilon > 0$ such that there is no other solution x_2 that dominates the solution x_1, with $x_2 \in \mathbb{R}^m \cap B(x_1, \epsilon)$, where $B(x_1, \epsilon)$ shows a bowl having a center x_1 and a radius ϵ.
- *Global optimality in the Pareto sense*: A solution x_1 is globally optimal in the Pareto sense, if there does not exist any vector x_2 that dominates the vector x_1.

The main difference between the global and local optimality lies in the fact that we do not have a restriction on the set $\mathbb{R}^m$ anymore. Thus, the *Pareto set* or the *efficient set* is the collection of all Pareto-optimal solutions and their corresponding images in the objective space is called the Pareto front.

These solutions are obtained using a precise approach based on a specific optimization method among a wide range of techniques available in the literature [2]. The choice of the method to be applied must be adapted to the optimization problem to be solved. This choice represents a significant difficulty because it is responsible for the success or not of the adopted method.

In the following, we focus on the optimization methods that can be used in this context.

5.6 Optimization methods

As discussed in the previous section, depending on the case (whether a single-variable or multi-variable, continuous or discrete problem, etc.), an appropriate optimization method is carefully chosen to solve the problem in an efficient way [259]. Several methods have been developed to solve mono-objective problems such as exact, heuristics, meta-heuristics, hybrid, etc. [2]. However, not all of them are appropriate for the multi-objective optimization problems, therefore a reasonable choice should be made according to the nature and complexity of the given problem.

There are different classifications of these various optimization methods for solving multi-objective optimization problems in literature. According to DA Van Veldhuizen [260], optimization methods can be grouped into four main categories:

- *No preference methods* which are methods where the decision maker is not considered and the problem is saved using a simple method that presents directly the solution to the decision maker.
- *A priori optimization methods* which are methods where the compromises and preferences that we would like to optimize are determined before the execution of the optimization method. In this method, the preferences of the decision maker are asked and then the best solution according to the given preferences is found.
- *A posteriori methods* which aim to generate a representative set of Pareto-optimal solutions and the decision maker chooses the best one among them. This method allows the decision maker to choose the solution that is suitable for his problem by comparing it with the rest of the solutions.
- *Interactive methods* called also *progressive optimization methods* allow the interaction of the decision maker during the optimization process. This interaction is done by asking questions with the decision maker.

In the following sections, we will only present the most used methods. To do this we follow the classification proposed by Colette et al. in their work [1] to group all the optimization methods into four categories: the interactive methods, scalar methods, fuzzy methods, and methods based on metaheuristics.

- *Interactive methods:* As discussed in the previous section, these methods are considered as progressive methods which allow searching a single solution. They are based on the interaction with the decision maker who can choose the solution that best corresponds to his preferences. In this category, we cite the STEP method, the substitution compromise method, the Fandel method, etc.
- *Scalar methods:* They include a set of methods which are the most evident approach for solving multi-objective optimization problems. This approach called 'naïve approach' aims to reformulate the optimization problem in order to return to a mono-objective optimization. This process is frequently achieved by using the aggregation (or the sum) of all objective functions in a single function. A weighting coefficient is assigned to each objective function to indicate its importance in the global objective function. By aggregating these weighted objective functions, we obtain a single objective function. In this category of methods, we can also find the Keeney-Raiffa method [261], the distance to a reference objective method, etc.
- *Fuzzy methods:* These methods are inspired by fuzzy logic theory. They aim to neglect the binary logic (true or false) and consider a certain degree of flexibility in order to accept the uncertainty and imprecision of human knowledge. In this category, we find the Sakawa method, the Reardon method, etc.

For more detail, we can refer to the works of Colette et al. [1] in which they list, detail and formalize all these methods.

5.6.1 The methods based on a metaheuristic approach

These methods have been developed since the 1980s. They include tabu search methods, simulated annealing, genetic algorithms, ant colony algorithms, etc. The concept of *metaheuristic* was introduced

by Fred Glover in [262] who defined it as follows: '*A metaheuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality*'. As well as, another definition was proposed in [263], '*metaheuristics are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space*'.

Their major objective is to solve difficult optimization problems. In fact, metaheuristics were initially dedicated to solving the difficulty of optimization problems. In contrast to traditional optimization methods (which take the path of the biggest slope to quickly find the local minima, but once they get they found the local minimum they do not search anymore), the main advantage of these methods is their ability to avoid local minima. In fact, traditional methods start from an initial configuration and search for the best value through a series of iterations, while metaheuristics tolerate the momentary deterioration of the situation. This idea has proven to be effective in avoiding local minima (or maxima) for solving complex problems and become the basis of all neighbourhood metaheuristics (simulated annealing, tabu search, etc.). There are also other metaheuristics so-called distributed metaheuristics, such as evolutionary algorithms which also have very particular techniques to solve local minima (or maxima) problems.

In the next sections, we review metaheuristics cited above introducing their properties and functioning.

5.6.1.1 Simulated annealing

The simulated annealing method [264] is inspired by the annealing process and based on the principle of starting from a given configuration to achieve the desired result. This result concerns the global optimum (or a result that is close to it) achieved by applying elementary transformations into a finite number of iterations.

In practice, this method uses the Metropolis algorithm which defines and represents the behaviour of a system in *thermodynamic equilibrium*at a specific temperature (T) [264]. Starting from a given configuration, this method changes the system by modifying its basic elements such as the translation of a component or the exchange of two components. If this modification generates an improvement to the desired objectives (for example a reduction of the objective function), it is accepted. On the other hand, even if this same change leads to a deterioration, it will be accepted but only with a certain probability of $e^{-\delta E/T}$. Much that the higher the temperature, the more highly to keep the deteriorated state. This procedure is repeated by keeping the temperature T constant until the thermodynamic equilibrium is reached after a number of iterations. Once this goal has been reached, the whole process is repeated for a new lower temperature value. A series of transformations is then performed on this value again.

Figure 5.4 presents the main steps of the simulated annealing algorithm [265]. In this figure, the process described above is summarised and presented under a diagram summarizing the main steps followed when applying an optimization method based on simulated annealing. There are several variants of the simulated annealing method such as the simulated diffusion, micro-canonic annealing, threshold method, flooding method or record-breaking trip method.

5.6.1.2 Tabu search

The tabu search method was formalized in 1986 by F. Glover in [266]. The main advantages of this method are based on techniques inspired by human memory. In contrast to the simulated annealing method which is completely devoid of memory, the tabu search saves the history of the different steps of its application. It is able to react based on the lessons of the past. The actions performed in the current iteration are affected by the previous actions that constitute the history of its execution. In addition, the tabu search method is simple. In fact, its procedure is similar to the simulated annealing method and consists in applying a series of changes to several iterations of the process. The tabu search method works with only one configuration at the same time. Initially, this configuration is obtained arbitrary and then locally updated (only small changes at each step). At each iteration, the mechanism of switching from a given configuration c to a successor configuration c' follows these steps [267]:

1. The construction of all the neighbours of c. This set represents all possible and achievable configurations accessible in a single elementary movement applied to c. If this set is too large, a well-defined technique is applied to reduce its size. Among these techniques: the selection of a list of candidates

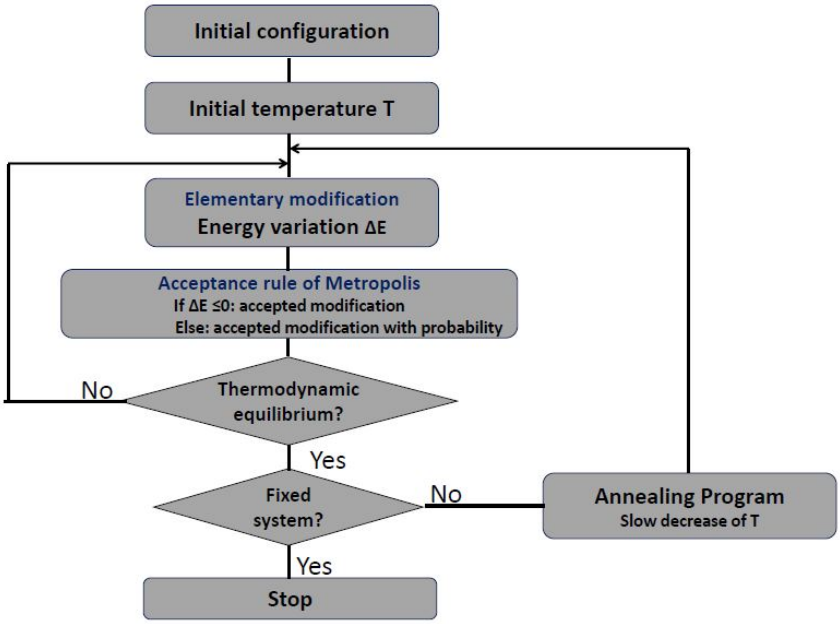


Figure 5.4 – Diagram illustrates the process of the simulated annealing [2].

or the random extraction of a subset of fixed-size neighbours. $V(c)$ denotes the set (or the subset) of these neighbours.

2. The evaluation of the objective function f in each of the configurations belonging to $V(c)$. In terms of optimizing the objective function, the configuration $c^{'}$ that succeeds c would be the best among those constituting the set $V(c)$. In addition, this same configuration $c^{'}$ can be taken into account even if it proves to be less good than the first one c ($f(c^{'}) > f(c)$). This emphasizes the particularity that constitutes the main advantage of metaheuristics (especially the tabu search method in this case) which consists in avoiding the local minima (or maxima) of the objective function f.

This procedure is sometimes ineffective because there is a frequent risk of returning to a previous configuration already retained in a previous iteration. This generates a cycle and consequently the blockage on an infinite loop. In order to avoid this problem, the tabu search method introduces the concept of the prohibited actions resulting in the construction of a list of prohibited movements. At each iteration, this list which is continuously updated is used. The tabu search method takes its name from this list which is itself called the *tabu list*. This list contains a finite number m of prohibited movements ($c \rightarrow c^{'}$) applied to c to have the $c^{'}$ configuration. Figure 5.5 presents a detailed diagram for the execution of the tabu search method.

The tabu search method is effective and provides excellent results in solving some optimization problems [2]. In addition, this method has fewer parameters in its basic form and is easier than the simulated annealing method. However, if we add the mechanisms associated with this method such as intensification and diversification, we can see an increase in its complexity.

5.6.1.3 Evolutionary Algorithms

Evolutionary algorithms were introduced in the 1950s [268]. They use research methods inspired by the biological evolution of species. They have initially attracted limited interest due to their high computational cost. However, interest in these techniques has considerably increased over the last two decades thanks to the increase in the performance of computers. The principle of an evolutionary algorithm is simple. Indeed, it is a question of considering a set of N points chosen randomly in a space of initial search. This set constitutes the initial population. Each individual x in the population has a certain level of performance which measure its adaptation to the target objective.

The basic principle of an evolutionary algorithm is to progressively change (by successive generations of individuals) the composition of the population while maintaining its constant size. Over the generations,

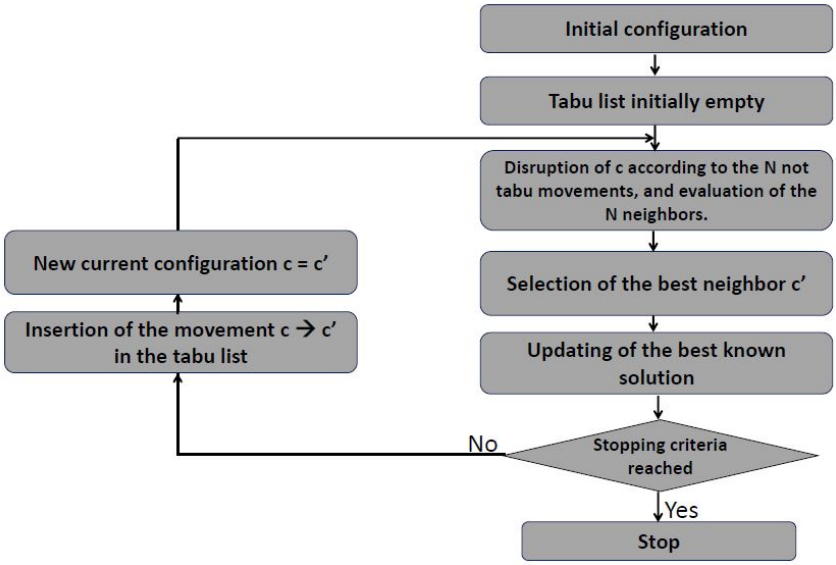


Figure 5.5 – Diagram illustrates the process of the tabu search [2].

the main objective is to improve the overall performance of individuals. To achieve this objective, the evolutionary algorithm uses two main mechanisms that determine the evolution of living organisms. These mechanisms are defined by C. Darwin's theory [269]:

1. *The selection* mechanism that focuses on reproduction and survival of the best performing individuals.

2. *The reproduction* mechanism that consists of mixing, recombining and changing the hereditary characteristics of parents in order to form descendants with new potentialities.

In practice, a well-defined representation of the individuals in a population must be chosen. For example, for combinatorial problems, an individual can be classically assimilated to a list of integers. However, for numerical problems manipulating variables in continuous spaces an individual is represented by a vector of real numbers, a string of binary numbers in the case of Boolean problems, or a combination of these representations in more complex structures. At each iteration of the algorithm execution, there are four phases of transition from one generation to another:

- A *selection phase* which consists of identifying and selecting chromosomes from the population to be parents to crossover. According to Darwin's evolution theory, the best ones should survive and create new offspring. There are many methods how to select the best chromosomes such as the roulette wheel selection, the rank selection, etc.

- A *reproduction phase* consists of applying specific variation operators to the copies of the selected individuals in order to generate new ones. Among these operators, the crossing (or recombination) that allows producing one or two descendants from two parents, and the mutation that allows producing a new individual from a single individual. The structure of the variation operators is based on the representation chosen for individuals that strongly depends on their coding.

- An *evaluation phase* aims to evaluate the performance of the new individuals generated in the previous phase according to the desired objectives.

- A *replacement phase* that concludes the process of defining a new generation of solutions and determines the choice of its members. For example, the lowest performing individuals in the population can be replaced by the best (the highest performing) individuals produced. The algorithm is stopped after a certain number of generations according to a stop criterion specified by the user.

If the optimization problem is based on an objective function with several global optima, evolutionary algorithms are particularly suited to propose a set of diverse solutions. Indeed, they can provide a variety

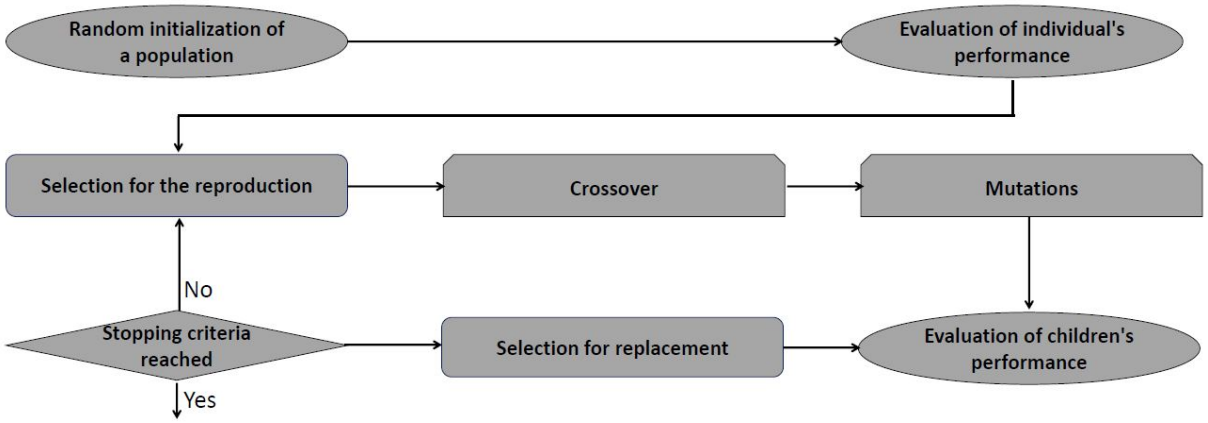


Figure 5.6 – Diagram illustrates the process of the evolutionary algorithm [2].

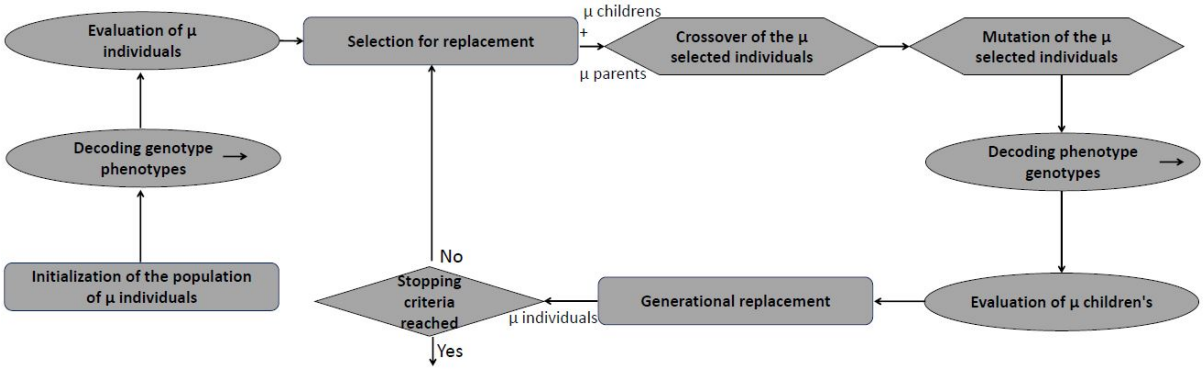


Figure 5.7 – Diagram illustrates the process of the genetic algorithm [2].

of compromise solutions for resolving the multi-objective problem. Figure 5.6 provides a description of the evolutionary algorithm execution. One of the most important variants of evolutionary algorithms is the genetic algorithms [269]. These algorithms are inspired by the Genetics (phenotype of natural genetics). This is the step that directly precedes the evaluation of the individual performance. A phenotype is the set of characteristics that can be observed in an individual. A genotype is associated with a string of binary symbols. This chain is then decoded in order to build a solution to the problem represented in its natural formalism (phenotype). This phenotype is then evaluated using a *Fitness function* to give a performance value that can be used by the selection operators. Figure 5.7 illustrates the basic components of a simple genetic algorithm. Variation operators work on genotypes which are represented in the form of binary chains makes them easier to treat by the crossover and mutation operators. The crossover operator is an essential research operator, and the mutation is applied with a low rate (probability of mutation: P_m). These operators maintain the diversity in the research area.

5.6.1.4 Ant colony

The ant colony method was introduced by Colorni Dorgio and Maniezzo in [270]. It is based on the behaviour of ant colonies and aims to simulate their collective capacity to solve certain problems. It should be noted that the various members of ant colonies have very limited capacities. Several studies have focused on the habits of ants which are considered as one of the most prosperous species. The process followed by ants emphasizes the collective faculty, i.e. the ability and the ease of a community to quickly find the shortest way [271]. Ant colonies algorithm have several important characteristics such as [271]:

- Flexibility: an ant colony is characterized by a high degree of flexibility to easily adapt its behaviour when the environment changes,

- Robustness: a colony is able to maintain its activity if some of its individuals are deficient,
- Decentralization: this characteristic emphasizes the distributed aspect of the approach (a colony is not centralized),
- Self-organization: a colony has a certain degree of autonomy because it can find its own solution that is not known in advance.

Thanks to its several advantages, this approach is effective and particularly suitable for distributed problems that evolve dynamically, or that require a high tolerance to failures. The implementation of these algorithms require a preliminary study and should be the subject of a specific treatment which can be more or less difficult.

Metaheuristics have proven to be successful because of their advantages which have been lacking in traditional optimization methods that have been proven ineffective to solve the difficulties of complex problems. After the success of the different metaheuristics, other types of difficulties emerged emphasizing the notion of complementary of these new methods among them, and with other approaches, give rise to a new concept of hybrid methods [1]. Moreover, to be really effective, it is usually necessary to use specific representations and operators according to the problem.

5.7 Optimization problems in system biology

Several problems in biology can be considered and formulated as optimization problems. In this section, we review and describe some applications of optimization in systems biology. We will classify these existing works according to their application area or topic. Then, we will describe the major contributions made in these topics. A summary of these works is given in the comparison Table 5.1.

5.7.1 Optimization in the design of optimal dynamic experiments

Among the various research problems that have been addressed in systems biology, we can cite the optimal experimental design of dynamic experiments [272] which consists of the determination and the modelling of the stimuli profiles that maximize the amount and quality of information extracted from the experiments.

In this topic, we found the works of Faller et al. [273] who propose an optimization solution to compute polynomial input profiles in order to enhance the parameter estimation accuracy for a mitogen-activated protein kinase cascade. Also, Kutalik et al. [274] propose the calculation of optimal sampling times in order to minimize the variation of the parameter estimates. Balsa-Canto et al. [275] propose a multimodal non-linear programming problem that aims to maximize the ratio quantity/quality of information for model calibration. The applicability of these approaches was illustrated through various examples related to the modelling of cell signalling cascades.

5.7.2 Optimization in the parameter estimation in cell systems modelling

Another topic that was the subject of several optimization problems is the structure and parameter estimation in cell systems modelling.

Various approaches have been developed in this area. Among them the works of Romero-Campero et al. [276] who propose an approach based on evolutionary algorithms to optimize both the kinetic parameters and the structure of their cell model. Their method consists of a P system[2] integrated into a stochastic simulation algorithm. Rodriguez-Fernandez et al. [277] propose a mixed-integer nonlinear programming-based optimization approach for evaluating and reducing the parameters of cellular systems modelling. Zomorrodi et al. [278] propose a constraint-based model enabling the maximization of an ecosystem objective function. In this same topic, Budinich Marko et al. [279] propose an approach based on Pareto optimality to describe all the feasible solutions of a microbial genome-scale considering metabolic constraints.

[2] https://en.wikipedia.org/wiki/P_system

5.7.3 Optimization in biological network alignment

Here we present the role of optimization in an important problem in systems biology: the biological networks alignment. In fact, network alignment aims to compare, match and align the nodes of diverse biological networks in order to identify sub-networks with similar nodes which could share the same functions, structure, or common evolutionary history.

In this topic, a wide array of applications has been proposed such as the works of Yang et al. [280] who develop a mixture of the global and local algorithm for network alignments so-called BinAligner. In this work, the alignment problem is formulated as an assignment problem solved by a combinatorial optimization algorithm (the Hungarian method). The proposed algorithm was applied and validated in aligning the protein-protein interaction network of two viri: the varicella-zoster virus and Kaposi's sarcoma virus. In addition, the network alignment problem was transferred into a linear or quadratic integer programming problem and solved through linear relaxation [281], Lagrangian relaxation [282], and ILOG CPLEX [283]. For example, in [281] authors develop an efficient algorithm for aligning molecular networks based on both molecule similarity and architecture similarity using integer quadratic programming. Klau G. W. [282] introduces the maximum structural matching formulation for network alignment using Lagrangian relaxation algorithm.

5.7.4 Optimization of biochemical reaction networks

Optimization approaches have been also used in systems biology to tackle the problem of optimizing biochemical reaction networks. Indeed, different optimization methods have been developed in this area which can be divided into two categories of applications: metabolic control analysis [284, 285] and biochemical systems theory [286].

In metabolic control analysis, Heinrich and Schuste [284, 285] provide an overview of the work's interest on metabolic control analysis which serves to measure the extent which different enzymes limit the flux under particular conditions. This analysis provides also a framework for experimental investigations and elucidates the regulatory properties of metabolic pathways. In the category of biochemical systems theory, we can cite the works of Torres and Voit [286] who present several optimization researches applied to metabolic networks in order to understand the biochemical processes that are involved in the synthesis of the desired product.

In these works, linear programming has been the engine behind metabolic flux balance analysis to represent the metabolic phenotype under certain conditions.

5.7.5 Optimization in the sequence alignment problem

Several optimization algorithms have been proposed to solve the multiple sequence alignment problem [287]. This problem consists to assign a function to genes with the goal of reducing the similarity among genes. This task is solved by comparing the corresponding sequences of nucleotides or amino acids to obtain a possible alignment between similar sequences[3].

Several sequence alignment methods have been developed to minimize the number of insertions or deletions (gaps). These optimization methods are based on simulated annealing [288], iterative algorithms [289], relaxation methods [290], genetic algorithms [291], tabu search [292] and Monte Carlo optimization [293]. As well as, a number of optimization approaches have been proposed for predicting and analysing the process of protein folding in simplified models. These optimization approaches were also based on Monte Carlo methods [294], tabu search [295], estimation of distribution algorithms [296] and genetic algorithms [297].

5.7.6 Optimization in inferring networks

Optimization problems have been used for inferring biomolecular networks which are also called *reverse engineering*. This problem operate in the different biological networks such as transcriptional regulatory networks [298], gene regulatory networks [299], signalling pathways [300], and protein-protein interaction networks [301].

[3] `https://www.cs.us.es/~fran/students/julian/sequence_alignment/sequence_alignment.html`

In transcriptional regulatory networks, we cite Wang Rui-Sheng et al. [298] who propose a linear programming problem which has a globally optimal solution for inferring transcriptional regulatory networks from gene expression data based on protein transcription complexes and mass action law. In gene regulatory networks, Thomas Reuben et al. [299] propose an optimization-based regulatory network inference approach that uses time-varying data from DNA microarray analysis. In signalling pathways, Lin Xiaoxia et al. [300] use mixed integer linear programming techniques to identify the network topology of the glucose signalling pathway in yeast. Finally, in protein-protein interaction networks Han Soohee et al. [301] present an optimization-based inference scheme to unravel the functional interaction structure of biomolecular components within a cell. Villaverde and Banga [302] review these optimization applications in reverse engineering and detail their strategies, perspectives and challenges in systems biology.

5.7.7 Optimization in the network controllability

Recently, a new area of systems biology was also solved by optimization theory. This is the task of steering complex biomolecular networks also called the *control theory* [28, 303]. Only a few studies have been focused on this problem.

Among them we cite the works of Wen-Xu Wang et al. [304] who propose a general approach to optimize the controllability of complex networks by minimizing the structural perturbations. This approach consists to drive the biological network using minimum signals of perturbation. Therefore, they aim to minimize the number of signals to be applied to the biological network rather than using a signal for each node. As well as, they propose to use only one control signal to achieve the optimal controllability of networks by adding a minimum number of links. To do this, they formulate their approach into an optimization problem using the concept of *matching path* which is based on the maximum matching algorithm. In the same topic, Kim et al. [305] propose an optimization algorithm for searching the minimum steering node set. This algorithm is used only with Boolean networks.

According to, Gao et al. [306], it is not necessary to control the whole network but it would be preferable to explore the target control which is a preselected subset of nodes. To do this, they minimize the number of driver nodes needed for target control. Their optimization approach is based on the 'k-walk' theory for directed networks, and on a greedy algorithm for treating general networks.

Wu et al. [307, 308, 309, 310] addressed the problem of drug target identification by formulating it as a problem of finding steering kernel in the network. To accomplish this goal, they propose a graph-theoretic algorithm to find a minimum set of steering nodes in biomolecular networks which can be a potential set of drug targets.

However, Wang Le-Zhi et al. [311] consider that the optimization methods cited previously (which control the networks using the minimum set of driver nodes) can cause the development of an unexpected phenomenon. As well as, they have seen that a network cannot be controlled with a small number of drivers, but it is necessary to balance the number of driver nodes and control cost. To overcome this difficulty, they propose a physical controllability framework based on the probability of achieving real control by increasing the set of input signals on properly chosen nodes.

5.8 Comparison among these optimization tools and problems

There are several reviews that cover the optimization problems in systems biology such as [312, 313, 314, 315, 316]. Indeed, as presented in Table 5.1 all these optimization and mathematical programming techniques differ in the type of problem they have to solve, their intended purpose and their methods adopted to solve their given problem. Moreover, we note that no method is better than the rest, but only more suitable for a particular problem. Each of these optimization techniques has its own uses and is best suited for solving specific problems with certain difficulty. However, most of these technique interest on different categories of problems such as sequence alignment problems, inferring networks problems, parameter estimation problems, comparison networks problems, etc. and only a few works focus on the driving of complex biomolecular networks. As well as, all these works are mono-objective and neglect other criteria for steering these biomolecular networks. Therefore, we conclude that to understand how these networks change we have to take into account more criteria such as the minimization of the number of input signals, the minimization of their total cost, the minimization of the number of target nodes, and the minimization of the patient discomfort.

Table 5.1 – Summary of optimization approaches in systems biology.

Topic/application area	Description/objectives	Examples/references
Design of optimal dynamic experiments	Enhance the parameter estimation of network's models. Maximize the quality of information extracted from experiments.	[272, 273, 274, 275]
Structure and parameter estimation in cell systems biology modelling	Minimize the number of kinetic parameters. Optimize the structure and parameter estimation of cell models.	[276, 277, 278, 279]
Biological network alignment	Compare, match and align biological networks. Identify sub-networks.	[280, 281, 282, 283]
Biochemical reaction networks	Analyse and control metabolic networks. Understand biochemical processes.	[284, 285, 286]
Sequence alignment problem	Minimize the similarity among molecule components. Minimize the number of insertions or deletions.	[287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 317]
Reverse engineering	Identify the network topology. Infer biological networks.	[298, 299, 300, 301, 302]
Network controllability	Minimize the set of target nodes. Minimize the set of control signals.	[304, 306, 307, 308, 309, 310, 311]

5.9 Thesis contribution in this field

As discussed in the previous section (5.7.7), a few studies have started to address the dynamic aspects of biological systems through the 'controllability' [28] of a network, where the ability to steer a complex directed network from any initial state toward any other desired state is measured by the minimum number of required driver nodes (nodes with the ability to steer the entire network). It has been shown that in order to achieve complete controllability, the minimum number of driver nodes is 80% of the nodes in a regulatory biomolecular network.

This result led other groups to develop a theoretic framework for studying transitions between two specific states of directed complex networks, a concept they call 'transittability' [6]. In general, this concept expresses the idea of steering the complex biomolecular network from an unexpected state to a desired state. The theorems were developed with continuous time-invariant linear systems, and applied to 4 different biological systems consisting of up to 17 molecules and 40 interactions.

With this idea in mind, we think that understanding the transittability of complex biomolecular networks should take into account more criteria such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of input signals, the minimization of the cost of these signals, the minimization of the number of target nodes, the minimization of patient discomfort.

That is why we hope to contribute to this discipline by proposing a multi-objective genetic algorithm for optimizing the transittability of complex biomolecular networks. This approach will provide the best set of external stimuli for driving the network. This will be the topic of our contribution to Chapter 9.

5.10 Summary

In this chapter, we have introduced a brief reminder of the basic concepts of optimization theory. Then, we focused on multi-objective optimization problems and define their basic principles. We also presented the appropriate optimization methods for solving these multi-objective optimization problems. We briefly described them according to the classification proposed by Colette et al. [1]. Also, we reviewed and described some applications of optimization in systems biology. We classified these existing works according to their application area and gave a comparative study. In the last section, we discussed the limits of some researches that address the dynamic aspects of biomolecular networks, and suggest a multi-objective genetic algorithm-based approach that might be useful for optimizing the transittability of these networks.

Part II

Contributions

*After presenting the context and the state-of-the-art of our works in the first part of this manuscript, this second part is devoted to our contributions on the design and development of a platform to simulate the state changes of complex biomolecular networks with the hope of understanding and steering their behaviour. This platform consists of four basic modules: (*i*) the modelling module to formalize the dynamic behaviour of biomolecular networks, (*ii*) the ontological module to provide a rich description of cellular entities and their interactions with each other, (*iii*) the simulation module to reproduce the dynamic behaviour of each network's component over the time and (*iv*) the optimization module to provide a set of transition sequences proposing the best steering of the biomolecular network from a given state to another. These contributions are organized by specific domain. According to each module of our proposed approach, four contributions have been made in this work. Therefore, we develop these contributions according to four chapters:*

Chapter 6

Logical-based modelling of complex biomolecular networks

Contents

6.1 Introduction

As discussed in the first part of this dissertation, several formalisms have been proposed in recent years for the modelling of biological networks. In Chapter 2, we have detailed some of these modelling approaches and have compared their characteristics. We concluded that most of them focus only on modelling isolated parts of this network, such as the metabolic network or the gene regulatory network, and do not study the dynamics of the network as a whole. Indeed, they do not examine the interactions among all the intervening molecules considering their types. As a result, these modelling approaches are impractical to understand the transittability of complex biomolecular networks. To do so, it is necessary to take into account the analysis of the structure and dynamics of the whole cell rather than just focusing on isolated parts.

In this chapter, we present a logical-based approach for modelling the dynamic behaviour of biomolecular networks. This formalism is based on the three levels of analysis defined by the systems theory: structural, functional and behavioural modelling. Indeed, it aims at describing and analysing all the properties and mechanisms of complex biomolecular networks. This logic-based modelling will form the basic element for modelling and understanding the transittability of these complex networks.

In the first section, we will present and define the functioning of an applied case study, the autoregulation of the bacteriophage T4 gene 32. We will use this simple and small example throughout the contribution chapters (Chapter 6 to Chapter 9) to explain the different notions of our proposed approaches. The second section of this chapter presents a brief introduction to systemic approach by presenting its different axes. In the third section, we will present our proposed logical-based modelling by detailing its triple levels: structure, function and behavioural modelling. Finally, the last section is dedicated to applying this modelling on the case study of the bacteriophage T4 gene 32.

6.2 Motivating example: the bacteriophage T4 gene 32

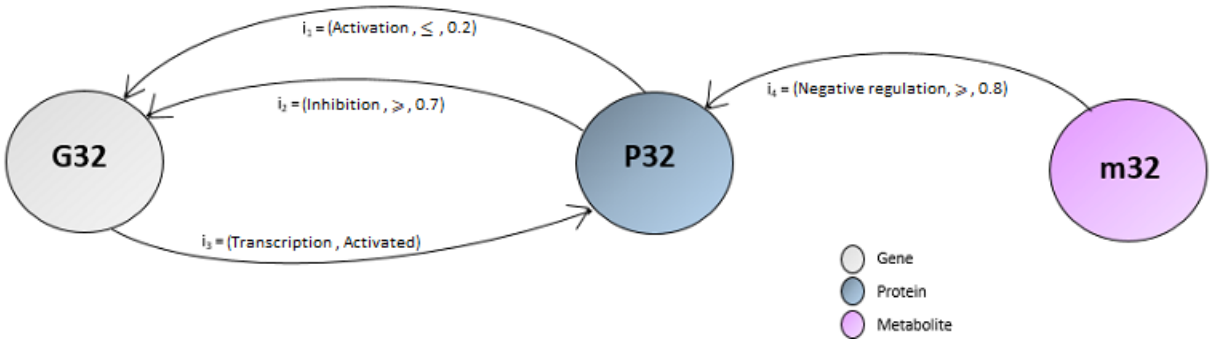


Figure 6.1 – The bacteriophage T4 gene 32 use case.

The bacteriophage T4 gene 32[1] encodes a single-stranded DNA binding protein required for T4 DNA replication, recombination, and repair [318]. It is a single polypeptide chain of 301 amino acid residues that consists of three structural domains, each of which has a binding function. Despite its role in DNA metabolism, the gene product 32 autoregulates its synthesis at the level of translation [319]. During the infection, the gene product 32 is produced in large amounts to perform its function of binding all available DNA at the replication fork, recombination nodes and at lesions in DNA resulting from damage [320]. When all the available DNA is bound, free gene product 32 accumulates within the cell until it reaches a certain concentration which then attenuates further synthesis of gene product 32 [321]. More detail about the bacteriophage T4 gene 32 can be found in [318, 320, 321].

As illustrated in Figure 6.1, this biomolecular network consists of three nodes a **gene G32** coding for a **protein p32** and a **metabolite m32** which can negatively regulates the protein synthesis of the protein $p32$.

[1] `http://genes.atspace.org/10.11.html`

It should be noted that the original example is composed of only the gene 'G32' and the protein 'p32'. However, to highlight the different level of the cell's components (gene, protein and metabolite), we added a metabolite 'm32'. This metabolite negatively regulate the protein synthesis of the protein p32. This negative regulation reaction is activated when the concentration of the metabolite m32 reached the threshold $S_{m32} = 0.8\ 10^{-6}\ mol/dm^3$ [322].

Therefore, in this network, the concentration of $p32$ is self-regulated and normally should remain between $0.2\ 10^{-6}\ mol/dm^3$ and $0.7\ 10^{-6}\ mol/dm^3$. When the concentration of $p32$ exceeds the threshold $S_{p32} = 0.7\ 10^{-6}\ mol/dm^3$, it is called an **Inhibition**, i.e. the protein $p32$ inhibits, or deactivates, the translation of its gene $G32$. However, when the concentration of $p32$ decreases and becomes lower than the threshold $S_{p32} = 0.2\ 10^{-6}\ mol/dm^3$, it is called an **Activation**, i.e. the protein $p32$ activates the translation of its gene $G32$. When the gene $G32$ is activated by the protein $p32$, it is called a **Translation**, in which we have a production of $p32$ thus increasing the value of its concentration. When the concentration of $m32$ exceeds the threshold $S_{m32} = 0.8\ 10^{-6}\ mol/dm^3$, the metabolite $m32$ negatively regulate the protein synthesis of $p32$ thus decreasing the value of its concentration, called a **Negative regulation**.

6.3 System theory

As discussed in part I, several approaches have been proposed for modelling and simulating biomolecular networks. Nevertheless, the majority of those approaches concern only a specific level of the network, such as metabolic or protein-protein interaction networks. The system theory seems to answer that need.

The logical-based modelling proposed in this chapter is then based on the systemic perspective for modelling complex biomolecular networks taking account of their multi-level aspect, and the heterogeneity of their molecular components and the diversity of the interactions among them. The system theory enables the definition of the system from the different axis of the system, allowing a better description of the dimensions and highlighting the relationship between them [323]. This theory aims at a successful performance of a collaborative simulation.

6.3.1 Complex systems

Various levels of system complexity have been proposed in the literature. The most cited classifications of complex systems are those of von Bertalanffy [324] and Le Moigne [325]. These classifications are both based on a model composed of nine levels of complexity of a system imagined by Kenneth E. Boulding in 1956. Table 6.1 presents these two classifications.

Table 6.1 – Levels of system complexity [7].

Von Bertalanffy [324]			**Le Moigne** [325]		
Level		*Description*	*Level*		*Description*
1	Static structures	Atoms, molecules, crystals, etc.	1	Passive system	It has nothing to do but being
2	Watchmaking movements	Clocks, solar systems, etc.	2	Active system	It is characterized by its activity
3	Self-regulatory mechanisms	Thermostat, servomechanisms, etc.	3	Regulated system	Emergence of regularities in its activity
4	Open Systems	Flames, cells and organisms in general, etc.	4	Informed system	Emergence of information in its representation
5	Low-level organizations	Plant type organisms, etc.	5	System decides	Emergence of decision-making processes
6	Animals	Increasing importance of information traffic, etc.	6	Memory system	Emergence of memory and importance of communication
7	Human	Symbolism, consequences, etc.	7	System is coordinated	Emergence of coordination or steering
8	Socio-cultural systems	Populations and organisms, etc.	8	System self-organizing	Emergence of the imagination and capacity for self-organization
9	Symbolic systems	Language, logic, mathematics, etc.	9	System autofinalizes	Emergence of consciousness and ability to finalize itself

Sharif and Irani [326] summarizes this notion of complexity into four concepts:

- *Self-organization.* The organization is the structuring of a whole according to the distribution of its elements on different levels. The system can create and recreate its structure.
- *Non-linearity.* Behaviours and responses are not deterministic and are influenced by the presence of non-linear relationships and feedback loops.
- *Order.* The implicit ability to exhibit linear or non-linear behaviours is a function of response to the stimulus.
- *Emerging behaviours.* The non-linear or self-organized interactions result in emerging properties and complex behaviours.

Table 6.2 depicts the main differences between simple and complex systems according to Glouberman and Zimmerman in [327].

Table 6.2 – Comparison between simple and complex systems.

Cluster	*Simple systems*	*Complex systems*
Theory	Linearity Absence of noise External system solution Adaptation in a static environment	Non-linearity Presence of noise Internal solution (part of the system) Interaction with the dynamic environment
Causality	Simple causality Determinist Certainty Focus on components Relationships determined by structures	Mutual causality Probabilistic Uncertainty Focus on relationships Interactive structures and relationships
Justification	Reductionism - Analysis Effectiveness, alignment and best practice measures	Holism - Synthesis Functioning of current relationships and feedback loops
Planning	Convergent Reducing characteristics Decision as an event An important issue involves a major change	Divergent Emerging characteristics Decision as emerging The size of the issue does not determine the size of the change

6.3.2 System theory objectives

As we have seen in the first part and referring to the definition of complex systems proposed in the previous section, we can note that biomolecular networks are also considered as complex systems. Indeed, they are highly structured (composed of several interconnected subnetworks), subject to strong variations (internal or external stimuli) and difficult to predict (their behaviour emerges according to their states and the state of their environment).

This reality explains the presence of the science of complexity that aims to 'overcome the simplifications and idealizations that lead to unrealistic views' [328]. This theory has been used in several fields because it provides a set of concepts to model the different characteristics of a complex system. According to Le Moigne, the analysis of a system requires to:

- give a systemic representation by defining an organization of structured sub-systems.
- ensure that the system behaviour can be simulated.
- improve the functioning of the system.

Definitions presented in this section refer to the work of Celine Bérard in [7].

6.3.3 System theory axes

The system theory was initially introduced in 1990 by Jean-Louis Le Moigne in his work [3]. From Le Moigne's point of view, a complex system cannot be reduced to a model composed of a set of equations describing its evolution (analytical, causal or deterministic). However, it confirms that any complex system is in constant evolution, and therefore can be defined according to four main axes: teleological, genetic, functional and ontological.

- The teleological axis including the structure of the system defines the objectives of the system. It consists of (*i*) defining the lifecycle, and (*ii*) understanding the structure of the system that will be used to predict its future evolution.
- The genetic axis including the behaviour of the system defines the evolution of the system over time.
- The functional axis describes the functions of the system components and the system itself: what the system is supposed to do.
- The ontological axis concerns the description and semantics of system resources. It aims to facilitate the human-computer interaction and access to system resources. As well as to make information more meaningful to the system and users.

Figure 6.2 depicts these axes and their objectives (this figure is inspired by the Systems theory in [3]).

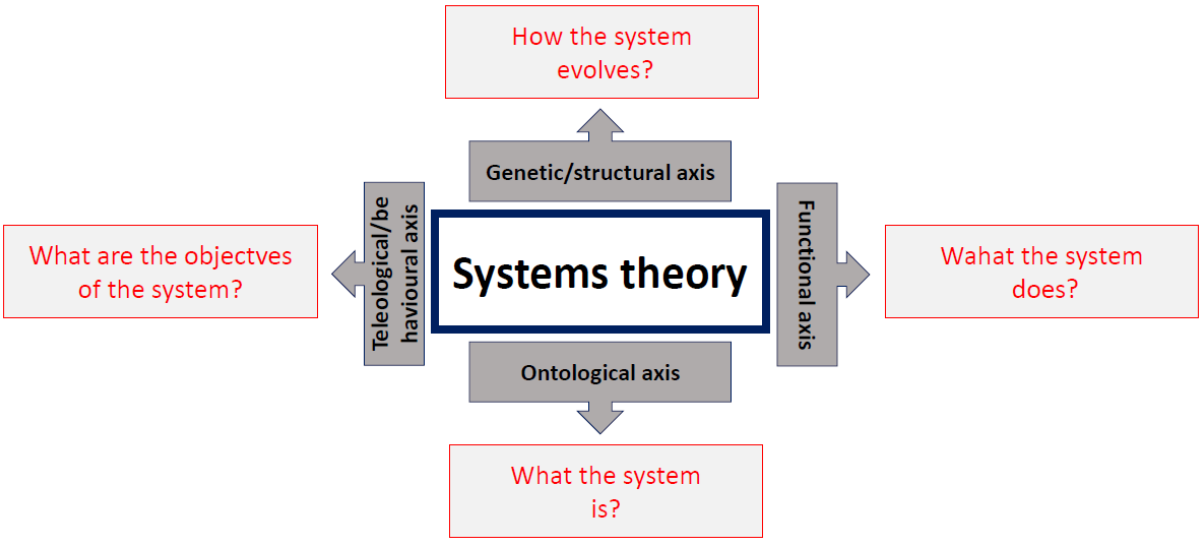


Figure 6.2 – The four axes of Systems theory according to Le Moigne [3].

6.4 Logic-based approach for modelling biomolecular networks

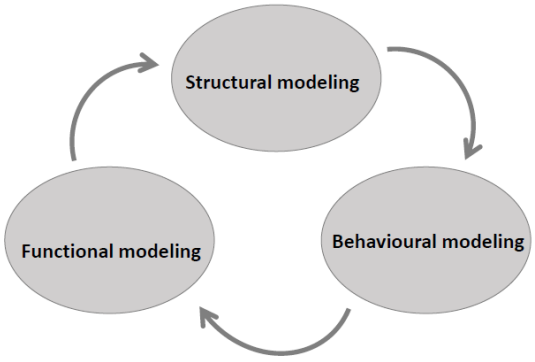


Figure 6.3 – The three axes of our proposed logical-based modelling.

One of the rules for complexity management during the modelling task involves separating knowledge of different nature. Here, we use a systemic approach to include the three types of knowledge required to describe a system [3] (Figure 6.3):

- **Structural modelling:** to describe the architecture of the network;
- **Functional modelling:** to describe the activities of each component of the network, and the associated conditions;
- **Behavioural modelling:** to describe how the network and its individual components evolve over time.

Hence, a biomolecular network BN can be represented by its *structure* SR, its *function* FR and its *behaviour* $CR_{[t_0,t_n]}$ that evolves over time t (generally a simulation interval $[t_0, t_n]$). Therefore, mathematically, the network BN is defined as follows:

$$BN = (SR, FR, CR_{[t_0,t_n]})$$

6.4.1 Structural modelling

The *structure* SR of the network is a directed graph defined by:

$$SR = (M, I) \qquad \text{where:}$$

- M represents all the molecules composing the network and denotes a finite set of nodes $M = \{m_1, m_2, \ldots, m_n\}$. We distinguish a tripartite partition of M: M_G the set of genes, M_P the set of proteins and M_M the set of metabolites.

$$M = M_G \cup M_P \cup M_M$$
$$M_x \cap M_y = \emptyset \qquad \text{where: } x, y \in \{G, P, M\} \text{ and } x \neq y.$$

- I represents the set of interactions among the network's molecules and denotes a finite set of edges $I = \{i_1, i_2, \ldots, i_m\}$. An edge $i = (m_s, m_d)$, (where $m_s, m_d \in M$) which starts at m_s (origin) and ends at m_d (destination) is also noted $m_s \rightarrow m_d$. Thus, for an edge $i \in I$, we denote by $s(i)$ the starting node and $d(i)$ the destination node.
  The partition of the graph nodes induces a partition into a range of different types of interactions:
  - three interactions among nodes of the same type (intraomic interactions): I_{GG} denotes the interactions (activation or inhibition) among genes, I_{PP} denotes the stable or transitional associations among proteins and I_{MM} denotes the interactions between metabolites (type of chemical reaction among reactants and products).
  - four interactions among the nodes of different types (interomic interactions): I_{GP} denotes the translation of genes encoding proteins, I_{PG} denotes the action of proteins (e.g. transcription factors) on genes, I_{PM} denotes the proteins acting on chemical reactions of metabolites (e.g. catalysis or hydrolysis), I_{MP} denotes the action of metabolites on proteins (e.g. negative or positive regulation).
  - two interactions I_{GM} and I_{MG} are not taken into account because there is no direct interaction between the genes and metabolites and vice versa.

$$I = I_{GG} \cup I_{PP} \cup I_{MM} \cup I_{GP} \cup I_{PM} \cup I_{MP} \cup I_{PG}$$
$$I_x \cap I_y = \emptyset \qquad \text{where: } x, y \in \{GG, PP, MM, GP, PM, MP, PG\} \text{ and } x \neq y.$$

6.4.2 Functional modelling

The *function* FR of the network associates the graph edges $i_{m_s,m_d} \in I$ with an interaction type and the condition that activates it. It depends on the type of the starting node m_s:

$$FR : \left| \; i_{m_s,m_d} \overset{FR}{\mapsto} \begin{cases} (TypeInteraction, Activation) \\ \quad \text{if} \;\; m_s \in M_G. \\ (TypeInteraction, \leqslant OR \geqslant, Threshold) \\ \quad \text{if} \;\; m_s \in M_P \cup M_M. \end{cases} \right.$$

- If the starting node is a gene ($m_s \in M_G$), the function FR associates to each edge $i_{m_s,m_d} \in I$, a couple consisting of a label $TypeInteraction$ that indicates whether the interaction is triggered on the activation or on the deactivation of the gene.
- If the starting node is a protein or a metabolite ($m_s \in M_G \cup M_M$), the function FR associates to each edge $i_{m_s,m_d} \in I$, a triplet consisting of a label $TypeInteraction$ representing the type of the interaction, a comparison operator ($\leqslant$ or $\geqslant$) that is used to compare the concentration of the starting node m_s to the threshold associated with this edge, and finally, the *Threshold* which defines the condition for activating the interaction i_{m_s,m_d} depending on the concentration of the starting node m_s.

In both cases, the label $TypeInteraction$ belongs to the set of concepts of the Interaction Ontology proposed by Van Landeghem et al. [4] (Figure 6.4). As shown in Table 6.3, the possible types depend on the type of the edge.

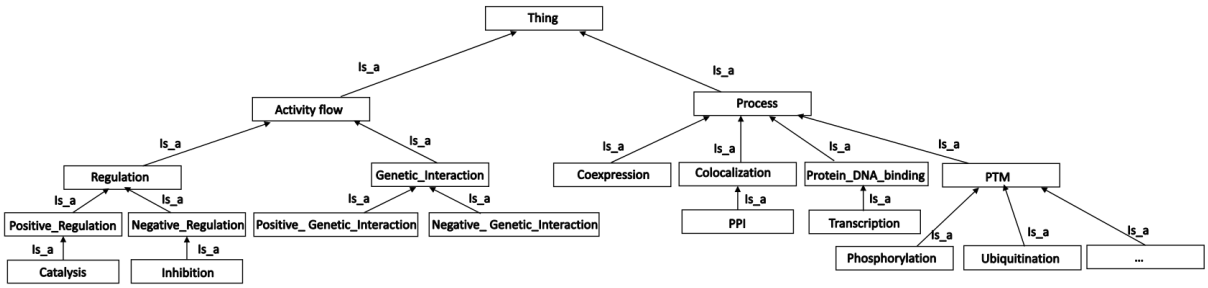


Figure 6.4 – A subset of the taxonomy of the Interaction Ontology [4].

Table 6.3 – Possible interaction types depending on the type of graph edge.

TypeInteraction	*Intraomic_Interactions*			*Interomic_Interactions*			
	I_{GG}	I_{PP}	I_{MM}	I_{GP}	I_{PG}	I_{PM}	I_{MP}
Positive Regulation (Catalysis/Hydrolysis)	-	✓	✓	-	✓	✓	✓
Negative Regulation (Inhibition)	-	✓	✓	-	✓	✓	✓
Positive genetic interaction	✓	-	-	-	-	-	-
Negative genetic interaction	✓	-	-	-	-	-	-
Colocalization	-	✓	✓	-	-	✓	✓
Coexpression	-	✓	✓	✓	✓	✓	✓
Transcription	-	-	-	-	✓	-	-
Phosphorylation	-	✓	-	-	-	-	✓
Dephosphrylation	-	✓	-	-	-	-	✓

6.4.3 Behavioural modelling

Complex biomolecular networks are dynamic systems characterised by continuous interactions.

Thus, in order to model the dynamic evolution of the network and reproduce its behaviour over time, we have implemented a behavioural simulation following the discrete-event formalism. This simulation allows studying the behaviour of the network through successive transitions.

6.4.3.1 State of the network

The *state of the network* at a given time is defined by a function $en(m,t)$ which assigns a state to each node at time t.

$$en : \left| (m,t) \overset{en}{\mapsto} \begin{cases} Activation \in \{True, False\} \\ \quad \text{if } m \in M_G. \\ [c_m(t)] \in \mathbb{R} \\ \quad \text{if } m \in M_P \cup M_M. \end{cases} \right.$$

- For all $m \in M_P \cup M_M$: $en(m,t) = [c_m(t)] \in \mathbb{R}$ where: $c_m(t)$: the value of the concentration of the molecule denoted by the node m at a given time t.
- For all $m \in M_G$: $en(m,t) = Activation$ where: $Activation \in \{True, False\}$. Associating a gene with a concentration is not meaningful. Instead, a gene may have two specific states, activated or not.

We define $ER(t)$, the state of the network at time t, by a set representing the states of all components in the network at time t.

$$ER(t) = \langle en(m_1,t), en(m_2,t), ..., en(m_n,t) \rangle$$

6.4.3.2 Transition of the network state

For a node $m \in M$, we define $ie(m)$ (resp. $oe(m)$) the set of incoming edges (resp. outgoing edges) on m, defined as follows:

$$ie(m) = \{i \mid d(i) = m\}$$
$$oe(m) = \{i \mid s(i) = m\}$$

We also define $Pred(m)$ the set of predecessor nodes on the node m such that:

$$Pred(m) = \{n \in M;\ \exists i \in I \mid s(i) = n \text{ and } d(i) = m\}$$

The state of a node at time $t+1$ depends on its state at time t, as well as the possible influence of each of its incoming edges. This influence depends on the state of the starting node of the edges in question.

For each node m, we define an *aggregate function* A_m (relating to the node m) which computes the evolution of the node status between two successive generations of the simulation. This *aggregate function* A_m depends on the current state of the node m, the state of its predecessor nodes $Pred(m)$ and the characteristics of its incoming edges $ie(m)$.

$$en(m,t+1) = A_m(en(m,t),\ ie(m),\ en(n,t)\ ; n \in Pred(m))$$

6.4.3.3 Steering the network to a given state

A state transition in the network occurs by changing the state of at least one of its nodes. The changes of a node state (e.g. changes in the molecule concentration) can occur either by a *internal stimulus* modelled by the aggregate function described above, or by a *external stimulus* generated outside the cell.

We define a stimulus as an event that causes changes in the state of the molecule on which it acts and therefore changes the state of the whole network.

An *external stimulus* S is represented by a triplet $[t, m, \Delta_c]$, where:

- t is the time of introduction of the stimulus S.
- m is the node targeted by the stimulus S.
- Δ_c is the change in concentration caused by the stimulus S and depends on the type of the node:
  - If $m \in M_G$, Δ_c determines the activation or deactivation of a gene: $\Delta_c \in \{Activated, Deactivated\}$.
  - Else, if $m \in M_P \cup M_M$, Δ_c represents the concentration change caused by the stimulus S: $\Delta_c \in \mathbb{R}$.

We denote $ER(t)$, with $t \in \mathbb{N}$, the state of the network at time $T(t) = t_0 + t.\Delta T$ (where ΔT is the time step and t_0 the initial time of the simulation).

To simulate the different transition states of a network, we specify a state $ER(0)$ at time t_0 and a time step size ΔT. Then, the successive states $ER(t+1)$ are computed from the current state $ER(t)$ according to the interactions and the aggregate functions defined by the network, and the external stimuli.

At a given time $t+1$, for each $m \in M$ we have:

- If there are no external stimuli in time t for the node m then:

$$en(m, t+1) = A_m(en(m,t),\ ie(m),\ en(n,t)) \quad \text{where: } n \in Pred(m)$$

  - Else If $m \in M_G$: $\quad en(m, t+1) = \Delta_c$
  - Else ($m \in M_P \cup M_M$):

$$en(m, t+1) = A_m(en(m,t),\ ie(m),\ en(n,t)\ ) + \Delta_c \qquad \text{where: } n \in Pred(m)$$

6.4.3.4 Behaviour

The *behaviour* of the network $CR_{[t_0,t_n]}$ is given by the sequence of its successive states during the simulation time.

$$CR_{[t_0,t_n]} = [ER(0), ER(1), ..., ER(n)]$$

Thus, the behaviour of the network extends between two distinct times t_0 and t_n forming the simulation interval $[t_0, t_n]$.

6.5 Application to the motivating example

Table 6.4 presents the logic modelling of the example network shown in Figure 6.1 and presented in Section 6.2.

6.6 Summary

The logic modelling presented in this chapter aims to provide biologists wishing to study complex biomolecular networks with a simple and comprehensive modelling approach to assist them in building their networks. This method consists of formalizing, building and analysing the biological knowledge of the different elements on which the biomolecular network is based.

In this chapter, we have focused only on the structural, functional and behavioural quality of biomolecular networks. However, in order to have a complete and more realistic modelling of these networks, it is essential to cover all the knowledge that manages the rules of their behaviour and organization. This is what semantic modelling invites us to do in the next chapter.

<table>
<tr><td>Structure</td><td>Nodes: $M = \{ G32 , p32 , m32 \}$
$\{G32\} \in M_G$; $\{p32\} \in M_P$; $\{m32\} \in M_M$</td></tr>
<tr><td></td><td>Edges: $I = \{ i_1 , i_2 , i_3 , i_4 \}$
$\{i_1, i_2\} \in I_{PG}$; $\{i_3\} \in I_{GP}$ and $\{i_4\} \in I_{MP}$
$i_1 :$ $s(i_1) = p32$ et $d(i_1) = G32$
$i_2 :$ $s(i_2) = p32$ et $d(i_2) = G32$
$i_3 :$ $s(i_3) = G32$ et $d(i_3) = p32$
$i_4 :$ $s(i_4) = m32$ and $d(i_4) = p32$</td></tr>
<tr><td>Function</td><td>Edges: $i_1 \overset{FR}{\mapsto} (Activation, \leq, 0.2)$
$i_2 \overset{FR}{\mapsto} (Inhibition, \geq, 0.7)$
$i_3 \overset{FR}{\mapsto} (Transcription, Activation)$
$i_4 \overset{FR}{\mapsto} (NegativeRegulation, \geq, 0.8)$</td></tr>
<tr><td>Behaviour</td><td>Aggregate functions
A_{p32}:
<table>
<tr><td colspan="2">Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_3</td><td>i_4</td><td>State of c_{p32}</td></tr>
<tr><td>Deactivated</td><td>< 0.8</td><td>$\Delta_1 = 0$</td></tr>
<tr><td>Activated</td><td>< 0.8</td><td>$\Delta_2 > 0$</td></tr>
<tr><td>Deactivated</td><td>≥ 0.8</td><td>$\Delta_3 < 0$</td></tr>
<tr><td>Activated</td><td>≥ 0.8</td><td>Δ_4</td></tr>
</table>
A_{G32}:
<table>
<tr><td colspan="2">Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_1</td><td>i_2</td><td>State of $G32$</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td><td>Maintained state</td></tr>
<tr><td>Activated</td><td>Deactivated</td><td>Activated</td></tr>
<tr><td>Deactivated</td><td>Activated</td><td>Deactivated</td></tr>
</table>
A_{m32}:
<table>
<tr><td>Incoming edges</td><td>Evolution</td></tr>
<tr><td>No incoming edges</td><td>State of $m32$</td></tr>
<tr><td>—</td><td>Maintained state</td></tr>
</table>
States:
$CR = \{\langle 0, [min_{p32}; 0.2[, [min_{m32}, 0.8[\rangle, \langle 0, [min_{p32}; 0.2[, [0.8, max_{m32}[\rangle,$
$\langle 1, [min_{p32}; 0.2[, [min_{m32}, 0.8[\rangle, \langle 1, [min_{p32}; 0.2[, [0.8, max_{m32}[\rangle,$
$\langle 0, [0.2, 0.7[, [min_{m32}, 0.8[\rangle, \langle 0, [0.2, 0.7[, [0.8, max_{m32}[\rangle,$
$\langle 1, [0.2, 0.7[, [min_{m32}, 0.8[\rangle, \langle 1, [0.2, 0.7[, [0.8, max_{m32}[\rangle,$
$\langle 0, [0.7, max_{p32}[, [min_{m32}, 0.8[\rangle, \langle 0, [0.7, max_{p32}[, [0.8, max_{m32}[\rangle,$
$\langle 1, [0.7, max_{p32}[, [min_{m32}, 0.8[\rangle, \langle 1, [0.7, max_{p32}[, [0.8, max_{m32}[\rangle\}$</td></tr>
</table>

Table 6.4 – Logical modelling of the autoregulation of the bacteriophage T4 gene 32.

Chapter 7

Semantic modelling of complex biomolecular networks

Contents

7.1 Introduction

In the previous chapter, we proposed a logical-based modelling that provides the different elements on which the biomolecular network is based. However, to obtain an optimal and more realistic modelling, we want to enhance it with an additional semantic layer. Semantic technologies, especially ontologies, are one of the tools frequently used for this purpose. In fact, they are indispensable for understanding the semantic knowledge about the functioning of cells at a molecular level.

In this chapter, we present a semantic approach for modelling biomolecular networks and describe the proposed Biomolecular Network Ontology (BNO) created specially to address the needs of analysing the complex biomolecular network's behaviour. This ontology provides a foundation for qualitative simulation of these networks. The BNO ontology is freely available at `https://github.com/AliAyadi/The-Biomolecular-Network-Ontology`.

The first section of this chapter focuses on the proposed semantic approach for modelling the semantics of complex biomolecular networks. We detail each one of the ontologies that constitute it and the relationship among them. A second section is dedicated especially to describe in detail the main components of the BNO ontology on which the transittability of complex biomolecular networks are meant to be contextualised. The last section presents the application of the proposed ontology through the case study related to the biological domain, the bacteriophage T4 gene 32 already used in Chapter 6.

7.2 Semantic approach for analysing the transittability of complex biomolecular networks

Modelling the behaviour of complex biomolecular networks requires, first and foremost, to formalize the domain knowledge. However, it is not sufficient to simply describe it. Certainly, the behaviour of biomolecular networks is investigated through appropriate semantic structures for the description of their components that must not be overlooked. Thus, the use of a formalized language such as ontologies provides a rich description but also allows to perform reasoning. Therefore, in this section, we propose a semantic architecture composed of four ontologies: three of them already exist in the literature, the Gene Ontology (GO) [170, 329], the Simple Event Model Ontology (SEMO) [330], the Time Ontology (TO) [331] and we are developing the Biomolecular Network Ontology (BNO). Linked together, these ontologies provide the necessary concepts for modelling the dynamic behaviour and the transition states of a complex biomolecular network. We will briefly present the general architecture of the ontological process and describe the set of ontologies which compose our approach.

7.2.1 The global architecture

We propose a semantic approach that aims to enrich the structural description of biomolecular networks by contextual knowledge concerning their state transitions, the events that can steer these transitions but also their entire temporal context linked to this information. Thus, we present an approach for understanding the transittability [6] of biomolecular networks which is basically composed of four ontologies: the Gene Ontology (GO) [170, 329], the Simple Event Model Ontology (SEMO) [330], the Time Ontology (TO) [331] and our development, the Biomolecular Network Ontology (BNO).

Figure 7.1 describes the global architecture of our semantic approach for analysing the transittability of complex biomolecular networks.

This semantic architecture is based and follows the logical-based modelling of complex biomolecular networks detailed in Section 6.4. In fact, this correspondence between the logical and semantic modelling is presented in Figure 7.2. The Biomolecular Network Ontology describes the static structure of the biomolecular network which has already been presented in Section 6.4.1. Merging with the Simple Event Model Ontology, the Biomolecular Network Ontology describes what can be carried out by each component of the biomolecular network and the conditions for these activities, this notion was detailed in Section 6.4.2. Finally, the Biomolecular Network Ontology, the Simple Event Model Ontology and the Time Ontology describe how the biomolecular network and its individual components evolve over time which is clearly mentioned in the previous Section 6.4.3.

These ontologies are described in more detail in the sections below.

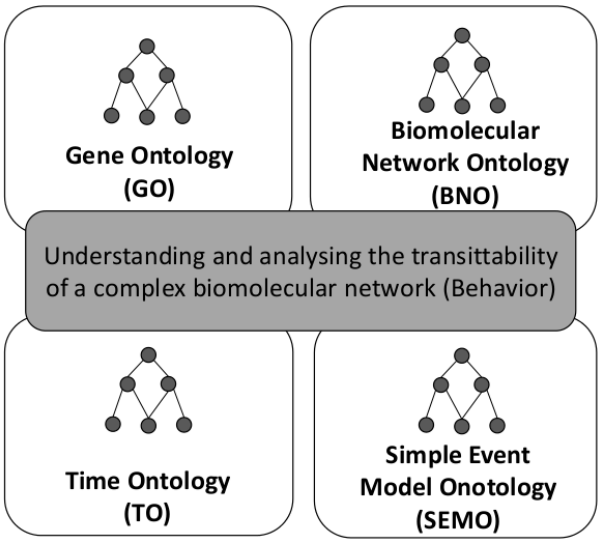


Figure 7.1 – Global architecture of our proposed semantic modelling.

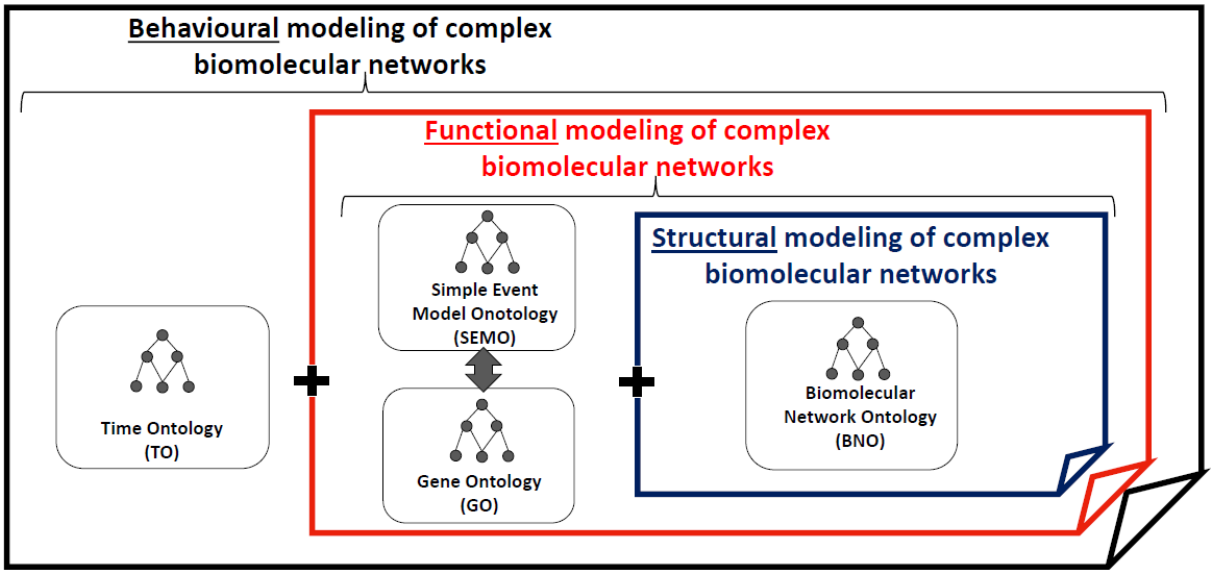


Figure 7.2 – Correspondence between the logical and semantic modelling.

7.2.2 The Gene Ontology (GO)

In this study, the Gene ontology[1] is considered as a core ontology. It ensures the description and the classification of cellular components. As well as, it provides a structured terminology for the description of gene functions and processes, and the relationships among these components [332].

We chose to use the Gene Ontology for the following reasons, (1) it is an initiative of several genomic databases such as the Saccharomyces Genome Database (SGD), the Drosophila genome database (FlyBase), etc. to build a generic ontology for describing the role of genes and proteins, (2) it is the most developed and most used in biology (since 2000), and (3) it provides annotation files about large number of cellular entities.

[1] `http://www.geneontology.org`

7.2.3 The Simple Event Model Ontology (SEMO)

The Simple Event Model ontology[2] proposed by Van Hage et al. [330] provides the necessary knowledge for the description of events. The ontological architecture of the Simple Event Model ontology consists of four basic concepts: *Event* that specifies what is happening, *Actor* that indicates the participants of an event, *Place* that describes the location where the event happened, and *Time* that describes the moment.

We chose to use the Simple Event Model ontology because it provides the necessary concepts to describe and model events in various subject domains.

7.2.4 The Time Ontology (TO)

The Time ontology[3] developed by Hobbs and Pan [331] enables a more intuitive use of the time dimension while making the most of semantic knowledge. It gives a rich vocabulary to describe the topological relationships that may exist between time points and intervals and also provides information about time.

The main concepts of this temporal ontology can be summarized as *TemporalEntity* which consists of two sub-classes *Instant* and *ProperInterval*, *DurationDescription*, *DateTimeDescription*, *TemporalUnit*, etc. Also, it contains several proprerties such as *hasDurationDescription*, *intervalStarts*, *hasDateTimeDescription*, etc.

We chose to use the Time Ontology because of its basic structure that is not specific to a particular application and because it is simple to adapt it in our context.

7.2.5 The Biomolecular Network Ontology (BNO)

We developed an ontology the '*Biomolecular Network Ontology (BNO)*' which aims to describe the domain knowledge of complex biomolecular networks in their static state. This ontology provides information on the biomolecular network and its components (nodes, interactions, states, transition states, etc.) and an indication of the network's context such as the type of sub-network, the type of node, the conditions and nature of interactions, etc. This allows to precisely analyse and interpret the semantic context in order to achieve intelligent modelling of biomolecular networks and their state changes.

The BNO ontology is the major contribution of this chapter, that is why an entire section has been devoted to detail this ontology (Section 7.3).

7.2.6 The relations among these ontologies

Concepts in the Biomolecular Network ontology are linked to the Gene ontology concepts. In fact, the concepts of the Gene ontology are used to enrich the definitions of the concepts of the Biomolecular Network ontology by two relations: an equivalence relation *owl:equivalenceClass* and a specification relation *owl:subClassOf*. Some instances of these relations are shown in Figure 7.3. For example, as described in Figure 7.3a, after inference the concept *BNO:Protein* will be specialized by the concept *GO:beta-galactosidase* (GO: 0009341) because the *BNO:Node* concept is equivalent to the concept *GO:cellular_component* (GO: 0005575). Other examples of these links are illustrated by Table 7.1. The Biomolecular Network ontology is also linked with the Simple Event Model Ontology through the *BNO:Node* concept, in fact, a *SEM:event* can stimulate a molecular entity (represented by the concept *BNO:Node*). The Simple Event Model ontology will be used to describe the states of *BNO:Node* and its behaviour.

Moreover, the Time Ontology (TO) has been integrated into the Simple Event Model ontology. The concept *SEM:Time* was made equivalent to the concept *TO:TemporalEntity* which represents the root of the Time ontology. Hence, the property *SEM:hasTime* will connect the Simple Event Model ontology to the Time ontology and, as a consequence, the diverse types of temporal concepts will be defined as specializations of the class *SEM:Time*. Figure 7.3b shows the use of this principle. Thus, we can exploit the wealth of temporal concepts provided by this temporal ontology to describe the *SEM:event* class.

Using these relationships it is possible to merge these ontologies to formalize the necessary knowledge to study the state changes of the biomolecular network's behaviour.

[2] http://semanticweb.cs.vu.nl/2009/11/sem/
[3] https://www.w3.org/TR/owl-time/

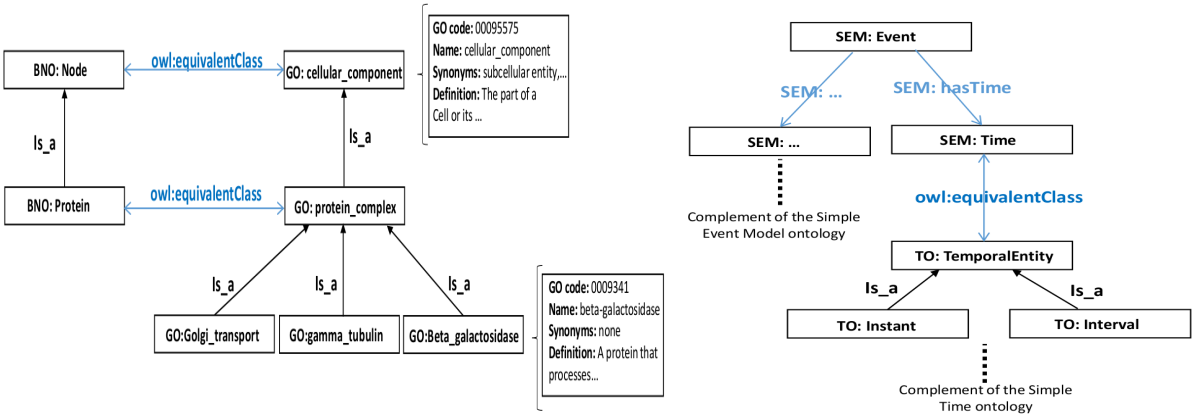


Figure 7.3 – Example of merging: 7.3a The Gene ontology concepts to the Biomolecular Network ontology concepts. 7.3b The Time ontology within the Simple Event Model ontology.

Table 7.1 – Linking of Gene Ontology concepts to the Biomolecular Network ontology.

Type of relationship	Biomolecular Network Ontology concept name	Gene Ontology concept name
Equivalence: $BNO\ 'owl:equivalenceClass'\ GO$	$BNO:Node$	$GO:cellular\_component$
	$BNO:Protein$	$GO:protein\_complex$
Subclass: $BNO\ 'owl:subClassOf'\ GO$	$BNO:Interaction$	$GO:biological\_process$

7.3 The Biomolecular Network Ontology

To study the dynamic behaviour and the transition states of biomolecular networks, it is required to model their domain knowledge. Therefore, we developed the Biomolecular Network ontology. This ontology is the major contribution of this chapter, it is intended to describe exhaustively the field of complex biomolecular networks by describing the static aspect of their structure. It was defined in collaboration with domain experts.

Figure 7.15 presents the Biomolecular Network ontology. We use the graphical notation for OWL ontologies defined by Brockmans et al. [333] and Bārzdiņš et al. [334] where boxes are OWL concepts; full lines are object properties and dotted lines are data properties. Full lines can be labelled to indicate restrictions meaning that the range of the relationship is specialized.

7.3.1 Development

As described in Figure 7.15 and 7.4, we have developed the BNO ontology using the OWL-language [335] under the Protégé editor. Concepts, relations, and attributes were modelled as *concepts*, *object properties* and *data properties*, respectively. Axioms were represented in Protégé using diverse OWL restrictions (existential restrictions, universal restrictions, cardinality restrictions, hasValue restrictions), characteristics of object property, and datatype restrictions.

7.3.2 The key concepts

Only a few of the object properties restrictions are displayed in Figure 7.15 for the sake of clarity. This domain ontology consists of five main concepts:

- The *Biomolecular_Network* class: This class includes the different types of complex biomolecular networks. As mentioned earlier in Section 1.4, the complex biomolecular network can be composed by Gene Regulatory networks (GRNs), Protein-Protein Interaction networks (PPINs) and Metabolic networks (MNs) which correspond to the following concepts: *Genomic_Network*, *Proteomic_Network* and *Metabolomic_Network*.

  These types of networks can be connected to the other ontology's concepts through three properties, *has_node* that depicts its cellular components, *has_interaction* that describes the interactions

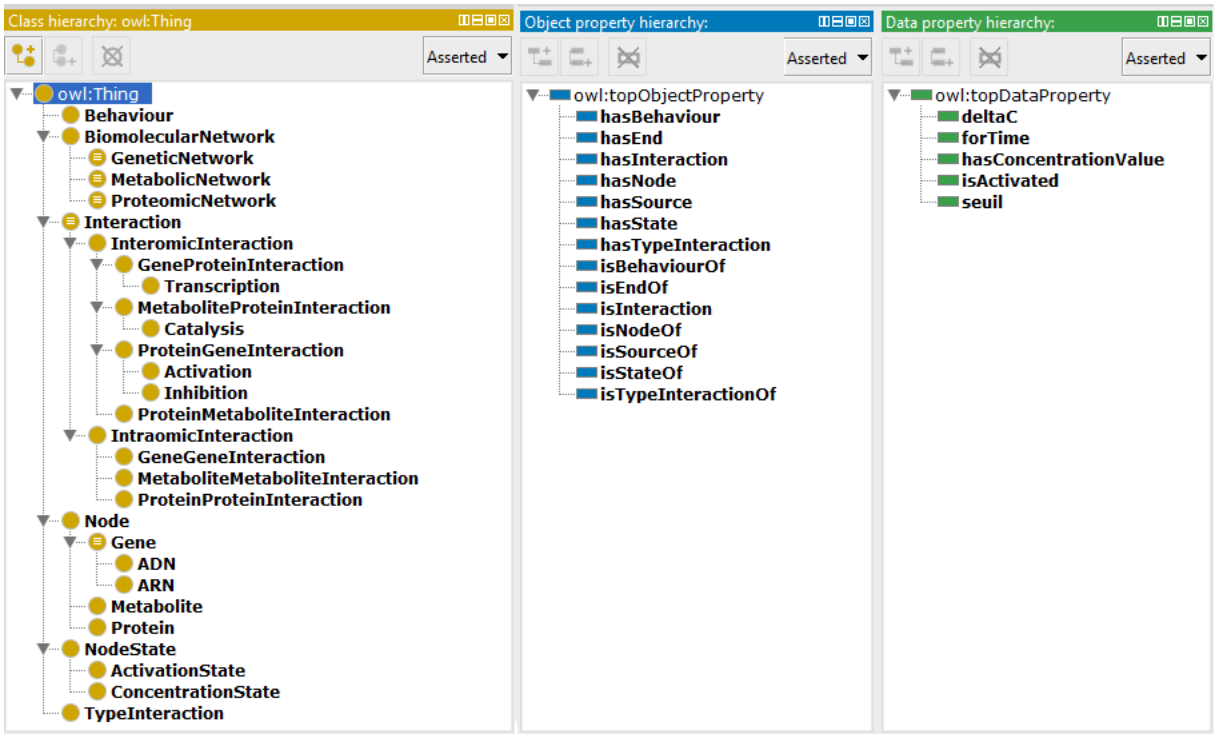


Figure 7.4 – The Biomolecular Network Ontology: hierarchy of concepts, hierarchy of properties and hierarchy of data properties.

linked to its components and the property *has_node only* that specifies exactly the nature and type of its components.

The instances of these concepts will be defined later, among these instances we will focus on the $BacteriophageT4G32$ instance in Section 7.4.1.

- The *Node* class: This class contains the different types of cellular entities M that constitute the biomolecular network. In fact, we can identify three sub-classes: the *Gene* which describes the set of genes M_G, the set of proteins *Protein* which models the set M_P and the *Metabolite* which describes the set of metabolites M_M. The *Gene* sub-class is itself divided into two types the $BNO : DNA$ and $BNO : RNA$. The class *Node* is connected with the *Node_State* through the property *has_state*. These BNO concepts are detailed on the logical-based modelling in Section 6.4.1.

- The *Interaction* class: This class covers all the diverse types of interactions that can be operated among the different types of nodes of the biomolecular network. This class consists of two sub-classes, *Intraomic_Interactions* that covers the interactions between molecular components of the same type and the class *Interomic_Interaction* that describes the interactions between molecular components of the different type. This class is connected to the *Node* class via two properties, *has_source* and *has_end*. These BNO concepts are developed through the logical-based modelling in Section 6.4.2.

- The *Node_State* class contains the possible states of the nodes. This class is composed of two sub-classes, the *ConcentrationState* and the *ActivationState*.

- The *Interaction_Type* class allows to specify the types and the nature of the interaction among cellular components. This class is linked to the *BNO:Interaction* class through the properties *Has_type*.

  The instances of this class belong to the set of concepts of the Interaction Ontology proposed by Van Landeghem et al. [4] (in Figure 6.4. To successfully integrate the main Interaction ontology concepts (*IO:Activity_flow* and *IO:Process*) with the Biomolecular Network ontology, we create an

abstract BNO UML *BNO:Interaction_ Type* to generalise those two Interaction ontology concepts (Figure 7.15).

Table 7.2 and Figure 7.4 show the most important BNO concepts.

Table 7.2 – A summary of concepts in the Biomolecular Network ontology. The left column presents the five major concepts and their immediate sub-classes. The right column presents the description of these concepts.

BNO ontology concepts	Description
BNO:BiomolecularNetwork	defines the different kinds of complex biomolecular networks.
BNO:GenomicNetwork	defines the interactions among genes forming Gene Regulatory networks.
BNO:ProteomicNetwork	defines the interactions among proteins forming Protein-Protein Interaction networks.
BNO:MetabolomicNetwork	defines the interactions among proteins forming Metabolic networks.
BNO:Node	defines the different types of cellular entities.
BNO:Gene	describes the set of genes M_G.
BNO:DNA	describes the set DNA.
BNO:RNA	describes the set of RNA.
BNO:Protein	describes the set proteins M_P.
BNO:Metabolite	describes the set metabolites M_M.
BNO:Interaction	defines all the types of interactions operated among the nodes.
BNO:IntraomicInteraction	defines the interactions between molecular components of the same type.
BNO:I_GG	defines the interactions between genes.
BNO:I_PP	defines the interactions between proteins.
BNO:I_MM	defines the interactions between metabolites.
BNO:InteromicInteraction	defines the interactions between molecular components of the different type.
BNO:I_GP	defines the interactions between genes and proteins.
BNO:I_PG	defines the interactions between proteins and genes.
BNO:I_PM	defines the interactions between proteins and metabolites.
BNO:I_MP	defines the interactions between metabolites and proteins.
BNO:NodeState	defines the possible states of the nodes.
BNO:ActivationState	defines the states of the genes.
BNO:ConcentrationState	defines the concentration of the proteins and metabolites.
BNO:InteractionType	defines the nature of the interaction among cellular components.

7.3.3 The major properties and data types

After the definition of the major concepts of the BNO ontology and in order to describe the semantic relations among them, we define the domain, range, property type and inverse properties as constraint conditions. The different properties and data types of the BNO ontology are explained below.

- $hasBehaviour(object1, object2)$: where object1 is a $BiomolecularNetwork$ and object2 is a $Behaviour$.
- $hasInteraction(object1, object2)$: where object1 is a $BiomolecularNetwork$ and object2 is an $Interaction$.
- $hasNode(object1, object2)$: where object1 is a $BiomolecularNetwork$ and object2 is a $Node$.
- $hasSource(object1, object2)$: where object1 is an $Interaction$ and object2 is a $Node$.
- $hasEnd(object1, object2)$: where object1 is an $Interaction$ and object2 is a $Node$.
- $hasState(object1, object2)$: where object1 is a $Node$ and object2 is a $NodeState$.
- $hasTypeInteraction(object1, object2)$: where object is an $Interaction$ and object2 is a $TypeInteraction$.
- $deltaC(object, dc)$: where object is an $Interaction$ and dc is a $float$ representing the change in concentration caused by the interaction.
- $forTime(object, t)$: where object is a $NodeState$ and t is a int representing its time.
- $hasConcentrationValue(object, c)$: where object is a $Protein$ or a $Metabolite$ and c is a $float$ representing the value of its concentration.

- *isActivated*(*object*, *bv*): where object is an *Gene* and *bv* is a *boolean* equal to true if the gene is activated.
- *threshold*(*object*, $t^{'}$): where object is the threshold of an *Interaction* and $t^{'}$ a comparison operator ($\leq$ or $\geq$) determining the minimum and maximum threshold, respectively.

Table 7.3 summarises the major properties of the BNO ontology, including their domain, range and inverse.

Table 7.3 – A summary of the properties, including their domain, range and inverse.

BNO ontology properties	Domain	Range	Inverse
hasBehaviour	BiomolecularNetwork	Behaviour	isBehaviourOf
hasInteraction	BiomolecularNetwork	Interaction	isInteractionOf
hasNode	BiomolecularNetwork	Node	isNodeOf
hasSource	BiomolecularNetwork	Node	isSourceOf
hasEnd	Interaction	Node	isEndOf
hasState	Interaction	State	isStateOf
hasTypeInteraction	Interaction	TypeInteraction	isTypeInteractionOf

7.4 Application to the motivating example: the bacteriophage T4 gene 32

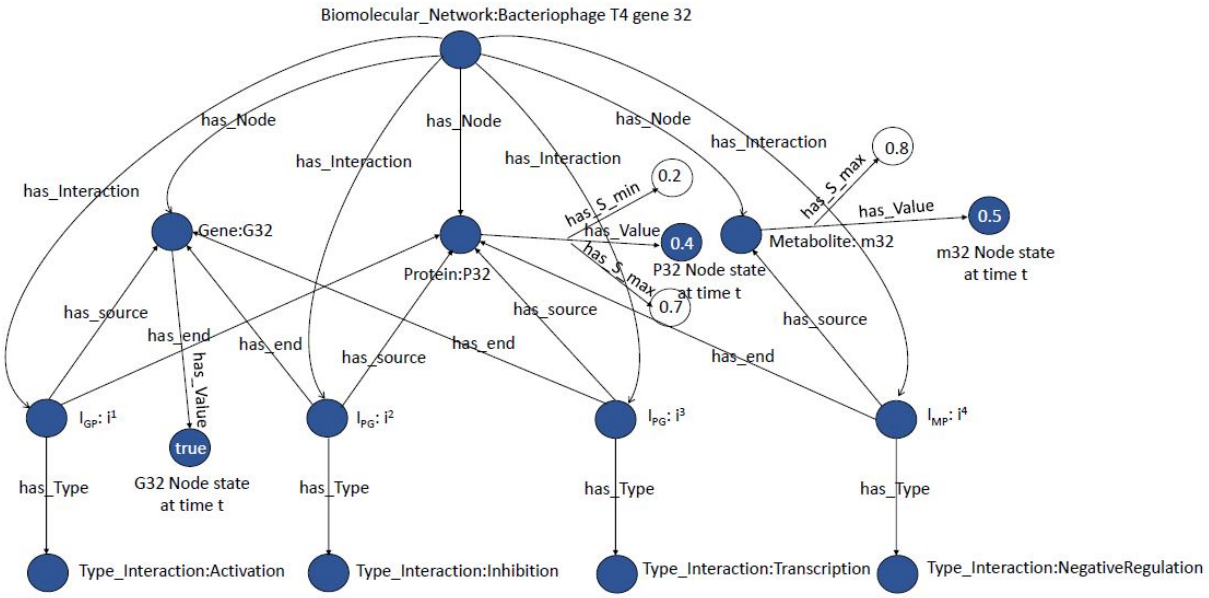


Figure 7.5 – Instantiation of the BNO ontology for the given example.

The aim of this section is to illustrate the proposed BNO ontology for reasoning and inferring new knowledge with sets of rules expressed in SWRL [335]. To do this, we test its performance by using the real example presented in Section 6.2, the bacteriophage T4 gene 32.

7.4.1 Instantiation of the BNO ontology

Figure 7.5 presents the instantiation of the BNO ontology for the given example of the bacteriophage T4 gene 32. The BNO ontology provides detailed and rigorous semantics to model this biomolecular network.

We use the Protégé editor to instantiate the BNO ontology for the bacteriophage T4 gene 32. Figure 7.6 illustrates the nodes instantiations respectively, the gene $G32$, protein $p32$ and metabolite $m32$. The instantiations of the four reactions are detailed in Figure 7.7.

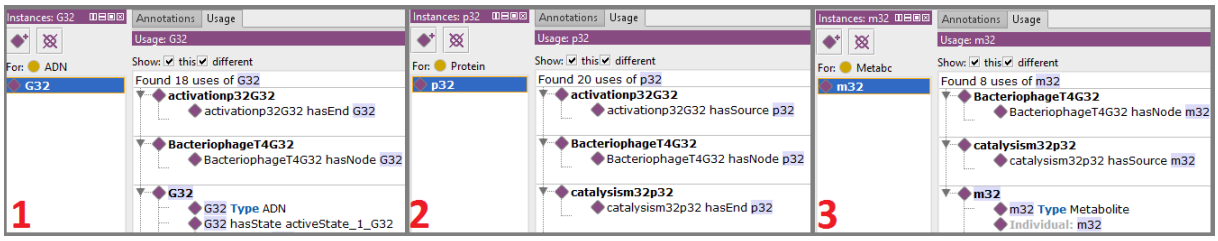


Figure 7.6 – A snapshot look at the BNO node instances associated with the given example displaying respectively: (1) the gene $G32$, (2) the protein $p32$ and (3) the metabolite $m32$.

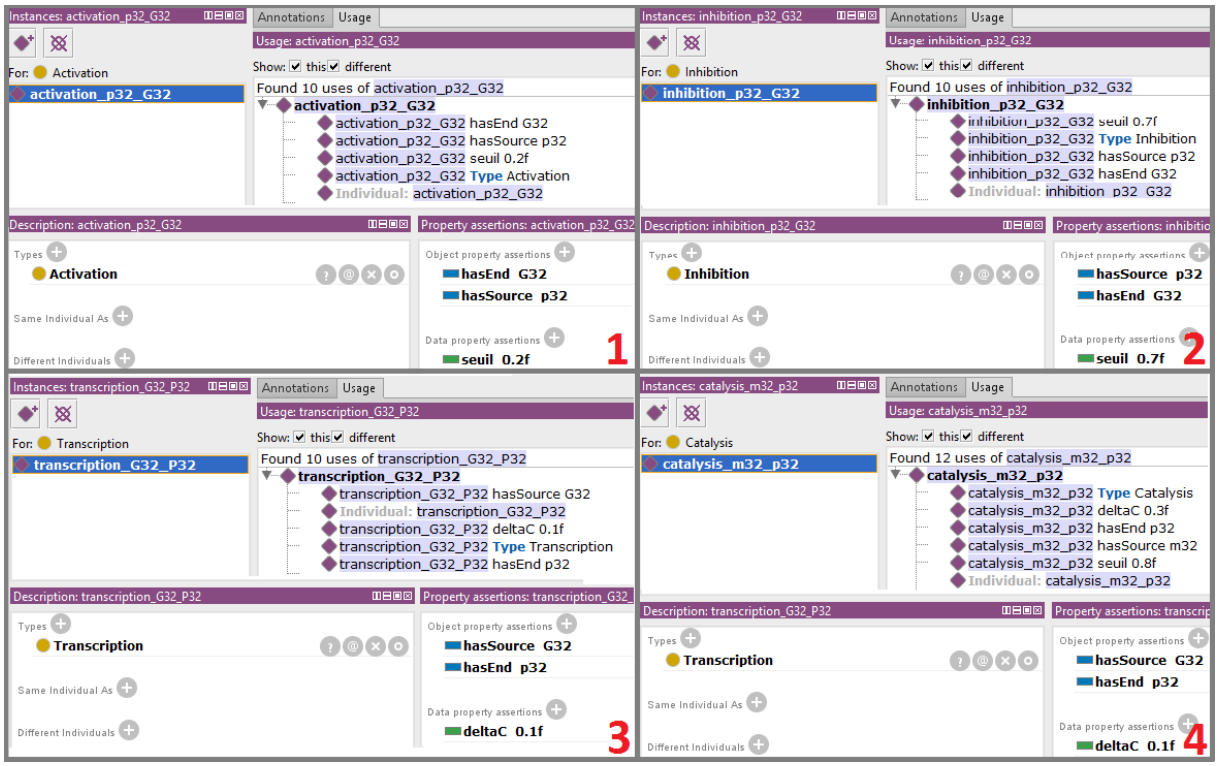


Figure 7.7 – A snapshot look at the BNO interaction instances associated with the given example displaying respectively: (1) Activation, (2) Inhibition, (3) Transcription and (4) Catalysis.

7.4.2 SWRL rule-based reasoning

As detailed in Section 3.6.1, the Semantic Web Rule Language (SWRL) is an ontological language based on OWL-DL and OWL-Lite that expresses the rule description language based on OWL [336]. SWRL can be used to write rules to reason about OWL individuals and infer new knowledge about those individuals. The rules in SWRL are implication rules, and follow this syntax: *antecedent* $\rightarrow$ *consequent*. This form means that the consequent must be true when the antecedent is satisfied. In SWRL rules, the symbol '$\wedge$' means conjunction, '$?x$' is a variable, '$\rightarrow$' means implication. A symbol without the leading '?' denotes the name of an instance (an individual) in the ontology. These SWRL rules can provide additional expressiveness to OWL-based ontologies. Thus we adopt these SWRL rules to build the reasoning rules

in order to represent the dynamic aspect of the biomolecular network. During this reasoning, inferences are made, classifying the instances of the BNO ontology and associating new properties to create instances while maintaining logical consistency.

7.4.2.1 Inhibition SWRL rule

The following rule models the *inhibition* interaction. When the concentration of the protein $p32$ exceeds the threshold 0.7 10^{-6} Mol/L^{-1}, it inhibits the translation of its gene $G32$.

$$ADN(?g) \wedge hasState(?g,\ ?gs1) \wedge forTime(?gs1,\ ?t) \wedge hasState(?g,\ ?gs2) \wedge forTime(?gs2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \wedge Protein(?p) \wedge Activation(?activ) \wedge hasSource(?activ,\ ?p) \wedge hasEnd(?activ,\ ?g) \wedge hasState(?p,\ ?ps) \wedge forTime(?ps,\ ?t) \wedge hasConcentrationValue(?ps,\ ?c) \wedge swrlb{:}greaterThanOrEqual(?c,\ 0.7) \rightarrow isActivated(?gs2,\ false)$$

As depicted in Figure 7.8, the results of this rule means that, *If there is a gene g having a state gs equal to true at a given time t and there is a protein p having a state ps*1 *and a concentration c at this time t, and these two molecules g and p are related by an Inhibition interaction, and if the concentration of p exceeds a threshold equal to* 0.7, *then the state of g move to false at time* $t + 1$.

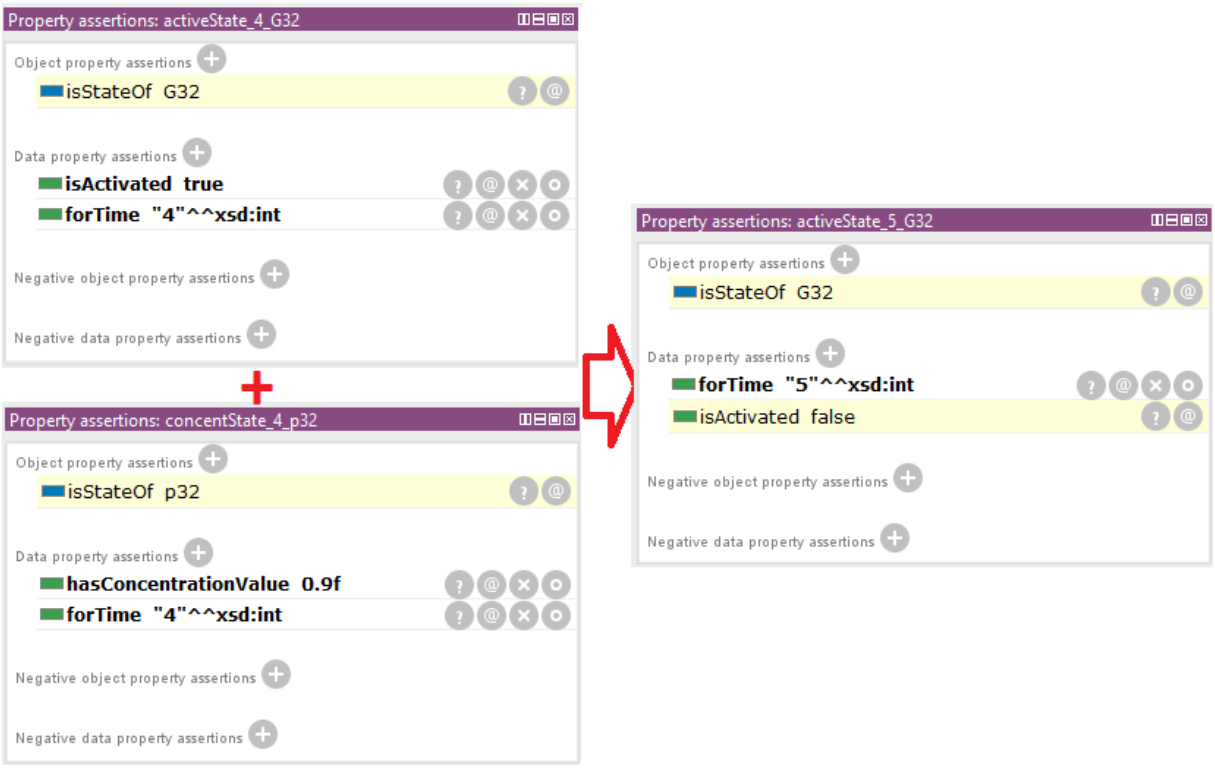


Figure 7.8 – Results of the reasoning process for the Inhibition SWRL rule.

7.4.2.2 Activation SWRL rule

In contrast to the first rule, this rule models the *activation* interaction. When the concentration of the protein $p32$ becomes less than the threshold 0.2 10^{-6} Mol/L^{-1}, it activates the translation of the gene $G32$.

$$ADN(?g) \wedge hasState(?g,\ ?gs1) \wedge forTime(?gs1,\ ?t) \wedge hasState(?g,\ ?gs2) \wedge forTime(?gs2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \wedge Protein(?p) \wedge Activation(?activ) \wedge hasSource(?activ,\ ?p) \wedge hasEnd(?activ,\ ?g) \wedge hasState(?p,\ ?ps) \wedge forTime(?ps,\ ?t) \wedge hasConcentrationValue(?ps,\ ?c) \wedge swrlb{:}lessThanOrEqual(?c,\ 0.2) \rightarrow isActivated(?gs2,\ true)$$

As described in Figure 7.9, the results of this rule means that, *If there is a gene g having a state gs equal to false at a given time t and there is a protein p having a state ps1 and a concentration c at this time t, and these two molecules g and p are related by an Activation interaction, and if the concentration of p is under a threshold equal to* 0.2, *then the state of g move to true at time* $t + 1$.

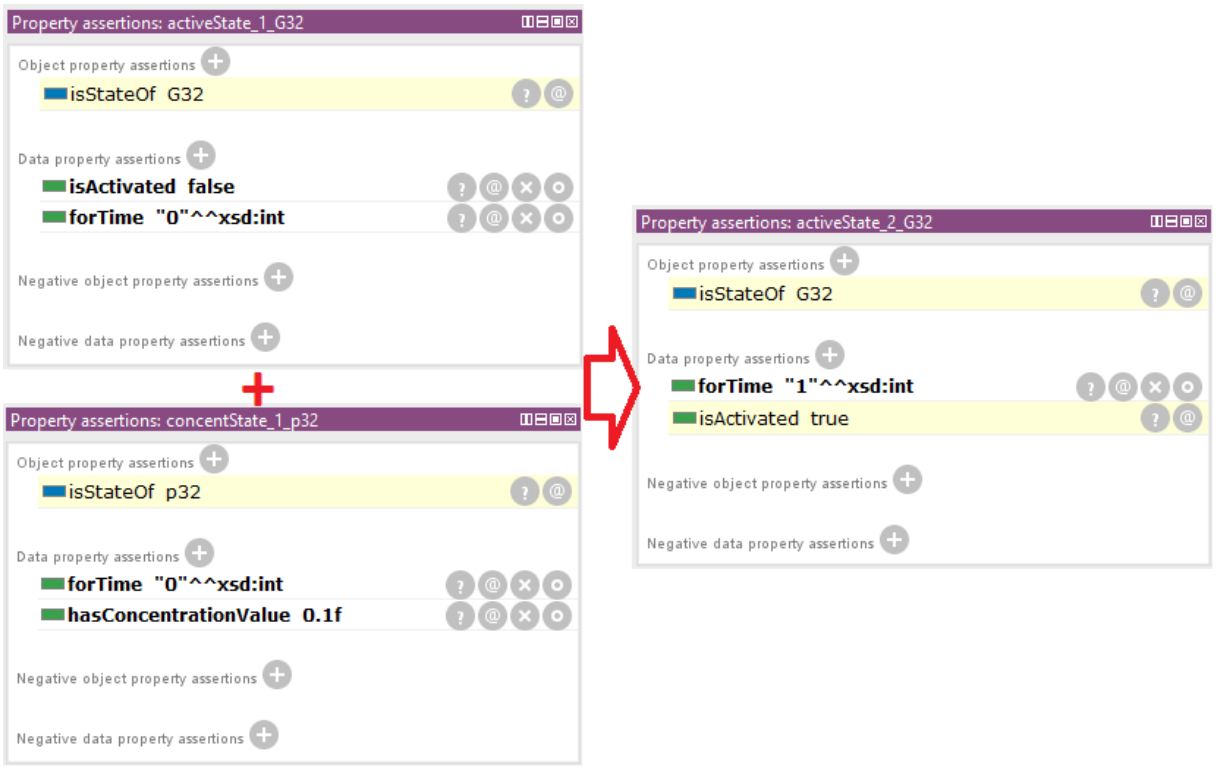


Figure 7.9 – Results of the reasoning process for the Activation SWRL rule.

7.4.2.3 Transcription SWRL rule

The following rule represents the gene *transcription*. In fact, if the gene $G32$ is activated, this one generates the protein synthesis and produces an increase in the concentration of this protein $p32$.

$$ADN(?g) \wedge hasState(?g,\ ?gs) \wedge forTime(?gs,\ ?t) \wedge isActivated(?gs,\ true) \wedge Protein(?p) \wedge Transcription(?trans) \wedge hasSource(?trans,\ ?g) \wedge hasEnd(?trans,\ ?p) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \rightarrow hasConcentrationValue(?ps2,\ ?c2)$$

The result of this rule is interpreted as (Figure 7.10), *If there is a gene g having a state gs equal to true at a given time t and there is a protein p having a state ps1 and a concentration c at this time t, and these two molecules g and p are related by a Transcription interaction, then the concentration of the protein p increases at time* $t + 1$. In the opposite case, we have this rule:

$$ADN(?g) \wedge hasState(?g,\ ?gs) \wedge forTime(?gs,\ ?t) \wedge isActivated(?gs,\ false) \wedge Protein(?p) \wedge Transcription(?trans) \wedge hasSource(?trans,\ ?g) \wedge hasEnd(?trans,\ ?p) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \rightarrow hasConcentrationValue(?ps2,\ ?c1)$$

The result of this rule means that (Figure 7.11), *If there is a gene g having a state gs equal to false at a given time t and there is a protein p having a state ps1 and a concentration c at this time t, and these two molecules g and p are related by a Transcription interaction, then the concentration of the protein p remains stable at time* $t + 1$.

Results of the Transcription SWRL rule and its opposite rule are presented in Figure 7.10 and Figure 7.11, respectively.

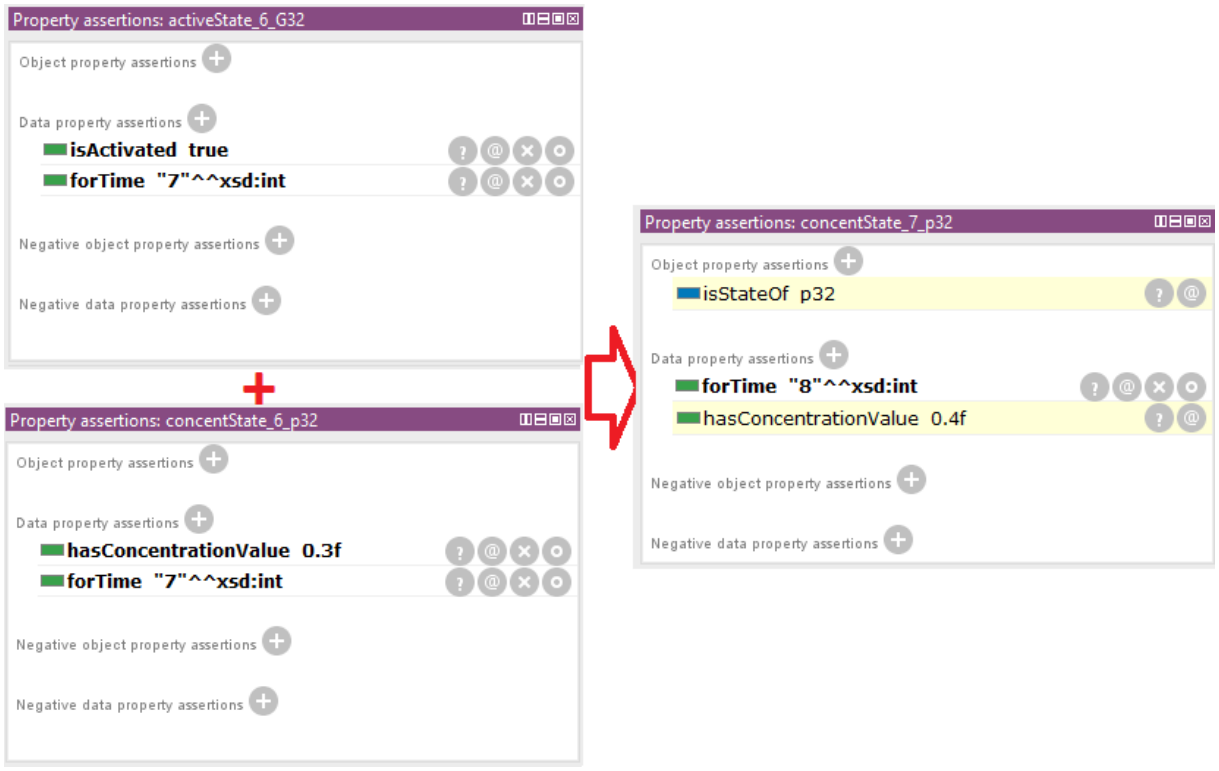


Figure 7.10 – Results of the reasoning process for the Transcription SWRL rule.

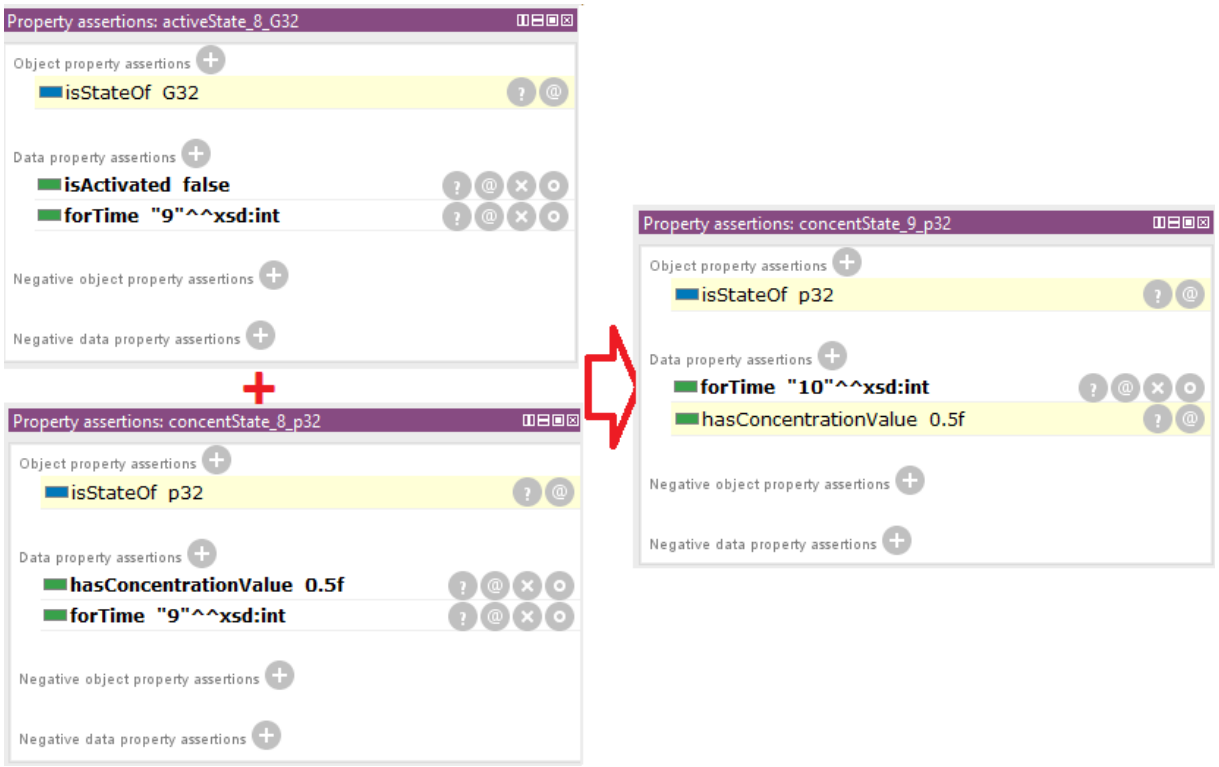


Figure 7.11 – Results of the reasoning process for the inverse of Transcription SWRL rule.

7.4.2.4 Negative regulation SWRL rule

As well, following the increase of the concentration of the protein $p32$, a *negative regulation* interaction resulted to create hormone balance. This reaction is ensured by the following rule:

$$Metabolite(?m) \wedge hasState(?m,\ ?ms) \wedge hasConcentrationValue(?ms,\ ?c) \wedge forTime(?ms,\ ?t) \wedge Protein(?p) \wedge Negative\_regulation(?negreg) \wedge hasSource(?negreg,\ ?m) \wedge hasEnd(?negreg,\ ?p) \wedge deltaC(?negreg,\ ?delta) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb:add(?t2,\ ?t,\ 1) \wedge swrlb:greaterThanOrEqual(?c,\ 0.8) \wedge swrlb:subtract(?c2,\ ?c1,\ ?delta) \rightarrow hasConcentrationValue(?ps2,\ ?c2)$$

The meaning of this rule is (Figure 7.12), *If there is a metabolite m having a state ms associated to a concentration value c at a given time t and there is a protein p having a state $ps1$ and a concentration $c1$ at this time t, and these two molecules m and p are related by a negative regulation interaction, and if the concentration of m exceeds a threshold equal to* 0.8*, then the concentration of the protein p decreases at time $t+1$.*

In contrast, when the concentration of the metabolite $m32$ is less than 0.8 we applied the following rule:

$$Metabolite(?m) \wedge hasState(?m,\ ?ms) \wedge hasConcentrationValue(?ms,\ ?c) \wedge forTime(?ms,\ ?t) \wedge Protein(?p) \wedge Negative\_regulation(?negreg) \wedge hasSource(?negreg,\ ?m) \wedge hasEnd(?negreg,\ ?p) \wedge deltaC(?negreg,\ ?delta) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb:add(?t2,\ ?t,\ 1) \wedge swrlb:lessThan(?c,\ 0.8) \rightarrow hasConcentrationValue(?ps2,\ ?c1)$$

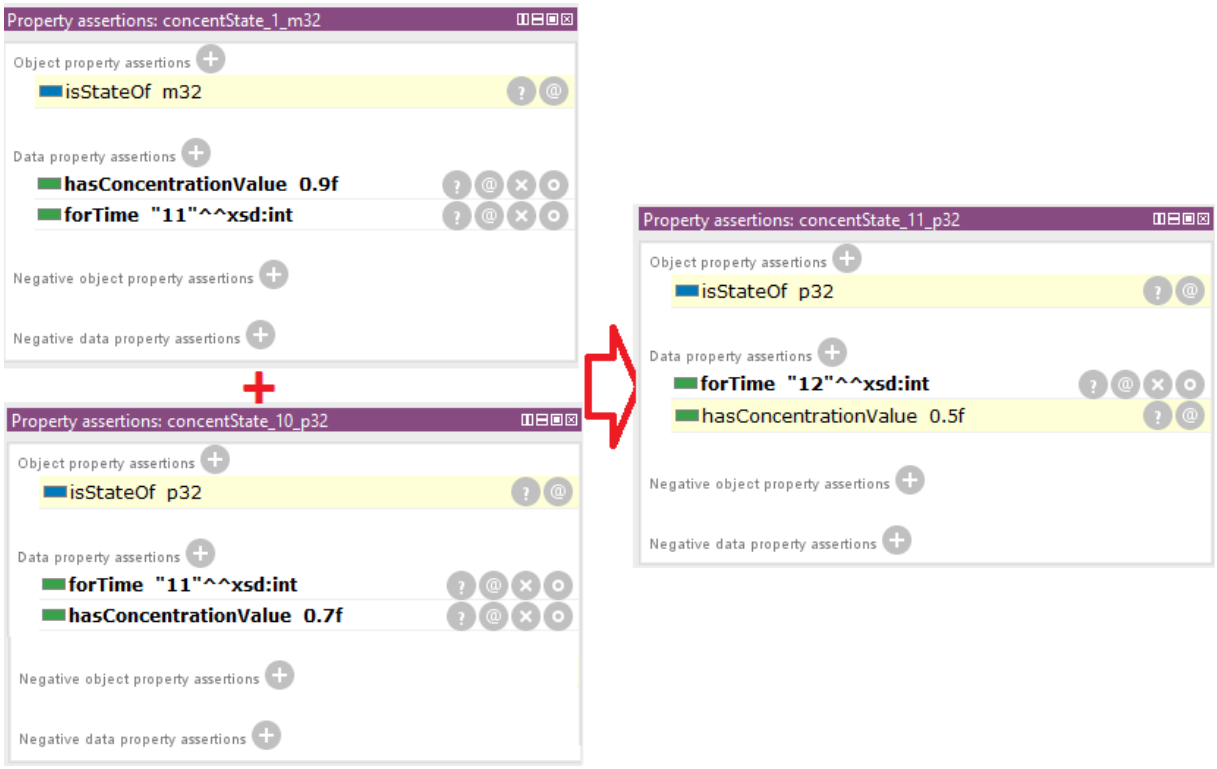


Figure 7.12 – Results of the reasoning process for the Negative regulation SWRL rule.

Which means (Figure 7.13), *If there is a metabolite m having a state ms associated to a concentration value c at a given time t and there is a protein p having a state $ps1$ and a concentration $c1$ at this time t, and these two molecules m and p are related by a negative regulation interaction, and if the concentration of m is under a threshold equal to* 0.8*, then the concentration of the protein p remains stable at time*

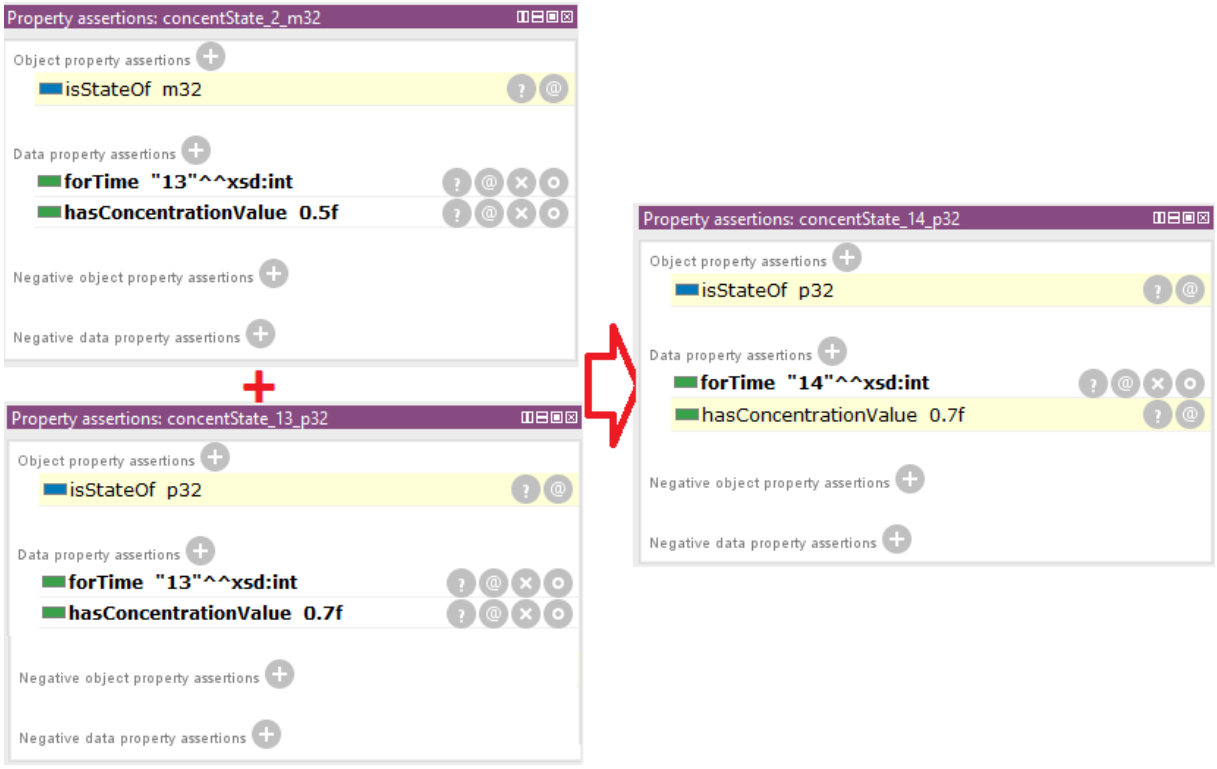


Figure 7.13 – Results of the reasoning process for the inverse of the Negative regulation SWRL rule.

$t+1$. Results of the Negative regulation SWRL rule and its opposite rule are presented in Figure 7.12 and Figure 7.13, respectively.

To conclude, the case study with OWL-SWRL rules represents a "proof of concept" since it demonstrates the logical consistency of the BNO ontology and validates its relevance. To check the inconsistencies and violations of these SWRL rules, we used the latest version of HermiT reasoning plugin in the Protégé 5 environment[4] version 1.3.8.3. Obtained results prove that the BNO ontology is consistent, credible and effective in describing relevant knowledge required in understanding the behaviour of complex biomolecular networks and their state changes. However, we must emphasise that, even if this ontology provides useful knowledge and rich semantics allowing biologists to understand the dynamical behaviour of complex biomolecular networks, it can not simulate large-scale networks. That is why more efficient simulation tools should be used for scaling up and reason on large biomolecular networks.

7.4.3 Rule-based qualitative reasoner within MATLAB

Despite the SWRL rule-based reasoning, extensive experiments were conducted to validate our proposed ontology including the implementation of the rule-based qualitative reasoner. This rule-based reasoner is implemented under the MATLAB development environment and can be freely downloaded at `https://github.com/AliAyadi/Qualitative\ReasoningInMATLAB`. The reasoner is based on a qualitative simulation algorithm 1, with specific reference to the SWRL rules defined in the previous Section 7.4.2.

Algorithm 1 provides a high-level description of the general reasoning algorithm of a complex biomolecular network. The main steps of this algorithm are: (1) The definition of the set of SWRL rules and their thresholds. (2) The initialization time and the state of all molecular components; (3) Evaluate the node state; (4) Launch the specific reaction defined by the corresponding SWRL rule when the node state reached a threshold; and (5) Update the novel value of the state node.

Figure 7.14 depicts the individual qualitative behaviour of the biomolecular components. Indeed, for each node, a description of its dynamical time-evolution is graphically presented. The evolution is

[4] http://www.hermit-reasoner.com/

Algorithm 1 Pseudocode of the qualitative simulation algorithm

```
Definition of the set of SWRL rules and their thresholds.

Initialization of time and network's state.
for All time step from begining to end_of_simulation do
    for Each molecular component do
        Evaluate the node state
        if the component's state achieves one threshold then
            Execution of the reaction defined by the SWRL rule corresponding to this threshold.
            ▷ Measure the state if it is a gene and the concentration if it is a protein or a metabolite
            Update the novel state of the node.
        end if
    end for
end for
```

displayed as a graph to easily see when and how the molecular component evolves during the simulation and to precisely detect changes in the state in time. These results are not difficult to interpret, because at each simulation step a qualitatively meaningful network state is reached. The results are close to human reasoning. In fact, this qualitative reasoning is based on the SWRL rules presented above which represent a set of constraints equations describing the relevant structural and functional relationships in the biomolecular network. The possible behaviours and states of the network may be predicted from these constraints rules and an initial state. The behavioural description of the biomolecular network and its individual components may be used to explain a set of hypothesis and determine the change directions of the network behaviour.

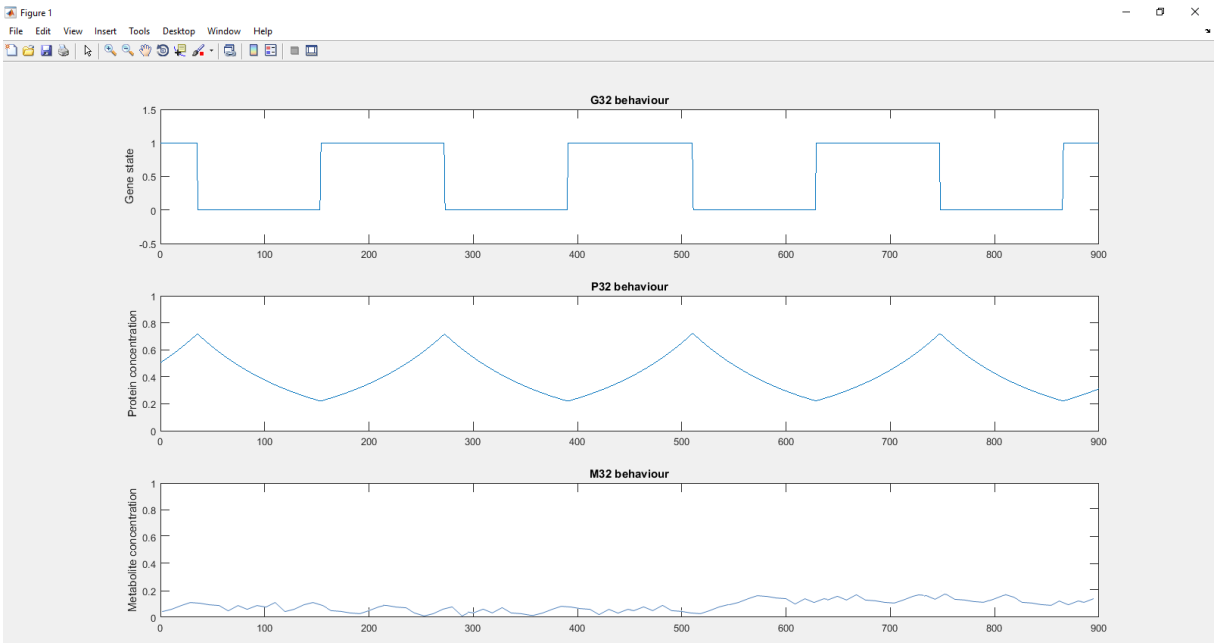


Figure 7.14 – Simulation results plotted with the MATLAB environment: the individual qualitative behaviour of the biomolecular components.

7.5 Summary

This chapter proposes an effective approach for analysis and understanding the behaviour of complex biomolecular networks over time. The use of a semantic approach based on merging different ontologies

can overcome issues of study the state changes of complex biomolecular networks and their behaviour. Indeed, we developed the Biomolecular Network Ontology (BNO) to describe the static structure of complex biomolecular networks and merge it with the Gene Ontology (GO) to provide structured terminologies for the description of cellular components. We also chose the Simple Event Model Ontology (SEMO) to describe events and stimuli which can stimulate the network's components and integrate the Time Ontology (TO) to study the different states of the biomolecular network and its nodes over time.

The Biomolecular Network Ontology developed in this chapter aims to describe the domain knowledge of complex biomolecular networks in their static state. This ontology provides information on the biomolecular network and its components (nodes, interactions, states, transition states, etc.) and an indication of the network's context such as the type of sub-network, the type of node, the conditions and nature of interactions, etc. This allows to precisely analyse and interpret the semantic context in order to achieve intelligent modelling of biomolecular networks and their state changes. These state changes can be computed with a rule-based system.

The SWRL rule-based reasoning and rule-based qualitative reasoner within MATLAB have been used to validate the BNO ontology and to demonstrate how it is capable of providing useful knowledge and rich semantics allowing biologists to understand and simulate the dynamical behaviour of complex biomolecular networks. However, the BNO ontology can not simulate large-scale networks. That is why more efficient simulation tools should be used for scaling up and reason on large biomolecular networks, this is the topic of the next chapter.

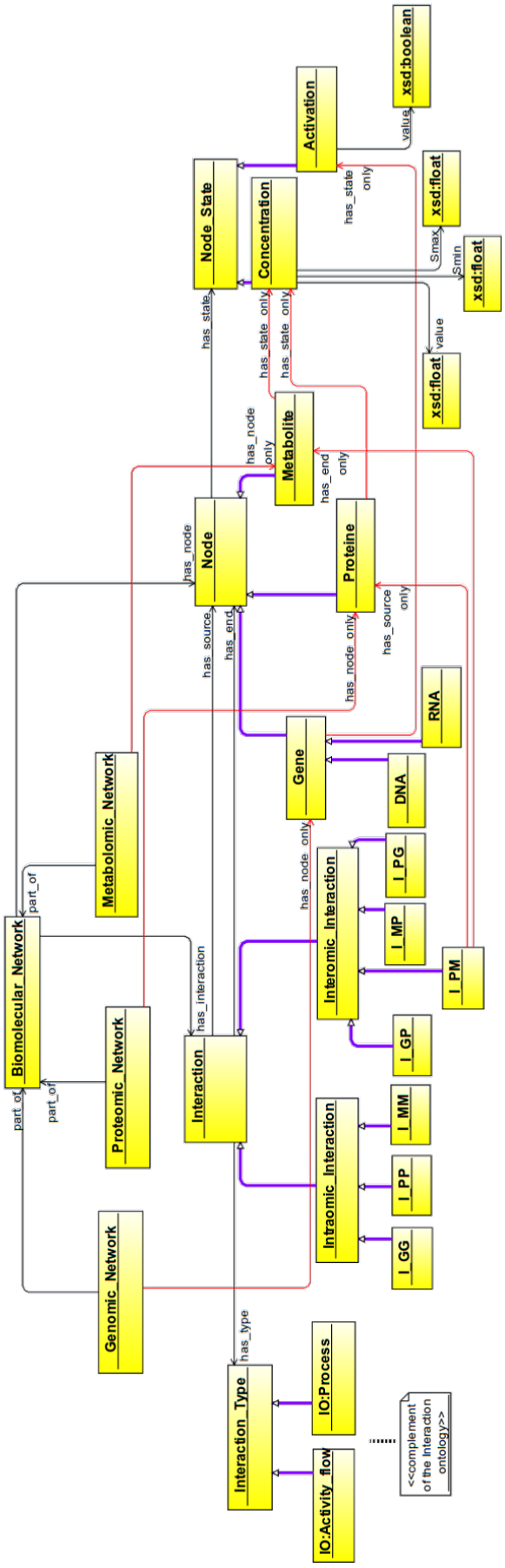


Figure 7.15 – The Biomolecular Network Ontology (BNO).

Chapter 8

Qualitative, discrete-event simulation simulation of complex biomolecular networks

Contents

8.1 Introduction

The complexity of biomolecular networks is firstly due to their large number of coupled components, but also to the diversity of these molecular components and to their intricate interactions. Indeed, biomolecular networks consist of various subnetworks which themselves are composed of several molecular components interacting in turn with each other, producing a complex global behaviour. The complexity and large size of these networks have prevented a fully quantitative simulation. Thus, biologists require tools allowing them to gain insights into the behaviour of complex biomolecular networks by simulating the different states of their components over time.

In this chapter, we propose two kinds of simulation: a qualitative and a discrete-event simulation. The first one responds to the complexity of calculating the quantitative reasoning methods which sometimes are impossible to implement. Then, the second method is an integrative discrete-event simulation considering that the behaviour of the complex biomolecular network emerges from the network-level interaction.

The first section of this chapter presents the qualitative simulation method and detail all its construction steps. Then, the second section of this chapter presents a discrete-event simulation approach that is able to reproduce the behaviour of complex biomolecular networks and their components over time. This simulation is based on the combination of the logical-based modelling of complex biomolecular networks (which is presented in Chapter 6) and a discrete-event simulation algorithm inspired by the discrete-event system specification formalism (detailed in Section 4.3.4). Moreover, we enrich and explain these simulation methods by applying them with the case study of the bacteriophage T4 gene 32.

8.2 Qualitative simulation model

In our works, the explicit representation of the network behaviour evolution, between two instants t_0 and t_n is essential. Therefore, we must link the logical-based modelling defined in Chapter 6 to a qualitative simulation mechanism. This simulation allows executing the model in order to simulate the network evolution and its components over time.

8.2.1 Qualitative reasoning

The reasoning is a mental activity that humans practice to solve difficulties they confront in their life. This reasoning is often performed in the lack of quantitative knowledge and is then called *qualitative reasoning*. In literature, we distinguish two types of reasoning, *heuristic reasoning* which is a mental shortcut and *causal reasoning* which is based on a model [8]. The second case of reasoning is based on the modelling of the system. Such reasoning is based on a model of causal type because it combines the effects and causes, such as a causal graph. It solves a problem by reasoning about the structure and function of the object in an application environment and their behaviour over time [337, 338].

We chose to use qualitative reasoning for two reasons: (1) To understand the overall functioning and properties of complex biomolecular networks through the analysis and simulation of the dynamical model (explained in the previous section), and the interpretation of the obtained knowledge. (2) To steer these networks by allowing to evaluate their simulation at any time.

8.2.2 Basic concepts

In the following sections, we will define the basic concepts of qualitative simulation [8] and detail the major phases of construction.

8.2.2.1 The causal graph

The qualitative simulation model is based on the development of a causal graph whose nodes denote variables that are related to this simulation and edges denote causality relations among these variables. By analogy with the logical-base modelling presented in Chapter 6, the causal graph is itself the biomolecular network SR where its nodes represent causal states of network's molecular components and its edges represent the types of interactions that can occur among these components.

8.2.2.2 Quantitative variables & Quantity space

A *variable* is a characteristic of interest. For example, in our case the variables of the qualitative model denote the **state of the molecular components** at a given moment denoted by $en(m,t)$. These variables are qualitative because they are represented by qualities (nominal or ordinal).

The set of these qualitative values and their corresponding intervals constitutes the *quantity space* of the variable $en(m,t)$, denoted by $EQ_{en(m,t)}$. Each variable $en(m,t)$ takes its qualitative value in its ordered set of qualitative values $EQ_{en(m,t)} = \{vq_1, vq_2, ..., vq_n\}$. In fact, the quantity space is a partition of the domain of a variable values into behaviour regions that are qualitatively homogeneous.

As defined in Algorithm 2, the partition of the quantity space $EQ_{en(m,t)}$ depends on the type of node:

- If $m \in M_G$: $en(m,t) = \{Deactivated, Activated\}$, its states can be '*Activated*' or '*Deactivated*'. So, we assign to its $EQ_{en(m,t)}$ the qualitative values 0 and 1 meaning respectively '*Deactivated*' and '*Activated*'.

$$en(m,t) = \{Deactivated, Activated\} \Rightarrow EQ_{en(m,t)} = \{0,1\}$$

- If $m \in M_P \cup M_M$: $EQ_{en(m,t)}$ depends on the outgoing arcs $oe(m)$ starting from the node m. In fact, as illustrated in Figure 8.1 for a quantity n of outgoing arcs, there will be $n+1$ qualitative values that are defined by an order relation on $EQ_{en(m,t)}$, creating an ordered set of qualitative values $EQ_{en(m,t)} = \{vq_1, vq_2, ..., vq_n\}$.

$$en(m,t) = \{[min_m, Threshold_1[, [Threshold_1, Threshold_2[, ..., [Threshold_n, max_m]\} \Rightarrow EQ_{en(m,t)} = \{vq_1, vq_2, ..., vq_n\}$$

To resolve the conflicts of partitioning the $EQ_{en(m,t)}$, we present the following algorithm.

Algorithm 2 Pseudocode of the $EQ_{en(m,t)}$ partitioning algorithm

Require: $m \in M$, $oe(m)$, min_m, max_m, $EQ_{en(m,t)} \leftarrow \emptyset$
Ensure: Partition of $EQ_{en(m,t)}$
1: **if** $(m \in M_P \cup M_M)$ **then**
2: **for** outgoing edges $i \in oe(m)$ **do**
3: Read its $Threshold$;
4: Sort the threshold values;
$$Threshold_1 < Threshold_2 <, ..., < Threshold_n$$
5: Quantitative partitioning of $EQ_{en(m,t)}$;
$$EQ_{en(m,t)} = \{[min_m; Threshold_1], [Threshold_1, Threshold_2], ..., [Threshold_n, max_m]\}$$
6: Translate quantitative measures into qualitative values;
$$EQ_{en(m,t)} = \{vq_1, vq_2, ..., vq_{n+1}\}$$
Where: $vq_1 = [min_m, Threshold_1]$ and $||EQ_{en(m,t)}|| = ||oe(m)|| + 1$
7: **end for**
8: Return the quantity space
$$EQ_{en(m,t)} = \{vq_1, vq_2, ..., vq_{n+1}\}$$
9: **else**
10: **if** $(m \in M_G)$ **then**
11: Boolean partitioning of $EQ_{en(m,t)}$;
$$EQ_{en(m,t)} = \{true, false\}$$
12: Translate boolean measures into qualitative values;
$$EQ_{en(m,t)} = \{vq_1, vq_2\}$$
Where: $vq_1 = 0$, $vq_2 = 1$ and $||EQ_{en(m,t)}|| = 2$
13: Return the quantity space
$$EQ_{en(m,t)} = \{vq_1, vq_2\}$$
14: **end if**
15: **end if**

Figure 8.1 displays the execution of the $EQ_{en(m,t)}$ partitioning algorithm in both cases. In addition

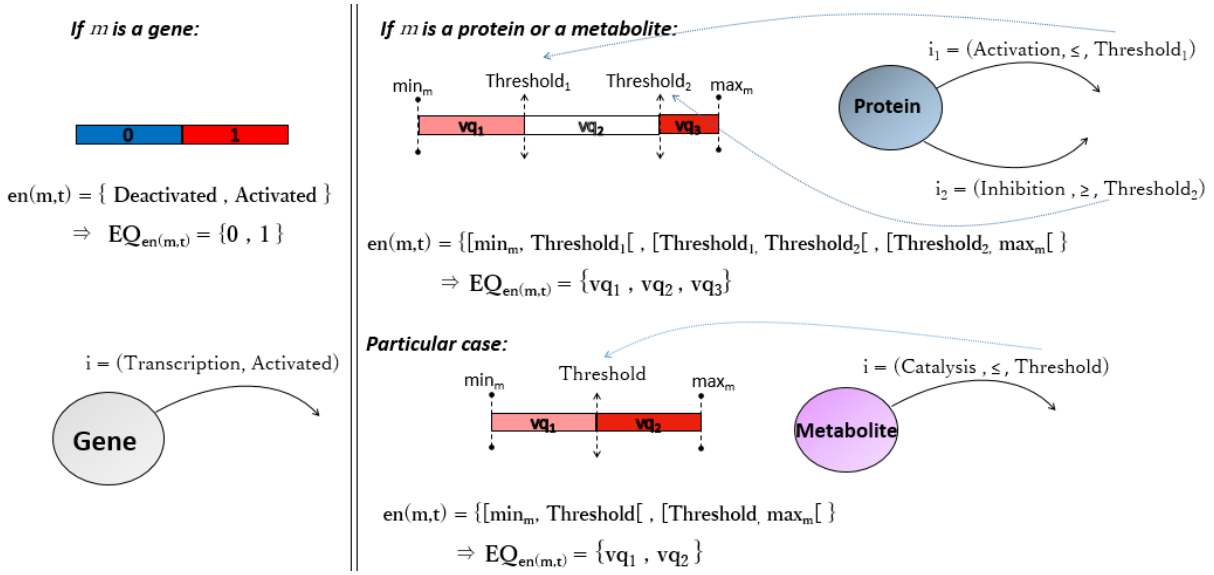


Figure 8.1 – Description of the $EQ_{en(m,t)}$ partitioning algorithm.

to the quantity space of its variables, a qualitative reasoning method also includes algebraic relations (constraints, influences, etc.) that act among these quantity space.

8.2.2.3 Operations and rules

The operations In [8], the authors define six operations for calculating the quantity spaces of the variables. Among them, we will just use the three unary operations shown in Table 8.1: the *incrementation* (*incr*), the *decrementation* (*decr*) and the *inverse* (*inv*) of a qualitative variable vq_i.

Using these operators, we can combine several variables together to create our own operations as a specific combination table.

The partition and propagation rules Based on the work presented in [8], we adapt a qualitative reasoning mechanism to compute the qualitative value of the nodes. As shown in Figure 8.2, this mechanism is based on both the partition rules and the propagation rules. These rules are used to compute the value of the target variable ($en(m, t+1)$) at the next time $t+1$ based on its qualitative value ($en(m, t)$) and the value of its predecessors ($en(Pred(m), t)$) at the current time t.

- *Partition rules* allow the translation of quantitative measures of the variables ($en(m, t)$) into qualitative values. They match a quantitative (real) interval with its corresponding qualitative value belonging to the quantity space $EQ_{en(m,t)}$. They are defined by the pseudo code of the algorithm 2.

- *Propagation rules* compute the propagation of the qualitative values from the source components to the target components of the causal graph. They are defined by the aggregate functions A_m which compute the evolution of the node status between two successive instants of the simulation (this function is detailed in Section 6.4.3). These rules are expressed by combining the operations presented in Table 8.1.

8.2.3 Application to the motivating example: the bacteriophage T4 gene 32

In order to demonstrate the different notions of this qualitative mechanism and to test its performance, we apply it to the example of the bacteriophage T4 gene 32 detailed in Section 6.2.

Operations on EQ
Unary operations $\forall [en(m,t)] \in EQ_{en(m,t)} = \{vq_1, vq_2, vq_3, vq_4, vq_5\}$ and $n \in \mathbb{N}$
Incrementation '$incr$' $incr_0([m]) = [en(m,t)]$ $[en(m,t)]$: vq_1 vq_2 vq_3 vq_4 vq_5 - $incr_1([en(m,t)])$: vq_2 vq_3 vq_4 vq_5 vq_5 $incr_n([en(m,t)]) = incr_{n-1}(incr_1([en(m,t)]))$
Decrementation '$decr$' $decr_0([en(m,t)]) = [en(m,t)]$ $[en(m,t)]$: vq_1 vq_2 vq_3 vq_4 vq_5 - $decr_1([en(m,t)])$: vq_1 vq_1 vq_2 vq_3 vq_4 $decr_n([en(m,t)]) = decr_{n-1}(decr_1([en(m,t)]))$
Inverse 'inv' $[en(m,t)]$: vq_1 vq_2 vq_3 vq_4 vq_5 - $inv([en(m,t)])$: vq_5 vq_4 vq_3 vq_2 vq_1

Table 8.1 – Unary operations on quantity spaces presented in [8].

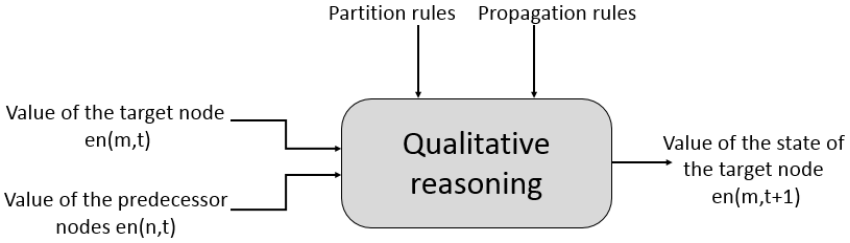


Figure 8.2 – Qualitative reasoning mechanism.

8.2.3.1 The variables

In the example presented in Figure 6.1, we have three variables $en(G32,t)$, $en(p32,t)$ and $en(m32,t)$ that respectively represent the state of the gene $G32$, the protein $p32$ and the metabolite $m32$.

8.2.3.2 The causal graph

We can use the structure of the biomolecular network as the causal graph of our example. This structure is defined in more details in Section 6.4.1.

8.2.3.3 The partition rules

$en(G32,t) \in \{Deactivated\ ,\ Activated\}$,
$\Rightarrow EQ_{en(G32,t)} = \{0\ ,\ 1\}$.
$en(p32,t) \in \{[min_{p32}, 0.2[\ ,\ [0.2, 0.7[\ ,\ [0.7, max_{p32}[\}$,
$\Rightarrow EQ_{en(p32,t)} = \{vq_1\ ,\ vq_2\ ,\ vq_3\}$.
$en(m32,t) \in \{[min_{m32}, 0.8[\ ,\ [0.8, max_{m32}[\}$,
$\Rightarrow EQ_{en(m32,t)} = \{vq_1\ ,\ vq_2\}$.

8.2.3.4 The propagation rules

For reasons of clarity, we note $[m]^t$ the qualitative value of the state of the component m at time t. It means that the notation $[m]^t \equiv [en(m,t)] \in EQ_{en(m,t)}$. Now, let us define the aggregate rules of each variables.
For the variable $G32$:

$$\begin{aligned}[G32]^{t+1} = A_{G32}([G32]^t, \{i_1, i_2\}, [p32]^t) : \\ & if\ ([p32]^t = vq_1)\ then \\ & [G32]^{t+1} = 1 \\ & else\ if\ ([p32]^t = vq_2)\ then \\ & [G32]^{t+1} = [G32]^t \\ & else\ if\ ([p32]^t = vq_3)\ then \\ & [G32]^{t+1} = 0\end{aligned}$$

For the variable $p32$:

$$\begin{aligned}[p32]^{t+1} = A_{p32}([p32]^t, \{i_3, i_4\}, [G32]^t, [m32]^t) : \\ & if\ ([m32]^t = vq_1)\ \wedge\ ([G32]^t = 0)\ then \\ & [p32]^{t+1} = [p32]^t \\ & else\ if\ ([m32]^t = vq_2)\ \wedge\ ([G32]^t = 0)\ then \\ & [p32]^{t+1} = decr([p32]^t) \\ & else\ if\ ([m32]^t = vq_1)\ \wedge\ ([G32]^t = 1)\ then \\ & [p32]^{t+1} = incr([p32]^t) \\ & else\ if\ ([m32]^t = vq_2)\ \wedge\ ([G32]^t = 1)\ then \\ & [p32]^{t+1} = [p32]^t\end{aligned}$$

For the variable $M32$:

$$\begin{aligned}[m32]^{t+1} = A_{m32}([m32]^t) \\ \Rightarrow [m32]^{t+1} = [m32]^t\end{aligned}$$

8.2.3.5 The simulation

Let us define the initial state of the network at t_0: $ER(t_0) = \langle [G32]^{t_0}, [p32]^{t_0}, [m32]^{t_0} \rangle$.
We chose the initial qualitative values of the components as: $ER(t_0) = \langle 0, vq_1, vq_1 \rangle$.
Then, we have performed a series of simulations to assess the evolution of the network over time:

$$\begin{aligned}ER(t_0+1) &= \langle [G32]^{t_0+1}, [p32]^{t_0+1}, [m32]^{t_0+1} \} \\ &= \langle 1, vq_1, vq_1 \rangle \\ ER(t_0+2) &= \langle [G32]^{(t_0+1)+1}, [p32]^{(t_0+1)+1}, [m32]^{(t_0+1)+1} \rangle \\ &= \langle 1, vq_2, vq_1 \rangle\end{aligned}$$

8.2.3.6 The behaviour

$$\begin{aligned}CR_{[t_0,t_2]} &= \{ER(0), ER(1), ER(2)\} \\ &= \{\langle 0, vq_1, vq_1 \rangle, \langle 1, vq_1, vq_1 \rangle, \langle 1, vq_2, vq_1 \rangle\}\end{aligned}$$

Figure 8.3 presents the possible states of each molecular components during the simulation. This is the qualitative simulation of the given example according to the possible initial states of the components. The possible molecular states presented in this Figure referred to the set of the network states presented in the last column of Table 6.4.

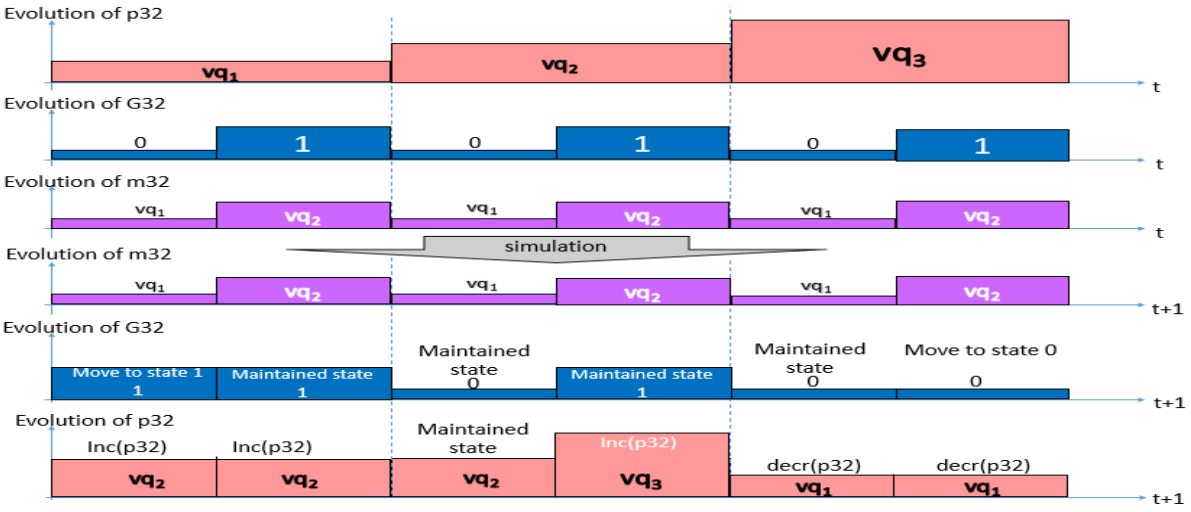


Figure 8.3 – All possible simulation results of our example.

8.3 Discrete-event simulation model

In this section, we present an approach for simulating the behaviour of complex biomolecular networks inspired by the DEVS formalism [339] detailed in Section 4.3.4, a formalism for supporting the modelling of complex systems.

8.3.1 Mapping the logical based modelling with the DEVS formalism

Before proposing our discrete-event algorithm, it is necessary to link the different parts of the logical modelling with their corresponding notions of the DEVS formalism. As discussed in Section 6.4, the logical-based modelling is based on three basic modelling pillars: (1) **The structural modelling** SR, to describe the architecture of the biomolecular network. (2) **The functional modelling** FR, to describe what can carry out each component of the biomolecular network, specifying the conditions for these activities. (3) And **the behavioural modelling** $CR_{[t_0,t_n]}$, to describe how the biomolecular network and its individual components evolve during the simulation period $[t_0, t_n]$. Therefore, the biomolecular network BN is defined as: $BN = (SR, FR, CR_{[t_0,t_n]})$.

We follow this tripartite classification to make the mapping between the logical-based modelling and the DEVS formalism. The structure of the biomolecular network SR (Section 6.4.1) is composed by the set of nodes M which corresponds to the set of DEVS components D, and the set of interactions I corresponds to the set of the internal links among DEVS components IC. The function of the biomolecular network, represented by the function FR, corresponds to the internal transition function δ_{int}. In the logical-based modelling, the behaviour of complex biomolecular networks is represented with multiple parameters. The first parameter consists of the function $en(m, t)$ that defines the state of each component, and the function $ER(t)$ that defines the state of the network at time t. These functions correspond to the set of DEVS components S in the DEVS formalism. The aggregate function A_m corresponds to a set of functions in DEVS formalism which consists of the internal transition function δ_{int}, the external transition function δ_{ext}, the confluent transition function δ_{con} and the time advance function ta. The external stimuli represented by S corresponds in DEVS formalism to the set X of input events. Finally, the behaviour of the network $CR_{[t_0,t_n]}$ corresponds in DEVS formalism to the set Y of output events which represent the evolution of each DEVS component during the period of simulation.

8.3.2 Discrete-event simulation algorithm

In this section, we describe the basic model of the proposed discrete-event simulation algorithm, with specific reference to the logical-based modelling developed in Chapter 6) and following the approach presented in Zeigler et al. [339] (detailed in Section 4.3.4). Algorithm 6 provides a high-level description

of the general simulator algorithm of a complex biomolecular network. The main steps of this algorithm are: (1) The initialization time and the state of all molecular components; (2) The execution of the aggregate function to compute the evolution of the node in the next iteration; (3) Evaluate the node state; (4) Launch the specific reaction if the node state reached a threshold; and (5) Update the value of the node.

Algorithm 3 Pseudocode of the general simulator algorithm

```
1: Initialization: t_0, ER(t_0)                          ▷ Initialisation of time and network's state.
2: for All time step t from begining to end_of_simulation do
3:     for Each component m_i ∈ M do
4:         Execution of the aggregate function A_{m_i}    ▷ Launch the aggregate function manages the
   evolution of the node m_i
5:         (Value, Threshold) = TestState(en(m_i, t), FR(m_i))        ▷ Evaluate the node state
6:         if Value = true then                  ▷ If the state of a node state achieves a threshold
7:             LaunchReaction(FR(en(m_i, t)))    ▷ Launch the reaction defined by the function FR
8:             Update the node's state (en(m_i, t))          ▷ Update the novel state of the node
9:         end if
10:    end for
11: end for
```

At the beginning of the simulation, the simulator initialises the network's state $ER(0) = \langle en(m_1, 0), en(m_2, 0), ..., en(m_n, 0)\rangle$, and time $t = 0$.

After each iteration, the simulator evaluates the state of each node. This step is done by the *TestState* function 4. This function requires the specification of two parameters: the state of the node m_i at this time t which is provided by the function $en(m_i, t)$ (see Section 6.4.3) and its function $FR(m_i)$. This function compares the value of the state with the set of Thresholds defined by its function FR. If the value of $en(m_i, t)$ reached a threshold, it returns a boolean value equal to true and the reached threshold, else it returns the boolean value false.

Algorithm 4 Pseudocode of the TestState function

```
function TestState(en(m_i, t), FR(m_i))
2:     for Each Threshold ∈ ae(m_i) do    ▷ Compare the state of the molecule m_i with all thresholds
           if en(m_i, t) reached Threshold then
4:             return (true, Threshold);      ▷ If a threshold is reached it returns true and the reached
   threshold
           else
6:             return (false, —);
           end if
8:     end for
   end function
```

Once the comparison has been done and according to the result returned by the *TestState* function, the simulator runs the *LaunchReaction* procedure defined by Algorithm 5. This procedure requires the specification of two parameters: the *Threshold* returned by the *TestState* function and the function $FR(m_i)$. According to these two parameters, the procedure launch the specific reaction corresponding to the type of the interaction defined by the function FR (Section 6.4.2). This label *TypeInteraction* represents the type of the interaction defined by the edge i_{m_i,m_d} (with m_i the actual node and m_d the target node). These types of interactions belong to the set of concepts of the Interaction Ontology (which is detailed in Section 6.4.1). After applying the specific reaction, the novel value of the node m_i is updated. This process will continue for all the nodes M and until the end of the simulation. Finally, the simulator returns the sequence of the successive states during the simulation time $CR_{[t_0,t_{end\_of\_simaulation}]} = [ER(0), ER(1), ..., ER(end\_of\_simaulation)]$ defining the behaviour of the biomolecular network. These results are presented graphically.

Algorithm 5 Pseudocode of the LaunchReaction procedure

```
procedure LaunchReaction(Threshold, FR(m_i))
    switch Threshold do
        case TypeInteraction_1
            assert(Launch the reaction corresponding to this type of interaction)
        case TypeInteraction_2
            assert(Launch the reaction corresponding to this type of interaction)
        case TypeInteraction_..
            assert(Launch the reaction corresponding to this type of interaction)
        case TypeInteraction_n
            assert(Launch the reaction corresponding to this type of interaction)
end procedure
```

8.3.3 Application to the motivating example: the bacteriophage T4 gene 32

Before simulation of the given example starts, it is necessary to define its structure in a text file that follows the structure SR of a biomolecular network defined in Section 6.4.1. This text file represents the data input of the simulator. This simulator starts by reading the network data from the defined text file. Figure 8.4 shows the input file containing the necessary elements describing the structure of the bacteriophage T4 gene 32 network.

bacteriophage_T4_G32-definition.txt — network_definition

```
Molecule Network ::=(<Comment>)*
                    <NetworkName>
                    (<Node> | <Link> | <Comment>)*
<Node> ::= ("Molecule" | "Gene" | "Protein" | "Metabolite") <NodeId> <Value> <Comment>
<Link> ::= "Link" <LinkId> <NodeId> <NodeId> <Value>

<NetworkName> ::="NetwrorkName" (String | space |Number)*
<NodeId> ::=(String | Number)*
<LinkId> ::=(String | Number)*
<Value> ::= Number
<Comment> ::= "#" String


Defintion of the autoregulation T4 gene32:

#This is a comment for my molecular definition
NetworkName  BioMolecular Network

# Definition of molecules
Gene G32 1        # gene G1 is activated
Protein p32 2.3    # protein with its concentration
Metabolite m32 7.1 # protein with its concentration


# Add some link between our molecules
Link G32p32 G32 p32 1 # add link (nammed G32p32) between the gene G32 and the protein p32 with a value 1.
Link G32p32 p32 G32 0.2 # add link (nammed G32p32) between the gene G32 and the protein p32 with a value 0.2 (threshold).
Link G32p32 p32 G32 0.7 # add link (nammed G32p32) between the gene G32 and the protein p32 with a value 0.7 (threshold).
Link G32p32 m32 p32 0.8 # add link (nammed G32p32) between the gene G32 and the protein p32 with a value 0.8 (threshold).
```

Problems Javadoc Declaration Console Progress Call Hierarchy

Figure 8.4 – Definition of the necessary elements describing the structure of the bacteriophage T4 gene 32 network.

To simulate the behaviour of the given network, we implement the algorithm presented in the previous section and define the different interactions. For example, we define the activation interaction by the following rule: 'If the value of the concentration of the protein $p32$ rises above a certain threshold, then the transcription of the gene $G32$ is switched on.' When the gene $G32$ is activated, a transcription interaction is activated creating a protein production which means that the output of this interaction is a production of the protein $p32$ with an increase of its concentration associated to a specific rate of production. Further to this activation interaction and once the gene is activated, a transcription interaction will automatically occur by creating the increased concentration of the protein $p32$, which will additively increase its production to $\Delta_c\%$ (the change in concentration caused by the production

interaction). It is the same with the inhibition interaction. We define it by the following rule: 'If the value of the concentration of the protein $p32$ rises above a certain threshold, then the transcription of the gene $G32$ is switched on.' When the gene $G32$ is deactivated, the transcription interaction is also deactivated, as well as stopping the protein production which means that the production reaction is inert and never performs any actions. This allows maintaining a stable level of the protein p or its degradation. Further to this inhibition interaction and once the gene is deactivated, the transcription interaction is automatically stopped enabling the degradation of the concentration of the protein $p32$.

In cooperation with our biological collaborators (the Complex Systems and Translational Bioinformatics CSTB team[1] – ICube Laboratory), we define the values of the thresholds for each interaction, for example, 0.2 for the activation reaction and 0.7 for the inhibition reaction. We also estimate the value of a set of parameters needed to simulate the interactions with the discrete-event algorithm simulation proposed in Section 8.3.2. Among them the production rate (which describes the rate of production of the target protein per unit time when the activation interaction occurs), the degradation rate (which describes the rate of attenuation of the target protein per unit time when the inhibition interaction occurs).

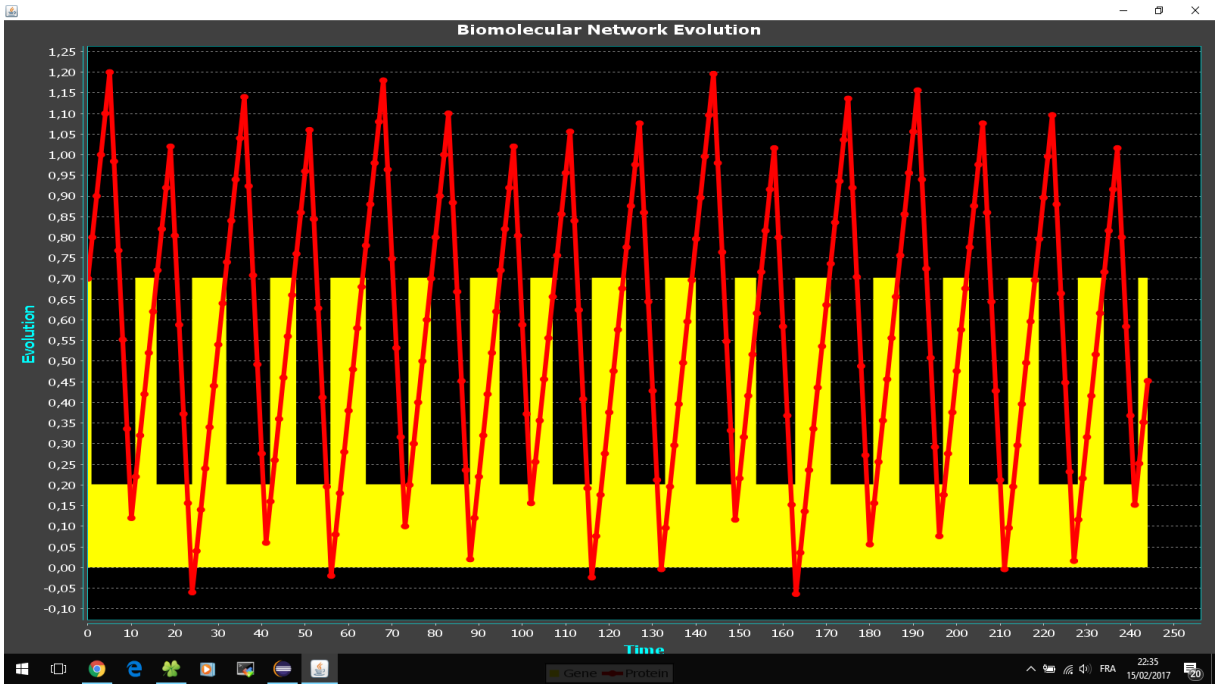


Figure 8.5 – The simulator's graphical interface. Evolution of the component's behaviour during the simulation period: the red curve represents the different values of the protein $p32$ during the period of simulation and the yellow surface represents the different states of the gene $G32$.

As presented in Figure 8.5, the simulation results of the given example were synthetically represented in graphical form. During the simulation, the concentration of the protein $p32$ is represented by the red curve and the state of the gene $G32$ by the yellow surface that appears when the gene is activated. We run the simulation with different starting states and observe its results. We note that this simulator can successfully reproduce the behaviour of the bacteriophage T4 gene 32 network. In fact, expert biologists agree with the simulator results. Moreover, with the current graphical interface, the user can easily analyse and observe the different states of the network components and consequently deduce the behaviour of the biomolecular network. In addition, these results correspond with results which were obtained earlier with the qualitative reasoning method presented in the previous Section 8.2.3 and the semantic approach based on an SWRL rule-based system in Section 7.4.2.

[1] `http://icube-cstb.unistra.fr/en/index.php/Home`

8.4 Summary

In this chapter, we draw inspiration from the works of [8] to propose a qualitative reasoning method to simulate the behaviour of the biomolecular network. This method is completely based on the logical-based modelling presented above in Chapter 6 that can be assimilated to a causal model. This qualitative simulation clearly demonstrates all the elements that we need to understand the evolution of biomolecular networks. Moreover, we integrate a discrete-event simulation algorithm inspired by the DEVS formalism, into the logical-based modelling of biomolecular networks. This approach aims at providing biologists with a flexible tool for simulating biomolecular networks by reproducing their behaviour and the state of their components over time and consequently allows them to analyse and understand simulated cell phenomena. These approaches have been verified on the bacteriophage T4 gene 32 biomolecular network use case. Simulation results obtained were formally treated and validated by expert biologists. Indeed, these results correspond to their domain knowledge. More important examples will be presented in the Experiments and discussion part (Part III).

In the next chapter, we will introduce a multi-objective genetic algorithm for optimizing the transittability of complex biomolecular networks which will provide the best set of external stimuli for driving the network.

Chapter 9

A multi-objective optimization method for solving the transittability of complex biomolecular networks

Contents

9.1 Introduction

The computation of the transittability of complex biomolecular networks can be considered as an optimization problem. As discussed in Chapter 5, only a few studies have been conducted on this problem. Most of them focused only on the minimization of the required nodes to steer the entire network, and others considered the minimization of the number of stimuli to be applied on the network. However, this assumption is not very realistic, because steering complex biomolecular networks are, in general, a multi-objective optimization problem. It requires finding appropriate trade-offs among various objectives, for example among the distance between the simulated final network state and the desired network state, the appropriate nodes to be stimulated and the number of external stimuli to be used, their cost and the patient comfort.

In this chapter, we firstly propose a multi-objective mathematical formulation for optimizing the transittability of complex biomolecular networks in which we take into account more criteria such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of external stimuli, the minimization of their cost, the minimization of the number of target nodes, the minimization of the patient discomfort. Then, we propose a two-step multi-objective optimization approach for solving this multi-objective problem. Our proposed approach is strongly based on the combination of both Non-dominated Sorting Genetic Algorithm (NSGA-II) [340] to obtain the set of Pareto-optimal solutions, and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method [341] to provide the decision-maker with the best compromise solution according to its preferences.

The first section of this chapter presents a brief description of the problem tackled in this study. Then, the second section proposes the theoretical and mathematical modelling of this problem by introducing its parameters, decision variables, objective functions and constraints. The third section explains and lists the steps of the proposed optimization approach used for solving the given problem, followed by the concluding comments in the last section.

9.2 Problem statement

According to the transittability notion, a complex biomolecular network can be steered from a state to another one through appropriate stimuli. These stimuli can be internal, such as the changes of the physical and chemical properties of the cell, or external such as environmental effects and drugs. Therefore, we can define a stimulus as an action or condition that interferes with a node, which can, in turn, affect other nodes and consequently causes the transition of the entire network. The transition of the biomolecular network starts when one (or more) stimuli trigger one (or more) nodes. Indeed, when the stimulus triggers a node, the state or the concentration of this molecular component will change to reach a specific node threshold. This threshold defines the type of interaction that will occur and the condition that activates it. The state change of a node provokes as well as the change of the overall network state (changing a node automatically modifies other network nodes) creating the stimulus-response behaviour of the biomolecular network. Thus, the state of the biomolecular network at an instant t is a set represented by the set of the states of all components in the network at time t.

Formally, the biomolecular network is defined as an undirected graph denoted by $BN = (M, I)$. Where a node $m_i \in M$ corresponds to a molecular component which can be a gene, protein or metabolite. And an edge $i = (m_i, m_j) \in I;\ m_i, m_j \in M$ expresses the different type of interaction among the molecular components. These interactions can be categorized in the following way: there are three interactions among molecular components of the same type (intraomic interactions), four interactions (among the 6 possibilities) between the nodes belonging to different networks (interomic interactions), and two interactions are not taken into account because there is no direct interaction between the genes and metabolites and vice versa.

For each molecular component m_i, we associate a concentration c_i^t which represents the value of its concentration at time t. The concentration level c_i^t should be inside the interval $[c_i^{min}, c_i^{max}]$, where c_i^{min} and c_i^{max} represent the minimum and maximum concentration value of the molecular component m_i, respectively.

The state transition of the biomolecular network occurs by changing at least the concentration of one of its nodes. These changes in the concentration of the molecule can occur either by an *internal stimulus* (for example, due to reactions that are internal to the cell) already seen with the aggregate function

presented in Chapter 6) or by an *external stimulus* generated outside the cell (for example, because of a medicine taken by the patient). Let S be the external stimuli set. Each stimulus interferes with a node at a time of introduction $S^t_{k,i}$, changing its concentration with a certain concentration $\Delta^t_{c,k,i}$ leading to an increase (or a decrease) of its actual concentration. Moreover, for each stimulus, we associate a cost $CostStim_k$ and the patient discomfort $DiscomfortStim_k$ caused by the stimulus k which should not exceed the maximum discomfort a patient can endure $Discomfort_P^{max}$.

Our goal is to optimize simultaneously the different criteria involved in the transittability of biomolecular networks, in particular, the distance between the simulated final network state and the desired network state, the number of external stimuli, their cost, the number of target nodes and the patient discomfort. Especially by finding the best compromise among these criteria to optimize the steering of the biomolecular network from an unexpected state to a desired state.

More details about the logical definition of this problem can be found in Chapter 6.

9.3 Proposed multi-objective mathematical model

In this section, we propose a multi-objective mathematical model for optimizing the transittability of complex biomolecular networks considering diverse criteria, such as: the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of external stimuli, the minimization of their total cost, the minimization of the number of target nodes, and the minimization of the patient discomfort. It is important to note that the first objective function that focuses on the minimization of the distance between the simulated final network state and the desired network state will be estimated using the simulator defined in the precedent chapter. The notation, parameters, decision variables and constraints of the model are presented in the following sections.

9.3.1 Parameters

Table 9.1 enumerates the parameters of the proposed multi-objective mathematical model.

Table 9.1 – Nomenclature used in the proposed mathematical model.

Symbol	***Description***
P	a patient
$BN = (M, I)$	the complex biomolecular network of nodes M and edges I
$M = \{1, \ldots, m\}$	the set of all the molecular components of the network
$I = \{1, \ldots, n\}$	the set of all the interaction among the molecular components of the network
$S = \{1, \ldots, k\}$	the set of external stimuli
$t = \{1, \ldots, T\}$	the time period
$StartTransi_{BN}$	the starting time of the biomolecular network's transition
$FinishTransi_{BN}$	the finishing time of the biomolecular network's transition
$S^t_{k,i}$	the time of introduction of the stimulus k to the node i
$e_{k,i}$	the execution time of the stimulus k on the node i
c^t_i	the level of concentration of the node i at time t
c^{min}_i	the minimum level of concentration of the node i
c^{max}_i	the maximum level of concentration of the node i
$\Delta^t_{c,k,i}$	the change in concentration caused by the stimulus k on the node i at time t
$DiscomfortStim_k$	the amount of discomfort associated and caused by the stimulus k
$Discomfort_P^{max}$	the maximum amount of discomfort that a patient P can endure

9.3.2 Decision variables

- $x^t_{k,i}$: Binary variable equal to 1 if and only if the stimuli k affect the molecular component i at time t, 0 otherwise.
- $CostStim_k$: Real variable corresponding to the cost of the stimuli k which affect the molecular component i at time t.

- $DiscomfortStim_k$: Nominal variable denotes the level of the discomfort of the patient during the stimulation done by the stimulus k. As described in Section 9.3.3.5, this variable is categorized as $DiscomfortStim_k = 0 : No\ discomfort$; $DiscomfortStim_k = 1 : Light\ discomfort$; $DiscomfortStim_k = 2 : Medium\ discomfort$; $DiscomfortStim_k = 3 : Strong\ discomfort$; $DiscomfortStim_k = 4 : Extremediscomfort$.

9.3.3 Objective functions

In this section, we detail each one of the different criteria considered to optimize the transittability of complex biomolecular networks.

9.3.3.1 Minimizing the distance between the simulated final network state and the desired network state

Before defining this objective function, let us briefly describe how this optimization module is integrated into the proposed simulator presented in the precedent chapter. First of all, we have as an input the initial state of our network and its desired future state. Our goal is to use some stimuli to steer the network from its initial state to the desired state. Thus, we launch our simulator with the initial state and the list of stimuli for a period of time. After this time simulation, the simulator generates a simulated final network state. Now, the problem is how to measure and to evaluate the proximity of the obtained simulated final network state (outputted by the simulator) and the desired future state (provided by the user as an input). Indeed, our main objective is that: the simulated final network state should be as close as possible to the desired future state.

Therefore, to assess the degree of proximity of these two states, we need to define a function that evaluates the degree of goal attainment. In other words, we need a function that evaluates the degree of proximity between the two states: the simulated final network state ($SFNS$) and the desired network state (DNS). This evaluation function should evaluate and assess the individuals of our genetic algorithm. We recall that in our optimization approach an individual represents a list of stimuli over time (see Section 9.4.1.3).

That is why we defined here an objective function that minimizes the difference between the simulated final network state as computed by the simulator and the desired network state which is the goal of the network optimization.

For the sake of clarity, we just use the Euclidean distance between the concentration vectors of the simulated final network state ($SFNS$) and the desired network state (DNS). We recall that the state of the network is defined by the value of the concentration of its nodes. Thus, we compute the distance between these two vectors ($\overrightarrow{m_{SFNS}}$ and $\overrightarrow{m_{DNS}}$) using the following formula:

$$d(\overrightarrow{m_{SFNS}}, \overrightarrow{m_{DNS}}) = \sqrt{\sum_{i=1}^{M}([m_{SFNS,i}] - [m_{DNS,i}])^2}$$

For the moment and in order to check our approach we want to have solutions. That is why we have defined a proximity threshold to be able to decide if the simulated final network state is close enough of the desired network state. However, we are aware that this measure could be improved, for example, by giving different weights to the different nodes according to their biological importance and the more or less important effects of the deviation between the desired concentration and the concentration computed by the simulator.

It is important to note that we treat all the nodes by using concentrations, even with genes (which have states and no concentrations). This explains why we are working on standardized concentrations. It is also important to remember that the simulated final network state is evaluated by the simulator defined in the previous chapter.

To do this, we define the objective function Z_1 which aims to minimize the distance between the simulated final network state denoted by $SFNS$ and the desired network state denoted by DNS, using Eq. (1).

$$Z_1 : \quad Min\ (Distance(SFNS, DNS)) \qquad (1)$$

9.3.3.2 Minimizing the number of external stimuli

As discussed in the previous sections, external stimuli called also *input signals* or *structural perturbations* are necessary for steering biomolecular networks from their actual state to a desired state. Indeed, external stimuli are the key element since they are responsible for steering biomolecular networks.

Therefore, the goal of this objective function is to identify the minimum number of stimuli that are most likely to steer the global biomolecular network from the initial state to the desired state. In other words, this criteria aims to give priority to the quality of the external stimuli than their quantity. Figure 9.1 explain this notion through an example of a biomolecular network which can be steered to the desired state through three possibilities. These possibilities here are explained from the point of view of the number of indispensable external stimuli for steering the network from a state to another one. We can reach the desired state via three different stimulation strategies (a, b and c). In the strategy a, each node receives an external stimulus. In the strategy b, we use three external stimuli. And in the strategy c, we only use two external stimuli. As a result, we note that the strategy c is the best one because we use the minimum number of external stimuli for steering the network.

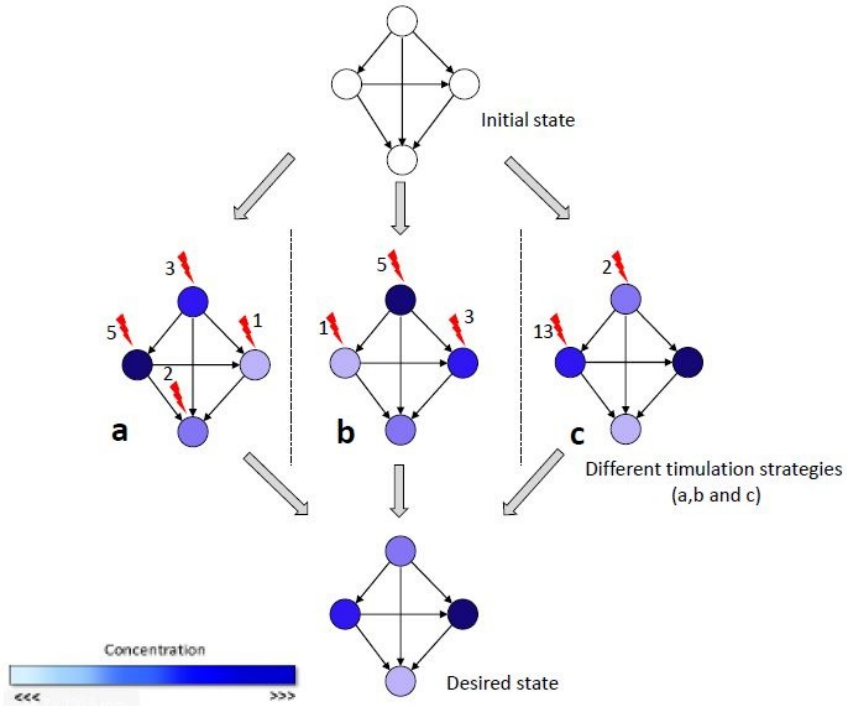


Figure 9.1 – A simple illustration of the transittability of complex biomolecular networks from the number and cost of external stimuli perspective.

To do this, we define the objective function Z_2 which aims to minimize the number of external stimuli for achieving the transittability of complex biomolecular networks using Eq. (2).

$$Z_2 : \quad Min \left(\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^{t}\right) \qquad (2)$$

9.3.3.3 Minimizing the cost of the external stimuli

This criteria is related to the previous objective function (Section 9.3.3.2). In fact, the cost of external stimuli can be proportional to the number of external stimuli. So, if we have a number of external stimuli equal to the number of nodes and all the external stimuli have the same cost, the transittability process of the complex biomolecular network will be very expensive. For these reasons, this criteria aims to find the best compromise between the quality of the external stimuli and their cost. We clarify this objective in the same example used in the first Section 9.3.3.2 (Figure 9.1). Here, we focus only on the cost of the stimuli. In the strategy a, the total cost of external stimuli is 11 $(5 + 3 + 1 + 2)$. In the strategy b, the cost of external stimuli is 9 $(5 + 1 + 3)$. And in the strategy c, the total cost of external stimuli is 15 $(2 + 13)$. Consequently, the strategy b offers the best cost of external stimuli for steering the network.

Thus, we define the objective function Z_3 which aims to minimize the cost of the external stimuli considering their quality for achieving the transittability of complex biomolecular networks using Eq. (3).

$$Z_3 : \quad Min \ (\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^{t} \times CostStim_k) \qquad (3)$$

9.3.3.4 Minimizing the number of target nodes

Several research studies have revealed that among all the nodes composing the biomolecular network, there are some specific nodes that have the ability to steer the network from its actual state to another specific state. Moreover, if we stimulate all the nodes of the network, there would be a probability that side and undesired effects of drugs on the biomolecular network [307, 308, 309]. Thus, instead of stimulating all the nodes randomly, it is better to have a stimulation strategy which targets a set of specific nodes. This will allow stimulating only a minimum number of nodes those allowing the transition of the network to the desired state, so-called the *minimum steering nodes*.

As discussed in Section 5.7.7, a number of researches focused on the minimum steering sets and prove that identify the minimum set of nodes to be affected by external stimuli is a primordial condition to study biomolecular network's transitions. According to Butcher et al. [342] and Yang et al. [343] there are some biomolecular networks for which the perturbation of only a subset nodes of all network molecules can contribute to their transition from a state to a specific state, such as the promyelocytic leukaemia network [343]. However, this is not the case for all biomolecular networks. That is why we have to select only indispensable nodes (driver nodes) among neutral nodes (not profitable nodes). Figure 9.1 explain this notion through an example of a biomolecular network which can be steered to the desired state through three possibilities. These possibilities here are explained from the point of view of the number of indispensable nodes for steering the network from its initial state to the desired state. We can reach the desired state via three different stimulation strategies (a, b and c). In the strategy a, we interfere with all the network nodes. In the strategy b, we interfere three nodes among four. And in the strategy c, we only perturb two nodes. Therefore, we note that the strategy c is the best one because there are only two indispensable nodes for steering the network.

So, we define the objective function Z_4 which aims to identify the minimum number of nodes that are indispensable for steering the network from a state to another using Eq. (4).

$$Z_4 : \quad Min \ (card(TN)) \qquad (4)$$

Notice that TN is the set of the target nodes affected by the stimuli, defined by: $TN = \{ \ i \in M; \ \exists \ k \in S; \ \exists t \in \{1, \ldots, T\}; \ x_{k,i}^{t} = 1\}$.

9.3.3.5 Minimizing the patient discomfort

The transittability of a biomolecular network can potentially be uncomfortable. By way of example, let's take the chemotherapy which is an anti-cancer treatment that consists of acting on cancer cells through toxic drugs (either by injection or sometimes in the form of infusion) until they die and disappear. This treatment corresponds to the transittability of a biomolecular network which generally causes acute pain, vomiting, dizziness, fatigue and stress. As well as, it has been proved that the patient discomfort negatively impacts the emotional and mental health of patients, the quality of their life and increases the use of healthcare resources [344, 345]. For all these reasons, we must consider this important criterion in the transittability process.

Therefore, our objective here is to reduce the patient discomfort during a certain treatment (while finding the best compromise with the other objective functions cited previously). In our context, the patient discomfort encompasses different aspects such as patient pain, stress, vomiting, dizziness, anxiety, fatigue, etc.

Based on the IPREA questionnaire proposed by Kalfon et al. [346] which focuses on the assessment of discomfort perceived by patients related to their intensive care, we define the uncomfortable level felt by a patient as an integer between 0 and 4. The patient is asked to mark his uncomfortable level on the line between the two extremities. These levels are defined as follows:

- Level 0 corresponds to *no discomfort*,
- Level 1 corresponds to *light discomfort*,

- Level 2 corresponds to *medium discomfort*,
- Level 3 corresponds to *strong discomfort*,
- Level 4 corresponds to *extreme discomfort*.

Similar to the minimization of the stimuli cost, this criterion focuses on the minimization the patient discomfort is related to the first objective function (Section 9.3.3.2). Indeed, we note that in our context, the patient discomfort is treated in two distinct forms: (*i*) Explicitly and directly by associating it with the stimulus. For example, swallowing a pill is more comfortable than an injection. This is explicitly considered by this objective function (Eq. (5)). Or (*ii*) implicitly by controlling the side effects of node concentrations. Indeed, for each node, its concentration is evaluated according to its minimum and maximum threshold concentrations. These threshold concentrations (min and max) are defined by the biologists and ensure to check and detect if the current concentration of the node becomes dangerous for the patient. This is handled by the constraint defined later by Eq. (15).

Therefore, the patient discomfort is one of the major aims to be achieved during the optimization of the transittability of biomolecular network. This goal is represented by the objective function Z_5 that aims to minimize the patient discomfort using Eq. (5).

$$Z_5 : \quad Min \ (\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^{t} \times DiscomfortStim_k) \qquad (5)$$

9.3.4 Constraints

In this section, all constraints required to steer the complex biomolecular network from a state to another are presented.

- Constraint (6) ensures that the time of introduction of the stimulus k on a node i is greater than (after) the starting time of the transittability process of the biomolecular network BN.

$$S_{k,i}^{t} > StartTransi_{BN} \quad \forall k \in S, i \in M, t \in T \qquad (6)$$

- Constraint (7) ensures that the time of introduction of the stimulus k on a node i is smaller than (before) the finishing time of the transittability process of the biomolecular network BN.

$$S_{k,i}^{t} < FinishTransi_{BN} \quad \forall k \in S, i \in M, t \in T \qquad (7)$$

- Constraint (8) ensures that the stimuli are introduced by order of time: stimulus $k+1$ begins after the stimulus k finished.

$$S_{k,i}^{t} + e_{k,i} \leq S_{k+1,i}^{t} \quad \forall k \in S, i \in M, t \in T \qquad (8)$$

- Constraint (9) ensures that both stimuli and nodes are acting simultaneously.

$$if \quad \sum_{k \in S} x_{k,i}^{t} = 1 \ \ then \ \ \sum_{i \in M} x_{k,i}^{t} = 1 \quad \forall t \in \{1 \dots T\} \qquad (9)$$

- Constraint (10) ensures the minimum number of indispensable nodes required for the transittability process.

$$\sum_{i=1}^{M} x_{i,k} \geq 1 \quad \forall k \in S \qquad (10)$$

- Constraint (11) ensures the minimum number of external stimuli required for the transittability process.

$$\sum_{k=1}^{S} x_{k,i} \geq 1 \quad \forall i \in M \qquad (11)$$

- Constraints (12) ensure the non-negativity constraints.

$$
\begin{array}{l}
c_i^t \geq 0 \\
\Delta_{c,k,i}^t \geq 0 \\
DiscomfortStim_k \geq 0 \\
CostStim_k \geq 0
\end{array} \qquad (12)
$$

- Constraint (13) represents the binary constraints.

$$x_{k,i}^t \in \{0,1\} \quad \forall k \in S, i \in M, t \in T \qquad (13)$$

- Constraint (14) ensures that the patient discomfort felt during the stimulation should not exceed the limit (maximum) of discomfort.

$$\sum_{k \in S} DiscomfortStim_k \leq Discomfort_P^{max} \quad \forall t \in T \qquad (14)$$

- Constraints (15) ensure that each stimulus affect only one node and each node is stimulated by only one stimulus at a time t.

$$
\begin{array}{ll}
\sum_{k=1}^{S} x_{k,i}^t = 1 & \forall i \in M, t \in T \\
\sum_{i=1}^{M} x_{k,i}^t = 1 & \forall k \in S, t \in T
\end{array} \qquad (15)
$$

- Constraints (16) ensure that the change in concentration applied by the stimulus k on the node i do not exceed both limits minimum and maximum of concentration of the node i.

$$
\begin{array}{ll}
\Delta_{c,k,i}^t + c_i^t \geq c_i^{min} & \forall k \in S, i \in M, t \in T \\
\Delta_{c,k,i}^t + c_i^t \leq c_i^{max} & \forall k \in S, i \in M, t \in T
\end{array} \qquad (16)
$$

According to the above assumptions, the proposed mathematical model for the transittability of complex biomolecular networks is as follows:

$$
\begin{array}{lll}
Z_1: & Min\ (Distance\ (SFNS, DNS)) & (1) \\
Z_2: & Min\ (\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^t) & (2) \\
Z_3: & Min\ (\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^t \times CostStim_k) & (3) \\
Z_4: & Min\ (card(TN)) & (4) \\
Z_5: & Min\ (\sum_{k \in S} \sum_{i \in M} \sum_{t=1}^{T} x_{k,i}^t \times DiscomfortStim_k) & (5)
\end{array}
$$

S.t.

$$
\begin{array}{lll}
S_{k,i}^t > StartTransi_{BN} & \forall k \in S, i \in M, t \in T & (6) \\
S_{k,i}^t < FinishTransi_{BN} & \forall k \in S, i \in M, t \in T & (7) \\
S_{k,i}^t + e_{k,i} \leq S_{k+1,i}^t & \forall k \in S, i \in M, t \in T & (8) \\
if \sum_{k \in S} x_{k,i}^t = 1 \ then \ \sum_{i \in M} x_{k,i}^t = 1 & \forall t \in \{1 \dots T\} & (9) \\
\sum_{i=1}^{M} x_{i,k} \geq 1 & \forall k \in S & (10) \\
\sum_{k=1}^{S} x_{k,i} \geq 1 & \forall i \in M & (11) \\
c_i^t \geq 0 & & \\
\Delta_{c,k,i}^t \geq 0 & & \\
DiscomfortStim_k \geq 0 & & \\
CostStim_k \geq 0 & & (12) \\
x_{k,i}^t \in \{0,1\} & \forall k \in S, i \in M, t \in T & (13) \\
\sum_{k \in S} DiscomfortStim_k \leq Discomfort_P^{max} & \forall t \in T & (14) \\
\sum_{k=1}^{S} x_{k,i}^t = 1 & \forall i \in M, t \in T & \\
\sum_{i=1}^{M} x_{k,i}^t = 1 & \forall k \in S, t \in T & (15) \\
\Delta_{c,k,i}^t + c_i^t \geq c_i^{min} & \forall k \in S, i \in M, t \in T & \\
\Delta_{c,k,i}^t + c_i^t \leq c_i^{max} & \forall k \in S, i \in M, t \in T & (16)
\end{array}
$$

9.4 Multi-objective optimization approach

So the first idea was to use some well-established genetic algorithm software such as the MOEA platform[1] (in Java, mostly not parallel) or the EASY platform[2] (in C and GPGPU cuda for highly parallel genetic algorithm computing) for directly generating executing and solving our multi-objective problem. This platform is a free and open source Java framework for solving multi-objective optimization problems. Indeed, it supports a variety of multi-objective evolutionary algorithms, including genetic algorithms, genetic programming, grammatical evolution, differential evolution, and particle swarm optimization.

However, this idea quickly changed. Indeed, we draw inspiration from the documentation provided by the MOEA platform and the works of Deb et al. [340] to develop our own genetic algorithm, in particular by implementing a non-dominated sorting genetic algorithm. This optimization module was integrated to the simulator presented in the precedent chapter for two main reasons: the first is that we want to have flexibility for individuals which are more specific for the population management (we focus particularly in this criterion). The second reason is to obtain a specific control on this criterion because it is useless to keep bad quality networks.

This section details our proposed optimization approach which consists of two steps. The first one is the search of the of the set of Pareto-optimal solutions. And, the second step is based on the use of a decision-making technique for generating the best compromise solution according to user preferences. As illustrated in Figure 9.2, these two steps are ensured by the combination of the non-dominated sorting genetic algorithm (NSGA-II) as proposed by Deb et al. [340] and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method [341], respectively.

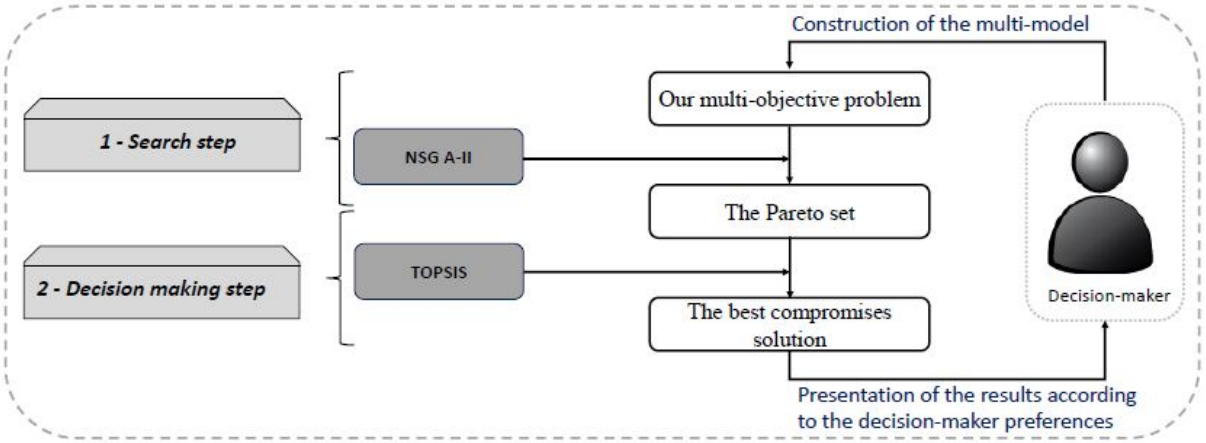


Figure 9.2 – Flowchart of the proposed resolution approach.

9.4.1 First step: search process

9.4.1.1 NSGA-II algorithm overview

Numerous methods such as the weighted-sum method, the goal programming, etc. have been proposed in the literature to solve multi-objective problems by combining their objectives to form a single objective problem and then the optimal solution is obtained [347]. However, in reality, different alternatives should be obtained according to the decision-maker preferences and these methods do not allow it. That is why we chose to use the NSGA-II algorithm which is a powerful metaheuristic to obtain the Pareto-optimal solutions. Moreover, the NSGA-II algorithm is characterized by its elitist strategy, its few parameters, and is less complicated than other variants of multi-objective algorithm [347].

[1] http://moeaframework.org/
[2] https://easyplatform.com/

9.4.1.2 NSGA-II algorithm operation

Similarly to a simple genetic algorithm, the NSGA-II algorithm starts by generating a random set of solutions called *population*. This population consists of a set of individuals also called *chromosome*. The population has a size Np which is an important parameter in the NSGA-II. Then, the objective functions are evaluated for each individual and ranked based on the concept of non-domination (if a solution cannot improve any objective value without degrading one or more of the other objective values). After that, the offspring population is created using the selection, crossover and mutation operators. Then, the best chromosomes are selected using the elitism operator. These steps are repeated until the stopped condition is reached. Finally, the output of the algorithm is a set of solutions that should be near the Pareto-optimal.

9.4.1.3 Genetic algorithm implementation

Here, we explain and detail the steps of the genetic algorithm implementation. The logical diagram of the employed multi-objective genetic algorithm on the basis of NSGA-II is given in Figure 9.3 and its operation is detailed in Algorithm 6. As well as, the genetic algorithm operators were carefully selected based on the requirements of the transittability problem.

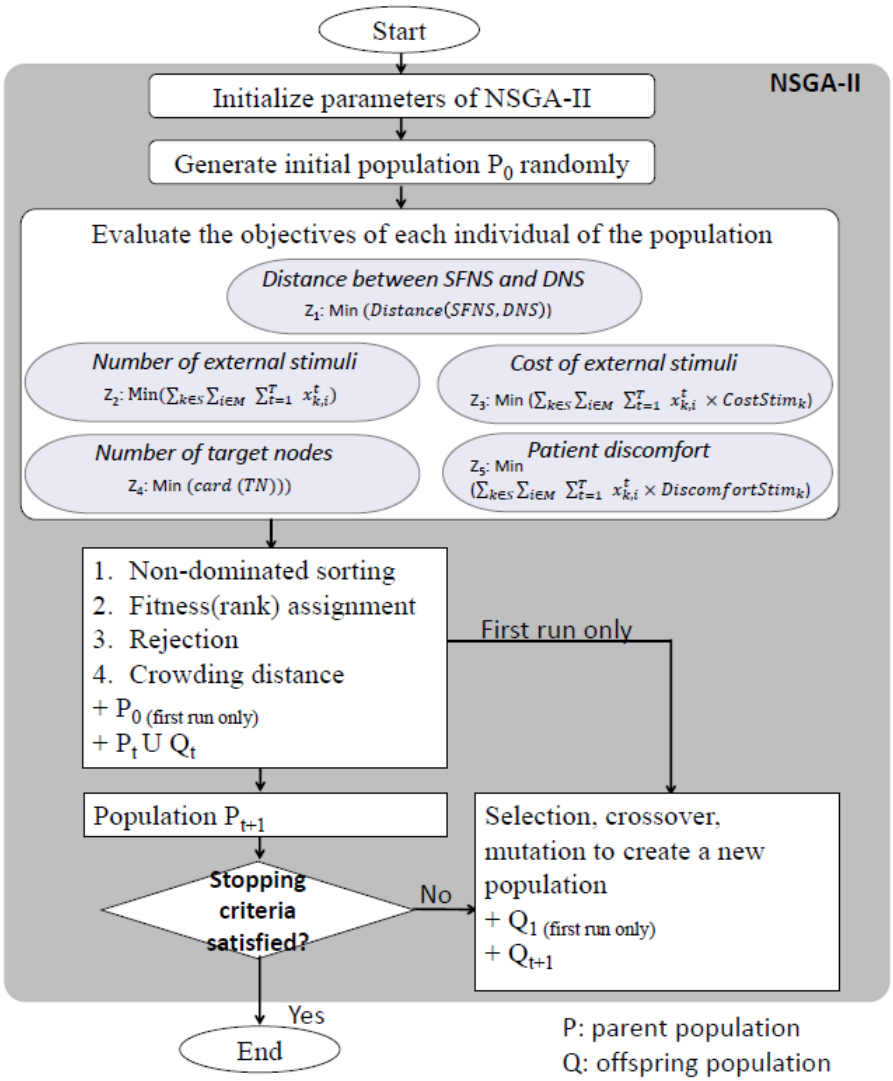
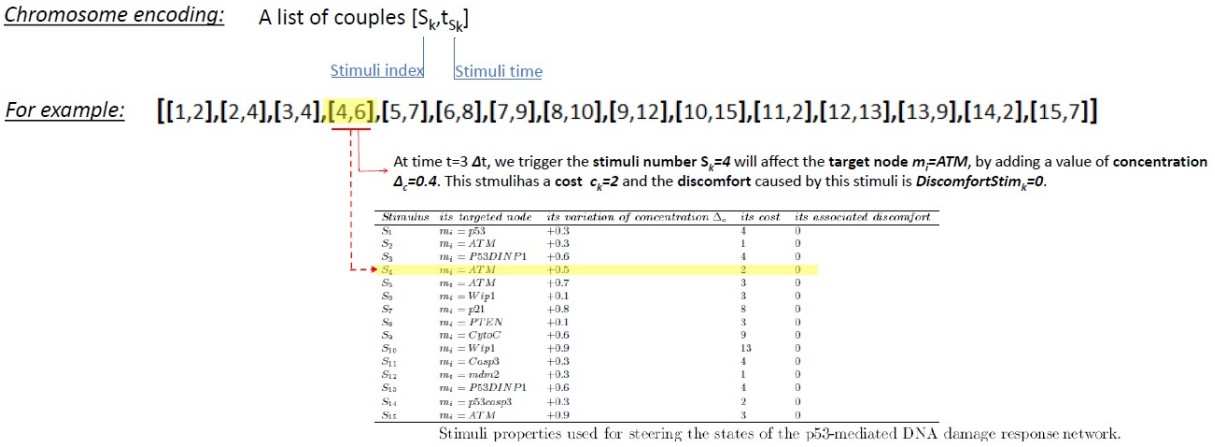


Figure 9.3 – Flowchart of the proposed multi-objective optimization method based on the NSGA-II algorithm.

Chromosome encoding In our context, the representation of a chromosome consists of a list of stimuli. An individual is a couple with the first part the stimuli and in the second part the time of its introduction. These stimuli are part of the biological data imputed by biologists and which are defined by different properties such as the index of the stimuli S_i, the time of introduction of the stimulus S_i into the node m_i, the target node m_i by the stimulus S_i, the variation of concentration caused by the stimulus S_i on the node m_i, the cost of the stimulus S_i, and the discomfort caused by the stimulus $DiscomfortStim_k$. Figure 9.4 illustrates the chromosome encoding considered for an example with $S = 15$. This example of stimuli and their properties can be found in Table 10.2.



Stimulus	*its targeted node*	*its variation of concentration* Δ_c	*its cost*	*its associated discomfort*
S_1	$m_i = p53$	+0.3	4	0
S_2	$m_i = ATM$	+0.3	1	0
S_3	$m_i = P53DINP1$	+0.6	4	0
S_4	$m_i = ATM$	+0.5	2	0
S_5	$m_i = ATM$	+0.7	3	0
S_6	$m_i = Wip1$	+0.1	3	0
S_7	$m_i = p21$	+0.8	8	0
S_8	$m_i = PTEN$	+0.1	3	0
S_9	$m_i = CytoC$	+0.6	9	0
S_{10}	$m_i = Wip1$	+0.9	13	0
S_{11}	$m_i = Casp3$	+0.3	4	0
S_{12}	$m_i = mdm2$	+0.3	1	0
S_{13}	$m_i = P53DINP1$	+0.6	4	0
S_{14}	$m_i = p53casp3$	+0.3	2	0
S_{15}	$m_i = ATM$	+0.9	3	0

Stimuli properties used for steering the states of the p53-mediated DNA damage response network.

Figure 9.4 – Chromosome encoding.

Initial population Initially, the chromosomes are generated randomly creating a first population P_0 ($gen = 0$) with a population of size Np. The objective functions from Equations (1) to (4) are evaluated for each chromosome respectively. Then, the parent population are ranked based on the non-domination concept. In a second step, a child population Q_0 ($gen = gen + 1$) of size Np is created from the parent population P_0 by the use of the selection, crossover and mutation operators.

Selection (NSGA-II) Considering the obtained chromosomes, the population is sorted based on the non-domination principle. This elitism method consists: (i) firstly in searching the dominated individuals in the population and ranking them according to their dominance using Equation (17) (where X and Y are two individuals and x_i, y_i are objective functions). Then, (ii) the selection of those which have the greater rank. In the case of two individuals with the same rank of dominance, we compute the crowding distance between them as defined by Equation (18) (where $d(k)$ is the crowding distance of individual k, f_j^k is the j^{th} objective function value of the k^{th} individual, and f_j^{min}, f_j^{max} are the minimum and maximum value of the j^{th} objective function, respectively). Indeed, the crowding distance value of a solution provides an estimation of the density of solutions surrounding that solution. In other words, the crowding distance value of a particular solution is the average distance of its two neighbouring solutions. It is a measure of "*density of solutions surrounding a particular solution in the population*" [340]. The individual having the greater crowding distance value is better than those having a small value.

$$\forall X = \{x_1, \ldots, x_M\} \quad and \quad Y = \{y_1, \ldots, y_M\} \quad Then \quad X \preceq Y \Leftrightarrow \forall i : x_i \leq y_i \; and \; \exists j : x_j \leq y_j \tag{17}$$

$$d(k) = \sum_{j=1}^{M} \frac{|f_j^{k+1} - f_j^{k-1}|}{f_j^{max} - f_j^{min}} \tag{18}$$

Crossover To perform the crossover, two parents are randomly selected. A point, representing here the time of introduction of a stimulus, is randomly selected and designated as the crossover point. Then, the stimuli that are very next to the crossover point are interchanged. Finally, offspring chromosomes

are obtained by copying the beginning of one parent to the crossover point and the rest is copied from the second parent. Figure 9.5 presents an example illustrating this single point crossover operator.

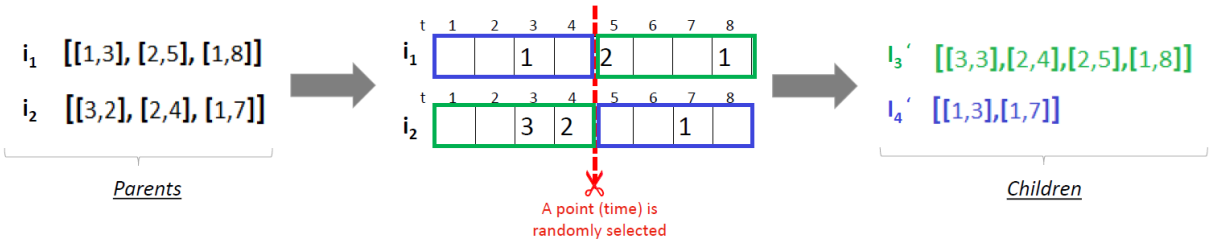


Figure 9.5 – Single point crossover.

Mutation In our work, we use a simple mutation operator where a P_M per cent of the chromosomes are randomly mutated. Our mutation operator operates in two major ways. Firstly, we randomly select an individual (by doing a random on the list of stimuli forming the chromosome). This random choice is applied to the stimuli or their time. Then, we remove or add a stimulus at this selected point time. This exploration phase consists of probing a much larger section of the search space looking for other promising solutions that are yet to be refined. This operation aims to diversify the search in order to avoid getting trapped in a local optimum. It is considered as a global search. Secondly, we randomly select an existing stimulus. Then, we change its time by adding a very small value Δt (+1 or -1) that allows increments as well as decrements the time to an optimum value. This method allows obtaining solution close to the optimum solution. This intensification phase allows refining our solutions looking for finding better solutions.

Fitness function For the sake of clarity, we just use the Euclidean distance between the concentration vectors of the simulated final network state ($SFNS$) and the desired network state (DNS). That is why our fitness function is the same as the first objective function Z_1 that aims to minimize the distance between the simulated final network state denoted by $SFNS$ and the desired network state denoted by DNS. We recall this function:

$$\text{Fitness function} = \quad Min\ (Distance(SFNS, DNS))$$

We remember that this evaluation function is done by the simulator defined in the previous chapter.

Stopping criteria The previous steps are repeated until reaching the stopping criterion. In our context, the stopping criterion is the limitation on the maximum number of generations.

9.4.2 Second step: decision making

In order to select the appropriate "optimal" solution among the set of Pareto-optimal set generated by the first step, we integrate a multi-criteria decision-making analysis method called *TOPSIS*.

9.4.2.1 TOPSIS method overview

The decision-making process requires decision-maker interaction. TOPSIS aims to rank a certain number of alternatives from the most preferred to the least preferred, with a view to supporting the decision-maker in its selection of the most appropriate alternative under uncertain criteria. This method is a part of the techniques used in the multiple criteria decision-making domain and it was developed by Hwang and Yoon in 1981 [341]. TOPSIS is based on two main features, *options* which represent the list of solutions that can be considered as a decision, and *criteria* which represent the criteria needed to make an optimal decision.

As presented above, in our multi-objective problem no single solution exists that simultaneously optimizes all the objectives. Besides, there exists a number of Pareto-optimal solutions generated by the genetic algorithm defined in the previous section. optimal solutions and decisions required the user

Algorithm 6 Pseudocode of the employed non-dominated sorting genetic algorithm (NSGA-II) [340]

Require: The parent population $P = \varphi$; The child population $Q = \varphi$; The collect population $R = \varphi$; The generation index population $gen = 0$; The maximum number of generation $MaxGen$
Ensure: The populations P are the non-dominated solutions
1: Randomly initialize the parent population P_0.
2: **while** stopping condition not satisfied ($gen < Maxgen$) **do**
3: Combine the parent and child populations $R_{gen} = P_{gen} \cup Q_{gen}$
4: Rank individuals of R_{gen} to obtain the non-dominated fronts: F = fast-non-dominated (R_{gen})
5: $P_{gen} + 1 = \varphi$ and $i = 1$
6: **while** the parent population size $|P_{gen} + 1| + |F_i| < N$ **do**
7: Compute the crowding-distance of F_i
8: Add the i^{th} non-dominated front F_i to the parent population $P_{gen} + 1$
9: $i = i + 1$
10: **end while**
11: Rank the F_i according to the crowding distance
12: Complete the parent population $P_{gen} + 1$ with the first $N - |P_{gen} + 1|$ elements of F_i
13: Generate the child population $Q_{gen} + 1$
14: $gen = gen + 1$
15: **end while**

intervention and need to be taken in the presence of trade-offs among all our objective functions. Such decision-making involves confronting trade-offs between multiple, conflicting objectives or criteria. In the transittability problem, the user is confronted with five objective functions, even contradictory, and with subjective and objective criteria in any mix. This requires to consider the user preferences. What makes the decision task more difficult. For these reasons, we integrate the TOPSIS method to our optimization module. Indeed, this method offers more flexibility and freedom to the decision-maker considering its preferences. The user selects amongst the objective functions which of them are to be used as objective functions, albeit he can use them all, but also has the possibility to associate weights according to their relative importance. Moreover, this method takes into account additional subjective preference information, this allows classifying the Pareto-optimal solutions obtained in the search step by sorting them in order of priority according to the decision-maker preferences.

9.4.2.2 TOPSIS method operation

According to the preferences given by the decision-maker, the TOPSIS method will select the appropriate "optimal" solution (among the set of Pareto-optimal solution) that is close to its preferences and requirements. In our context, the Pareto-optimal set constitutes the alternatives of the TOPSIS method. Its principle consists on compute firstly the distance measure among the different alternatives to define the ideal and negative-ideal solution. The distance between the ideal point and each alternative can be calculated using Eq. (19). Using the same separation measure, the distance between the negative ideal point and each alternative can be determined (Eq. (20)) [348]. The relative closeness to the ideal point can be calculated using Eq. (21). Where v_{ij} is the weighted standardized criterion value of the i^{th} alternative that is calculated by multiplying standardized criterion value by the corresponding weight, and v_{+j} is the ideal value and v_{-j} is the negative ideal value for the j^{th} criterion [348].

$$S_{i+} = [\sum_{j=1}^{n}(v_{ij} - v_{+j})^2] \qquad (19)$$
$$S_{i-} = [\sum_{j=1}^{n}(v_{ij} - v_{-j})^2] \qquad (20)$$
$$c_{i+} = \frac{S_{i-}}{S_{i+} + S_{i-}} \qquad (21)$$

Then, it associates to each alternative a numerical coefficient between 0 and 1 according to the Euclidean distances between each alternative on the one hand, and the ideal and negative-ideal solutions on the other hand. Next, it ranks the alternatives (their measures) according to the importance of the attribute starting by the appropriate alternative (that have the shortest distance from the ideal solution and the longest distance from the negative-ideal solution) to the bad one [349]. This is how the Pareto-optimal solutions are ranked, compared and proposed to the decision-maker order by its preferences.

Algorithm 7 Pseudocode of the TOPSIS technique

1: Establish a matrix of criteria and different alternatives
2: Normalize the decision matrix
3: Compute the weight of the normalized decision matrix
4: Determine the ideal solutions and nadir solution (negative ideal solution)
5: Compute the distance for each alternative
6: Compute the relative closeness to the ideal solution
7: Rank the preference order

Algorithm 7 depicted the pseudo-code of the TOPSIS decision-making technique. In our problem, the alternatives of the TOPSIS are the set of Pareto-optimal solution generated by the search step (they are the options which are to be evaluated for selection the best). The criteria are our five objective functions, the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of input signals, the minimization of the cost of these external stimuli, the minimization of the number of target nodes and the minimization of the patient discomfort (these will impact the selection of alternatives). In this method, two properties should be considered:

- The completeness: it is important to ensure that all the criteria are included.
- The operationality: it is important that each alternative can be judged against each criterion.

To each criterion, we associate a weight that will estimate the relative importance of this objective function. All the TOPSIS steps (summarized in Algorithm 7) are detailed in the Wikipedia encyclopedia[3].

9.5 Summary

The transittability of complex biomolecular networks is a multi-criteria problem by nature since there are several potentially conflicting criteria to consider while steering the network from an initial state to a desired state. In this chapter, five essential criteria, which need to be minimized simultaneously to steer complex biomolecular networks, are presented and described in detail. There is the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of input signals, the minimization of the cost of these external stimuli, the minimization of the number of target nodes and the minimization of the patient discomfort. Applying a minimum number of external stimuli on a minimum number of steering nodes has been already considered in existing mathematical models in the literature, however, these criteria are not sufficient for completely steering complex biomolecular networks. That is why other criteria have been considered in this proposed mathematical model.

Moreover, in this chapter, a multi-objective optimization approach for solving this problem has been proposed and detailed. This optimization approach consists of two steps: the *search* and *decision-making* steps. The search step is based on a powerful multi-objective genetic algorithm, the NSGA-II, to solve our problem and obtain a Pareto-optimal set. Indeed, we draw inspiration from the documentation provided by the MOEA platform and the works of Deb et al. [340] to develop our own genetic algorithm, in particular by implementing a non-dominated sorting genetic algorithm. While the decision-making step is based on a multi-criteria decision-making method, the TOPSIS, to compare the Pareto-optimal solutions and provide the decision-maker with the best compromise solution according to its preferences.

This optimization module was integrated to the simulator presented in the precedent chapter for two main reasons: the first is that we want to have flexibility for individuals which are more specific for the population management (we focus particularly in this criterion). The second reason is to obtain a specific control on this criterion because it is useless to keep bad quality networks.

The next part is devoted to the validation of our proposed approaches.

[3] https://en.wikipedia.org/wiki/TOPSIS

Part III

Experiments and discussion

From Chapter 6 to Chapter 9, we have introduced the four contributions for achieving the overall goal of this study, which is to propose a platform that enables biologists to simulate the state changes of biomolecular networks with the goal of steering their behaviours.
In order to verify the proposed approaches, a prototype intituled CBNSimulator, which combines the function of the approaches stated above in Part II, has been developed. In Chapter 10, we illustrate the potential of the prototype based on three case studies. This chapter also introduces the prototype interfaces and features. Then, in Chapter 11, we evaluate and discuss the performance of our proposed approaches according to the results obtained by exploiting the case studies.
This part is divided into two chapters:

Chapter 10

Prototype: the CBNSimulator

Contents

10.1 Introduction

From Chapter 6 to Chapter 9, we have detailed the different contributions for modelling, simulating, understanding and optimizing the transittability of complex biomolecular networks. Indeed, Chapter 6 introduces the proposed logic-based modelling for describing the different elements required for modelling the transittability of biomolecular networks following the systems theory. Chapter 7 enhances this logic-based modelling with an additional semantic layer to obtain an optimal and more realistic modelling describing the structure, function, behaviour and semantic of complex biomolecular networks. Both of these Chapters 6 and 7 are the foundations of the simulator presented in Chapter 8 that deals with a qualitative, discrete-time simulation for reproducing the behaviour of complex biomolecular networks. Finally, Chapter 9 presents a multi-objective optimization approach for solving the transittability of complex biomolecular networks.

In order to verify these contributions, we provide biologists with a flexible platform called the 'CBNSimulator' which combines the function of the four approaches stated above. This chapter focus on presenting the CBNSimulator as an innovative multi-domain collaborative platform for the modelling, simulating, understanding and optimizing of complex biomolecular networks. The first section presents its main objectives. The general architecture of the proposed platform is explained in the second section. The environment and tools for its implementation are illustrated in the third section. The last section illustrates the advantages of this platform by applying it to three widely studied biomolecular networks, the autoregulation of the bacteriophage T4 gene 32 network, the control of the phage lambda infection of bacteria and the p53-mediated DNA damage response network.

10.2 Aims of the CBNSimulator platform

The main goals of the CBNSimulator platform can be grouped into four important points:

1. *The logical modelling of complex biomolecular networks.*
   The main goal of the proposed CBNSimulator is to produce a whole-cell computational model. This computational model consists of a logic-based modelling that aims to represent, analyze and interpret the complex structure, the different functional aspects, and the dynamic behaviour of these multi-level biomolecular networks.

2. *The semantic modelling of complex biomolecular networks.*
   The second purpose of the CBNSimulator aims to provide a rich semantic description of the transittability of biomolecular networks. This semantic modelling should infer more knowledge about the functioning of biomolecular networks and provides the new knowledge required in understanding their behaviours and their state changes.

3. *The simulation of the behaviour of complex biomolecular networks.*
   The third goal of the CBNSimulator platform is to reproduce the conditions of the evaluated biomolecular network and its components over time. This simulation should predict the important properties of biomolecular networks even when quantitative data of such networks are unavailable or unknown.

4. *The optimization of the transittability of complex biomolecular networks.*
   The last goal of the CBNSimulator is to provide biologists with a tool that is able to control and guide the behaviour and the transition states of complex biomolecular networks. In other words, by optimizing the steering of complex biomolecular networks form their actual state to a desired state taking into account different constraints such as, the distance between the simulated final network state and the desired network state, the number of external stimuli, their cost, the number of target nodes and the patient discomfort.

10.3 Overview of the CBNSimulator platform

CBNSimulator provides a series of facilities (i) to model and simulate a given biomolecular network formalised through a logic-based model and semantically described through a semantic modelling, and

(*ii*) to steer these networks from their actual state to a desired state under specific and controlled conditions. As illustrated in Figure 10.1, the platform is organized into four main modules: the logic modelling module, the ontological module, the simulation module, and the optimization module.

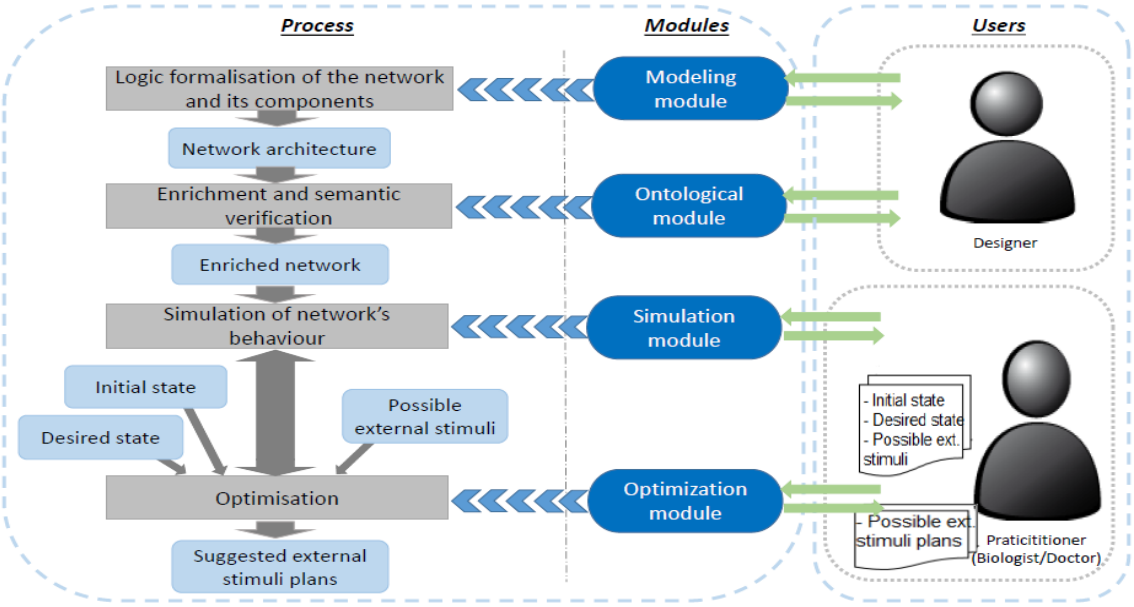


Figure 10.1 – Overall architecture of the CBNSimulator platform.

1. *The logic modelling module*: It represents the starting point for any new user-defined biomolecular network. It has been designed to provide all the necessary elements for modelling the biomolecular network by considering its different levels and molecular components. For example, the user can specify the list of molecular components and their corresponding type, the list of interactions among these components and their corresponding conditions that activate them, etc. This logic-based modelling takes into account the complexity and heterogeneity of these molecular components and their multilevel structure.

2. *The ontological module*: It ensures the management, modelling and sharing of expert knowledge. This module takes as input all the native information introduced by the expert (state of the network, its structure, etc.) through the logic-based formalization provided by the first module. Then, the ontological module provides output inferred network that is composed of native and inferred knowledge. This represents useful features, especially when the logic-based model lacks these details. Moreover, the additional information provided by this module about the network's elements can be used to identify new potential relationships among them.

3. *The simulation module*: It allows users to simulate and/or reproduce the dynamical behaviour of the network. Indeed, this simulator integrates all the information given by the expert (the enriched network with native and inferred knowledge) with other parameters in order to better reproduce the conditions of the evaluated biomolecular network and its components over time. The results generated during the simulation may be displayed to the users in graphical form to facilitate their interpretation, or used by the optimization module.

4. *The optimization module*: It provides a way to steer the states of the network from its actual state to another specific state. This optimization is performed by specifying the initial and desired states of the network, and all the possible external stimuli defined by the user. Then, based on the values of the evaluation criteria, this module provides the best transition sequences for steering the biomolecular network from its initial state to the desired state and, finally, presents the results according to the user preferences. This model allows also to optimize the transittability of the network by minimizing various criteria, such as the distance between the simulated final network

state and the desired network state, the number of external stimuli, their cost, the number of target nodes and the patient discomfort.

10.4 Development tools

In this section, we present the different software used for developing the CBNSimulator platform:

- *Java Platform, Standard Edition (Java SE)*[1] The platform uses Java programming language and is part of the Java software-platform family. Java is particularly well-suited for creating complex analytical applications and architecture.
- *Eclipse*[2] It is an integrated development environment for developing with Java, but it may also be used to develop applications in other programming languages via plug-ins. It is the most widely used Java IDE. Eclipse contains a base workspace and an extensible plug-in system for customizing the environment. Considering the features of the CBNSimulator platform, that is, a frame-based Java SE application, Eclipse Luna Service Release 2 (4.4.2) is chosen as the environment for its implementation.
- *Protégé*[3] It is a free, open source ontology editor and knowledge-base framework building intelligent systems. As well as the Protégé editor supports modelling ontologies through a web client or desktop client. Protégé supports various ontologies formats, such as, RDF, RDF Schema, XML Schema and OWL. In our researches, the proposed ontologies have been implemented within the Protégé 5 environment, version 1.3.8.3.

10.5 Experimental results

In order to better explain how CBNSimulator can be applied for modelling, simulation and optimization of complex biomolecular networks, we analysed, as case studies, the autoregulation of the bacteriophage T4 gene 32 [318], the control of lifecycle of bacteriophage lambda [350], and the p53-mediated DNA damage response network (cell fate decision) [6].

10.5.1 Case study 1: the bacteriophage T4 gene 32

This case study has already been used as a motivating example along the contributions clarifying their notions. Indeed, it should be mention that along with our contributions, this example has been presented as a three-node network consisting of the gene 'G32', the protein 'p32' and the metabolite 'm32'. The node m32 has been arbitrarily added in order to highlight the different level of cell's components (gene, protein and metabolite) and to better explain our contributions. Here, to conform to biology, we treat the original network which is only the gene 'G32' and the protein 'p32'. We chose this case study due to its simplicity and its well-known mechanism.

10.5.1.1 Description

The bacteriophage T4 gene 32[4] encodes a single-stranded DNA binding protein required for T4 DNA replication, recombination, and repair [318]. It is a single polypeptide chain of 301 amino acid residues that consists of three structural domains, each of which has a binding function. Despite its role in DNA metabolism, the gene product 32 autoregulates its synthesis at the level of translation [319]. During the infection, the gene product 32 is produced in large amounts to perform its function of binding all available DNA at the replication fork, recombination nodes and at lesions in DNA resulting from damage [320]. When all the available DNA is bound, free gene product 32 accumulates within the cell until it reaches a certain concentration which then attenuates further synthesis of gene product 32 [321]. More detail about the bacteriophage T4 gene 32 can be found in [318, 320, 321].

[1] https://www.oracle.com/fr/java/technologies/java-se.html
[2] https://www.eclipse.org/
[3] http://www.hermit-reasoner.com/
[4] `http://genes.atspace.org/10.11.html`

As illustrated in Figure 10.2, this biomolecular network consists of two nodes a **gene G32** coding for a **protein p32**, which in turn can inhibits or activates the gene G32 according to the value of its own concentration. Therefore, in this network, the concentration of $p32$ is self-regulated and normally should remain between 0.2 10^{-6} mol/dm^3 and 0.7 10^{-6} mol/dm^3. When the concentration of $p32$ exceeds the threshold S_{p32} = 0.7 10^{-6} mol/dm^3, it is called an **Inhibition**, i.e. the protein $p32$ inhibits, or deactivates, the translation of its gene $G32$. However, when the concentration of $p32$ decreases and becomes lower than the threshold $S_{p32} = 0.2\ 10^{-6}$ mol/dm^3, it is called an **Activation**, i.e. the protein $p32$ activates the translation of its gene $G32$. When the gene $G32$ is activated by the protein $p32$, it is called a **Translation**, in which we have a production of $p32$ thus increasing the value of its concentration.

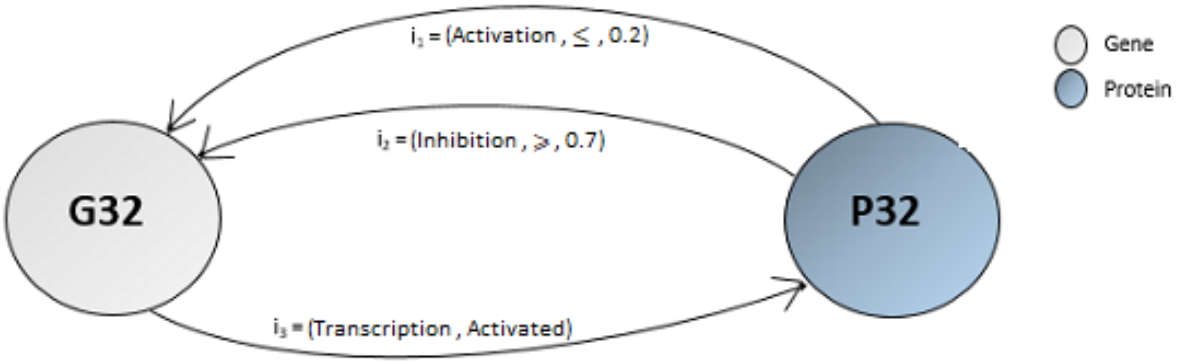


Figure 10.2 – The bacteriophage T4 gene 32 use case.

10.5.1.2 Logical modelling

The logic modelling of the bacteriophage T4 gene 32 is illustrated in Table 6.4 (Section 6.5).

10.5.1.3 Semantic modelling

The sematic modelling of the bacteriophage T4 gene 32 was already detailed in Section 7.4. Indeed, we present the instantiation of the BNO ontology for the given example in Figure 7.5. We highlight also the different properties of the instantiation its components (the gene $G32$, protein $p32$ and metabolite $m32$) in Figure 7.6. We describe also the instantiations of its four interactions in Figure 7.7.

Moreover, we use the BNO ontology to simulate the behaviour of its components. To do this, we use an SWRL rule-based reasoner, and we develop some SWRL rules such as the inhibition SWRL rule and its opposite rule, activation SWRL rule and its opposite, the transcription SWRL rule and its opposite rule, and the negative regulation SWRL rule and its opposite rule. Results of these rules are illustrated in Figures 7.8, 7.9, 7.10, 7.11, 7.12, 7.13, respectively.

10.5.1.4 Simulation under the CBNSimulator

The simulation results of the bacteriophage T4 gene 32, provided by the CBNSimulator, were synthetically represented in graphical form in Figure 8.5. These results were formally treated and validated by expert biologists. Indeed, these results correspond to their domain knowledge and the real behaviour of the network.

10.5.2 Case study 2: the control of the lifecycle of bacteriophage lambda

The control of the lifecycle of bacteriophage lambda infection is one of the best understood regulatory systems [350]. We use this the phage lambda because it is a simple and realistic case study with a well-known mechanism. Indeed, its functioning was investigated by several studies such as [350, 351, 352, 353, 354, 355].

10.5.2.1 Description

The phage lambda[5] is a virus that infects the bacteria Escherichia Coli. It is called a *temperate* bacteriophage, because it can alternate between two possible developmental pathways: the **lytic cycle** and the **lysogenic cycle** under certain conditions [355]. During the lytic cycle, the phage infects the bacteria: its DNA is replicated in large quantities within the bacteria to make lots of phages, and then kills the cell by making it explode (its destruction). During the dormant lysogenic cycle, the phage inserts its DNA into the bacterial chromosome allowing the phage DNA (called also prophage) to be copied and the cell continues to reproduce normally [350]. Figure 10.3 illustrates the switching between the lytic and lysogenic phase. Indeed, during the infection, the phage lambda have two choices: (*i*) replicate and kill the host cell (Lysis) or (*ii*) integrate into the bacterial chromosome, where it replicates as a part of the cell genome (Lysogeny) [5]. The phage lambda state generally in the lysogenic phase. However, when it is in the lysogen phase it can be switched to the lytic phase through ultraviolet stimuli.

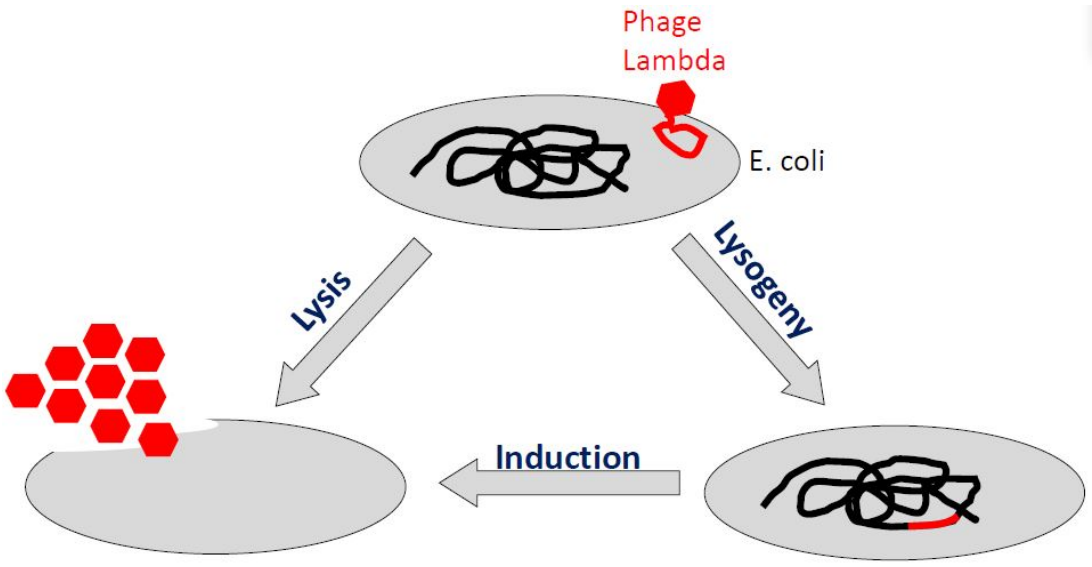


Figure 10.3 – The lifecycle of bacteriophage lambda. (inspired from [5])

As illustrated in Figure 10.4, this biomolecular network consists of six nodes: **four genes** G_CI, G_OR3, G_OR1, G_CRO, and **two proteins** P_CI, and P_CRO. During the lytic phase, the gene G_CI is activated (and G_CRO is deactivated). This activation of the gene G_CI generates its transcription by producing a protein P_CI. When the concentration of this protein reached a threshold equal to 0.1, it deactivates the gene G_OR1, which in turn deactivates the gene G_CRO (this provides the lysogenic phase). Then, when the gene G_CRO gene is deactivated, there is no production of the protein P_CRO. The expression of lysogenic genes (G_CRO and G_OR1) is therefore deactivated. In this case, we are totally in the lytic phase. After a certain time in this lytic state, the destruction of the host cell occurs. For reason of clarity, we reduce this functioning in a set of rules (the black arrows in Figure 10.4) as follows:

1. When the gene G_CI is activated, there is a transcription of the protein P_CI.
2. Once the protein concentration reached $P\_CI] > 0.1$, it inhibits the gene G_OR1.
3. When the gene G_OR1 is deactivated, it deactivates the gene G_CRO.
4. Once the gene G_CRO is deactivated, there is an absence of transcription.

On the other hand, in the lysogenic phase the gene G_CRO is activated (and G_CI is deactivated). This activation of the gene G_CRO generates its transcription by producing a protein P_CRO. When

[5] https://cshmonographs.org/index.php/monographs/issue/view/087969102.2

the concentration of this protein reached a threshold equal to 0.2, it deactivates the gene G_OR3, which in turn deactivates the gene G_CI (this provides the lytic phase). Then, when the gene G_CI is deactivated, there is no production of the protein P_CI. The expression of lytic genes (G_CI and G_OR3) is therefore deactivated. In this case, we are totally in the lysogenic phase. The state of the phage lambda stays at this state even in the absence of ultraviolet stimuli. For reason of clarity, we reduce this functioning in a set of rules (blue arrows in Figure 10.4) as follows:

1. When the gene G_CRO is activated, there is a transcription of the protein P_CRO.
2. Once the protein concentration reached $P\_CRO > 0.2$, it inhibits the gene G_OR3.
3. When the gene G_OR3 is deactivated, it deactivates the gene G_CI.
4. Once the gene G_CI is deactivated, there is anabsence of transcription.

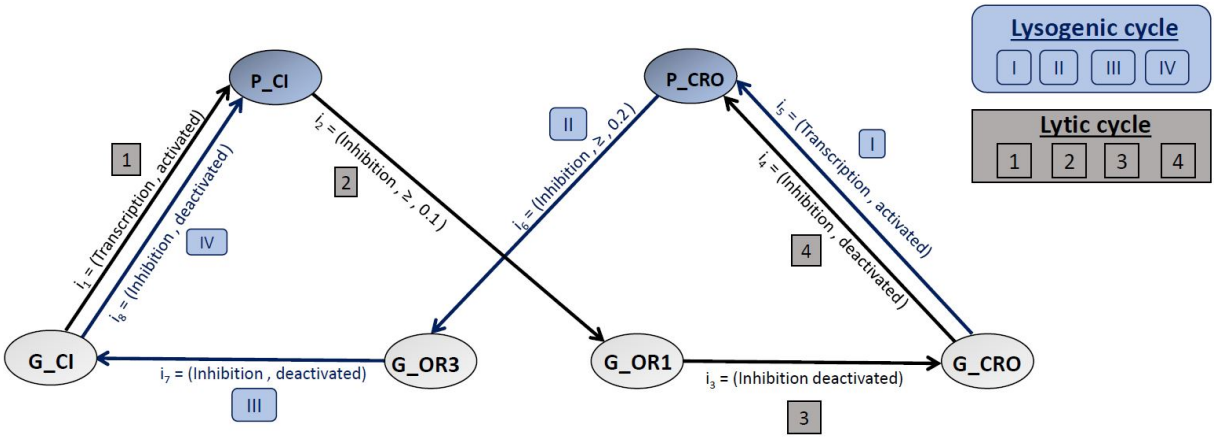


Figure 10.4 – Functioning rules of the phage lambda.

10.5.2.2 Logic-based modelling

Table 10.1 presents the structure and the function of the logical modelling of the phage lambda. Its causal graph is given by Figure 10.4. The possible states that can have the phage lambda network during the simulation are represented by the behaviour CR, as illustrated by Figure 10.5:

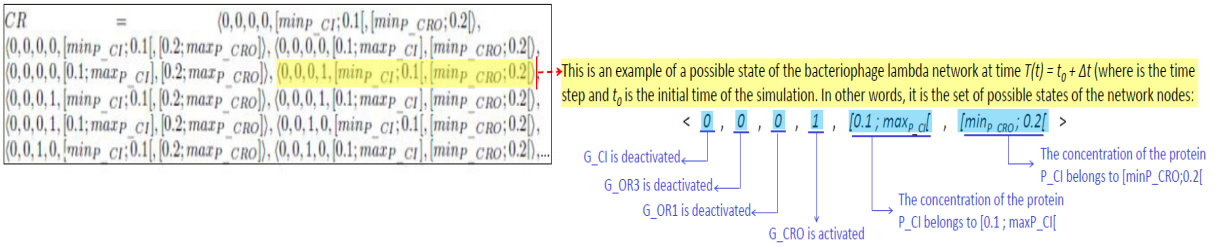


Figure 10.5 – An excerpt of the possible states that can have the phage lambda network during the simulation.

10.5.2.3 Semantic modelling

Here, we use OWL and SWRL rules for semantically modelling our case study and simulating its behaviour. To do this we instantiate our ontology for the given example. Figure 10.6 depicted the BNO individuals' of the phage lambda considering their various characteristics. The molecular components

and the interactions that occur among them are represented by individuals and corresponding relations. The relations are restricted by specifying domain and range. All the concepts and individuals are linked to primitive data (concentrations values, genes states, etc.) through the data properties. This semantic modelling provides all the elements for modelling the states and the individuals behaviours of phage lambda through the use of SWRL rules. This set of SWRL rules is able to reason and simulate the behaviour of the phage lambda. The following paragraphs demonstrate the simulation of the given case study using SWRL rules.

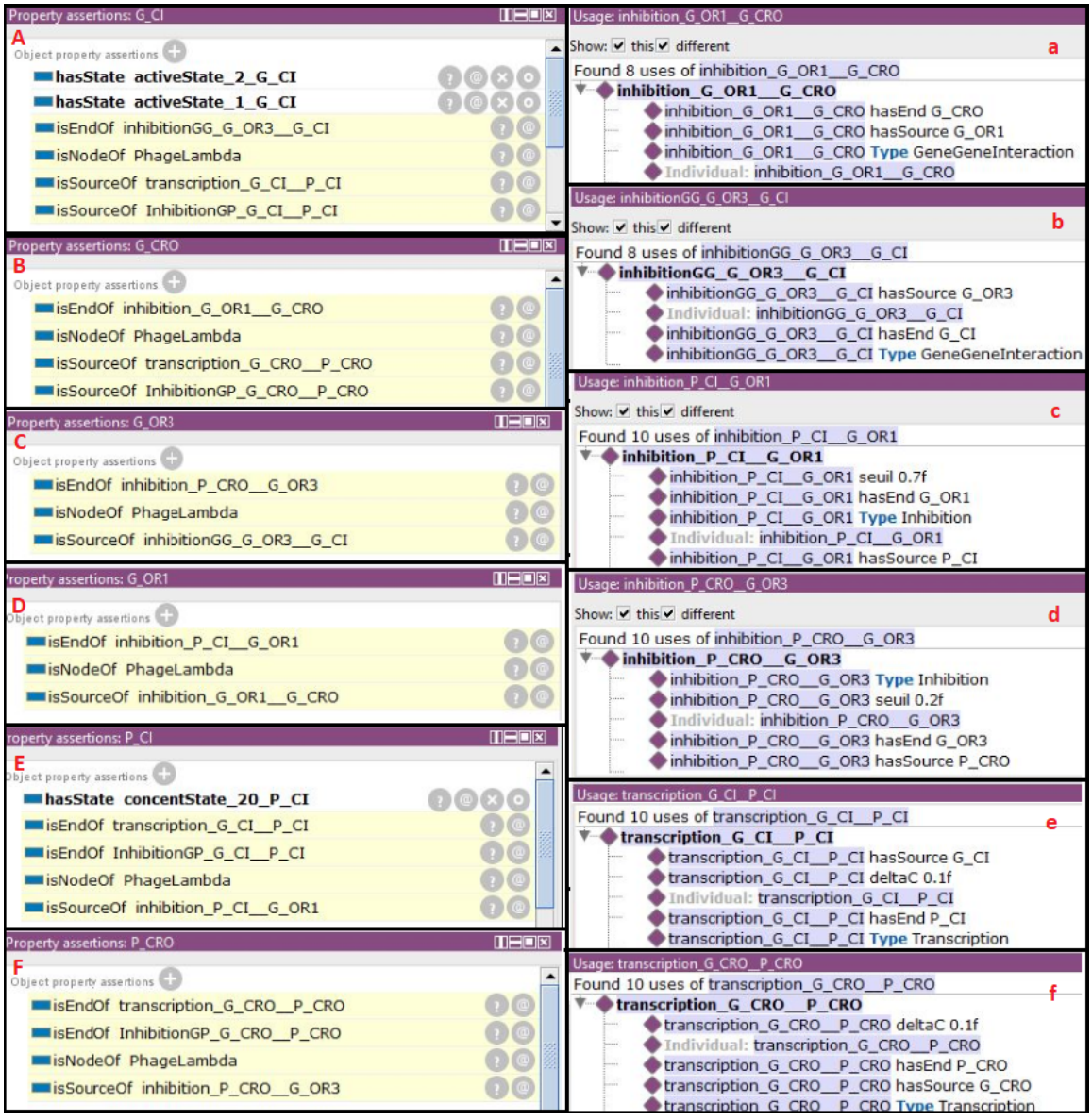


Figure 10.6 – Semantic modelling of the phage lambda within the Protégé editor. The molecular components: A- G_CI, B- G_CRO, C- G_OR3, D- G_OR1, E- P_CI, F- P_CRO. Some interactions: a- i_3, b- i_7, c- i_2, d- i_6, e- i_1, f- i_5.

Inhibition SWRL rule between proteins and genes (interactions i_2 and i_6) The following rule models the *inhibition* interaction that occurs between the proteins P_CI, P_CRO and the genes G_OR1, G_OR3, respectively. When the concentration of the proteins (P_CI and P_CRO) exceeds the threshold 0.2 10^{-6} Mol/L^{-1}, they inhibit the translation of their targeted genes (G_OR1 and G_OR3).

$$ADN(?g) \wedge hasState(?g,\ ?gs1) \wedge forTime(?gs1,\ ?t) \wedge hasState(?g,\ ?gs2) \wedge forTime(?gs2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \wedge Protein(?p) \wedge Inhibition(?inhi) \wedge hasSource(?inhi,\ ?p) \wedge hasEnd(?inhi,\ ?g) \wedge hasState(?p,\ ?ps) \wedge forTime(?ps,\ ?t) \wedge hasConcentrationValue(?ps,\ ?c) \wedge swrlb{:}greaterThanOrEqual(?c,\ 0.7) \rightarrow isActivated(?gs2,\ false)$$

The results of this rule means that, *If there is a protein p having a state ps equal to a concentration c at a given time t and there is a gene g having a state gs equal to true at this time t, and these two molecules p and g are related by an Inhibition interaction, and if the concentration c of the protein p exceeds a threshold equal to* 0.2, *then the state of g move to false at time* $t+1$. We treat for example the

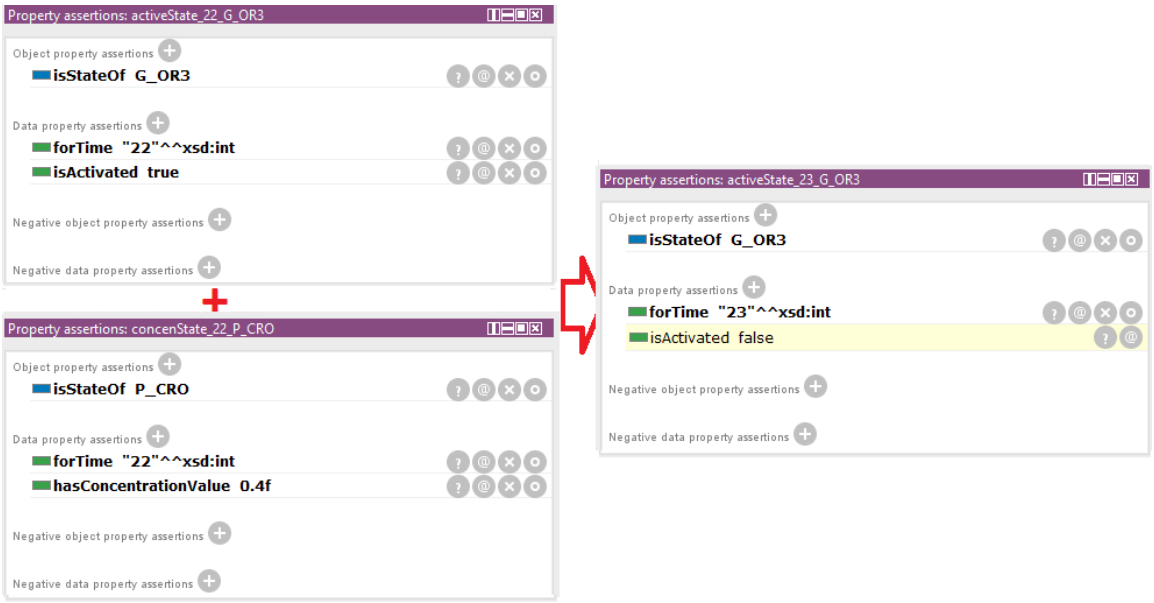


Figure 10.7 – Results of the reasoning process for the Inhibition SWRL rule between the proteins and their targeted genes.

case of the inhibition that occurs between the protein P_CRO and the gene G_OR3. As illustrated in Figure 10.7, we firstly, activate the gene G_OR3 and set the value of the concentration of the protein P_CRO at 0.4 10^{-6} Mol/L^{-1} at time $t = 22$. Then, when we launch the reasoner, we note that at time $t = 23$ the gene G_OR3 is automatically deactivated (because the value of the concentration of the protein P_CRO exceeds the threshold 0.2).

Inhibition SWRL rule between genes (interactions i_3 and i_7) The following rule models the *inhibition* interaction between the genes G_OR1 and G_CRO, and between G_OR3 and G_CI. When the concentration of the a gene (the staring node) is deactivated, it inhibits its targeted gene (deactivates the targeted node). This s the case of the two interactions i_3 and i_7.

$$ADN(?g) \wedge hasState(?g,\ ?gs) \wedge forTime(?gs,\ ?t) \wedge isActivated(?gs,\ false) \wedge swrlb{:}add(?t2,\ ?t,\ "1"8sd{:}int) \wedge ADN(?g2) \wedge InhibitionGG(?in) \wedge hasSource(?in,\ ?g) \wedge hasEnd(?in,\ ?g2) \wedge hasState(?g2,\ ?g1s) \wedge forTime(?g1s,\ ?t) \wedge isActivated(?g1s,\ true) \wedge hasState(?g2,\ ?g2s) \wedge forTime(?g2s,\ ?t2) \rightarrow isActivated(?g2s,\ false)$$

The results of this rule means that, *If there is a gene g having a state gs equal to true at a given time t and there is another gene g*1 *equal to false at this time t, and these two molecules g and g*1 *are related by an Inhibition interaction, then the state of the targeted gene g move to false at time* $t+1$. For example, we treat the case of the interaction i_7 that occurs between the genes G_OR3 and G_CI. As depicted in Figure 10.8, we firstly, activate the gene G_CI and deactivate the gene G_OR3 at time $t = 25$. Then, when we launch the reasoner, we note that at time $t = 26$ the gene G_CI is automatically deactivated.

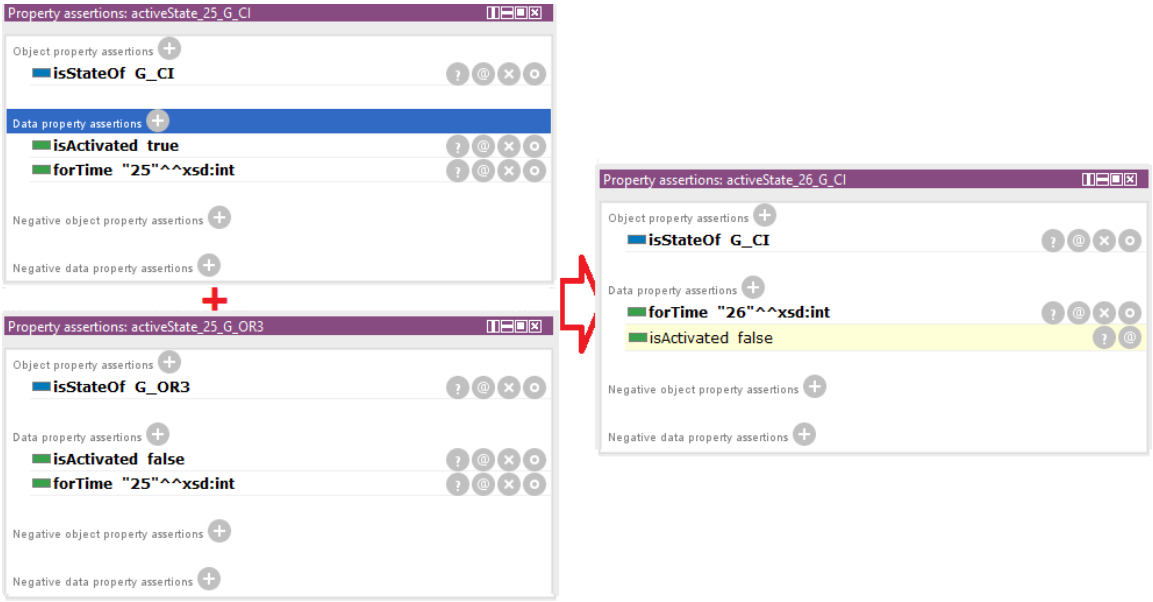


Figure 10.8 – Results of the reasoning process for the Inhibition SWRL rule between genes.

Transcription SWRL rule (interactions i_1 and i_5) The following rule represents the gene *transcription*. In our case study, we have two transcription interactions i_1 and i_5. In fact, if the genes G_CI and G_CRO are activated, these ones generates the proteins synthesis and produce an increase in the concentration of these proteins P_CI and P_CRO.

$$ADN(?g) \wedge hasState(?g,\ ?gs) \wedge forTime(?gs,\ ?t) \wedge isActivated(?gs,\ true) \wedge Protein(?p) \wedge Transcription(?trans) \wedge hasSource(?trans,\ ?g) \wedge hasEnd(?trans,\ ?p) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \rightarrow hasConcentrationValue(?ps2,\ ?c2)$$

The result of this rule is interpreted as follows, *If there is a gene g having a state gs equal to true at a given time t and there is a protein p having a state ps*1 *and a concentration c at this time t, and these two molecules g and p are related by a Transcription interaction, then the concentration of the protein p increases at time* $t + 1$. In Figure 10.9, we treat the transcription of the interaction i_1 that occurs between the gene G_CI and the protein P_CI. Firstly, we set the value of the concentration of the protein P_CI at 0.6 10^{-6} Mol/L^{-1} and activate the gene G_CI at time $t = 30$. Then, when we launch the reasoner, we note that the value of the concentration of the protein P_CI increases automatically to 0.7 10^{-6} Mol/L^{-1} at time $t = 31$.

In the opposite case, the case corresponding to the interactions i_4 and i_8 showing an inhibition interaction between the genes G_CI and G_CRO the proteins P_CI and P_CRO, respectively. We have this rule in which there is not transcription and the concentration of the protein is maintained stable:

$$ADN(?g) \wedge hasState(?g,\ ?gs) \wedge forTime(?gs,\ ?t) \wedge isActivated(?gs,\ false) \wedge Protein(?p) \wedge Transcription(?trans) \wedge hasSource(?trans,\ ?g) \wedge hasEnd(?trans,\ ?p) \wedge hasState(?p,\ ?ps1) \wedge forTime(?ps1,\ ?t) \wedge hasConcentrationValue(?ps1,\ ?c1) \wedge hasState(?p,\ ?ps2) \wedge forTime(?ps2,\ ?t2) \wedge swrlb{:}add(?t2,\ ?t,\ 1) \rightarrow hasConcentrationValue(?ps2,\ ?c1)$$

The result of this rule means that, *If there is a gene g having a state gs equal to false at a given time t and there is a protein p having a state ps*1 *and a concentration c at this time t, and these two molecules g and p are related by a Transcription interaction, then the concentration of the protein p remains stable at time* $t + 1$.

10.5.2.4 Simulation under the CBNSimulator

The CBNSimulator also provides a powerful simulator based on both (*i*) the qualitative, discrete-event simulation theory and (*ii*) the merging of the logic-based and semantic modelling. This simulation module

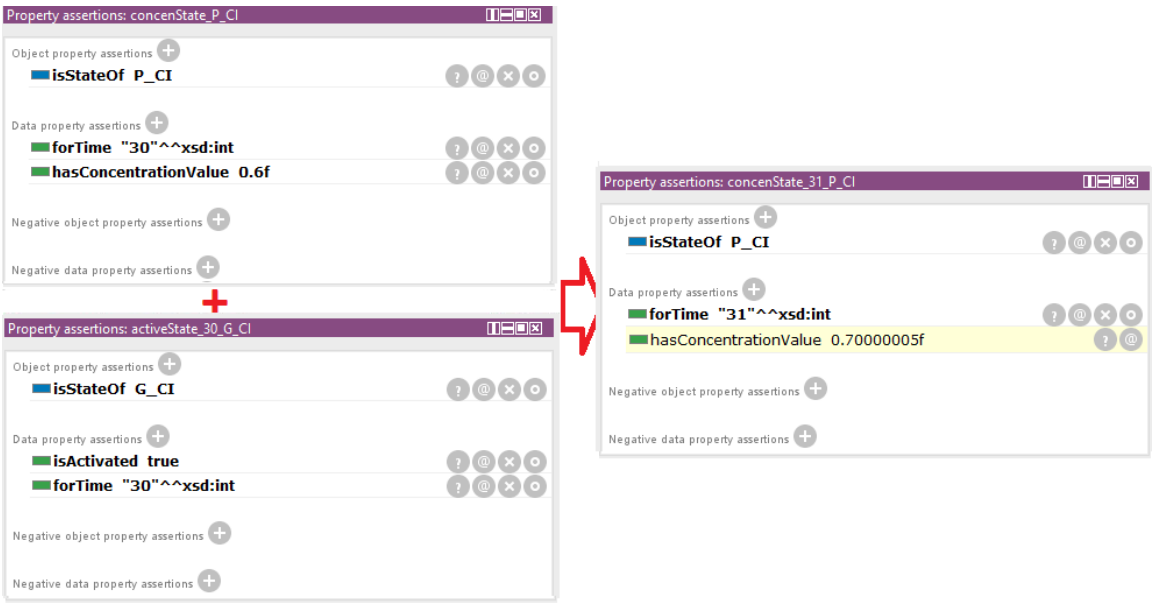


Figure 10.9 – Results of the reasoning process for the Transcription SWRL rule.

allows reproducing the behaviours of the biomolecular network and its different components. We use this simulator to simulate the behaviour of the phage lambda. In general, the simulator starts by setting the initial states and concentration values of the network components, and by defining the aggregation rules from the logic-based and the SWRL rules from the semantic modelling. These rules determine whether or not an interaction should be performing during the simulation process. After performing these interactions, the states and concentration values of the different molecular components are updated according to the corresponding interaction. Finally, the simulation results are displayed in a graphical form to the biologist.

Figure 10.10 and 10.11 depict the simulation of the two lifecycles of the phage lambda: the lysogenic cycle and the lytic cycle. In the first cycle (Figure 10.10), the molecular components of the phage lambda are fixed to *Activated*, 0.0, *Activated*, *Activated*, *Activated*, 0.0 corresponding to G_CI, P_CI, G_OR1, G_OR3, P_CRO, P_CRO, respectively. Then, when we launch the simulation we have a production of the protein P_CRO the red curve in the third display screen of the simulator (this is explained by the transcription interaction). When the concentration of this protein P_CRO reaches the threshold 0.2, it inhibits the gene G_OR3. In fact, in the second display screen, we note at $t = 4$ an inhibition of the gene G_OR3 that switch from the activated state to the deactivated state (represented by the red curve). In turn, the deactivation of the gene G_OR3 induces the deactivation of the gene G_CI at time $t = 6$ as represented by the yellow surface in the first display screen. The deactivation of this gene G_CI generates the degradation of its protein P_CI represented by the red curve in the first display screen, that is why we observe a decreasing of the concentration of the protein directly after the deactivation of its coding gene.

In the second cycle (Figure 10.11), the molecular components are fixed to *Activated*, 0.0, *Activated*, *Activated*, *Activated*, 0.0 corresponding to G_CI, P_CI, G_OR1, G_OR3, P_CRO, P_CRO, respectively. Then, when we launch the simulation, we have a production of the protein P_CI the red curve in the first display screen of the simulator (this is explained by the transcription interaction). When the concentration of this protein P_CI reaches the threshold 0.1, it inhibits the gene G_OR1. In fact, in the second display screen, we note at $t = 12$ an inhibition of the gene G_OR1 that switch from the activated state to the deactivated state (represented by the yellow surface). In turn, the deactivation of the gene G_OR1 induces the deactivation of the gene G_CRO at time $t = 13$ as represented by the yellow surface in the third display screen. The deactivation of this gene G_CRO generates the degradation of its protein P_CRO represented by the red curve in the third display screen, that is why we observe a decreasing of the concentration of the protein directly after the deactivation of its coding gene.

As shown in Figures 10.10 and 10.11, the CBNSimulator is able to reproduce the different states and

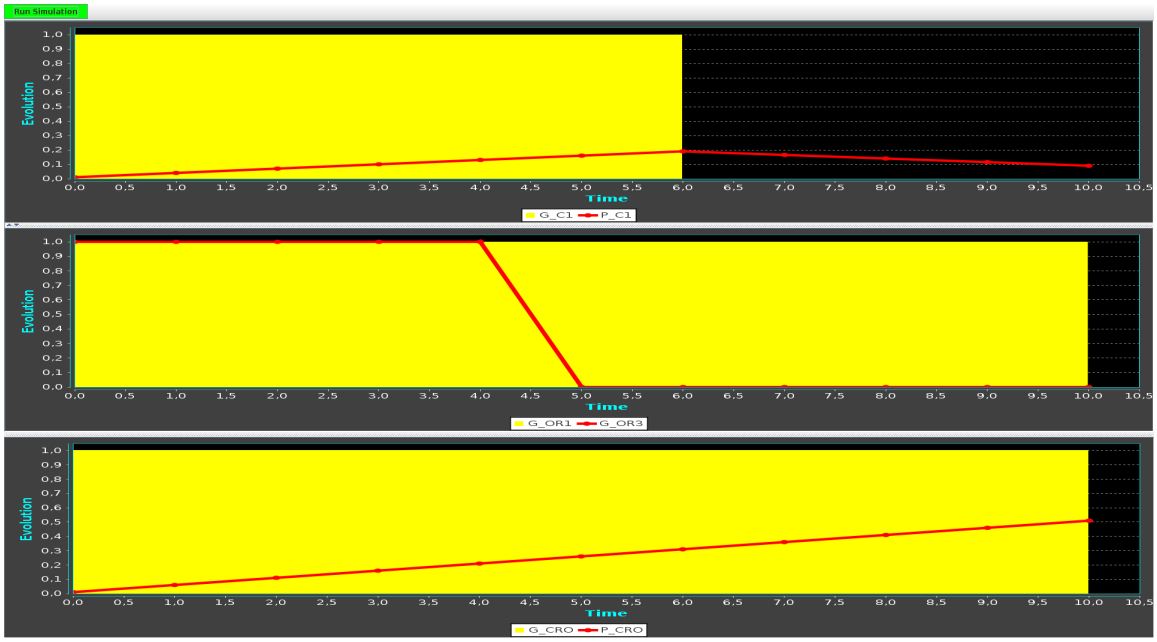


Figure 10.10 – The CBNSimulator's graphical interface. Evolution of the component's behaviour during the lysogenic cycle of the phage lambda.

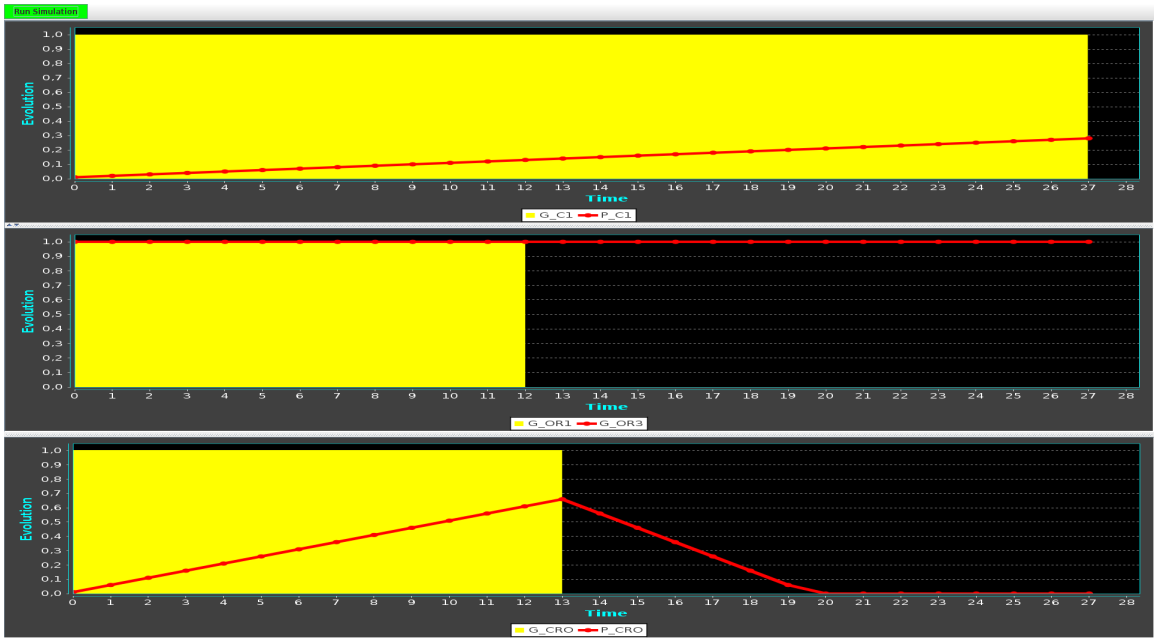


Figure 10.11 – The CBNSimulator's graphical interface. Evolution of the component's behaviour during the lytic cycle of the phage lambda.

behaviours of the phage lambda and its components over time. Indeed, the obtained results confirm that the simulation performed by the CBNSimulator is consistent and in accordance with the simulation results reported by researches [350, 354, 355] who developed this biomolecular network using other methods. Moreover, the simulation results are displayed in a graphic form facilitating the analysis of the results.

10.5.3 Case study 3: the p53-mediated DNA damage response network

We use this case study in order to see how the CBNSimulator is able to simulate and optimize the transition states and behaviours of complex biomolecular networks. This case study was already treated in the literature by Wu et al. [6] and Zhang et al. [356], therefore, we can compare our results with them. In the following sections, we will only focus on the simulation and the optimization of the p53-mediated DNA damage response network.

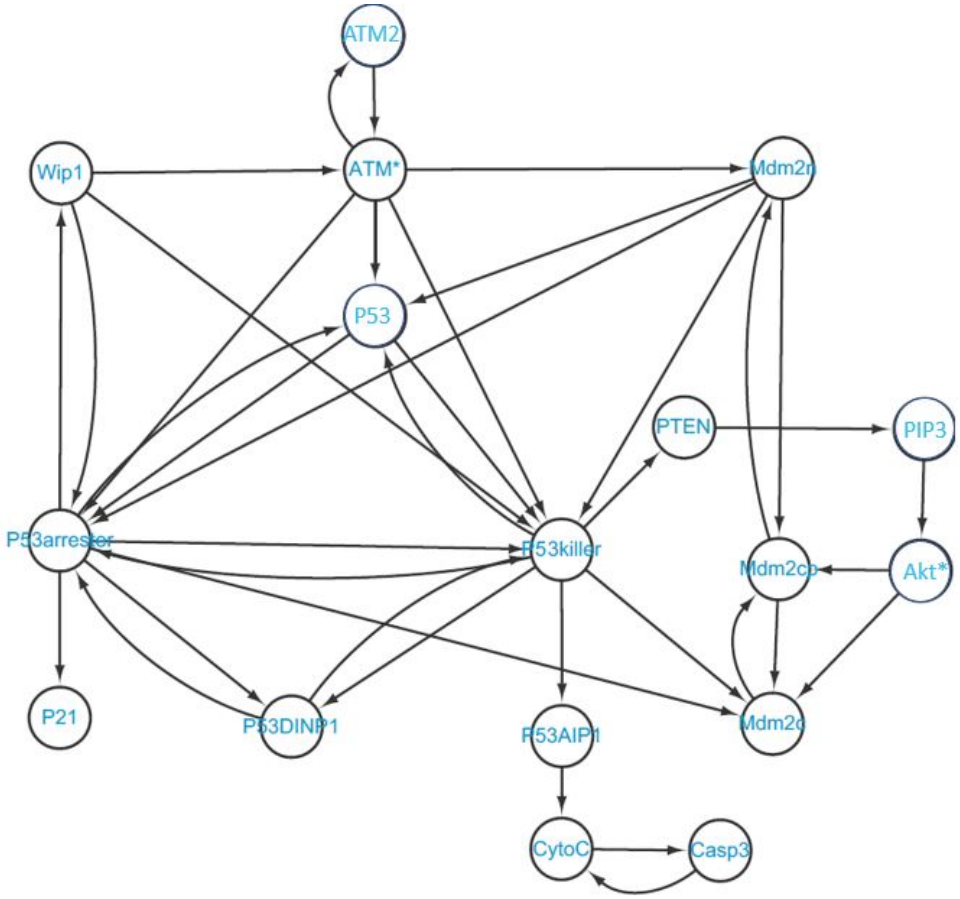


Figure 10.12 – The p53-mediated DNA damage response network [6].

10.5.3.1 Description

The tumor protein p53 is a key mediator of cellular response to diverse stresses [356] (external stimuli) such as *ultraviolet* (UV) or *infrared radiation* (IR) which can damage DNA in the form of DNA strand breaks. In response to DNA damage, this network can stay at three states (called also phenotypes): the normal, cell cycle arrest and apoptosis states [356]. This system is known as the *p53-mediated DNA damage response network* [6]. This biomolecular network is composed of seventeen molecular components and forty interactions. The seventeen nodes are: $ATM2$, ATM^*, $PTEN$, $p53$, $p53^*$, $p53_{killer}$, $p53_{arrester}$, $Mdm2$, $CytoC$, $casp3$, $p21$, $Wip1$, $P53DINP1$, $P53AIP1$, $CytoC$, Akt^*, and $PIP3$ which constitute the set of nodes M. The schematic description of this network is illustrated in Figure 10.12 [6].

This network has three states:

The normal state: If there are no external stimuli (DNA damages) the network remains at the normal state. In this state, the $ATM2$ and ATM^* are deactivated. As a consequence, the $p53$ remains deactivated, and there is no product.

In the presence of external stimuli the network can steer to two different states depending on the number or the dose of external stimuli:

The cell cycle arrest state: This state is the halt of the cell cycle progression in the case of unfavourable conditions, stress or DNA damage. In this state, the $ATM2$ is activated through the stimuli, and produces also the ATM^*. In turn, the ATM^* activates the $p53$. At this stage, the $p53$ produces the $p53^*$ in order to activate the $p53_{arrester}$. When the $p53_{arrester}$ is produced, this one activates the $p21$. It is this module, the $p21$ responsible for the cell cycle arrest. Thus, when we obtain a high concentration of $p21$ we conclude that we are in the cell arrest state.
The apoptosis state: This state is the process of the death of cells which occurs for maintaining the health of the body by eliminating old cells. If the external stimuli still applied to the network, the $ATM2$ still produces the ATM^*, which still activates the $p53$. As well as, the $p53$ still produces the $p53^*$ but in order to activate the $p53_{killer}$. Indeed, when the $p53^*$ is produced for a long time, most of it form the $p53_{killer}$. And once the $p53_{killer}$ is produced, it activates the $p53AIP1$ which in turn activates the $Casp3$. Finally, the presence of $Casp3$ provides the cell apoptosis because when the $Casp3$ is present it activates also the $PTEN$ that fully activates the $p53$. Thus, when we obtain a high concentration of $Casp3$ we conclude that we are in the cell apoptosis state.

10.5.3.2 Simulation under the CBNSimulator

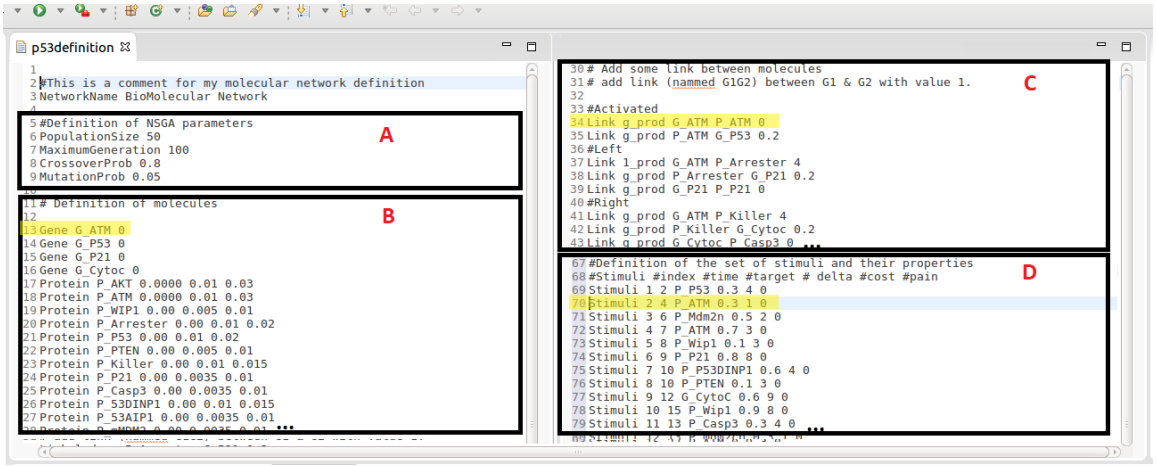


Figure 10.13 – The CBNSimulator's input file. Definition of the necessary elements describing the simulation parameters of the p53-mediated DNA damage response network.

The p53 network has been simulated within the CBNSimulator under different doses of external stimuli (IR). This simulator starts by reading the network data from the defined text file. Figure 10.13 shows the different part of the input file to the CBNSimulator. In this figure, we highlight the important elements required to simulate and optimise our case study. For example, the black frame **A** presents an excerpt of our genetic algorithm parameters such as the population size is fixed at 50, the maximum generation is fixed at 100, the crossover rate which is set at 0.8, and the mutation rate which is set at 0.05. The black frame **B** lists the different nodes of the p53-mediated DNA damage response network, their type and their state: for example, the highlighted line in this frame means that we define a node called G_ATM which is a *gene* and its state is *Deactivated* (0). The black frame **C** lists the interactions that occur among the nodes. For example, the highlight line in this frame means that there is an interaction between the gene G_ATM (starting node of the interaction) and the protein P_ATM (targeted node of the interaction) and their will we a production of P_ATM when the gene G_ATM is activated (its state is greater than 0). The black frame **D** lists the different stimuli designed to stimulate the nodes of our network. The complete list of these stimuli is presented in Table 10.2. For example, the highlighted line in this frame means that the stimulus number 2 will affect the node P_ATM at time $t = 4$ with a variation of concentration $\Delta_c = 0.3$, the cost of this stimulus is set to 1 and there is no discomfort feeling by the patient during this stimulation. Since the number of nodes is large (seventeen nodes), we display only the most important of them: ATM, $p53_{arrester}$, $p53_{killer}$, $p21$, $Casp3$, $PTEN$, and $Wip1$. The normal state of the p53-mediated DNA damage response network is illustrated in Figure 10.14. In this Figure, all the

components of the network are deactivated (proteins values setting at 0.0 and gene states *Deactivated*). This normal state corresponds to the state when there are no stimuli and the ATM is not affected, by consequence, is not activated.

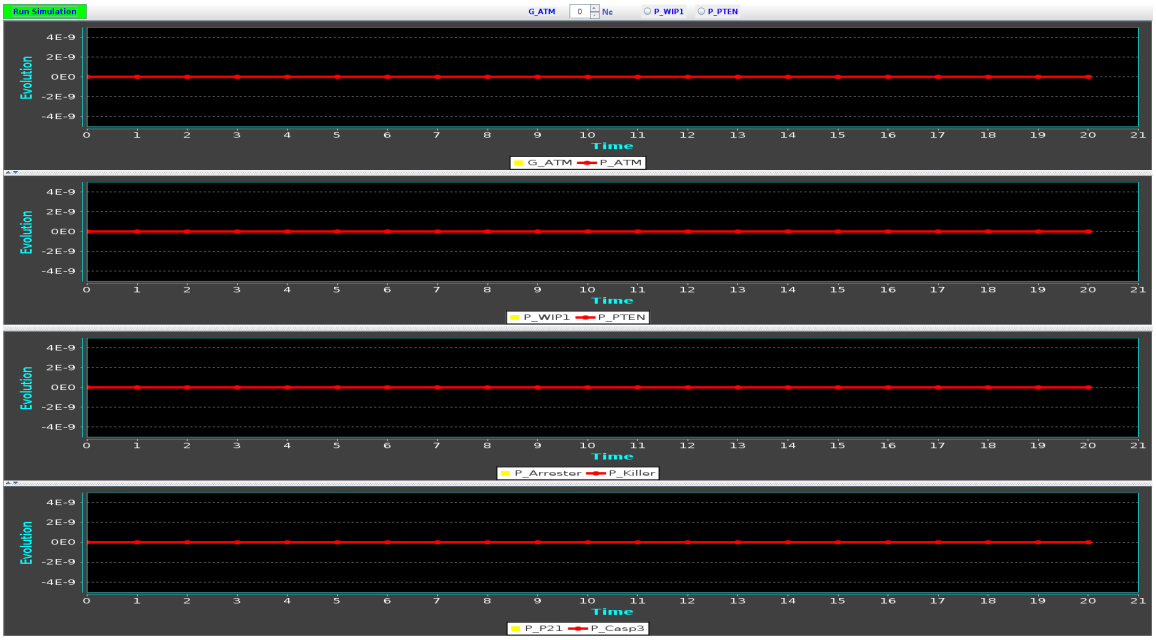


Figure 10.14 – The CBNSimulator's graphical interface. The p53-mediated DNA damage response network at the normal state.

To validate our simulator, we chose to repeat the same experiments of Zhang et al. [356] by simulating the p53 network with two different doses of infrared radiation: one less than 4Gy and the second greater than 4Gy. The simulation results are shown in Figures 10.15 and 10.16.

As illustrated in Figure 10.15, we simulate the p53-mediated DNA damage response network by affecting the ATM with three stimuli. These stimuli consisting on IR at doses less than 4Gy. As depicted in the figure 10.15, we note that at $t = 0$, the ATM is activated by the stimuli inducing the production of the protein ATM, they are displayed in the first display screen by the yellow surface and the red curve, respectively. In turn, the ATM activates the $P53$ which produces the $p53_{arrester}$. This is displayed in the third display screen, at $t = 3$ there is a production of the $p53_{arrester}$ represented by the yellow surface. In the same screen, we note that there is no production of $p53_{killer}$. Then, when the $p53_{arrester}$ is produced, it induces a production of the protein $p21$ at $t = 24$ represented by the yellow surface in the fourth display screen. We observe also that there is no production of $Casp3$ because there is no production of $p53_{killer}$. Here, we demonstrated through the CBNSimulator that we are in the cell cycle arrest state. We note also that the $Wip1$, $PTEN$ and $P53DINP1$ are not activated in this simulation (the second display screen). So, to conclude, when we affect the p53 network with three stimuli less than 4Gy, it steers to the cell cycle arrest state. These simulation results correspond to the results obtained by Zhang et al. [356].

In Figure 10.16, we increase the number of external stimuli (IR) precisely with 5 stimuli at doses greater than 4Gy. As illustrated by the figure 10.16, we note that at $t = 0$, the ATM is activated by the stimuli inducing the production of the protein ATM, they are displayed in the first display screen by the yellow surface and the red curve, respectively. In turn, the ATM activates the $P53$ which produces the $p53_{killer}$. This is displayed in the third display screen, at $t = 2$ there is a high production of the $p53_{killer}$ represented by the red curve. In the same screen, we note that there is no production of $p53_{arrester}$. Then, when the $p53_{killer}$ is produced, it induces a production of the protein $Casp3$ at $t = 23$ represented by the red curve in the fourth display screen. We observe also that there is no production of $p21$ because there is no production of $p53_{arrester}$. Here, we demonstrated through the CBNSimulator that we are in the apoptosis state. We note also that the $Wip1$, $PTEN$ and $P53DINP1$ are not activated in this simulation (the second display screen). So, to conclude, when we affect the p53 network with five stimuli

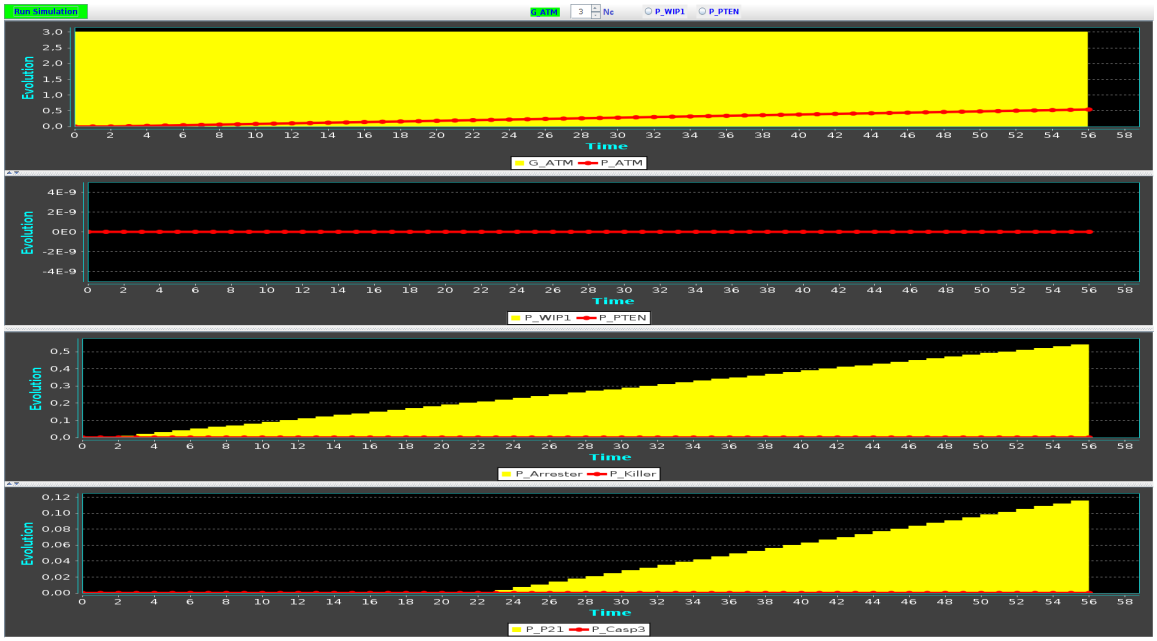


Figure 10.15 – The CBNSimulator's graphical interface. Steering the p53-mediated DNA damage response network from the normal state to the cell cycle arrest state (using three stimuli less than 3 Gy).

greater than 4Gy, it steers to the apoptosis state. These simulation results also correspond to the results obtained by Zhang et al. [356].

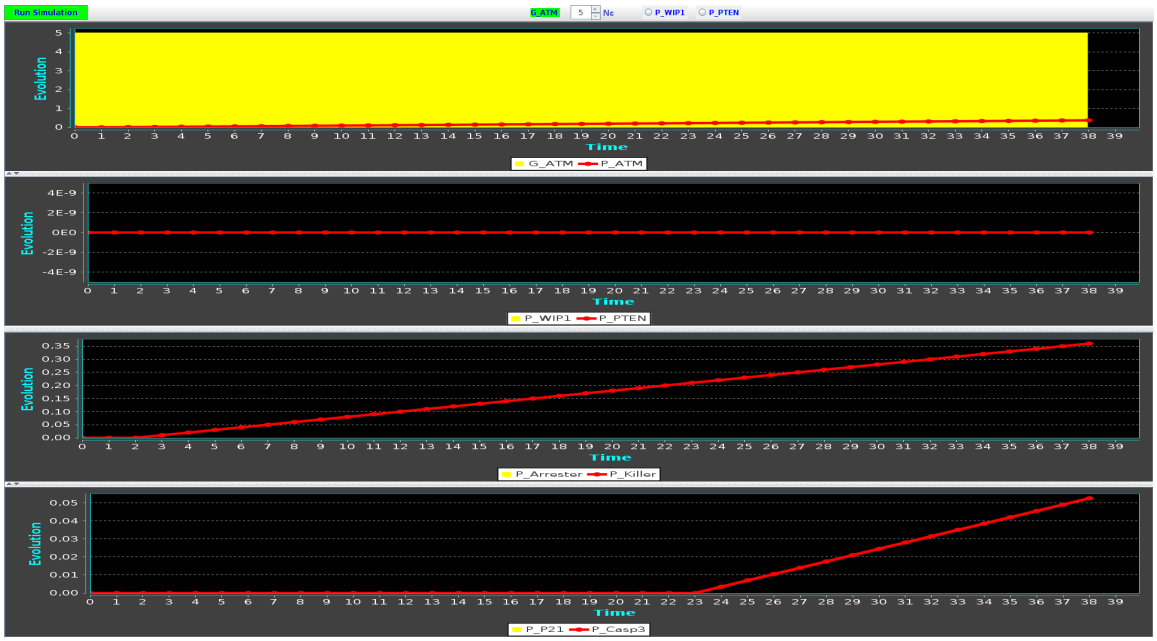


Figure 10.16 – The CBNSimulator's graphical interface. Steering the p53-mediated DNA damage response network from the normal state to the apoptosis state (using 5 stimuli greater than 4 Gy).

In addition to the two simulations presented above and with the goal to further check the logic of the CBNSimulator, we launch the first simulation using three stimuli, then we added progressively two others stimuli. This experiment is depicted by Figure 10.17. Indeed, we start the simulation with the same conditions of the first experiment presented above. As presented in the first display screen of Figure

10.17: at $t = 0$, we affect the p53 network with three stimuli, then at $t = 63$ we add the fourth stimulus and at $t = 68$ we add the fifth stimulus. We observe in the third display screen that at $t = 68$ there has been a change: a degradation of $p53_{arrester}$ and a production of $p53_{killer}$ represented by the yellow surface and the red curve, respectively. This generates another change at the level of $p21$ and $Casp3$, indeed we observe in the fourth display screen that at $t = 88$ the protein $p21$ declines and the protein $Casp3$ increases (represented by the yellow surface and the red curve, respectively). This is explained by the fact that the stable activation of the ATM (for a long time or a high dose of IR) by these stimuli activates constantly the $p53$ and $p53^*$ creating the $p53_{killer}$ rather than the $p53_{arrester}$ which in turn provides the protein $Casp3$ and also activates the $PTEN$ to maintain the full activation of the ATM for a certain time. Therefore, we conclude that when there are high doses of IR, the p53 network transits to the apoptosis state.

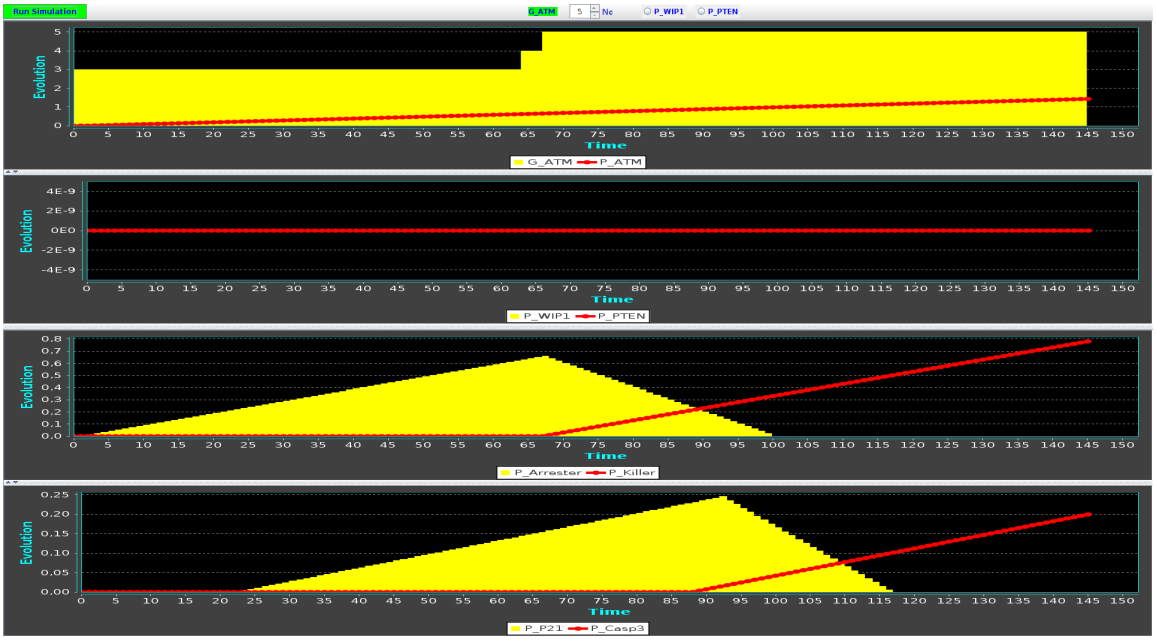


Figure 10.17 – The CBNSimulator's graphical interface. The transition of the p53-mediated DNA damage response network from the cell cycle arrest state to the apoptosis state (by progressively adding stimuli).

The correctness of the obtained results has been validated by matching and comparing our results with the experimental results reported in the literature. The obtained simulation results of the p53 transition states are in great agreement with those of Zhang et al. [356]. In fact, our simulation results for the cell fate decision via the p53-mediated DNA damage response network confirm that at the low IR doses (less than 4Gy) there is a production of $p21$ and the network transit to the cell cycle arrest state, and when the IR doses are greater than 4Gy, there is a production of $Casp3$ and the network transit to the apoptosis state. Based on these simulation results, we can note that minor DNA damage only induces the cell cycle arrest and severe DNA damage induces the apoptosis. Through these simulations, we notice that the CBNSimulator can successfully reproduce and optimize the transition states of the p53-mediated DNA damage response network, in particular, by explaining how the p53 coordinates and regulates the cell fate decision. Obtained results are very nearly with the results provided by Zhang et al. [356].

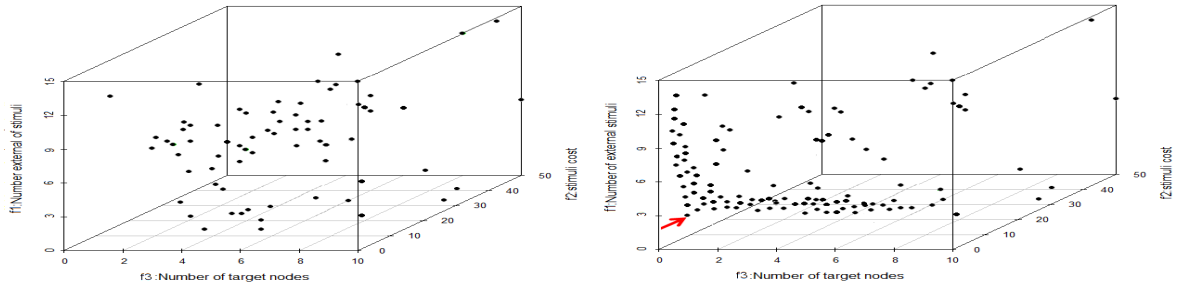
10.5.3.3 Optimization of the p53-mediated DNA damage response network

To optimize the steering of the p53-mediated DNA damage response network, we present its mathematical model as follows. Let S be the set of 15 external stimuli that represent the infrared radiation to be applied on the network during the transittability process. All the properties about these stimuli are shown in Table 10.2: the index of the stimuli S_i, the time of introduction of the stimulus S_i into the node m_i, the target node m_i by the stimulus S_i, the variation of concentration caused by the stimulus S_i on the node m_i, the cost of the stimulus S_i, and the discomfort caused by the stimulus $DiscomfortStim_k$. The

molecular components to be targeted are randomly chosen among all the nodes composing the network (the set M as listed above). Our objective is first to steer this network from its normal state to the apoptosis state by minimizing the number of external stimuli, their cost and the number of nodes to be stimulated. It must be noted that for this case study, we do not consider the patient discomfort. Then, our second goal is to steer this same network from its normal state to the arrest state by minimizing the same objectives.

To solve this case study, we apply the NGSA-II as detailed in Section 9.4.1.3. The population size is $Np = 50$. The number of generation is $MaxGen = 100$. The crossover rate is $P_c = 0.8$, and the mutation rate is $P_m = 0.05$. We suppose that the decision-maker preferences are equal (because of the simplicity and the small size of our biomolecular network). The simulation results were performed on a personal computer Core i5 with a speed of 3.20 GHz $\times$ 4 and 15.5GB RAM running Ubuntu 16.04 LTS.

To get an idea of how the Pareto front looks like, and how the network components are evolving in a visual way, we choose the 3D and 2D plot tool of R software[6] (The R Foundation for Statistical Computing), version 3.5.0 alpha (2018-03-25 r74463) to visualize them as shown in Figures 10.18 and 10.21. Therefore, the obtained results of the optimization approach are shown in Figure 10.18 as a three-dimensional (3D) chart. Figure 10.18a depicts the solution obtained in the first generation. We note that the distribution of the initial population is not uniform and it is difficult to provide good individuals. This can be explained by the fact that the first population was generated randomly. Figure 10.18b depicts the solution obtained in the last generation highlighting the trade-offs between the number of external stimuli, their costs, and the number of target nodes objectives to reach the apoptosis state. For all the objectives the ideal solution is the minimum value. Consequently, the optimal trade-off satisfying all three objectives is indicated by the red arrow in the second figure.



(a) (b)

Figure 10.18 – Trade-offs between the number of external stimuli, their costs, and the number of targeted nodes objectives for the given example. 10.18a Obtained results in the first generation. 10.18b Obtained results in the last generation.

The best compromises solution is obtained after the search and decision-maker methods. Among all the stimuli presented in Table 10.2, we can only treat the network with two stimuli to steer it from the normal state to the apoptosis state or to the cell cycle arrest state. The number of nodes to be stimulated for each state is two nodes corresponding to the nodes $PTEN$ and $P53DINP1$ for the apoptosis state, and the nodes $Wip1$ and $P53DINP1$ for the cell cycle arrest state. Indeed, we have integrated these results provided by the optimization algorithm with the simulator module. The simulation results are shown in Figures 10.19 and 10.20.

In the first simulation (Figure 10.19) we do not activate the ATM, but we directly stimulate the nodes $Wip1$ and $P53DINP1$. We observe that there is nothing in the first display screen because the ATM is not activated. In the second display screen, we note a production of the $Wip1$ at t=2 (yellow surface in the second display screen). This production of $Wip1$ induces at $t = 23$ a production of $p53_{arrester}$ as presented in the third display screen by the yellow surface. Then, at $t = 44$ we observe a production of the protein $p21$ as presented in the fourth display screen by the yellow surface. Thus, we confirm the results of the optimization approach: the stimulation of both $Wip1$ and $P53DINP1$ provides a production of $p53_{arrester}$ and $p21$, and the p53 network steers to the cell cycle arrest state.

[6] https://www.r-project.org/

Figure 10.19 – The CBNSimulator's graphical interface. Steering the p53-mediated DNA damage response network from the normal state to the cell cycle arrest state (with IR dose greater than 4 Gy).

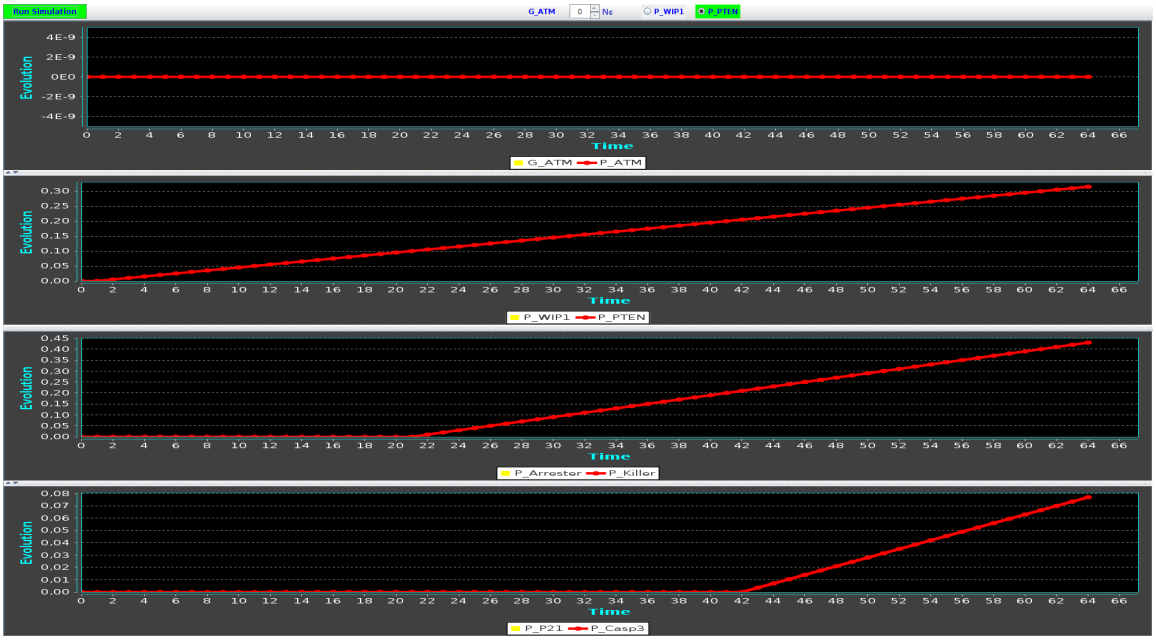


Figure 10.20 – The CBNSimulator's graphical interface. Steering the p53-mediated DNA damage response network from the normal state to the cell cycle arrest state (with IR dose greater than 4 Gy).

In the second simulation (Figure 10.20) we do not activate the ATM, but we directly stimulate the nodes $PTEN$ and $P53DINP1$. We observe that there is nothing in the first display screen because the ATM is not activated. In the second display screen, we note a production of the $PTEN$ at t=0. This production of $PTEN$ induces at $t = 22$ a production of $p53_{killer}$ as presented in the third display screen by the red curve. Then, at $t = 42$ we observe a production of the protein $Casp3$ as presented in the fourth display screen by the red curve. Thus, we confirm the results of the optimization approach: the

stimulation of both $PTEN$ and $P53DINP1$ provides a production of $p53_{killer}$ and $Casp3$, and the p53 network steers to the apoptosis state. We conclude that the stimulation of $PTEN$ and $P53DINP1$ are sufficient to induce the apoptosis state, and the stimulation of $Wip1$ and $P53DINP1$ are sufficient to induce the cell cycle arrest state.

Moreover, we visualize the response of each component of the network during the optimization process and their evolution are shown in Figure 10.21. As discussed above, the evolution of these components is displayed using the 2D plot tool of R software[7], version 3.5.0 alpha (2018-03-25 r74463). Indeed, we can observe the different states of the important nodes under three and five stimuli. These results correspond also to the simulation results provided above by the simulation approach using different IR doses.

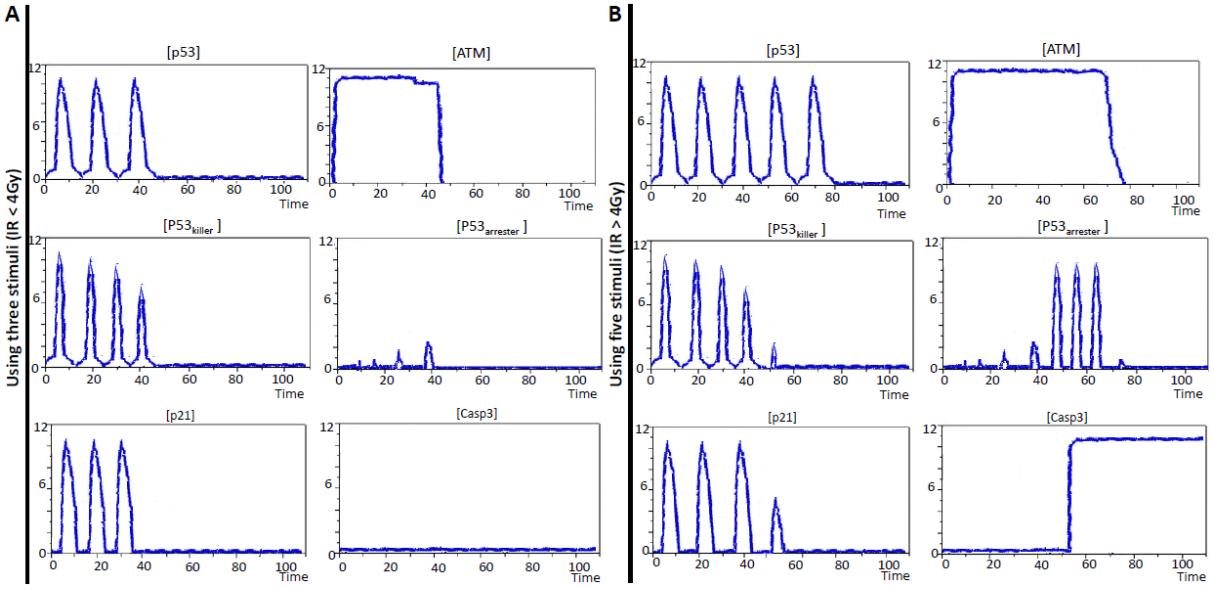


Figure 10.21 – The simulation results showing the response of the p53 system to three stimuli versus its response to five stimuli.

Our collaborators have qualitatively analysed and evaluated the obtained results in view of validating their significance. The obtained results demonstrate that the proposed optimization method provides a good quality of solutions minimizing the number of external stimuli, their cost and the number of targeted nodes. The obtained Pareto solution satisfies all the objectives.

In addition, our obtained optimization results correspond to Wu et al. [6] results. Our results propose the stimulation of the nodes $Wip1$ and $P53DINP1$ for the transition between the normal state and the cell cycle arrest state, and the stimulation of the nodes $PTEN$ and $P53DINP1$ for the transition between the normal state and the apoptosis state. However, it is important to mention that our optimization approach cannot manage the transition of the p53 network from cell cycle arrest state to the apoptosis state and vice versa. This case was usefully treated by the simulator through the progressive addition of stimuli, but the optimization module has not proposed a set of nodes to be affected for steering the network from cell cycle arrest state to the apoptosis state, and vice versa.

To conclude, the CBNSimulator platform can be considered as a powerful tool for modelling, simulating, analysing and optimizing the behaviour of complex biomolecular networks. It was checked and applied to three different networks. Obtained results agree with the literature results, in particular for the simulation and the optimization modules of this platform. However, it is important to mention the fact that this proposed platform was only tested with small examples. Results for larger biomolecular networks will be provided in the short future. Moreover, the time level of the different biological processes is not taken into account, this can be considered as a limit of the CBNSimulator.

[7] https://www.r-project.org/

10.6 Summary

In this chapter, we have briefly presented the objectives, the architecture of the CBNSimulator, and the development tools used for its implementation. The rest of the chapter is devoted for testing and applying our approaches (the main modules of the CBNSimulator) to three case studies: the bacteriophage T4 gene 32, the phage lambda and the p53-mediated DNA damage response network.

The CBNSimulator has been tested on these different case studies in terms of quality of solutions found and its efficiency to determine the optimal solutions for reproducing and optimising the transittability of complex biomolecular networks. Indeed, the three case studies presented in this chapter illustrate the strengths and limits of the CBNSimulator. Obtained results from these experiments were compared with those obtained by researches in literature such as Wu et al. [6], Zhang et al. [356], etc. The correspondence and the great agreement between the results obtained by the CBNSimulator and those obtained by these researches, in the third case study, shows the effectiveness and the robustness of this platform in optimizing the transittability of complex biomolecular networks and interpreting their transition states.

Critical discussions and evaluations about these approaches and their results are presented in the next chapter.

<table>
<tr><td>Structure</td><td>Nodes: $M = \{ G\_CI , P\_CI , G\_OR3 , G\_CRO , G\_OR1 , P\_CRO \}$
$\{G\_CI , G\_OR3 , G\_CRO , G\_OR1\} \in M_G$ and $\{P\_CI , P\_CRO\} \in M_P$
Edges: $I = \{ i_1 , i_2 , i_3 , i_4, i_5 , i_6 , i_7 , i_8\}$
$\{i_2, i_6\} \in I_{PG}$ and $\{i_1, i_3, i_4, i_5, i_7, i_8,\} \in I_{GP}$
$i_1 : \quad s(i_1) = G\_CI \quad$ and $d(i_1) = P\_CI$
$i_2 : \quad s(i_2) = P\_CI \quad$ and $d(i_2) = G\_OR1$
$i_3 : \quad s(i_3) = G\_OR1 \quad$ and $d(i_3) = G\_CRO$
$i_4 : \quad s(i_4) = G\_CRO$ and $d(i_4) = P\_CRO$
$i_5 : \quad s(i_5) = G\_CRO$ and $d(i_5) = P\_CRO$
$i_6 : \quad s(i_6) = P\_CRO$ and $d(i_6) = G\_OR3$
$i_7 : \quad s(i_7) = G\_OR3 \quad$ and $d(i_7) = G\_C1$
$i_8 : \quad s(i_8) = G\_C1 \quad$ and $d(i_8) = P\_C1$</td></tr>
<tr><td>Function</td><td>Edges:
$i_1 \overset{FR}{\mapsto} (Transcription, Activated)$
$i_2 \overset{FR}{\mapsto} (Inhibition, \geq, 0.1)$
$i_3 \overset{FR}{\mapsto} (Inhibition, Deactivated)$
$i_4 \overset{FR}{\mapsto} (Inhibition, Deactivated)$
$i_5 \overset{FR}{\mapsto} (Transcription, Activated)$
$i_6 \overset{FR}{\mapsto} (Inhibition, \geq, 0.2)$
$i_7 \overset{FR}{\mapsto} (Inhibition, Deactivated)$
$i_8 \overset{FR}{\mapsto} (Inhibition, Deactivated)$</td></tr>
<tr><td>Behaviour</td><td>Aggregate functions
A_{P_CI}:
<table>
<tr><td colspan="2">Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_3</td><td>i_4</td><td>State of c_{p32}</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td><td>$\Delta_1 < 0$</td></tr>
<tr><td>Activated</td><td>Deactivated</td><td>$\Delta_2 = 0$</td></tr>
<tr><td>Activated</td><td>Activated</td><td>$\Delta_3 > 0$</td></tr>
<tr><td>Deactivated</td><td>Activated</td><td>impossible</td></tr>
</table>
A_{P_CRO}:
<table>
<tr><td colspan="2">Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_3</td><td>i_4</td><td>State of c_{p32}</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td><td>$\Delta_1 < 0$</td></tr>
<tr><td>Activated</td><td>Deactivated</td><td>$\Delta_2 = 0$</td></tr>
<tr><td>Activated</td><td>Activated</td><td>$\Delta_4 > 0$</td></tr>
<tr><td>Deactivated</td><td>Activated</td><td>impossible</td></tr>
</table>
A_{G_CRO}:
<table>
<tr><td>Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_3</td><td>State of G_CRO</td></tr>
<tr><td>Activated</td><td>Activated</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td></tr>
</table>
A_{G_CI}:
<table>
<tr><td>Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_7</td><td>State of G_CI</td></tr>
<tr><td>Activated</td><td>Activated</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td></tr>
</table>
A_{G_OR1}:
<table>
<tr><td>Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_2</td><td>State of G_OR1</td></tr>
<tr><td>Activated</td><td>Activated</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td></tr>
</table>
A_{G_OR3}:
<table>
<tr><td>Incoming edges</td><td>Evolution</td></tr>
<tr><td>i_6</td><td>State of G_OR3</td></tr>
<tr><td>Activated</td><td>Activated</td></tr>
<tr><td>Deactivated</td><td>Deactivated</td></tr>
</table>
States: see Section 10.5.2.2</td></tr>
</table>

Table 10.1 – Logical modelling of the phage lambda.

Table 10.2 – Stimuli properties used for steering the states of the p53-mediated DNA damage response network.

Stimulus	*its time of introduction*	*its targeted node*	*its variation of concentration* Δ_c	*its cost*	*its associated discomfort*
S_1	$t = 2$	$m_i = p53$	+0.3	4	0
S_2	$t = 4$	$m_i = ATM$	+0.3	1	0
S_3	$t = 4$	$m_i = P53DINP1$	+0.6	4	0
S_4	$t = 6$	$m_i = ATM$	+0.5	2	0
S_5	$t = 7$	$m_i = ATM$	+0.7	3	0
S_6	$t = 8$	$m_i = Wip1$	+0.1	3	0
S_7	$t = 9$	$m_i = p21$	+0.8	8	0
S_8	$t = 4$	$m_i = PTEN$	+0.1	3	0
S_9	$t = 12$	$m_i = CytoC$	+0.6	9	0
S_{10}	$t = 15$	$m_i = Wip1$	+0.9	13	0
S_{11}	$t = 2$	$m_i = Casp3$	+0.3	4	0
S_{12}	$t = 13$	$m_i = mdm2$	+0.3	1	0
S_{13}	$t = 15$	$m_i = P53DINP1$	+0.6	4	0
S_{14}	$t = 2$	$m_i = p53casp3$	+0.3	2	0
S_{15}	$t = 7$	$m_i = ATM$	+0.9	3	0

Chapter 11

Discussion and evaluation

Contents

11.1 Introduction

In the previous chapter, we presented the experiments that have been conducted to verify and validate our contributions, the prototype CBNSimulator, its implementation and its application. This chapter proposes a discussion which aims to place our contributions in the context of similar works and to discuss their relative advantages and disadvantages.

This chapter is divided into four sections. In the first section, we discuss the efficiency of the logic-based modelling for formalizing complex biomolecular networks considering their structural, functional and behavioural aspects. In the second section, we present an evaluation study of the semantic modelling approach to check the ontology quality of the proposed Biomolecular Network Ontology. In the third section, we discuss the performance of the proposed simulator in reproducing the behaviour and state changes of complex biomolecular networks. And finally, the fourth section analyses the usefulness of the optimization approach in optimizing the transition states of complex biomolecular networks.

11.2 Logic-based modelling discussion and evaluation

With the goal of studying and simulating the dynamics of cells, various modelling techniques have been proposed in the literature. As detailed and discussed in Chapter 2, we can find continuous methods such as stochastic and differential equations which provides good precision and are therefore powerful techniques for describing and modelling a specific problem. However, these quantitative models have also notable disadvantages such as the fact that they require complete numerical and quantitative data (parameters, concentration levels, kinetic constants, timings, etc) which are usually unavailable or not up-to-date. For example, the production of structural analysis matrices in biology may require the mobilisation many of experts over several months. Also, quantitative data can be based on an inappropriate sample, using inadequate parameters or being misinterpreted in various ways. The excessive formalisation of these models can increase the level of complexity to solve the simulation of the network. Indeed, with large-scale biomolecular networks, the quantitative model cannot be solved due to its complexity and its size. Indeed, the high number of molecular components increases the complexity of solving the differential equation system modelling them.

As discussed in Chapter 2 biologists have various modelling techniques at their disposal, each with its advantages and disadvantages, depending on the desired objectives and available resources. If all the quantitative parameters are expected, a quantitative modelling is required. However, if global properties are the main concerns, qualitative logic-based modelling is required. A major advantage of qualitative logic-based models is that they do not require precise quantitative data and that they are well-suited for simulating and analysing large-scale biomolecular networks, in particular for understanding the transittability of complex biomolecular networks.

In the following paragraphs, we discuss the advantages and limits of the proposed logic-based modelling. This discussion focuses on the important logic-based modelling features, such as the modelling of the structure, function, behaviour of complex biomolecular networks. As well as, in the importance of modelling the stimuli-induced state changes.

The structural modelling The first step of logic-based modelling is the construction and definition of the molecular elements of the complex biomolecular networks. In this structure modelling, the essential elements of the biomolecular networks are formalized into nodes which are categorized into genes, proteins and metabolites. These nodes are represented either by Boolean values and real values. In fact, logical models in systems biology were initially developed based on Boolean networks starting with the works of Thomas [357], Glass [358], Kauffman [18], etc. However, the proposed logic-based modelling is not totally based on Boolean models. Indeed, in our proposed model, the nodes of the biomolecular network are formalized by Boolean variables (the case of genes which can be activated or deactivated), but also by real variable (the case of the concentration of proteins and metabolites). Therefore, our proposed logic-based modelling is able to model both discrete entities, that is valued in $\{0; 1\}$, and continuous entities, which are valued in $\mathbb{R}$. This property is considered useful because it allows adding further clarification and details. Thus, the results provided by the proposed logic-based modelling, while remaining qualitative, can be finer than those provided by the Boolean one.

Once defined, the nodes of the network are linked with edges. These edges represent the interaction

that can occur between each couple of nodes. Indeed, the partition of the graph nodes induces a partition into a range of different types of interactions which allow linking the different levels of the biomolecular network. Thus, we conclude that the proposed logic-based modelling is a multi-level model. In fact, in a biomolecular network, interactions can occur between molecular components at the same level (for example interactions between genes) or between different levels (for example between proteins and genes) forming a complex system with multiple spatial and temporal levels. Therefore, the proposed logic-based modelling enable to model and integrate the experimental information that exists at different levels. This is a hard and impossible task using quantitative models such as differential equations or stochastic models, because of the high number of molecular components inside the biomolecular network. In addition, this property has been validated experimentally through the case studies presented in the previous chapter. All of them are composed of molecules belonging to the different level and interacting with each other.

Thus, we note that the logic-based modelling is better suited for describing large-scale biomolecular networks, where detailed and quantitative knowledge are incomplete and where the heterogenous sub-cellular components belonging to different levels of the network can be represented in a single and complete formalization.

The functional modelling In our logic-based modelling, the molecular components of the network are not described by well-defined equations as in the quantitative models, but instead, rules are assigned to the participating elements to study the properties that emerge due to the interactions of the elements, usually considered as a rule-based modelling. Therefore, we note that this logic-based modelling is suitable for describing large-scale biomolecular networks with several molecular components, and are computationally much cheaper than the quantitative and continuous models. In addition, the proposed logic-based modelling allows the classification of the different types of interactions that occur among molecular entities. In order to add more clarification about the interactions and to precise the function of each interaction, the functional modelling of the logic-based modelling is associated with the concepts of the Interaction Ontology proposed by Van Landeghem et al. [4]. Indeed, through this ontology, we provide to the logic model the necessary and precise knowledge about the interactions. This additional feature allows to precisely define the type of the interaction and the condition that activates it. This precision contributes to the implementation of a valid model. This is essential for obtaining a multi-scale modelling close to the reality and for minimizing the gaps between the proposed model and the real experiments. We can see this precision with the modelling of the case studies. The function of each interaction is precise and clearly defined.

The behavioural modelling According to the case studies presented in the previous chapter, the proposed logic-based modelling showed its efficiency in describing and modelling the dynamic evolution of the biomolecular network and in reproducing its behaviour over time. This aspect is ensured by the use of aggregation functions which compute the state transitions of the network during the simulation. Indeed, for each molecular components, we define an aggregate function which computes its evolution based on its current state, the state of its predecessor components and the characteristics of its incoming interactions.

In our logic-based modelling, the molecular components follow simple rules (generally defined by expert biologists and represented by the aggregation functions), but their interactions may induce new behaviours (phenotypes) at the level of the network. Thus, it can be used to investigate emergent behaviours in complex biomolecular networks. The advantage is that it can be used for networks in which the molecular components cannot be well defined by a precise mathematical equation but can be defined with rules. Most importantly, they are computationally much simpler than the quantitative models. The limit is that it is not accurate for quantitative modelling.

Stimuli-induced state changes Complex biomolecular networks are subject to strong variations caused by internal or external stimuli. Therefore, it is difficult to predict their behaviour because it emerges according to their states and the state of their environment. A biomolecular network exposed to stimuli may exhibit more qualitatively different behaviours than a network working without stimuli. In other words, a phase transition or a transition state can be induced by these stimuli. This aspect has been considered and studied in the logic-based modelling. Most of the mathematical models presented in Chapter 2 cannot represent and take into account these stimuli and the state changes they cause.

This is a very important feature that makes the difference from our logic-based approach to other models. As shown in the second and third case studies presented in the previous chapter. The proposed logic formalism considers and models the state changes of biomolecular networks caused by stimuli. Indeed, the proposed logic-based modelling considers two types of stimuli: internal and external stimuli. In fact, the changes of a molecular state can occur either by an internal stimulus modelled by the aggregate function discussed above or by an external stimulus generated outside the cell. Therefore, we explicitly define this notion of stimuli as an event that causes changes in the state of the molecule on which it acts and therefore changes the state of the whole network. Then, we integrate it with the aggregation functions.

This point is very important and allows the logic-based modelling to understand the response of cell in the presence of external or internal stimuli, to understand drugs mode of action (through stimuli) and to predict the behaviour of complex biomolecular networks in response to drugs. This was proved by the third case study, the p53-mediated DNA damage response network, based on this logic-based modelling the CBNSimulator has successfully simulated and managed the transition states of the network from the normal state to the apoptosis state and/or from the normal state to the arrest state. This case study highlights the fact that our logic-based modelling can be also used for modelling and understanding the relation and the mode of action between drug design and biomolecular networks modelling.

To conclude, the logic-based modelling allows biologists to produce formal models of the biomolecular network of interest and then to simulate it on computers. All the properties discussed here were validated experimentally in the previous chapter (Chapter 10). Indeed, the case studies were formalized within the logic-based modelling and simulated within the CBNSimulator. The first example network, the bacteriophage T4 gene 32 is chosen for its simplicity, it is simple enough to be mentally computed in order to easily judge the produced results. However, for the other examples, we validate their modelling by comparison with results of other works. We conclude that the logic-based modelling provides all the elements necessary for formalizing and simulating the behaviours and transition states of these biomolecular networks. This logic model provides a formal and mathematical formalism to model and simulates the complex structure of biomolecular networks, the effect of stimuli which target some molecular components, and the simulation of the dynamic evolution of their behaviours.

11.3 Ontology discussion and evaluation

Ontology evaluation and validation is a very important issue to check the ontology quality. A large group of ontology evaluation approaches exists, among those, data-driven evaluation, task-based approach, automated consistency checking, etc. [359]. Table 11.1 provides an overview of the best-known ontology evaluation approaches and their limits. However, according to the current literature [360], there is no agreement on a methodology for validation and evaluation of ontologies. The choice of a suitable approach depends on the purpose of evaluation, the application in which the ontology is to be used, and on what aspect of the ontology we are trying to validate and evaluate [361]. For all these reasons, we have chosen to evaluate the BNO ontology by following different evaluation approaches. This choice of hybrid approaches inherits many of the advantages of each of these approaches. The goal is to evaluate our ontology in a different manner. Basing on the methods presented in Table 11.1, we adopted a combination of automated consistency checking, expert knowledge evaluation, criteria-based evaluation, and task-based evaluation for evaluating the BNO ontology.

Automated consistency checking The verification of the logical axioms is an essential task in ontology evaluation. Indeed, this evaluation ensures that the logical axioms are satisfiable and consistent. This satisfaction consists in: (*i*) checking the encoding of the specification, (*ii*) detecting errors such as class hierarchies, redundant axioms, etc., and (*iii*) confirming that the BNO ontology has been built according to certain specified ontology quality criteria. By definition, consistency checking ensures that an ontology does not include any contradictory facts. For definitions to be semantically consistent, they must be able to obtain consistent conclusions using the meaning of all definitions and axioms [368, 369].

To evaluate the BNO ontology and check the inconsistencies and violations of its SWRL rules, we used the latest version of the Description Logic reasoner HermiT reasoning plugin in the Protégé 5 environment [1] version 1.3.8.3. HermiT can not only determine whether or not the ontology is consistent

[1] `http://www.hermit-reasoner.com/`

Table 11.1 – A summary of ontology validation evaluation approaches.

Evaluation methods	***Description***	***Limitations***
Task-based approach	This approach evaluates an ontology by using it in tasks and assessing the performance. It is an effective approach to assess the capability of an ontology to achieve its purposes and objectives. It is a good method to evaluate the capacity an ontology to achieve its objectives [362].	This method does not evaluate the structure of an ontology and ignores deficits in its conceptualization.
Automated consistency checking	This approach evaluates the consistency of an ontology by using Description Logic reasoner [363]. Most popular DL reasoners are Hermit, Pellet, Fact++, fuzzyDL, etc.	This method checks the internal consistency of an ontology (the content does not contain contradictory information) and ignore its background knowledge.
Gold standard checking	This approach compares an ontology to a gold standard ontology (a benchmark ontology) and measures their conceptual and lexical similarities [364].	There may be errors in the methods of comparisons and a lack of ontology in the domain of study.
Criteria-based evaluation	This approach evaluates an ontology by using a set of predefined criteria, such as clarity, consistency, accuracy, Computational efficiency, conciseness, completeness, correctness, etc. [365].	Some criteria lack quantitative measures and are frequently rely on expert judgement.
Data driven evaluation	This approach compares an ontology with a source of data about the domain that is to be covered by the ontology [366].	This approach can not evaluate the correctness and the clarity of an ontology.
Expert knowledge evaluation	This approach evaluates an ontology by using expert knowledge who try to assess how well the ontology meets a set of predefined criteria, standards, requirements, etc. [367].	This approach lacks quantitative measures.

but also identify subsumption relationships between concepts and resolution of the error. In terms of the time, HermiT is as fast as other DL reasoners when classifying relatively easy-to-process ontologies, and usually much faster when classifying more difficult ontologies. In fact, HermiT can classify a number of ontologies which no other reasoner has previously been able to handle. Using its HermiT reasoner plugin, Protégé automatically checked the inferred concepts and relations and for hierarchies, domains, ranges, and conflicting disjoint assertions. Contradictory facts and inconsistent concepts are marked with red. During the development of the BNO ontology, the automated consistency checking process was iterative. Indeed, the BNO ontology was developed incrementally by adding new definitions and modifying old ones. Moreover, the HermiT reasoner was used to check the correctness of the SWRL rules edited in the SWRLTab (as shown in Chapter 7). Their consistency was verified by the results of our experiments as shown in Figures 7.8 and 7.9.

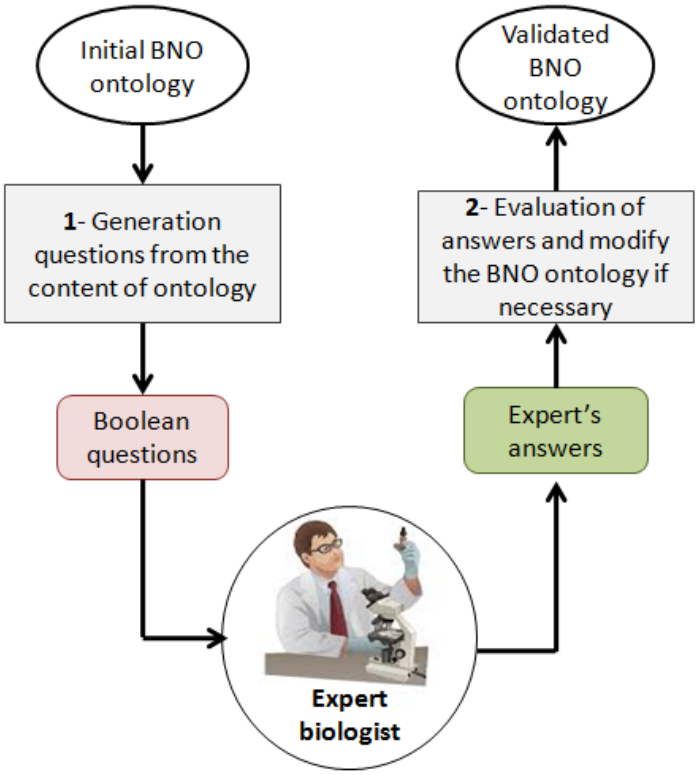


Figure 11.1 – Steps of the expert knowledge evaluation approach.

Expert knowledge evaluation Even if we have used best-known evaluation methods to test the consistency of an ontology, the intervention of domain experts is always necessary, especially if a quality level of the ontology is expected. The evaluation here focuses on the semantics of the BNO ontology content and not on its formalization. As a consequence, we proposed a method based on questions expressed in natural language and generated from the BNO ontology in order to test and, if necessary, to correct the content of the ontology using the answers that will be provided by the expert biologists. This questions and answers method facilitates the task of experts. We obtained the assistance and expertise of our collaborators from the CSTB team who have evaluated the BNO ontology and conclude that it is in accordance with their knowledge in the domain (expert knowledge).

As shown in Figure 11.1, the evaluation process consists of two steps:
Fist step - Generation of questions from the content of the BNO ontology: In this step, we generate a set of questions from the ontological elements of the BNO ontology. Table 11.2 contains examples of questions that we have defined in terms of ontology elements and their translation into boolean questions addressed to expert biologists.

Second step - Evaluation of the expert's answers: In this step, we evaluate the expert's answers in order to decide the validity of the BNO ontological elements tested or (if necessary) their modification to make them valid.

We study the correlation between the number of questions generated (in terms of the number of ontological elements to be evaluated) and the size of the BNO ontology. Table 11.3 presents the number of questions generated according to the size of the ontologies in terms of concepts, relations and individuals. The number of questions highlights the role and the intervention of the expert biologist and its knowledge to reduce the potential errors linked to the semantics of the content of the ontology.

The results obtained in Table 11.3 show that the number of questions generated depends on the number of ontological elements to be evaluated and validated.

The notion of validity in this method depends on the domain expert answering the generated boolean questions. Thus, from our point of view, the validity of the questions corresponds to the agreement between the knowledge of the domain expert and the semantic content of the ontology.

Table 11.2 – An excerpt of ontological questions and their translation into Boolean questions addressed to biologists.

Questions made by an expert in ontology	***Examples of its corresponding questions addressed to expert biologists***
Is CLASS a type of CLASS ?	Is the 'P04040 (CATA_HUMAN)' a type of 'Protein'? Is the 'G32' a type of 'Gene'?
Is INSTANCE an example of CLASS ?	Is the 'the bacteriophage T4 G32' an example of 'Biomolecular_Network'? Is the 'G32' an example of 'Gene'?
Is SUB-PROPERTY a type of PROPERTY ?	Is the 'the catalysis' is a type of 'Interaction'?

Table 11.3 – Number of questions generated according to the size of the BNO ontology.

Concepts	Properties	Individuals	Total	Questions generated
29	20	29	78	75

Criteria-based evaluation In addition to the evaluation conducted by biologists, we adopted a criteria-based evaluation method following the validation protocol proposed by Vrandecic in [365]. This validation protocol is essentially based on a number of criteria that will enable us to determine whether our ontology is relevant or not. This protocol is based on seven evaluation criteria, (1) the *accuracy*, (2) the *adaptability*,(3) the *clarity*, (4) the *completeness*, (5) the *computational efficiency*, (6) the *conciseness*, and (7) the *consistency*. We evaluate the BNO ontology against these criteria as follows.

Accuracy: The definitions and descriptions in the ontology agree with the expert's knowledge about the field. The information regarding the concepts of the BNO ontology was developed from the well-known Gene ontology (GO). Moreover, we obtained the assistance and expertise of our collaborators from the CSTB team who have evaluated the BNO ontology and conclude that it does not contain semantic mismatches, logical inconsistencies, and conceptual conflicts.

Adaptability: We have opted for developing the BNO ontology as part of a global semantic architecture composed of four ontologies that are related to each other: the Gene Ontology (GO), the Simple Event Model Ontology (SEMO), the Time Ontology (TO) and our development, the BNO ontology. This architecture aims at aligning and merging the BNO ontology with the rest of ontologies through equivalence *owl:equivalenceClass* or subclass *owl:subclassOf* relations. These relations among ontologies are detailed in Chapter 7. This choice enhances extensibility and reusability and makes the BNO ontology easily adaptable to dynamical contexts.

Clarity: In developing the BNO ontology, we have been careful to assign a clear and unambiguous description to define and categorize concepts and the relationships among concepts within our particular knowledge domain. This clarity is ensured by the use of the *rdfs:comment* that provides the obviously needed capability to annotate an ontology. In this manner, the BNO ontology communicates effectively the intended meaning of its terms.

Completeness: This criterion measures whether the ontology can answer all the questions that it should be able to answer. It provides an estimation of how the BNO ontology represents the domain of the complex biomolecular networks and their transittability. These questions were specified by the expert biologists of the CSTB team and it has been verified that all of them can be answered.

Computational efficiency: An ontology can be analysed by an inference system. In our case, the BNO ontology was treated by the two reasoning mechanisms detailed in the previous section. We concluded that the reasoning on the BNO ontology is consistent and allows inferences in a reasonable time. Moreover, the complexity of this operation is adequate.

Conciseness: The terms of the BNO ontology was checked with the help of expert biologists, we assume that the ontology does not contain any redundant terms. Moreover, we have used the OntOlogy Pitfall Scanner[2] tool to check for logical correctness of the *BNO* ontology and diagnostics of ontology-design errors. Analysis results have provided great evidence of the correctness of BNO.

Consistency: This criterion ensures that the logical axioms are satisfiable and consistent. The satisfaction of the logical axioms is recognized when it is possible to find a situation under which all the axioms are true, and their consistency when it is impossible to find a contradiction within the axioms. As detailed in the previous section, reasoning in the BNO ontology was performed using an SWRL rule-based reasonner.anisms. No inconsistencies or violations were found.

As discussed in section 11.3, there is no single best approach to evaluate an ontology. For this reason, we check the BNO ontology with different approaches. Firstly, we focus in particular on automated ontology evaluation, which is a necessary precondition for the healthy development of an ontology. Automated consistency checking was made through the Hermit reasoner. Based on the feedback of the reasoner, inconsistencies have been corrected along the development process of the BNO ontology in an iterative way. The final results made by the reasoner revealed that there are no inconsistencies in the BNO ontology. The BNO ontology has been also evaluated with different criteria in terms of accuracy, adaptability, clarity, completeness, etc. For each criterion, the BNO ontology is evaluated. The combination of these criteria allows us to check the BNO ontology from different levels. The final results of the criteria-based evaluation indicated that the BNO ontology was clear, extendable, and complete. Moreover, we evaluate the usefulness of the BNO ontology through the expert knowledge evaluation. The BNO ontology was used to model a set of case studies, among them the Bacteriophage T4 G32 case presented in this study. Results proved that the BNO ontology is able to deduce the main concepts of the case studies and their properties and is capable to infer new knowledge such as to compute the newt state of molecular components. These results proved that the BNO ontology is able to describe and model the transittability of a biomolecular network. However, it is important to note that the BNO ontology cannot be used directly through a logic reasoner to compute the transittability of large-scale networks. This would exceed the computing capabilities of current reasoners. Specific simulation tools must be designed for this task.

11.4 Simulation discussion and evaluation

The CBNSimulator is equipped with a qualitative, discrete-time simulator. This simulator allows simulating in silico the behaviour of complex biomolecular networks. It simulates and tests the different state changes of biomolecular networks under various experimental conditions. The proposed qualitative, discrete-event simulator can also help biologists to discover and detect the stimuli that regulate the biomolecular network such as drug effects to the biomolecular network.

As discussed in Chapter 4 and in the previous sections, simulation techniques can be categorized into two kinds: quantitative and qualitative techniques. Quantitative simulation provides the most precise prediction in describing the behaviour of specific molecular entities. Nevertheless, the lack of quantitative data limits its use with only specific case studies (when all the required quantitative data are available) but cannot be used in more general or large-scale networks. On the other hand, qualitative simulation simplifies the real simulation of the biomolecular network and is usually able to reproduce the network

[2] `http://oops.linkeddata.es/advanced.jsp`

behaviour. Moreover, this qualitative simulation can be used to explain and predict the behaviour and state changes of the biomolecular and its components through discrete simulations.

Because of the lack of quantitative data, various qualitative simulation techniques have been proposed in the literature (see Chapter 4) however, most of them cannot support the simulation of complex and multi-level biomolecular networks. As well as, most of them suffer from the lack of automation in simulation biomolecular properties [370].

Our proposed simulator is different from the existing techniques. Indeed, it is totally based on logic-based modelling. Therefore, the power of this simulator is essentially due to the efficient logic-based modelling of complex biomolecular networks which is expressive enough to integrate and capture the different elements and qualitative properties required to understand the dynamic behaviour and state changes of biomolecular networks. But also, in the representation and the simulation of multi-level biomolecular networks. Thus, the proposed qualitative, discrete-event simulation is able to simulate complex and multi-level biomolecular networks. This property has been verified through the different case studies in the previous chapter. All the case studies are composed of molecular components and interactions belonging to a different level within the biomolecular network.

In addition, our proposed qualitative, discrete-event simulation is able to perform automatic and efficient biomolecular networks simulation. Indeed, the simulation core rests on a discrete-event based framework (the DEVS formalism), which is used as an efficient and accurate simulation tool of complex systems at different levels of abstraction. This ensures both synchronous and asynchronous updating methods for simulation. In the absence of perturbations, the simulator sets the initial states of the molecular components and the initial time, then it updates the state of all the components at the same time. This synchronization is based on the interactions rules (aggregate functions) defined by the biologist. It was the case of both the first and second case studies in the previous chapter. However, the asynchronous simulation takes into account the different perturbations caused by the stimuli. When certain stimuli affect some molecular components, the simulator updated the state of these components then synchronizes the state of the other components. This has experimented through the third case study in the previous chapter.

Another important advantage of the proposed qualitative simulation mechanism is that it does not require precise quantitative data, but even more it can simulate the behaviour of biomolecular networks using only qualitative data. This simulation mechanism is well-suited to simulate and analyse large-scale biomolecular networks even with a lack of quantitative data. This proposed qualitative mechanism have been verified and illustrated for the example of the bacteriophage T4 gene 32, as detailed in Section 8.2.3. The simulation mechanism is based on the development of a causal graph whose nodes denote variables (representing the molecular components of the network) which this simulation is concerned and edges denote causality relations among these variables (representing the interactions among the molecular components). The state of the biomolecular network is described by a few qualitative distinct values corresponding to precise quantitative values (called the quantity space). To compute the qualitative value of the nodes, this mechanism is based on both the partition rules and propagation rules. These rules are used to compute the value of the target variable at the next time $t+1$ based on its qualitative value at the current time t and the value of its predecessors' nodes at the current time t. As detailed in Section 8.2.3, the application of the qualitative simulation to the given example shows that the proposed qualitative simulation mechanism is able to model and simulate complex biomolecular networks, predicting their different behaviours for different simulations constraints.

To conclude, the qualitative, discrete-event simulation performed by the CBNSimulator can be used to elucidate and predict the behaviour, state changes and properties of complex biomolecular networks. Moreover, the simulator displays the different states of each molecular component in a graphic form that is easy to interpret. We tested this simulator on the three case studies. The results show the performance simulation of the proposed simulator in modelling and simulating complex multi-scale biomolecular networks under different environmental conditions and by considering the perturbations caused by external stimuli.

While the proposed simulator has many advantages, it has the potential for significant further improvements. Indeed, this proposed qualitative simulation is not able to precise the different time scales. This can be considered as the main limit of our qualitative, discrete-event simulation. However, we hope to enhance our simulator by working on this point. This future direction is described in details in the perspective section of the next part. Moreover, integrating this qualitative, discrete-event simulation with another quantitative continuous simulation tool would allow to increase the performance and obtain

a semi-quantitative simulator including the advantages of both qualitative and quantitative simulations. This is one of the possible future directions for this work.

11.5 Optimization discussion and evaluation

Optimizing the transittability of complex biomolecular networks is one of the main objectives of the CBNSimulator. As discussed in 5.7.7, only a few studies have been focused on this problem and most of them considered it as a mono-objective optimization problem and neglect other criteria for steering these biomolecular networks. These researches only focus on the minimization of the number of required driver nodes for steering the network and/or on the minimization of the number of external stimuli to be applied on the network. However, even these two criteria are necessary conditions, they are not sufficient for completely steering complex biomolecular networks. Indeed, this assumption is not always realistic, because steering complex biomolecular networks are in general a multi-objective optimization problem. It requires finding appropriate trade-offs among various objectives, such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of external stimuli, the minimization of the cost of these stimuli, the minimization of the number of target nodes, and the minimization of the patient discomfort.

Therefore, in this area we firstly propose a multi-objective mathematical formulation for optimizing the transittability of complex biomolecular networks in which we take into account more criteria such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of stimuli, the minimization of the cost of these stimuli, the minimization of the number of target nodes, and the minimization of the patient discomfort. Indeed, these five objectives are the fundamental pillars for successfully steering the state of biomolecular networks. The first objective which is the most important is the minimization of the distance between the simulated final network state and the desired network state. This objective is ensured by the simulator and consists to compute the distance between the obtained network and the desired network. This objective aims to provide simulated network as close as possible to the desired network. The second one aims to identify the minimum number of stimuli that are most likely to steer the global biomolecular network from the initial state to the desired state. In other words, this criteria aims to give priority to the quality of the external stimuli over their quantity. The third objective is closely related to the first objective function and aims to minimize the total cost of the external stimuli to be applied to the network components. In fact, the cost of external stimuli maybe associated with the number of external stimuli. So, if we have a number of external stimuli equal to the number of nodes and all the external stimuli have the same cost, the transittability process of the complex biomolecular network will be very expensive. That is why this criterion aims to find the best compromise between the quality of the external stimuli and their cost. The fourth objective aims to identify the minimum set of nodes to be affected by the external stimuli. Indeed, several studies have revealed that, among all the nodes composing the biomolecular network, there are some specific nodes that have the ability to steer the network from its actual state to another specific state. Moreover, the stimulation of all the nodes of the network may place the patient at risk of developing additional adverse effects caused by the external stimuli such as ultraviolet irradiation. Thus, instead of stimulating all the nodes randomly, it is better to have a stimulation strategy which targets a set of specific nodes. This will allow stimulating only a minimum number of nodes thus allowing the transition of the network to the desired state. Then, the fifth objective aims to reduce the patient discomfort during a certain treatment (while finding the best compromise with the other objective functions cited previously). In our context, the patient discomfort encompasses different aspects such as patient pain, stress, vomiting, dizziness, anxiety, fatigue, etc. Indeed, the transittability of a biomolecular network can potentially be uncomfortable and negatively impacts the emotional and mental health of patients, the quality of their life and increases the use of health care resources. For all these reasons, we must consider this important criterion in the transittability process. It is important to mention that both the first and third objectives have been already used in the literature for steering biomolecular networks. However, for the second and fourth objectives, we have added them after a long discussion with expert biologists and because they are crucial for achieving the transittability of biomolecular networks.

Moreover, we propose in Chapter 9 a two-step multi-objective optimization approach for solving this multi-objective problem. Our proposed approach is strongly based on the combination of both Non-dominated Sorting Genetic algorithm (NSGA-II) to obtain the set of Pareto-optimal solutions, and

the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method to provide the decision-maker with the best compromise solution according to its preferences. The NSGA-II is one of the most widely used genetic algorithms for multi-objective problems. Thus, we think that it is interesting for us to chose and adapt this algorithm to the problem of steering complex biomolecular networks. Based on the results obtained in Chapter 10, we conclude that the NSGA-II is well suited to solve the transittability of complex biomolecular networks as a multi-objective optimization problem. Indeed, it is able to find a set of non-dominated solutions in a single run. However, even the obtained results have been satisfactory and positive, more can be done. In fact, we did not test our optimization problem with other methods. Maybe it is recommended to enhance the NSGA-II parameters and why not to solve the problem with other optimization heuristics with the goal of comparing and improving the obtained solutions. This is remaining as future work. Moreover, it is important to note that this work of optimizing the transittability of complex biomolecular networks is still the beginning phase in the literature. There are no other works treating the same problematic based on optimization tools, and there is no existing evaluation yet. We only compare our results according to the work of Wu et al. [6].

This proposed approach was tested and applied to solve the steering of the p53-mediated DNA damage response network with the goal of optimizing simultaneously the different criteria involved in its transittability, in particular, the distance between the simulated final network state and the desired network state, the number of external stimuli, their cost, the number of target nodes and the patient discomfort. We use this example because it has three states called also phenotypes, the normal, the apoptosis and the arrest state. This case study has been already studied and simulated in the work of Wu et al. [6] and Zhang et al. [356], this allows us to compare our results by referring to their experiments. Obtained results are in great agreement with those of [6] and [356]. Indeed, we have succeeded in finding the best compromise among the different criteria of the transittability and in optimizing the steering of the biomolecular network from the normal state to one of the desired state (apoptosis or arrest state). The experimental results illustrate the effectiveness of this approach in optimizing all the objective functions.

We compare our obtained results with the most representative works proposed for steering the biomolecular networks using the controllability notion in literature. We note that the proposed approach (in agreement with Wu et al. [6] results) proposes to affect the minimum set of nodes (not all nodes) rather than the entire nodes of the biomolecular networks. As a consequence, the proposed approach provides less external stimuli than the total number of the network nodes and consequently, the total cost of these stimuli will certainly be less costly than the use of controllability notion [28]. In addition, the proposed method considers as another objective the minimization of the patient discomfort. This increases the applicability of translational medicine for improving human health and disease, including genetic and environmental factors of patient's well-being. This is a great opportunity to understand diseases and find new diagnoses and treatments. Therefore, applications of the proposed optimization approach can be used in the design of treatments such as chemotherapy, the identification of potential drug targets in a signalling network of human cancer, or to study the phenotype transitions (for example, to direct a biomolecular network from its abnormal or disease phenotype to a healthy phenotype).

11.6 Summary

In this chapter, we have presented a discussion of our four proposed approaches. These approaches are combined together under the CBNSimulator platform. This platform enables the study of the behaviour of complex biomolecular networks through the creation of their logic modelling. As seen with the three case studies, the logic-based modelling provides all the necessary elements to model biomolecular networks. This logical formalism treats the biomolecular network based on three aspects: the structural, functional, and behavioural modelling of the network. Furthermore, with the goal of enriching this logic-based modelling, the CBNSimulator integrates a semantic level to this logic modelling through the semantic modelling. This semantic modelling aims to enrich and infer new knowledge, to detect more properties and relationships among the molecular components, and to suggest new inferring data in order to provide the logic model with complementary knowledge and data. This semantic approach joined to the logic-based modelling provides a powerful formalism for modelling and representing complex biomolecular networks. Moreover, the CBNSimulator provides a qualitative, discrete-event simulator. This simulation approach was used to elucidate and predict the behaviour, state changes and properties of the three case studies. Simulation results were displayed in a graphic form to facilitate their interpretation and

analysis. Finally, the CBNSimulator proposes an optimization module allowing biologists to steer complex biomolecular networks from their actual state to another specific state. The experimental results illustrate the effectiveness of this approach in optimizing simultaneously the different criteria involved in their transittability, in particular, the distance between the simulated final network state and the desired network state, the number of external stimuli, their cost, the number of target nodes and the patient discomfort. Thus, using the CBNSimulator, biologists can perform in silico experiments, in particular, the steering of the biomolecular network from a state to another one, which has the advantage of being less costly in time and resources than the in vitro and in vivo experiments. However, this platform must continue to be refined and enhanced. Interesting research directions are discussed in the next part.

General conclusion and future research

In this part, we conclude by briefly foregrounding our thesis' contributions, and suggest specific open questions and directions for future research.

Conclusion

An important objective of systems biology is the construction of predictive models for understanding the dynamic behaviour of complex biomolecular networks and their transittability. Indeed, existing studies have focused mainly on numerical models involving (highly non-linear) differential equations and using tools for estimating parameters [371] to formalize and simulate biomolecular networks. However, as discussed in Chapter 2 to Chapter 4 state-of-the-art quantitative models cannot be reused or merged with other models in a systematic fashion, and are limited to a small number of variables [371].

Moreover, with the recent development of high-throughput technologies, huge amounts of data have been generated to describe the complex processes and molecular mechanisms at work in the cell, through the study of cellular components on several levels.

With these advances, systems biology faces numerous novel challenges:

- The first challenge is how to extract important knowledge from all this data in order to understand and infer cellular functions and behaviours in different conditions.
- Another challenge is the modularity of complex biomolecular networks. It is not an easy task to model and combines large-scale genomic, proteomic and metabolic data in order to obtain a mixed model of the complex biomolecular network. This is a restriction to the re-use of models in systems biology because they are implemented for a specific context.
- An important challenge is to simulate the behaviour of biomolecular networks using logic and qualitative models to automate various forms of biological reasoning. Indeed, understanding the transition states of biomolecular networks and drug target discovery, require qualitative tools to assist the biologist when certain numerical data are unavailable.
- Another challenge for systems biology is to enrich existing quantitative models to include precise semantics of biomolecular networks behaviour and build formal methods to reason about them.
- A supplementary important challenge is to understand the dynamic aspects of these biomolecular networks in order to control and guide their behaviour. This issue is known through the "transittability" that focuses on the idea of steering the complex biomolecular network from an unexpected state to a desired state.

To address these challenges, the overall goal of this study is to propose an intelligent platform that enables biologists to simulate the state changes of biomolecular networks with the goal of steering their behaviours. The development of such a platform involves several related computing methods belonging to different domains such as mathematical systems modelling, knowledge engineering, computer simulation, and combinatorial optimization. Thus, the intelligence of our platform derives from the combination of diverse techniques and fields of study. The first element of intelligence is presented in this dissertation through the logic-based modelling approach that addresses the problem of modelling complex biomolecular networks considering the diversity and heterogeneity of their molecular components, and adopting a global vision which considers their multi-level aspects. This formalization focuses on the structure of the network (modelling of the diverse components and their interactions), network control (identification

of the function and role of each component) and network dynamics (observation of its behaviour over time). The second feature of our intelligent platform concerns the semantic approach, which provides the necessary knowledge for modelling and understanding the behaviour of complex biomolecular networks and their state changes. The Biomolecular Network Ontology presented in this dissertation formalizes the domain knowledge of complex biomolecular networks making it visible and accessible to all biologists working on this topic. Moreover, the reasoning and simulation system exploits all this formally encoded knowledge in order to reproduce the behaviour of complex biomolecular networks and their components over time even with incomplete knowledge. Finally, the proposed multi-objective optimization approach take into account several criteria, such as: the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of stimuli, the minimization of the cost of these stimuli, the minimization of the number of target nodes, and the minimization of the patient discomfort, for optimizing the transittability of complex biomolecular networks.

Contributions

We recall our four contributions here. For each contribution, we stress the results and general conclusions. More general perspectives of our contributions are left to the next section.

In Chapter 6, we propose a logic-based formalization which allows biologists to model the different elements that compose a complex biomolecular network, and to identify and categorize all its components and the nature of their interactions. Indeed, the logic-based formalization was tested with different case studies and provides a relatively simple modelling approach able to capture interesting and relevant behaviours in the cell. In particular, in the case of the poorly understood biomolecular network where quantitative data and parameters are often scarce and hard to obtain. However, to obtain an optimal and more realistic modelling, we need to enhance it with an additional semantic layer. For this reason, we propose a semantic architecture for providing and inferring more knowledge about the functioning of cells at a molecular level.

The proposed semantic architecture (Chapter 7) consists of four ontologies: three of them already exist in the literature, the Gene Ontology (GO), the Simple Event Model Ontology (SEMO), the Time Ontology (TO) and we are developing the Biomolecular Network Ontology (BNO). Linked together, these ontologies provide the necessary concepts for modelling the dynamic behaviour and the transition states of a complex biomolecular network. The Biomolecular Network Ontology describes the static structure of the biomolecular network. Merging with the Simple Event Model Ontology, the Biomolecular Network Ontology describes what can carry out each component of the biomolecular network and the conditions for these activities. Finally, the Biomolecular Network Ontology, the Simple Event Model Ontology and the Time Ontology describe how the biomolecular network and its individual components evolve over time.

In the same chapter, we focus on the implementation of the BNO ontology to describe the domain knowledge of complex biomolecular networks in their static state. This ontology provides information on the biomolecular network and its components (nodes, interactions, states, transition states, etc.) and an indication of the network's context such as the type of sub-network, the type of node, the conditions and nature of interactions, etc. This allows to precisely explain and interpret the semantic context in order to achieve intelligent modelling of biomolecular networks and their state changes. These state changes can be computed with a rule-based system using OWL-SWRL rules. These rules represent a "proof of concept" demonstrating the logical consistency of the approach and validating the relevance of the ontology. Additionally, we simulate the evolution of different biomolecular networks. Obtained results are encouraging and indicated that the BNO is consistent, credible and effective in describing relevant knowledge required in understanding the behaviour of complex biomolecular networks and their state changes. Nevertheless, more efficient simulation tools should be used to study larger biomolecular networks.

Moreover, it is important to note that the complexity of biomolecular networks is firstly due to their large number of coupled components, but also to the diversity of these molecular components and to their intricate interactions. Indeed, biomolecular networks consist of various subnetworks which themselves are composed of several molecular components interacting in their turn with each other, producing a complex global behaviour. The complexity and large size of these networks have prevented a fully quantitative simulation. Thus, biologists require tools allowing them to gain insights into the behaviour of complex

biomolecular networks by simulating the different states of their components over time.

Therefore, we propose a qualitative, discrete-event simulation in Chapter 8. The qualitative simulation predicts the set of possible states based on the logic-based modelling of the real network. The link between the logic model and the qualitative simulation is ensured by the partition and propagation rules. The value of the qualitative simulation comes from the ability to describe natural types of incomplete numerical knowledge, and the ability to deduce a complete set of possible states (using qualitative values rather than real numbers). We chose to use qualitative reasoning for two reasons: (*i*) To understand the overall functioning and properties of complex biomolecular networks through the analysis and simulation of the dynamical model, and the interpretation of the obtained qualitative knowledge. (*ii*) To steer these networks by allowing to evaluate their simulation at any time, even in the lack of quantitative knowledge.

In addition to the qualitative simulation, we propose a discrete-event simulation. This approach was inspired by the DEVS formalism [339], a formalism for supporting the modelling of complex systems, and based on the logic-based modelling. This simulation allows biologists to analyse and predict the effect of changes, the behaviour of the biomolecular network components. It also enables the study of the internal interaction of a subsystem with a complex system. This approach aims at providing biologists with a flexible tool for simulating biomolecular networks by reproducing their behaviour and the state of their components over time and consequently allows them to analyse and understand simulated cell phenomena. These approaches have been verified on diverse biomolecular networks, and the simulation results obtained were formally treated and validated by expert biologists. Indeed, these results correspond to their domain knowledge.

The computation of the transittability of complex biomolecular networks can be considered as an optimization problem. As discussed in Chapter 5, only a few studies have been carried out for this problem and most of them have focused only on the minimization of the nodes required to steer the entire network and on the minimization of the number of stimuli to be applied on the network. However, this assumption is not always realistic, because steering complex biomolecular networks are in general a multi-objective optimization problem. It requires finding appropriate trade-offs among various objectives, for example between the appropriate nodes to be stimulated and the number of external stimuli to be used, their cost and the patient discomfort.

Consequently, we propose (in Chapter 9) a multi-objective mathematical formulation for optimizing the transittability of complex biomolecular networks in which we take into account more criteria such as the minimization of the distance between the simulated final network state and the desired network state, the minimization of the number of stimuli, the minimization of the cost of these stimuli, the minimization of the number of target nodes, and the minimization of the patient discomfort. Moreover, we propose a two-step multi-objective optimization approach for solving this multi-objective problem. Our proposed approach is strongly based on the combination of both Non-dominated Sorting Genetic Algorithm (NSGA-II) [340] to obtain the set of Pareto-optimal solutions, and the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) method [341] to provide the decision-maker with the best compromise solution according to its preferences.

In order to validate all these methods proposed in Chapter 6, 7, 8, and 9, the prototype CBNSimulator has been developed in Chapter 10. The overall architecture of CBNSimulator is composed of four modules. Each module corresponds to one of the proposed approaches discussed above. As well as, we have applied all these approaches to various biomolecular networks, among them the autoregulation of the bacteriophage T4 gene 32, the phage lambda and the p53-mediated DNA damage response network. The logic-based modelling approach provides all the elements that we need to understand the structure, function and behaviour of these case studies, but also to study its evolution over time. In addition, the proposed semantic approach produces good quality (efficient) simulation and rapid reasoning of these biomolecular networks behaviour. The SWRL reasoner is able to reproduce the overall behaviour of these biomolecular networks induced by the SWRL rules defined by the expert biologists. It reproduces behaviours similar to those observed in real life. Indeed, the qualitative simulation method provides all possible simulation results of the given networks and their molecular components. Thus, based on this qualitative method, it is possible to predict the behaviour of these networks even in the absence of quantitative information. It is sufficient to have some essential information to predict their future behaviour. Moreover, the proposed multi-objective optimization approach for solving this multi-objective problem takes into account different criteria which are important to study the transittability of complex biomolecular networks. It provides a set of optimum solutions to the problem according to the user preferences. The results obtained from this study are encouraging and it is expected that they will

be useful in understanding the behaviour of complex biomolecular networks and in optimizing their transittability.

However, our proposed CBNSimulator has some important limitations. First, the complexity of real biomolecular networks means that a number of simplifications are necessary. For example, we have not included interactions between metabolites and genes, although it is known that specific metabolites, such as nicotine, do interact directly with genes. Second, in order to completely understand biomolecular networks, some quantitative aspects should be taken into account. To address this, the quantitative methods described in the State-of-the-art could be used in combination with our tool. Third, the cases studies used here are relatively small and medium-sized networks, and in fact, our semantic method cannot simulate very large networks. The main reason is the time constraint and the exponential number of interactions among the molecular components. It is difficult to compute the dynamics of all the interactions at the same time. For this reason, we integrate a framework for dynamic simulation of complex biomolecular networks based on the Discrete Event System Specification (DEVS) formalism [339], which allows the description of biomolecular networks at different levels. In order to validate all these methods proposed in Chapter 6, 7, 8, and 9, the prototype CBNSimulator has been developed in Chapter 10. The overall architecture of CBNSimulator is composed of four modules. Each module corresponds to one of the proposed approaches discussed above. As well as, we have applied all these approaches to various biomolecular networks, among them the autoregulation of the bacteriophage T4 gene 32, the phage lambda and the p53-mediated DNA damage response network. The logic-based modelling approach provides all the elements that we need to understand the structure, function and behaviour of these case studies, but also to study its evolution over time. In addition, the proposed semantic approach produces good quality (efficient) simulation and rapid reasoning of these biomolecular networks behaviour. The SWRL reasoner is able to reproduce the overall behaviour of these biomolecular networks induced by the SWRL rules defined by the expert biologists. It reproduces behaviours similar to those observed in real life. Indeed, the qualitative simulation method provides all possible simulation results of the given networks and their molecular components. Thus, based on this qualitative method, it is possible to predict the behaviour of these networks even in the absence of quantitative information. It is sufficient to have some essential information to predict their future behaviour. Moreover, the proposed multi-objective optimization approach for solving this multi-objective problem takes into account different criteria which are important to study the transittability of complex biomolecular networks. It provides a set of optimum solutions to the problem according to the user preferences. The results obtained from this study are encouraging and it is expected that they will be useful in understanding the behaviour of complex biomolecular networks and in optimizing their transittability.

However, our proposed CBNSimulator has some important limitations. First, the complexity of real biomolecular networks means that a number of simplifications are necessary. For example, we have not included interactions between metabolites and genes, although it is known that specific metabolites, such as nicotine, do interact directly with genes. Second, in order to completely understand biomolecular networks, some quantitative aspects should be taken into account. To address this, the quantitative methods described in the State-of-the-art could be used in combination with our tool. Third, the cases studies used here are relatively small and medium-sized networks, and in fact our semantic method cannot simulate very large networks. The main reason is the time constraint and the exponential number of interactions among the molecular components. It is difficult to compute the dynamics of all the interactions at the same time. For this reason, we integrate a framework for dynamic simulation of complex biomolecular networks based on the Discrete Event System Specification (DEVS) formalism [339], which allows the description of biomolecular networks at different levels.

Suggested future work

Our contributions detailed in the previous section suggest some specific open questions and directions for future research. The remainder of this section will lay out some of these perspectives. Indeed, although the dynamic synchronization among the four modules (corresponding to our contributions): logic-based modelling, semantic approach, qualitative and quantitative simulation, and the multi-objective optimization module to achieve the full development of our intelligent platform, the CBNSimulator, there are still some important improvements that can be made in further researcher.

- About our first contribution:

As mentioned in the previous section, our proposed logic-based modelling does not include interactions among metabolites and genes, although it is known that some specific metabolites, such as nicotine, do interact directly with genes. One possible direction to explore is to extend the logic-based modelling with the goal of integrating and considering this hypothesis.

- About our second contribution:

An interesting question which we are currently investigating is the problem of extending the semantic approach presented in this thesis by applying it to large and concrete biomolecular networks and enhancing its performance in order to apply it in the domain of drug discovery. This can be solved by integrating other ontologies and adding them to our semantic approach. Considering that SWRL rules provide powerful solutions for problems that cannot be solved with standard Description Logic-based reasoning, we plan to provide a user interface (by adding a functionality to the CBNSimulator) that allow biologists to define their specific SWRL rules to represent their preferences and take profit of the knowledge required for specific and personalized application.

- About our third contribution:

The components of the biomolecular network and their interactions are physically and temporally organized to ensure the production of a particular behaviour or phenotype. These spatiotemporal aspects are important to understand the functioning of a cell as a whole. The spatial organization of the network includes the location of the different molecular components as well as their positions and levels (proteomic, metabolic, etc.). The temporal organization includes the order of the sequence of state transitions of each component, the duration and frequency of these interactions. Both of these spatiotemporal aspects constitute the dynamic properties of the biomolecular network. In this work, to maintain simplicity we take into account the multi-level aspect of the network (considering the different levels of the biomolecular network), but we do not consider the different biological time levels. Indeed, the timing of processes levels with size: from nanosecond at the level of single molecules to microseconds at the level of proteins interaction, to hours at cellular processes, etc. We neglect this property and work within a fixed time level. To address this limitation, we must include our modelling with a timing generator that allows to follow the dynamics of the system and change among different time levels according to the given spatial level and the nature of the process. Moreover, to overcome these difficulties of time levels, a simulator based on agent-based systems may be proposed because they take into account the effects of different time levels.

- About our fourth contribution:

The transittability of complex biomolecular networks has different perspectives in terms of optimization criteria (minimization of total transittability time and other objectives), therefore, it will be interesting to improve and detail some objectives such as the patient comfort. Moreover, we plan to improve the performance of our NSGA-II algorithm by implementing a hybridisation with a local search algorithm, such as the Nelder-Mead simplex method, and compare its performance with other meta-heuristics in order to improve the computation time of our optimization algorithm and the quality of the solutions. We can also use the EASEA platform[3] (EAsy Specification of Evolutionary Algorithms) that ensures the parallel implementation of genetic algorithms on a computer network furnished with GPGPU cards (General-Purpose computation on Graphics Processing Units). Using this platform we can solve the transittability of complex and large-scale networks by taking the advantages of the massive parallelism of many-core architectures.

- About the prototype CBNSimulator:

Our prototype, the CBNSimulator, needs to be improved. First, we are actually working to make a more dynamic synchronization among the four modules, the logic-based modelling, semantic approach, qualitative and quantitative simulation, and the optimization module to improve the performance of our platform. As well as, we plan to extend the experiments in order to assess and analyse in more detail the impact of our different proposals. Indeed, simple cases studies are not enough to completely evaluate

[3] http://easea.unistra.fr/index.php/EASEA_platform

and validate our approaches, therefore, new experiments could be carried out within the application of the CBNSimulator. Moreover, the CBNSimulator platform currently offers only the simulation and optimization layer for complex biomolecular networks. The logic and semantic modelling modules have been implemented implicitly in the code and specifically for each example. Thus, we plan to add new interfaces allowing the user to define and explicitly design the logical and semantic modelling of his own biomolecular network. Solving these problems will contribute to the generality of our platform.

Detailed abstract in French

La biologie des systèmes représente un nouveau domaine de la biologie qui pourrait avoir des applications importantes en recherche biomédicale et en ingénierie biologique. En effet, ce domaine académique vise à proposer des modèles de fonctionnement intégrant différents niveaux d'informations pour décrire et comprendre le comportement d'une cellule, représentée par un réseau biomoléculaire complexe. En combinant les données biologiques expérimentales et les techniques de l'informatique, la biologie des systèmes offre la possibilité d'avoir une vraie compréhension du fonctionnement et des interactions entre les différents composants moléculaires d'une cellule à tous les niveaux de son organisation (génétique, protéomique et métabolique).

En effet, la cellule peut être considérée comme un système complexe composé de milliers d'entités moléculaires différentes (gènes, protéines et métabolites), qui interagissent physiquement, fonctionnellement et logiquement créant un réseau moléculaire. Pour réduire la complexité de ce réseau, la plupart des études traditionnelles se sont concentrées uniquement sur un niveau particulier du système cellulaire, comme les réseaux de régulation des gènes, les réseaux d'interaction protéine-protéine ou encore les réseaux métaboliques. Diverses approches ont été développées pour modéliser, analyser et comprendre ces réseaux, y compris les équations différentielles ordinaires, les méthodes stochastiques, les réseaux booléens, les réseaux bayésiens, les réseaux de Petri, etc. et des études comparatives de ces techniques ont été réalisées. Néanmoins, peu d'approches ont été développées pour étudier le système cellulaire dans son ensemble, et en particulier les interactions entre les différents types de réseaux moléculaires. De plus, la plupart des techniques de modélisation existantes ne tiennent pas compte de l'évolution dynamique du réseau ainsi que de ses différentes transitions d'états.

Récemment, certains auteurs ont commencé à aborder les aspects dynamiques et ont introduit des concepts tels que la "contrôlabilité" d'un réseau, où la capacité de diriger un réseau complexe d'un état initial vers un autre état désiré est mesurée par le nombre minimum de nœuds pilotes requis (nœuds ayant la capacité de diriger l'ensemble du réseau). Ils ont montré que pour avoir une contrôlabilité complète, le nombre minimum de nœuds pilote est de 80 % des nœuds d'un réseau biomoléculaire. Ce résultat a conduit d'autres groupes à développer un cadre théorique pour étudier les transitions entre deux états spécifiques de réseaux complexes, un concept qu'ils appellent la " transitabilité " du réseau. Cette transitabilité exprime l'idée de pouvoir piloter un réseau complexe d'un état initial vers un autre état désiré en stimulant un minimum de nœuds.

Notre thèse s'inscrit dans ce contexte. Par conséquent, nous proposons une plate-forme qui permet aux biologistes de simuler les changements d'état des réseaux biomoléculaires dans le but de piloter leurs comportements et de les faire évoluer d'un état non désiré vers un état souhaitable. Cette plate-forme considérée comme un système intelligent combine la modélisation logique, le raisonnement sémantique et qualitatif, un outil de simulation et un algorithme d'optimisation. Ses objectifs généraux sont: Caractériser les composants moléculaires d'une cellule; Comprendre les interactions dynamiques entre les composants moléculaires et les stimuli environnementaux; Fournir un outil aux biologistes pour reproduire le comportement de réseaux complexes; et Déduire un ensemble optimal de stimuli externes à appliquer pendant un intervalle de temps prédéterminé pour piloter le réseau de son état actuel à un état désiré.

Ainsi, cette plate-forme, que l'on a intitulé CBNSimulator, offre aux biologistes la possibilité de (*i*) modéliser et simuler un réseau biomoléculaire complexe en se basant sur le modèle logique qui est décrit sémantiquement par une modélisation sémantique, et (*ii*) guider et piloter la transition de ces réseaux de leur état actuel à un état souhaité dans des conditions spécifiques et contrôlées. CBNSimulator est composée de quatre modules: un module de modélisation logique, un module ontologique, un module de simulation et un module d'optimisation.

1. *Le module de modélisation logique*: Ce module représente le point de départ de tout nouveau réseau biomoléculaire défini par l'utilisateur. Cette modélisation logique a été conçue pour fournir tous les éléments nécessaires à la modélisation du réseau biomoléculaire en tenant compte de ses différents niveaux et composants moléculaires. Par exemple, l'utilisateur peut spécifier la liste des composants moléculaires et leur type, la liste des interactions entre ces composants et les conditions qui les activent, etc. Cette modélisation logique tient compte de la complexité et de l'hétérogénéité de ces composants moléculaires et de leurs structures multi-niveaux.

2. *Le module ontologique*: Ce module assure la gestion, la modélisation et le partage des connaissances des experts biologistes. Ce module prend en entrée toutes les informations natives introduites par l'expert (état du réseau, sa structure, etc.) à travers la modélisation logique fournie par le premier module. Ensuite, le module ontologique fournit, en sortie, un réseau composé de connaissances natives et déduites (inférées). Ce module joue un rôle très utile pour l'enrichissement de la formalisation du réseau, surtout lorsque la modélisation logique du réseau manque de détails. De plus, les informations supplémentaires fournies par ce module peuvent être utilisées pour identifier de nouvelles relations potentielles entre les composants moléculaires du réseau.

3. *Le module de simulation*: Ce module permet aux utilisateurs de simuler et/ou de reproduire le comportement dynamique du réseau. En effet, ce simulateur intègre toutes les informations fournit par l'expert (le réseau enrichi de connaissances natives et inférées) avec d'autres paramètres afin de mieux reproduire le comportement du réseau biomoléculaire et de ses composants dans le temps. Les résultats générés lors de la simulation sont affichés à l'utilisateur sous forme graphique afin de faciliter leur interprétation, et peuvent également être transmises au module d'optimisation.

4. *Le module d'optimisation*: Ce module fournit un moyen de diriger et piloter le réseau de son état actuel vers un autre état spécifique. Cette optimisation est réalisée en spécifiant les états initial et souhaité du réseau, ainsi que tous les stimuli externes possibles définis par l'utilisateur. Ensuite, ce module d'optimisation fournit les meilleures séquences de stimuli pour piloter le réseau biomoléculaire de son état initial à l'état désiré et, enfin, présente les résultats en fonction des préférences de l'utilisateur. Ce module permet également d'optimiser la transitabilité du réseau en minimisant divers critères, tels que: la distance de proximité entre l'état du réseau obtenu à la fin de la simulation et l'état du réseau désiré, le nombre de stimuli externes, leur coût global, le nombre de nœuds cibles et l'inconfort du patient.

Le manuscrit de thèse est divisé en trois parties et est composé de onze chapitres. Les parties "État de l'art" et "Contributions" ont été divisées chacune en quatre chapitres en fonction des modules de l'architecture de la plate-forme proposée. La première partie *Etat de l'art* vise d'abord à présenter l'environnement biologique dans lequel nous travaillons en explorant l'histoire conceptuelle de la biologie des systèmes et en définissant ses principaux concepts. Ensuite, à présenter les différents outils et approches qui ont été proposés dans les différents domaines de recherche couverts par cette thèse. À la fin de chaque chapitre, une section est consacrée à la définition de l'énoncé du problème lié à chaque domaine de recherche spécifique. Le premier chapitre présente l'environnement biologique et le contexte dans lequel nous travaillons. Nous nous concentrons principalement sur les réseaux biomoléculaires complexes et leur transitabilité. Le deuxième chapitre capitalise les notions de base des modèles mathématiques en biologie systémique, synthétise les plus importants d'entre eux, et présente la principale problématique abordée par notre thèse dans ce domaine: une modélisation logique des réseaux biomoléculaires complexes. Le troisième chapitre donne un aperçu sur les ontologies, décrit les principales bio-ontologies qui ont été proposées en biologie des systèmes et présente la principale problématique abordée par notre domaine: une ontologie de domaine pour décrire le domaine des réseaux biomoléculaires complexes. Le quatrième chapitre se focalise sur les notions de base de la simulation en biologie des systèmes, détaille les principaux outils et plates-formes de simulation dans la littérature, et présente la principale problématique abordée par notre thèse dans ce domaine: un simulateur qualitatif et à événements discrets pour comprendre le comportement des réseaux biomoléculaires complexes. Enfin, le dernier chapitre de cette partie présente les connaissances de base sur les outils d'optimisation, y compris une synthèse des travaux menés sur les problèmes d'optimisation en biologie des systèmes, et présente la principale problématique abordée par notre thèse dans ce domaine: un algorithme génétique pour résoudre et optimiser la transitabilité des réseaux biomoléculaires complexes.

Après avoir présenté le contexte et l'état d'avancement de nos travaux dans la première partie de ce manuscrit, la deuxième partie *Contributions* est consacrée à nos contributions sur la conception et le développement d'une plate-forme de simulation des changements d'état de réseaux biomoléculaires complexes dans l'espoir de comprendre et de piloter leur comportement. Cette plate-forme se compose de quatre modules de base: (*i*) le module de *modélisation* pour formaliser le comportement dynamique des réseaux biomoléculaires, (*ii*) le module *ontologique* pour fournir une description riche des entités cellulaires et les interactions ayant lieux entre elles, (*iii*) le module de *simulation* pour reproduire le comportement dynamique des composants de chaque réseau dans le temps et (*iv*) le module d'*optimisation* pour fournir un ensemble de stimuli externes proposant le meilleur pilotage du réseau biomoléculaire d'un état donné à un autre. Ces contributions sont organisées par domaine spécifique. Ainsi, selon chacun des modules de l'approche que nous proposons, quatre contributions ont été apportées à ce travail. C'est pourquoi cette partie est divisée en quatre chapitres, chacun présentant nos contributions dans un domaine de recherche spécifique. Le sixième chapitre propose une approche logique pour modéliser le comportement dynamique des réseaux biomoléculaires. Ce formalisme s'appuie sur les trois niveaux d'analyse définis par la théorie des systèmes: la modélisation structurelle, fonctionnelle et comportementale. En effet, il vise à décrire et à analyser toutes les propriétés et les mécanismes des réseaux biomoléculaires complexes. Cette modélisation basée sur la logique constituera l'élément de base pour la modélisation, la simulation et la compréhension de la transitabilité de ces réseaux complexes. Le septième chapitre présente une approche sémantique pour la modélisation des réseaux biomoléculaires et décrit l'implémentation de la Biomolecular Network Ontology (BNO), une ontologie créée spécialement pour répondre aux besoins de l'analyse du comportement des réseaux biomoléculaires complexes. Cette ontologie fournit une base pour la simulation qualitative de ces réseaux. Le huitième chapitre propose deux types de simulation: une simulation qualitative et une simulation à événements discrets. La première répond à la complexité du calcul des méthodes de simulation quantitative qui sont parfois impossibles à mettre en œuvre. La deuxième méthode est une simulation intégrative à événements discrets qui considère que le comportement du réseau biomoléculaire complexe émerge de l'interaction des différents niveaux du réseau. Enfin le dernier chapitre de cette partie, présente tout d'abord une formulation mathématique multi-objectifs pour optimiser la transitabilité des réseaux biomoléculaires complexes considérant divers critères tels que la minimisation de la distance de proximité entre l'état du réseau obtenu à la fin de la simulation et l'état du réseau désiré, la minimisation du nombre de stimuli externes, la minimisation de leur coût global, la minimisation du nombre de nœuds cibles, et la minimisation de l'inconfort du patient. Ce chapitre présente également une approche d'optimisation multi-objectifs pour résoudre ce problème. Cette approche est basée sur la combinaison de l'algorithme génétique de tri non dominé (NSGA-II) qui génère l'ensemble des solutions optimales de Pareto-optimal, et de la méthode TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) pour fournir au décideur la meilleure solution de compromis en fonction de ses préférences.

La troisième partie *Validation* présente, détaille et discute nos résultats. Cette partie est divisée en deux chapitres. Le dixième chapitre qui présente un prototype de plate-forme, intitulé CBNSimulator, que nous avons développé pour valider nos contributions ainsi que les expériences que nous avons menées pour déterminer les performances de ce prototype. Cette plate-forme est basée sur le modèle logique et sémantique du réseau biomoléculaire ainsi que sur les performances de simulation à événements discrets pour comprendre l'évolution et le comportement des réseaux biomoléculaires complexes dans le temps. Le onzième et dernier chapitre discute les résultats des diverses expériences que nous avons menées afin d'évaluer nos contributions, tout en comparant nos résultats à ceux de la littérature.

Ainsi pour résumer, cette thèse présente quatre contributions:

En première partie, nous proposons une approche de modélisation logique permettant de décrire et modéliser les réseaux biomoléculaires complexes. Cette modélisation logique se base sur la séparation des trois axes de la théorie des systèmes, à savoir: l'aspect structurel, fonctionnel et comportemental du réseau. Cette séparation des connaissances de nature différente simplifie la complexité de la tache de modélisation des réseaux biomoléculaires complexes.

Par la suite, une deuxième contribution porte sur une approche sémantique composé de quatre ontologies, Gene Ontology, Simple Event Ontology, Time ontology et Biomolecular Network Ontology. Conjointement, ces ontologies permettent d'enrichir la modélisation logique des réseaux biomoléculaires en rajoutant une couche sémantique fournissant les concepts nécessaires à la modélisation du comportement dynamique et des états de transition d'un réseau biomoléculaire complexe. En particulier, nous avons développé la Biomolecular Network Ontology (BNO) pour décrire les connaissances du domaine

des réseaux biomoléculaires complexes et fournir toutes les connaissances et les éléments nécessaires à la réalisation d'une modélisation intelligente des réseaux biomoléculaires et de leurs changements d'état. Ces changements d'état sont simulés grâce à un système de raisonnement à base de règles SWRL.

En troisième partie, nous présentons une simulation qualitative à événements discrets, basée sur la modélisation logique et sémantique, pour simuler qualitativement le réseau biomoléculaire et interpréter son comportement (ainsi que celui de ses différents composants moléculaires) dans le temps. Cette technique de simulation facilite la compréhension, l'analyse et l'interprétation du fonctionnement global et des propriétés des réseaux biomoléculaires complexes. Elle permet également de simuler ces réseaux en permettant d'évaluer leur état à tout moment, même en cas d'absence de connaissances quantitatives.

Finalement, dans la quatrième contribution nous proposons une formulation mathématique multi-objectifs pour optimiser la transitabilité des réseaux biomoléculaires complexes dans laquelle nous prenons en compte davantage de critères tels que la minimisation de la distance de proximité entre l'état du réseau obtenu à la fin de la simulation et l'état du réseau désiré, du nombre de stimuli externes, la minimisation du coût de ces stimuli, la minimisation du nombre de nœuds cibles et la minimisation de l'inconfort du patient. Afin de résoudre ce problème, nous avons aussi proposé une approche d'optimisation multi-objectifs basée sur la combinaison de l'algorithme NSGA-II (Non-dominated Sorting Genetic Algrorithm) permettant d'obtenir l'ensemble des solutions Pareto-optimales, et de la méthode TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) fournissant le meilleur compromis entre les différents objectifs du décideur.

En se basant sur les quatre contributions citées précédemment, un prototype logiciel intitulé CBN-Simulator a été développé. En outre, nous avons évalué ce prototype et ses différents modules en les appliquant à divers réseaux biomoléculaires, à savoir, le bactériophage T4 gene 32, le phage lambda et le réseau de signalisation p53.

Nous avons montré sur les trois études de cas que l'approche de la modélisation logique des réseaux biomoléculaires complexes démontre et décrit clairement tous les éléments dont nous avons besoin pour comprendre la structure, la fonction et le comportement de ces réseaux biomoléculaires, mais aussi pour en étudier leur évolution dynamique dans le temps.

La première étape de la modélisation logique est la construction et la définition des éléments moléculaires des réseaux biomoléculaires complexes. Dans cette modélisation structurelle, les éléments essentiels des réseaux biomoléculaires sont formalisés en nœuds qui sont classés en gènes, protéines et métabolites. Ces nœuds sont représentés soit par des valeurs booléennes, soit par des valeurs réelles. En fait, les modèles logiques en biologie des systèmes ont d'abord été développés à partir des réseaux booléens à partir des travaux de Thomas [357]. Cependant, la modélisation logique proposée n'est pas entièrement basée sur des modèles booléens. En effet, dans notre modélisation, les nœuds du réseau biomoléculaire sont formalisés par des variables booléennes (cas des gènes actifs ou inactifs), mais aussi par des variables réelles (cas de la concentration des protéines et métabolites). Par conséquent, notre modèle logique proposé est capable de modéliser à la fois des entités discrètes, qui sont évaluées en $\{0; 1\}$, et des entités continues, qui sont évaluées en domaine réel $\mathbb{R}$. Cette propriété est considérée utile parce qu'elle permet d'ajouter des clarifications et des détails supplémentaires. Ainsi, les résultats fournis par la modélisation logique proposée, tout en restant qualitatifs, peuvent être plus fins que ceux fournis par la modélisation booléenne.

Une fois définis, les nœuds du réseau sont reliés par des arcs. Ces arcs représentent l'interaction qui peut se produire entre chaque couple de nœuds. En effet, la partition des nœuds du graphe induit une partition en une série de différents types d'interactions qui permettent de relier les différents niveaux du réseau biomoléculaire. Nous concluons donc que la modélisation logique proposée est un modèle à plusieurs niveaux. En effet, dans un réseau biomoléculaire, des interactions peuvent se produire entre des composants moléculaires au même niveau (par exemple des interactions entre gènes) ou entre différents niveaux (par exemple entre protéines et gènes) formant un système complexe à plusieurs niveaux spatio-temporels. Par conséquent, la modélisation logique proposée permet de modéliser et d'intégrer les connaissances biologiques qui existent à différents niveaux. C'est une tâche difficile et impossible avec des modèles quantitatifs tels que les équations différentielles ou les modèles stochastiques, en raison du nombre élevé de composants moléculaires du réseau biomoléculaire.

De plus, cette propriété a été validée expérimentalement par les études de cas que nous avons présentées dans la partie validation. Tous nos études de cas sont composées de molécules appartenant à des niveaux différents et interagissant les unes avec les autres.

Ainsi, nous constatons que la modélisation basée sur la logique est mieux adaptée pour décrire les

réseaux biomoléculaires à grande échelle, où les données quantitatives sont incomplètes et où les composantes sous-cellulaires hétérogènes appartenant aux différents niveaux du réseau peuvent être représentées dans une seule et complète formalisation.

Dans notre modélisation logique, les composantes moléculaires du réseau ne sont pas décrites par des équations bien définies comme dans les modèles quantitatifs, mais des règles sont assignées aux composants moléculaires pour étudier les propriétés qui émergent par leur interaction, généralement considérées comme une modélisation fondée sur des règles. Par conséquent, nous notons que cette modélisation basée sur la logique convient pour décrire des réseaux biomoléculaires à grande échelle avec plusieurs composantes moléculaires, et qu'elle est beaucoup moins coûteuse en termes de calcul que les modèles quantitatif et continu.

De plus, la modélisation logique proposée permet de classer les différents types d'interactions qui se produisent entre les entités moléculaires. Afin de clarifier les interactions et de préciser la fonction de chacune d'elles, la modélisation fonctionnelle de de notre modèle vise à associer ces interactions moléculaires avec les concepts de l'ontologie des interactions proposés par Van Landeghem et al. [4]. En effet, à travers cette ontologie, nous fournissons au modèle logique les connaissances nécessaires et précises sur les interactions biologiques. Cette fonction supplémentaire permet de définir précisément le type d'interaction et la condition qui l'active. Cette précision contribue à la mise en œuvre d'un modèle valide. Ceci est essentiel pour obtenir une modélisation multi-échelles proche de la réalité et pour minimiser les écarts entre le modèle proposé et les expériences réelles. Cette précision se voit dans la modélisation des études de cas. La fonction de chaque interaction est précise et clairement définie.

Selon les études de cas présentées dans le chapitre précédent, la modélisation logique proposée a démontré son efficacité pour décrire et modéliser l'évolution dynamique du réseau biomoléculaire et pour reproduire son comportement dans le temps. Cet aspect est assuré par l'utilisation des fonctions d'agrégation qui calculent les transitions d'état du réseau pendant la simulation. En effet, pour chaque composant moléculaire, nous définissons une fonction agrégée qui calcule son évolution en fonction de son état actuel, de l'état des composants précédents et des caractéristiques de ses interactions entrantes.

Dans notre modélisation logique, les composants moléculaires suivent des règles simples (généralement définies par des experts biologistes et représentées par les fonctions d'agrégation), mais leurs interactions peuvent induire de nouveaux comportements (phénotypes) à l'échelle du réseau. Ces fonctions d'agrégations peuvent donc être utilisées pour étudier les comportements émergents dans les réseaux biomoléculaires complexes. L'avantage est qu'elles peuvent être utilisées pour des réseaux dont les composants moléculaires ne peuvent pas être bien définis par une équation mathématique précise mais peuvent être définis avec des règles. Plus important encore, ils sont beaucoup plus simples à calculer que les modèles quantitatifs. Leur seule limite est qu'elles ne sont pas exactes pour la modélisation quantitative.

Les réseaux biomoléculaires complexes sont soumis à de fortes variations dues à des stimuli internes ou externes. Il est donc difficile de prédire leurs comportements car ils émergent en fonction de leurs états et de l'état de leur environnement. Un réseau biomoléculaire exposé à des stimuli peut présenter des comportements qualitativement plus différents qu'un réseau travaillant sans stimuli. En d'autres termes, une transition de phase ou un état de transition peut être induit par ces stimuli. Cet aspect a été pris en compte et étudié dans la modélisation logique. La plupart des modèles mathématiques présentés dans la littérature ne peuvent représenter et prendre en compte ces stimuli et les changements d'état qu'ils provoquent.

C'est une caractéristique très importante qui fait la différence de notre approche logique par rapport aux autres modèles. Comme le montrent les deuxième et troisième études de cas présentées dans la partie validation. Le formalisme logique proposé prend en compte et modélise les changements d'état des réseaux biomoléculaires causés par les stimuli. En effet, le modèle logique proposé tient compte de deux types de stimuli: les stimuli internes et les stimuli externes. En fait, les changements d'un état moléculaire peuvent se produire soit par un stimulus interne modélisé par la fonction d'agrégation décrite ci-dessus, soit par un stimulus externe généré à l'extérieur de la cellule. Nous définissons donc explicitement cette notion de stimuli comme un événement qui provoque des changements dans l'état de la molécule sur laquelle il agit et donc change l'état de l'ensemble du réseau. Ensuite, nous l'intégrons aux fonctions d'agrégation.

Ce point est très important et permet à la modélisation logique de comprendre la réponse cellulaire en présence de stimuli externes ou internes, de comprendre le mode d'action des médicaments (par stimuli) et de prédire le comportement des réseaux biomoléculaires complexes en réponse aux médicaments. Ceci a été prouvé par la troisième étude de cas, le réseau de réponse aux dommages de l'ADN à médiation p53,

basé sur cette modélisation logique, le CBNSimulator a simulé et géré avec succès les états de transition du réseau de l'état normal à l'état apoptose et/ou de l'état normal à l'état d'arrêt. Cette étude de cas souligne le fait que notre modélisation basée sur la logique peut également être utilisée pour modéliser et comprendre la relation et le mode d'action entre la conception des médicaments et la modélisation des réseaux biomoléculaires.

En conclusion, cette modélisation logique permettra aux biologistes de produire des modèles formels des réseaux biomoléculaires puis de les simuler sur ordinateur. Toutes les propriétés discutées ici ont été validées expérimentalement dans la partie validation. En effet, les études de cas ont été formalisées dans le cadre de la modélisation fondée sur la logique et simulées dans le simulateur CBNSimulator. Le premier exemple de réseau, le bactériophage T4 gene 32 est choisi pour sa simplicité, il est assez simple pour être calculé mentalement afin de juger facilement des résultats produits. Cependant, pour les deux autres exemples, nous validons leur modélisation par comparaison avec les résultats d'autres travaux. Nous concluons que la modélisation logique fournit tous les éléments nécessaires à la formalisation et à la simulation des comportements et des états de transition de ces réseaux biomoléculaires. Ce modèle logique fournit un formalisme formel et mathématique pour modéliser et simuler la structure complexe des réseaux biomoléculaires, l'effet des stimuli qui ciblent certains composants moléculaires, et la simulation de l'évolution dynamique de leurs comportements.

Comme nous l'avons vu, il n'existe pas de meilleure approche pour évaluer une ontologie. Pour cette raison, nous vérifions l'ontologie BNO avec différentes approches. Tout d'abord, nous nous concentrons en particulier sur la validation automatisée de l'ontologie, qui est une condition préalable nécessaire au développement sain d'une ontologie. La vérification automatisée de la cohérence a été effectuée par l'intermédiaire du raisonneur Hermit. Sur la base des commentaires du raisonneur, les incohérences ont été corrigées de manière itérative tout au long du processus de développement de l'ontologie BNO. Les résultats finaux obtenus par le raisonneur étaient statistiquement significatifs et ont révélé qu'il n'y avait pas d'incohérences dans l'ontologie BNO.

L'ontologie BNO a également été évaluée selon différents critères en termes de précision, d'adaptabilité, de clarté, d'exhaustivité, etc. Pour chaque critère, l'ontologie BNO est évaluée comme suit. *(1) Précision*: Les définitions et les descriptions de l'ontologie correspondent aux connaissances des experts dans ce domaine. Les informations concernant les concepts de l'ontologie BNO ont été développées à partir de la célèbre ontologie génétique (Gene Ontology). De plus, nous avons obtenu l'aide et l'expertise de nos collaborateurs de l'équipe du CSTB qui ont évalué l'ontologie BNO et conclu qu'elle ne contient pas d'erreurs sémantiques, d'incohérences logiques et de conflits conceptuels. *(2) Adaptabilité*: Nous avons opté pour le développement de l'ontologie BNO dans le cadre d'une architecture sémantique globale composée de cinq ontologies reliées entre elles : l'ontologie génique (Gene Ontology), l'ontologie des interactions biologiques (Interaction Ontology), l'ontologie de la modélisation des évènements (Simple Event Model Ontology), l'ontologie du temps (Time Ontology) et celle que l'on a développée l'ontologie BNO. Cette architecture vise à aligner et fusionner l'ontologie BNO avec le reste des ontologies à travers les relations d'équivalence *owl:equivalenceClass* ou de spécialisation *owl:subclassOf*. Ce choix améliore l'extensibilité, la réutilisabilité et rend l'ontologie BNO facilement adaptable aux contextes dynamiques des réseaux biomoléculaires. *(3) Clarté*: Lors de l'élaboration de l'ontologie BNO, nous avons pris soin d'attribuer une description claire et non ambiguë pour définir et catégoriser les concepts et les relations qui existent entre eux. Cette clarté est assurée par l'utilisation du *rdfs:comment* qui permet d'annoter une ontologie. De cette manière, l'ontologie BNO communique efficacement le sens voulu de ses termes. *(4) Complétude*: Ce critère détermine si l'ontologie peut répondre à toutes les questions auxquelles elle est supposée être en mesure de répondre. Il fournit une estimation de la façon dont l'ontologie BNO représente le domaine des réseaux biomoléculaires complexes et leur transitabilité. Ces questions ont été précisées par les biologistes experts de l'équipe CSTB et il a été vérifié qu'il est possible d'y répondre à toutes ces questions. *(5) Efficacité informatique*: Une ontologie peut être analysée par un système d'inférence. Dans notre cas, l'ontologie BNO a été analysée par les deux mécanismes de raisonnement Hermit et Pellet. Nous avons conclu que le raisonnement sur l'ontologie BNO est cohérent et permet des inférences dans un délai raisonnable. De plus, la complexité de cette opération est adéquate en terme de temps de raisonnement. *(6) Concision*: Les termes de l'ontologie BNO ont été vérifiés sous l'assistance d'experts biologistes qui ont vérifié que l'ontologie ne contient pas de termes redondants. De plus, nous avons utilisé l'outil OntOlogy Pitfall Scanner `http://oops.linkeddata.es/advanced.jsp` pour vérifier que l'ontologie *BNO* est logiquement correcte. Les résultats d'analyse ont prouvé l'exactitude de l'ontologie BNO. *(7) Consistance*: Ce critère garantit que les axiomes logiques de l'ontologie sont

satisfaisants et cohérents. La satisfaction des axiomes logiques est constatée lorsqu'il est impossible de trouver une contradiction entre les axiomes. Comme nous l'avons expliqué précédement, les raisonneurs OWL et SWRL n'ont détecté aucune incohérence. Aucune incohérence ou violation n'a été constatée.

La combinaison de ces critères nous permet de contrôler l'ontologie BNO à partir de différents niveaux. Les résultats finaux de la validation fondée sur des critères ont indiqué que l'ontologie BNO était claire, extensible et complète.

De plus, nous évaluons l'utilité de l'ontologie BNO à travers la validation des connaissances expertes. L'ontologie BNO a été utilisée pour modéliser notre ensemble d'études de cas. Les résultats ont prouvé que l'ontologie BNO est capable de déduire les principaux concepts des études de cas et leurs propriétés et est capable de déduire de nouvelles connaissances telles que le calcul de l'état suivant des composants moléculaires.

Ces résultats ont prouvé que l'ontologie BNO est capable de décrire et de modéliser la transitabilité d'un réseau biomoléculaire. Cependant, il est important de noter que l'ontologie BNO ne peut pas modéliser la transitabilité des réseaux à grande échelle, et que des outils de simulation plus efficaces devraient être utilisés pour étudier les réseaux plus importants.

L'approche de simulation sémantique proposée a produit une simulation de bonne qualité (efficace) et un raisonnement rapide du comportement de ces réseaux biomoléculaires. Le raisonneur à base de règles SWRL est capable de reproduire le comportement global de ces réseaux biomoléculaires grâce aux règles SWRL définies par les biologistes experts. Comme il a été montré dans la partie validation, ce raisonneur reproduit des comportements similaires à ceux observés par les biologistes in vivo.

Le CBNSimulateur est équipé d'un simulateur qualitatif à événements discrets. Ce simulateur permet de simuler in silico le comportement des réseaux biomoléculaires complexes. Il simule et teste les différents changements d'état des réseaux biomoléculaires dans diverses conditions expérimentales. Le simulateur qualitatif à événements discrets proposé peut également aider les biologistes à découvrir et à détecter les stimuli qui régulent le réseau biomoléculaire, comme les effets des médicaments sur un réseau biomoléculaire. Comme on a pu le conclure au chapitre 4, les techniques de simulation peuvent être classées en deux catégories: les techniques quantitatives et qualitatives. La simulation quantitative fournit la prédiction la plus précise pour décrire le comportement d'entités moléculaires spécifiques.

Néanmoins, le manque de données quantitatives limite son utilisation à des études de cas spécifiques (lorsque toutes les données quantitatives requises sont disponibles), mais ne peut être utilisé dans des réseaux plus généraux ou à grande échelle. D'autre part, la simulation qualitative simplifie la simulation réelle du réseau biomoléculaire et est généralement capable de reproduire le comportement du réseau. De plus, cette simulation qualitative peut être utilisée pour expliquer et prédire le comportement et les changements d'état du biomoléculaire et de ses composantes au moyen de simulations discrètes. En raison du manque de données quantitatives, diverses techniques de simulation qualitative ont été proposées dans la littérature, mais la plupart d'entre elles ne peuvent soutenir la simulation de réseaux biomoléculaires complexes et multiniveaux. En outre, la plupart d'entre eux souffrent du manque d'automatisation dans la simulation des propriétés biomoléculaires. Le simulateur que l'on propose est différent des techniques existantes. En effet, il est entièrement basé sur une modélisation logique. Par conséquent, la puissance de ce simulateur est essentiellement due à l'efficacité de la modélisation basée sur la logique des réseaux biomoléculaires complexes qui est suffisamment expressive pour intégrer et capturer les différents éléments et propriétés qualitatives nécessaires pour comprendre le comportement dynamique et les changements d'état des réseaux biomoléculaires. Mais aussi, dans la représentation et la simulation des réseaux biomoléculaires multi-niveaux. Ainsi, la simulation qualitative à événements discrets proposée permet de simuler des réseaux biomoléculaires complexes et multiniveaux. Cette propriété a été vérifiée à travers les différentes études de cas du chapitre précédent. Toutes les études de cas sont composées de composants moléculaires et d'interactions appartenant à un niveau différent au sein du réseau biomoléculaire. De plus, la simulation qualitative à événements discrets que nous proposons est capable d'effectuer une simulation automatique et efficace des réseaux biomoléculaires.

En effet, le noyau de simulation repose sur un système discret basé sur les événements (le formalisme DEVS), qui est utilisé comme un outil de simulation efficace et précis des systèmes complexes à différents niveaux d'abstraction. Cela permet de garantir des méthodes de mise à jour synchrones et asynchrones pour la simulation. En l'absence de perturbations, le simulateur définit les états initiaux des composants moléculaires et le temps initial, puis met à jour l'état de tous les composants en même temps. Cette synchronisation est basée sur les règles d'interactions (fonctions agrégées) définies par le biologiste. C'était le cas des première et deuxième étude de cas du chapitre précédent. Cependant, la simulation asynchrone

tient compte des différentes perturbations causées par les stimuli. Lorsque certains stimuli affectent certains composants moléculaires, le simulateur met à jour l'état de ces composants puis synchronise l'état des autres composants. C'est ce que nous avons expérimenté dans la troisième étude de cas du chapitre précédent. Un autre avantage du mécanisme de simulation qualitative proposé est qu'il n'exige pas de données quantitatives précises, mais encore plus il permet de simuler le comportement des réseaux biomoléculaires en utilisant uniquement des données qualitatives. Ce mécanisme de simulation est bien adapté pour simuler et analyser des réseaux biomoléculaires à grande échelle, même en l'absence de données quantitatives. Ce mécanisme qualitatif proposé a été vérifié et illustré pour l'exemple du gène 32 du bactériophage T4. Le mécanisme de simulation repose sur l'élaboration d'un graphe causal dont les nœuds sont des variables (représentant les composants moléculaires du réseau) concernées par cette simulation et les arcs sont des relations de causalité entre ces variables (représentant les interactions entre les composants moléculaires). L'état du réseau biomoléculaire est décrit par quelques valeurs qualitatives distinctes correspondant à des valeurs quantitatives précises (appelées l'espace de quantité). Pour calculer la valeur qualitative des nœuds, ce mécanisme est basé à la fois sur les règles de partition et les règles de propagation. Ces règles sont utilisées pour calculer la valeur de la variable cible au moment suivant $(t+1)$ en fonction de sa valeur qualitative au moment actuel (t) et de la valeur des nœuds de ses prédécesseurs en ce moment(t). L'application de la simulation qualitative à l'exemple donné montre que le mécanisme de simulation qualitative proposé est capable de modéliser et de simuler des réseaux biomoléculaires complexes en prédisant leurs différents comportements pour différentes contraintes de simulation.

En conclusion, la simulation qualitative à événements discrets effectuée par le simulateur CBNSimulator peut être utilisée pour élucider et prédire le comportement, les changements d'état et les propriétés des réseaux biomoléculaires complexes. De plus, le simulateur affiche les différents états de chaque composant moléculaire sous une forme graphique facile à interpréter. Nous avons testé ce simulateur sur les trois études de cas. Les résultats montrent la simulation des performances du simulateur proposé dans la modélisation et la simulation de réseaux biomoléculaires complexes multi-échelles dans différentes conditions environnementales et en considérant les perturbations causées par des stimuli externes.

Bien que le simulateur proposé présente de nombreux avantages, il a le potentiel d'apporter d'autres améliorations importantes. En effet, la simulation qualitative proposée ne permet pas de préciser les différentes échelles de temps. Ceci peut être considéré comme la limite principale de notre simulation qualitative à événements discrets. Cependant, nous espérons améliorer notre simulateur en travaillant sur ce point. Cette orientation future est décrite en détail dans la section perspective de la partie suivante. De plus, l'intégration de cette simulation qualitative à événements discrets avec un autre outil de simulation quantitative continue permettrait d'augmenter les performances et d'obtenir un simulateur semi-quantitatif incluant les avantages des simulations qualitatives et quantitatives. C'est l'une des orientations possibles pour l'avenir de ce travail.

L'optimisation de la transitabilité des réseaux biomoléculaires complexes est l'un des principaux objectifs du CBNSimulator. En effet, comme indiqué précédemment, seules quelques études se sont concentrées sur ce problème et la plupart l'ont considéré comme un problème d'optimisation mono-objectif et ont négligé tout autre critère pour le pilotage de ces réseaux biomoléculaires. Ces recherches se concentrent uniquement sur la minimisation du nombre de nœuds nécessaires pour piloter le réseau et/ou sur la minimisation du nombre de stimuli externes à appliquer sur le réseau. Cependant, même si ces deux critères sont des conditions nécessaires, ils ne sont pas suffisants pour piloter complètement des réseaux biomoléculaires complexes. En effet, cette hypothèse n'est pas toujours réaliste, car le pilotage de réseaux biomoléculaires complexes est en général un problème d'optimisation multi-objectifs. Il faut trouver des compromis appropriés entre divers objectifs, comme la minimisation de la distance entre l'état final simulé du réseau et l'état souhaité du réseau, la minimisation du nombre de stimuli externes, la minimisation du coût de ces stimuli, la minimisation du nombre de nœuds cibles et la minimisation de l'inconfort du patient.

C'est pourquoi, dans ce domaine, nous proposons d'abord une formulation mathématique multi-objectifs pour optimiser la transitabilité des réseaux biomoléculaires complexes dans laquelle nous prenons en compte davantage de critères tels que la minimisation de la distance entre l'état final simulé du réseau et l'état souhaité du réseau, la minimisation du nombre de stimuli, la minimisation du coût de ces stimuli, la minimisation du nombre de nœuds cibles, et la minimalisation de l'inconfort du patient. En effet, ces cinq objectifs sont les piliers fondamentaux pour piloter avec succès l'état des réseaux biomoléculaires. Le premier objectif qui est le plus important est la minimisation de la distance entre l'état final simulé du réseau et l'état souhaité du réseau. Cet objectif est assuré par le simulateur et consiste à calculer la

distance entre le réseau obtenu et le réseau souhaité. Cet objectif vise à fournir un réseau simulé aussi proche que possible du réseau souhaité. Le second vise à identifier le nombre minimum de stimuli les plus susceptibles de faire passer le réseau biomoléculaire global de l'état initial à l'état souhaité. En d'autres termes, ce critère vise à donner la priorité à la qualité des stimuli externes sur leur quantité.

En effet, ces cinq objectifs sont les piliers fondamentaux pour piloter avec succès l'état des réseaux biomoléculaires. Le premier objectif qui est le plus important est la minimisation de la distance entre l'état final du réseau simulé et l'état souhaité du réseau. Cet objectif est assuré par le simulateur et consiste à calculer la distance entre le réseau obtenu et le réseau souhaité. Il vise à fournir un réseau simulé aussi proche que possible du réseau souhaité. Le second objectif vise à identifier le nombre minimum de stimuli les plus susceptibles de faire piloter le réseau biomoléculaire de l'état initial à l'état souhaité. En d'autres termes, ce critère vise à donner la priorité à la qualité des stimuli externes par rapport à leur quantité. Le troisième objectif est étroitement lié à la fonction du premier objectif et vise à minimiser le coût total des stimuli externes à appliquer aux composants du réseau. En fait, le coût des stimuli externes peut être associé au nombre de stimuli externes. Ainsi, si nous avons un nombre de stimuli externes égal au nombre de nœuds et que tous les stimuli externes ont le même coût, le processus de transitabilité du réseau biomoléculaire complexe sera très coûteux. C'est pourquoi ce critère vise à trouver le meilleur compromis entre la qualité des stimuli externes et leur coût. Le quatrième objectif vise à identifier l'ensemble minimal de nœuds devant être affectés par les stimuli externes. En effet, plusieurs études ont révélé que, parmi tous les nœuds composant le réseau biomoléculaire, il existe des nœuds spécifiques qui ont la capacité de faire passer le réseau de son état actuel à un autre état spécifique. De plus, la stimulation de tous les nœuds du réseau peut exposer le patient au risque de développer des effets indésirables supplémentaires causés par les stimuli externes tels que l'irradiation ultraviolette. Ainsi, au lieu de stimuler tous les nœuds au hasard, il est préférable d'avoir une stratégie de stimulation qui cible un ensemble de nœuds spécifiques. Cela permettra de ne stimuler qu'un nombre minimum de nœuds, permettant ainsi la transition du réseau vers l'état désiré. Ensuite, le cinquième objectif vise à réduire l'inconfort du patient lors d'un certain traitement (tout en trouvant le meilleur compromis avec les autres fonctions objectives citées précédemment). Dans notre contexte, l'inconfort du patient englobe différents aspects tels que la douleur, le stress, les vomissements, les étourdissements, l'anxiété, la fatigue, etc. En effet, la transitabilité d'un réseau biomoléculaire peut être potentiellement inconfortable et avoir un impact négatif sur la santé émotionnelle et mentale des patients, sur leur qualité de vie et sur l'utilisation des ressources médicales. Pour toutes ces raisons, nous devons tenir compte de cet important critère dans le processus de transitabilité.

Il est important de mentionner que le premier et le troisième objectif ont été déjà utilisés dans la littérature pour le pilotage des réseaux biomoléculaires. Cependant, pour le deuxième et quatrième objectifs, nous les avons ajoutés après une longue discussion avec des experts biologistes et parce qu'ils sont essentiels pour atteindre la transitabilité des réseaux biomoléculaires.

De plus, nous proposons une approche d'optimisation multi-objectifs en deux étapes pour résoudre ce problème multi-objectifs. Notre approche proposée est fortement basée sur la combinaison de l'algorithme génétique NSGA-II (Non-Dominated Sorting Genetic Algorithm II) pour obtenir l'ensemble des solutions de Pareto-optimal, et de la méthode TOPSIS (Technique d'ordre de préférence par similarité de solution idéale) pour fournir au décideur la meilleure solution en fonction de ses préférences.

L'algorithme génétique NSGA-II est l'un des algorithmes génétiques les plus utilisés pour les problèmes multi-objectifs. Ainsi, nous pensons qu'il est intéressant pour nous de choisir et d'adapter cet algorithme à la problématique du pilotage de réseaux biomoléculaires complexes. En nous basant sur les résultats obtenus, nous concluons que le NSGA-II est bien adapté pour résoudre la problématique de la transitabilité des réseaux biomoléculaires complexes en tant que problème d'optimisation multi-objectif. En effet, il est capable de trouver un ensemble de solutions non dominées en un seul essai. Cependant, même si les résultats obtenus sont satisfaisants et positifs, il est possible de faire davantage. En effet, nous n'avons pas testé notre problème d'optimisation avec d'autres méthodes. Il est peut-être recommandé d'améliorer les paramètres NSGA-II et pourquoi pas de résoudre le problème avec d'autres heuristiques d'optimisation dans le but de comparer et d'améliorer les solutions obtenues. Il s'agit là d'un travail futur. De plus, il est important de noter que ce travail d'optimisation de la transitabilité des réseaux biomoléculaires complexes en est encore à ses débuts dans la littérature. Il n'existe pas d'autres travaux traitant la même problématique à partir d'outils d'optimisation, et il n'existe pas encore d'évaluation. Nous ne comparons nos résultats qu'en fonction des travaux de Wu et al. [6] et de Zhang et al. [356].

Cette approche proposée a été testée et appliquée pour résoudre le pilotage du réseau de la régulation

de la protéine p53 face aux dommages de l'ADN suite à des stimuli exteres « the p53-mediated DNA damage response network » dans le but d'optimiser simultanément les différents critères impliqués dans sa transitabilité, en particulier la distance entre l'état final du réseau simulé et celui souhaité, le nombre de stimulus externes, leur coût, le nombre de points cibles et les gênes pour le patient. Nous utilisons cet exemple parce qu'il a trois états appelés aussi phénotypes : l'état normal, l'apoptose et l'état d'arrêt. Cette étude de cas a déjà été étudiée et simulée dans les travaux de Wu et al. [6] et Zhang et al. [356], ce qui nous permet de comparer nos résultats en référence à leur résultat. En effet, nos résultats sont en accord avec les résultats obtenus par ces deux derniers. Nous avons réussi à trouver le meilleur compromis entre les différents critères de transitabilité et à optimiser le pilotage du réseau biomoléculaire de l'état normal à l'état souhaité (apoptose ou arrêt). Les résultats expérimentaux illustrent l'efficacité de cette approche dans l'optimisation de toutes les fonctions objectives. L'approche d'optimisation multi-objectifs proposée a permis de résoudre avec succès ce problème tout en fournissant un ensemble de solutions optimales au problème de la tranitabilité des réseaux biomoléculaires complexes. La solution optimale rendu à l'utilisateur est sélectionnée en fonction des préférences de l'utilisateur par la technique TOPSIS.

Nous avons comparé les résultats obtenus avec les travaux les plus représentatifs proposés pour le pilotage des réseaux biomoléculaires en utilisant la notion de contrôlabilité de la littérature. Nous notons que l'approche proposée (et qui est en accord avec les résultats de Wu et al. [6]) propose d'affecter l'ensemble minimal de nœuds (pas tous les nœuds) plutôt que l'ensemble des nœuds des réseaux biomoléculaires. Par conséquent, l'approche proposée fournit moins de stimuli externes que le nombre total de nœuds du réseau et, par conséquent, le coût total de ces stimuli sera certainement moins coûteux que l'utilisation de la notion de contrôlabilité. De plus, la méthode proposée considère également la minimisation de l'inconfort du patient comme un objectif primordial à atteindre. Cela accroît l'applicabilité de la médecine translationnelle pour améliorer la santé et les maladies humaines, y compris les facteurs génétiques et environnementaux du bien-être du patient. C'est une excellente occasion de comprendre les maladies et de trouver de nouveaux diagnostics et traitements.

Par conséquent, les applications de l'approche d'optimisation proposée peuvent être utilisées dans la conception de traitements médicamenteux tels que la chimiothérapie, l'identification de cibles médicamenteuses potentielles dans un réseau de signalisation du cancer humain ou l'étude des transitions phénotypiques (par exemple, pour diriger un réseau biomoléculaire depuis son phénotype anormal ou malade vers un phénotype sain).

Bien que les résultats de cette thèse soient satisfaisant et répondent à l'objectif attendu: le fait de pourvoir proposer une plate-forme qui permet aux biologistes de simuler les changements d'état des réseaux biomoléculaires dans le but de piloter leurs comportements et de les faire évoluer d'un état non désiré vers un état souhaitable, néanmoins un effort important reste à faire afin d'améliorer la qualité des services proposés par cette plate-forme.

En effet, les contributions présentées dans cette thèse ouvrent plusieurs orientations de recherche. Par exemple:

- Il sera intéressant d'étudier les interactions pouvant avoir lieu entre le niveau métabolique et le niveau génétique. Comme mentionné dans le sixième chapitre, la modélisation logique que nous proposons n'inclut pas les interactions entre les métabolites et les gènes, bien que l'on sache que certains métabolites spécifiques, comme la nicotine, interagissent directement avec les gènes. Une perspective possible à explorer est d'étendre la modélisation logique dans le but d'intégrer et de considérer cette hypothèse.

- Il sera intéressant de pouvoir étendre l'approche sémantique sur d'autres domaines proches de la transitabilité telque le domaine des médicaments biologiques afin de découvrir les différents types de molécules sur lesquelles agissent ces médicaments. Une question intéressante que nous étudions actuellement est le problème de l'extension de l'approche sémantique présentée dans cette thèse en l'appliquant à de grands réseaux biomoléculaires et en améliorant sa performance afin de l'appliquer dans le domaine de la découverte des médicaments biologiques. Ceci peut être résolu en alignant l'ontologie BNO avec d'autres ontologies et en les ajoutant à notre approche sémantique.

- Il sera intéressant d'étudier les différentes échelles de temps au sein du réseau biomoléulaire. En effet, les composantes du réseau biomoléculaire et leurs interactions sont organisées temporellement: de la nano-seconde au niveau des molécules individuelles, à la micro-seconde au niveau de l'interaction

des protéines, aux heures au niveau des processus cellulaires, etc. Dans ce travail, pour des raisons de simplicité, nous prenons en compte l'aspect multi-échelle du réseau, mais nous ne considérons pas les différentes échelles de temps biologiques. Nous négligeons cette propriété et travaillons à l'intérieur d'une échelle de temps fixe. Pour faire face à cette limite, nous devons améliorer notre modélisation en incluant un générateur de temps qui permet de suivre la dynamique du système et de changer entre différents niveaux de temps en fonction du niveau spatial donné et de la nature du processus.

- Il sera interessant d'améliorer l'optimisation de la transitabilité des réseaux biomoléculaires complexes. Cette dernière approche necessite d'améliorer et de détailler certaines fonctions objectives telle que l'inconfort du patient. Dans l'approche proposée la minimisation de l'inconfort du patient prends en compte divers crières tels que la minimisation des douleurs, la panique, etc. Nous souhaitons expliciter ce critère et le détailler afin de donner plus de clarté à la transitabilité du réseau. Nous prévoyons également d'améliorer les performances de l'algorithme NSGA-II en implémentant une hybridation avec un algorithme de recherche local, comme la méthode simplex de Nelder-Mead afin d'améliorer le temps de calcul de notre algorithme d'optimisation et la qualité des solutions.

- Il sera intéressant d'améliorer le prototype CBNSimulator. Cette amélioration portera sur la synchronisation des differents modules de la plate-forme. En effet, nous travaillons actuellement à rendre plus dynamique la synchronisation entre les quatre modules du CBNSimulator dans le but d'améliorer sa performance. En outre, nous prévoyons d'étendre les expériences afin d'évaluer et d'analyser plus en détail l'impact de nos différentes propositions sur des réseaux biomoléculaires plus complexes. En effet, les études de cas que nous avons utilisé ne suffisent pas pour évaluer et valider complètement nos approches, ainsi de nouvelles expériences pourraient être réalisées dans le cadre de l'application du CBNSimulateur. De plus, une amélioration des interfaces de la plate-forme sont nécessaires pour pouvoir la rendre plus simple et plus générique aux biologistes.

Bibliography

[1] Yann Collette and Patrick Siarry. *Optimisation multiobjectif: Algorithmes*. Editions Eyrolles, 2011.

[2] Johann Dréo, Alain Pétrowski, Patrick Siarry, and Eric Taillard. *Métaheuristiques pour l'optimisation difficile*. Eyrolles, 2003.

[3] Jean-Louis Le Moigne. *La théorie du système général: théorie de la modélisation*. jeanlouis le moigne-ae mcx, 1994.

[4] Sofie Van Landeghem, Thomas Van Parys, Marieke Dubois, Dirk Inzé, and Yves Van de Peer. Diffany: an ontology-driven framework to infer, visualise and analyse differential molecular networks. *BMC Bioinformatics*, 17(1):1–12, 2016.

[5] Ido Golding. Decision making in living cells: lessons from a simple system. *Annual review of biophysics*, 40:63–80, 2011.

[6] Fang-Xiang Wu, Lin Wu, Jianxin Wang, Juan Liu, and Luonan Chen. Transittability of complex networks and its applications to regulatory biomolecular networks. *Scientific reports*, 4, 2014.

[7] Céline Bérard. *Le processus de décision dans les systèmes complexes: une analyse d'une intervention systémique*. PhD thesis, Université du Québec à Montréal, 2009.

[8] Louise Travé-Massuyès. *Le raisonnement qualitatif pour les sciences de l'ingénieur (coll. diagnostic et maintenance)*. Hermes Science Publications, 1997.

[9] Clive G Bowsher, Margaritis Voliotis, and Peter S Swain. The fidelity of dynamic signaling by noisy biomolecular networks. *PLoS computational biology*, 9(3):e1002965, 2013.

[10] Georgios A Pavlopoulos, Anna-Lynn Wegener, and Reinhard Schneider. A survey of visualization tools for biological network analysis. *Biodata mining*, 1(1):12, 2008.

[11] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Peña, Julia K Goodrich, Jeffrey I Gordon, et al. Qiime allows analysis of high-throughput community sequencing data. *Nature methods*, 7(5):335–336, 2010.

[12] Vanessa Medina Villaamil, Guadalupe Aparicio Gallego, Isabel Santamarina Cainzos, Manuel Valladares-Ayerbes, and Luis M Antón Aparicio. State of the art in silico tools for the study of signaling pathways in cancer. *International journal of molecular sciences*, 13(6):6561–6581, 2012.

[13] Alan Aderem. Systems biology: its practice and challenges. *Cell*, 121(4):511–513, 2005.

[14] Gerald Karp. *Biologie cellulaire et moléculaire: Concepts and experiments*. De Boeck Supérieur, 2010.

[15] Alexander V Ratushny, Stephen A Ramsey, and John D Aitchison. Mathematical modeling of biomolecular network dynamics. *Network Biology: Methods and Applications*, pages 415–433, 2011.

[16] Daniel T Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of computational physics*, 22(4):403–434, 1976.

[17] Darren J Wilkinson. *Stochastic modelling for systems biology*. CRC press, 2011.

[18] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[19] Dirk Husmeier. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17):2271–2282, 2003.

[20] Claudine Chaouiya, Hanna Klaudel, and Franck Pommereau. A modular, qualitative modeling of regulatory networks using petri nets. In *Modeling in Systems Biology*, pages 253–279. Springer, 2011.

[21] Abdelghani Bellouquid and Marcello Delitala. *Mathematical modeling of complex biological systems*. Springer, 2006.

[22] Avi Ma'ayan. Introduction to network analysis in systems biology. *Science signaling*, 4(190):tr5, 2011.

[23] Khalid Raza and Rafat Parveen. Evolutionary algorithms in genetic regulatory networks model. *arXiv preprint arXiv:1205.1986*, 2012.

[24] Ali Najafi, Gholamreza Bidkhori, Joseph H Bozorgmehr, Ina Koch, and Ali Masoudi-Nejad. Genome scale modeling in systems biology: algorithms and resources. *Current genomics*, 15(2):130–159, 2014.

[25] Hans Peter Fischer. Mathematical modeling of complex biological systems: from parts lists to understanding systems behavior. *Alcohol Research & Health*, 31(1):49, 2008.

[26] Hidde De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103, 2002.

[27] Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I Glass, and Markus W Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401, 2012.

[28] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[29] C Zanni-Merk. Knowledge technologies for problem solving in engineering. *Mémoire d'Habilitation À Diriger des Recherches. Université de Strasbourg*, 2014.

[30] James D Watson, Francis HC Crick, et al. A structure for deoxyribose nucleic acid. *Nature*, 171(4356):737–738, 1953.

[31] Ralf Dahm. Friedrich miescher and the discovery of dna. *Developmental biology*, 278(2):274–288, 2005.

[32] Lewis J et al. Alberts B, Johnson A. The structure and function of dna. *Molecular Biology of the Cell*, (24), 2002.

[33] B Alberts, A Johnson, J Lewis, M Raff, K Roberts, and P Walter. Molecular biology of the cell, (garland science, new york, 2008). *Google Scholar*, page 652, 2002.

[34] Melissa J Moore and Nick J Proudfoot. Pre-mrna processing reaches back totranscription and ahead to translation. *Cell*, 136(4):688–700, 2009.

[35] Bruce M Alberts. The function of the hereditary materials: Biological catalyses reflect the cell's evolutionary history. *American Zoologist*, 26(3):781–796, 1986.

[36] Francisco J Iborra, Dean A Jackson, and Peter R Cook. Coupled transcription and translation within nuclei of mammalian cells. *Science*, 293(5532):1139–1142, 2001.

[37] Suzanne Clancy and William Brown. Translation: Dna to mrna to protein. *Nature Education*, 1(1):101, 2008.

[38] Claudia Manzoni, Demis A Kia, Jana Vandrovcova, John Hardy, Nicholas W Wood, Patrick A Lewis, and Raffaele Ferrari. Genome, transcriptome and proteome: the rise of omics data and their integration in biomedical sciences. *Briefings in bioinformatics*, page bbw114, 2016.

[39] Terence A Brown. Transcriptomes and proteomes. 2002.

[40] J Gregor Sutcliffe, Pamela E Foye, Mark G Erlander, Brian S Hilbush, Leon J Bodzin, Jayson T Durham, and Karl W Hasel. Toga: an automated parsing technology for analyzing expression of nearly all genes. *Proceedings of the National Academy of Sciences*, 97(5):1976–1981, 2000.

[41] Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

[42] Shao-En Ong, Leonard J Foster, and Matthias Mann. Mass spectrometric-based approaches in quantitative proteomics. *Methods*, 29(2):124–130, 2003.

[43] Miyako Kusano, Takayuki Tohge, Atsushi Fukushima, Makoto Kobayashi, Naomi Hayashi, Hitomi Otsuki, Youichi Kondou, Hiroto Goto, Mika Kawashima, Fumio Matsuda, et al. Metabolomics reveals comprehensive reprogramming involving two independent metabolic responses of arabidopsis to uv-b light. *The Plant Journal*, 67(2):354–369, 2011.

[44] Richard P Horgan and Louise C Kenny. 'omic'technologies: genomics, transcriptomics, proteomics and metabolomics. *The Obstetrician & Gynaecologist*, 13(3):189–195, 2011.

[45] Uwe Sauer, Matthias Heinemann, and Nicola Zamboni. Getting closer to the whole picture. *Science(Washington)*, 316(5824):550–551, 2007.

[46] Lesley T MacNeil and Albertha JM Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome research*, 21(5):645–657, 2011.

[47] Ariel Bensimon, Albert JR Heck, and Ruedi Aebersold. Mass spectrometry-based proteomics and network biology. *Annual review of biochemistry*, 81:379–405, 2012.

[48] Hiroaki Kitano et al. *Foundations of systems biology*. MIT press Cambridge, 2001.

[49] Georgios A Pavlopoulos, Anna-Lynn Wegener, and Reinhard Schneider. A survey of visualization tools for biological network analysis. *Biodata mining*, 1(1):1, 2008.

[50] Vanessa Medina Villaamil, Guadalupe Aparicio Gallego, Isabel Santamarina Cainzos, Manuel ValladaresAyerbes, and Luis M Antón Aparicio. State of the art in silico tools for the study of signaling pathways in cancer. *International journal of molecular sciences*, 13(6):6561–6581, 2012.

[51] Alexander J Gates and Luis M Rocha. Control of complex networks requires both structure and dynamics. *Scientific reports*, 6:24456, 2016.

[52] Cecilia Zanni-Merk, Stella Marc-Zwecker, Cédric Wemmert, and François de Bertrand de Beuvron. A layered architecture for a fuzzy semantic approach for satellite image analysis. *International Journal of Knowledge and Systems Science (IJKSS)*, 6(2):31–56, 2015.

[53] Hiroaki Kitano. Looking beyond the details: a rise in system-oriented approaches in genetics and molecular biology. *Current genetics*, 41(1):1–10, 2002.

[54] Daniel Machado, Rafael S Costa, Miguel Rocha, Eugénio C Ferreira, Bruce Tidor, and Isabel Rocha. Modeling formalisms in systems biology. *AMB express*, 1(1):45, 2011.

[55] René Thomas and Richard d'Ari. *Biological feedback*. CRC press, 1990.

[56] Stuart A Kauffman. The origins of order: Self-organization and selection in evolution. In *Spin Glasses and Biology*, pages 61–100. World Scientific, 1992.

[57] A Stéphanou and Vitaly Volpert. Hybrid modelling in biology: a classification review. *Mathematical Modelling of Natural Phenomena*, 11(1):37–48, 2016.

[58] Assieh Saadatpour and Réka Albert. A comparative study of qualitative and quantitative dynamic models of biological regulatory networks. *EPJ Nonlinear Biomedical Physics*, 4(1):1–13, 2016.

[59] Geoffrey Koh, David Hsu, and PS Thiagarajan. Component-based construction of bio-pathway models: The parameter estimation problem. *Theoretical Computer Science*, 412(26):2840–2853, 2011.

[60] Luis Mendoza, Denis Thieffry, and Elena R Alvarez-Buylla. Genetic control of flower morphogenesis in arabidopsis thaliana: a logical analysis. *Bioinformatics (Oxford, England)*, 15(7):593–606, 1999.

[61] Lucas Sánchez, Jacques van Helden, and Denis Thieffry. Establishement of the dorso-ventral pattern during embryonic development ofdrosophila melanogaster: a logical analysis. *Journal of theoretical biology*, 189(4):377–389, 1997.

[62] Clare E Giacomantonio and Geoffrey J Goodhill. A boolean model of the gene regulatory network underlying mammalian cortical area development. *PLoS computational biology*, 6(9):e1000936, 2010.

[63] Rebekka Schlatter, Kathrin Schmich, Ima Avalos Vizcarra, Peter Scheurich, Thomas Sauter, Christoph Borner, Michael Ederer, Irmgard Merfort, and Oliver Sawodny. On/off and beyond-a boolean model of apoptosis. *PLoS computational biology*, 5(12):e1000595, 2009.

[64] Song Li, Sarah M Assmann, and Réka Albert. Predicting essential components of signal transduction networks: a dynamic model of guard cell abscisic acid signaling. *PLoS biology*, 4(10):e312, 2006.

[65] Ranran Zhang, Mithun Vinod Shah, Jun Yang, Susan B Nyland, Xin Liu, Jong K Yun, Réka Albert, and Thomas P Loughran. Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences*, 105(42):16308–16313, 2008.

[66] René Thomas. Regulatory networks seen as asynchronous automata: a logical description. *Journal of theoretical biology*, 153(1):1–23, 1991.

[67] Melody K Morris, Julio Saez-Rodriguez, Peter K Sorger, and Douglas A Lauffenburger. Logic-based models for the analysis of cell signaling networks. *Biochemistry*, 49(15):3216–3224, 2010.

[68] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, et al. A logical model provides insights into t cell receptor signaling. *PLoS computational biology*, 3(8):e163, 2007.

[69] Regina Samaga, Julio Saez-Rodriguez, Leonidas G Alexopoulos, Peter K Sorger, and Steffen Klamt. The logic of egfr/erbb signaling: theoretical properties and analysis of high-throughput data. *PLoS computational biology*, 5(8):e1000438, 2009.

[70] Julio Saez-Rodriguez, Leonidas G Alexopoulos, MingSheng Zhang, Melody K Morris, Douglas A Lauffenburger, and Peter K Sorger. Comparing signaling networks between normal and transformed hepatocytes using discrete logical models. *Cancer research*, 71(16):5400–5411, 2011.

[71] Laurence Calzone, Laurent Tournier, Simon Fourquet, Denis Thieffry, Boris Zhivotovsky, Emmanuel Barillot, and Andrei Zinovyev. Mathematical modelling of cell-fate decision in response to death receptor engagement. *PLoS computational biology*, 6(3):e1000702, 2010.

[72] L Farinas de Cerro and K Inoue. Logical modeling of biological systems, 2014.

[73] Namrata Tomar, Olivia Choudhury, Ankush Chakrabarty, and Rajat K De. An integrated pathway system modeling of saccharomyces cerevisiae hog pathway: a petri net based approach. *Molecular biology reports*, 40(2):1103–1125, 2013.

[74] Maurizio Adriano Strangio. Graph-based exploratory analysis of biological interaction networks. In *Advanced Technologies*. InTech, 2009.

[75] Derek Ruths, Melissa Muller, Jen-Te Tseng, Luay Nakhleh, and Prahlad T Ram. The signaling petri net-based simulator: a non-parametric strategy for characterizing the dynamics of cell-specific signaling networks. *PLoS computational biology*, 4(2):e1000005, 2008.

[76] Ina Koch, Björn H Junker, and Monika Heiner. Application of petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber. *Bioinformatics*, 21(7):1219–1226, 2004.

[77] Hartmann Genrich, Robert Küffner, and Klaus Voss. Executable petri net models for the analysis of metabolic pathways. *International Journal on Software Tools for Technology Transfer (STTT)*, 3(4):394–404, 2001.

[78] Venkatramana N Reddy, Michael N Liebman, and Michael L Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Computers in biology and medicine*, 26(1):9–24, 1996.

[79] Claudine Chaouiya. Petri net modelling of biological networks. *Briefings in bioinformatics*, 8(4):210–219, 2007.

[80] Lingxi Li and Hiroki Yokota. Application of petri nets in bone remodeling. *Gene regulation and systems biology*, 3:105, 2009.

[81] Simon Hardy and Pierre N Robillard. Modeling and simulation of molecular biology systems using petri nets: modeling goals of various approaches. *Journal of bioinformatics and computational biology*, 2(04):619–637, 2004.

[82] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[83] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[84] David Heckerman. Bayesian networks for data mining. *Data mining and knowledge discovery*, 1(1):79–119, 1997.

[85] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational intelligence*, 5(2):142–150, 1989.

[86] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[87] David Orlando, Charles Y Lin, Allister Bernard, Edwin S Iversen, Alexander J Hartemink, and Steven B Haase. A probabilistic model for cell cycle distributions in synchrony experiments. *Cell Cycle*, 6(4):478–488, 2007.

[88] Amira Djebbari and John Quackenbush. Seeded bayesian networks: constructing genetic networks from microarray data. *BMC systems biology*, 2(1):57, 2008.

[89] Nicole Krämer, Juliane Schäfer, and Anne-Laure Boulesteix. Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC bioinformatics*, 10(1):384, 2009.

[90] David Edwards. *Introduction to graphical modelling*. Springer Science & Business Media, 2012.

[91] Nicolas Verzelen and Fanny Villers. Tests for gaussian graphical models. *Computational Statistics & Data Analysis*, 53(5):1894–1905, 2009.

[92] Hiroyuki Toh and Katsuhisa Horimoto. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.

[93] Shisong Ma, Qingqiu Gong, and Hans J Bohnert. An arabidopsis gene network based on the graphical gaussian model. *Genome research*, 17(11):1614–1625, 2007.

[94] Anja Wille, Philip Zimmermann, Eva Vranová, Andreas Fürholz, Oliver Laule, Stefan Bleuler, Lars Hennig, Amela Prelić, Peter von Rohr, Lothar Thiele, et al. Sparse graphical gaussian modeling of the isoprenoid gene network in arabidopsis thaliana. *Genome biology*, 5(11):R92, 2004.

[95] Daniel E Zak, Gregory E Gonye, James S Schwaber, and Francis J Doyle. Importance of input perturbations and stochastic gene expression in the reverse engineering of genetic regulatory networks: insights from an identifiability analysis of an in silico network. *Genome research*, 13(11):2396–2405, 2003.

[96] Hidde De Jong, Jean-Luc Gouzé, Céline Hernandez, Michel Page, Tewfik Sari, and Johannes Geiselmann. Hybrid modeling and simulation of genetic regulatory networks: A qualitative approach. In *International Workshop on Hybrid Systems: Computation and Control*, pages 267–282. Springer, 2003.

[97] Grégory Batt, Damien Bergamini, Hidde De Jong, Hubert Garavel, and Radu Mateescu. Model checking genetic regulatory networks using gna and cadp. In *SPIN*, volume 2989, pages 158–163. Springer, 2004.

[98] FILIPA ALVES and RUI DILÃO. A software tool to model genetic regulatory networks: applications to segmental patterning in drosophila. In *BIOMAT 2005*, pages 71–88. World Scientific, 2006.

[99] Riccardo Porreca, Samuel Drulhe, Hidde de Jong, and Giancarlo Ferrari-Trecate. Structural identification of piecewise-linear models of genetic regulatory networks. *Journal of Computational Biology*, 15(10):1365–1380, 2008.

[100] Hidde De Jong and Michel Page. Search for steady states of piecewise-linear differential equation models of genetic regulatory networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):208–222, 2008.

[101] Rui Dilão and Daniele Muraro. A software tool to model genetic regulatory networks. applications to the modeling of threshold phenomena and of spatial patterning in drosophila. *PloS one*, 5(5):e10743, 2010.

[102] William S Hlavacek and Michael A Savageau. Rules for coupled expression of regulator and effector genes in inducible circuits. *Journal of molecular biology*, 255(1):121–139, 1996.

[103] John J Tyson and Bela Novak. Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions. *Journal of theoretical biology*, 210(2):249–263, 2001.

[104] Jean-Christophe Leloup, Albert Goldbeter, et al. Modeling the molecular regulatory mechanism of circadian rhythms in drosophila. *BioEssays*, 22(1):84, 2000.

[105] Hans Meinhardt and Alfred Gierer. Pattern formation by local self-activation and lateral inhibition. *Bioessays*, 22(8):753–760, 2000.

[106] John Reinitz, David Kosman, Carlos E Vanario-Alonso, David H Sharp, et al. Stripe forming architecture of the gap gene system. *Developmental Genetics*, 23(1):11–27, 1998.

[107] George Von Dassow, Eli Meir, Edwin M Munro, and Garrett M Odell. The segment polarity network is a robust developmental module. *Nature*, 406(6792):188–192, 2000.

[108] Delphine Ropers, Hidde De Jong, Michel Page, Dominique Schneider, and Johannes Geiselmann. Qualitative simulation of the carbon starvation response in escherichia coli. *Biosystems*, 84(2):124–152, 2006.

[109] Ping Ao, Lik Wee Lee, Mary E Lidstrom, Lan Yin, and Xiaomei Zhu. Towards kinetic modeling of global metabolic networks: Methylobacterium extorquens am1 growth as validation. *Chinese Journal of Biotechnology*, 24(6):980–994, 2008.

[110] Andreas Dräger, Andreas Zell, Hannes Planatscher, Jørgen B Magnus, Jochen Supper, Marcel Kronfeld, Marco Oldiges, Michael J Ziller, and Oliver Kohlbacher. Modeling metabolic networks in c. glutamicum: a comparison of rate laws in combination with various parameter optimization strategies. *BMC Systems Biology*, 3(1):5, 2009.

[111] Anand R Asthagiri and Douglas A Lauffenburger. A computational study of feedback effects on signal dynamics in a mitogen-activated protein kinase (mapk) pathway model. *Biotechnology progress*, 17(2):227–239, 2001.

[112] Bernd Binder and Reinhart Heinrich. Interrelations between dynamical properties and structural characteristics of signal transduction networks. *Genome Informatics*, 15(1):13–23, 2004.

[113] Alexander Y Mitrophanov, Gordon Churchward, and Mark Borodovsky. Control of streptococcus pyogenes virulence: modeling of the covr/s signal transduction system. *Journal of theoretical biology*, 246(1):113–128, 2007.

[114] John Von Neumann and Arthur Walter Burks. *Theory of self-reproducing automata*. University of Illinois Press Urbana, 1996.

[115] Stephen Wolfram. *A new kind of science*, volume 5. Wolfram media Champaign, 2002.

[116] Rob J de Boer, Jan D van der Laan, and Pauline Hogeweg. Randomness and pattern scale in the immune network. 1992.

[117] Hugo de Garis. "cam-brain∝ atr's billion neuron artificial brain project a three year progress report. In *World Wisepersons Workshop*, pages 215–243. Springer, 1995.

[118] Lemont B Kier, C-K Cheng, Bernard Testa, and Pierre-Alain Carrupt. A cellular automata model of enzyme kinetics. *Journal of molecular graphics*, 14(4):227–231, 1996.

[119] Advait Apte, Danail Bonchev, and Stephen Fong. Cellular automata modeling of fasl-initiated apoptosis. *Chemistry & biodiversity*, 7(5):1163–1172, 2010.

[120] Steven C Bankes. Agent-based modeling: A revolution? *Proceedings of the National Academy of Sciences*, 99(suppl 3):7199–7200, 2002.

[121] Gary An, Qi Mi, Joyeeta Dutta-Moscato, and Yoram Vodovotz. Agent-based models in translational systems biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(2):159–171, 2009.

[122] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

[123] Michael Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[124] Mark Pogson, Rod Smallwood, Eva Qwarnstrom, and Mike Holcombe. Formal agent-based modelling of intracellular chemical interactions. *Biosystems*, 85(1):37–45, 2006.

[125] Gary An. A model of tlr4 signaling and tolerance using a qualitative, particle–event-based method: Introduction of spatially configured stochastic reaction chambers (scsrc). *Mathematical biosciences*, 217(1):43–52, 2009.

[126] Matthew B Biggs and Jason A Papin. Novel multiscale modeling tool applied to pseudomonas aeruginosa biofilm formation. *PLoS One*, 8(10):e78011, 2013.

[127] Drew Endy and Roger Brent. Modelling cellular behaviour. *Nature*, 409(6818):391–395, 2001.

[128] Julio Collado-Vides and Ralf Hofestädt. *Gene regulation and metabolism: postgenomic computational approaches*. MIT Press, 2004.

[129] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[130] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[131] Guy Pierra, Hondjack Dehainsala, Yamine Ait Ameur, and Ladjel Bellatreche. Bases de données à base ontologique. principe et mise en oeuvre. *Ingénierie des systemes d'information*, 10(2):91–115, 2005.

[132] Asunción Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update: Knowledge Based Systems and Applied Artificial Intelligence*, 2(3):33–43, 1999.

[133] Nicola Guarino. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3):293–310, 1997.

[134] Riichiro Mizoguchi. A step towards ontological engineering. In *12th National Conference on AI of JSAI*, pages 24–31, 1998.

[135] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[136] Stefan Schulz, Elena Beisswanger, Joachim Wermter, and Udo Hahn. Towards an upper-level ontology for molecular biology. In *AMIA Annual Symposium Proceedings*, volume 2006, page 694. American Medical Informatics Association, 2006.

[137] Riichiro Mizoguchi. Part 1: introduction to ontological engineering. *New generation computing*, 21(4):365–384, 2003.

[138] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2):93–136, 1996.

[139] Riichiro Mizoguchi. Tutorial on ontological engineering part 2: Ontology development, tools and languages. *New Generation Computing*, 22(1):61–96, 2004.

[140] Mariano Fernández-López. Overview of methodologies for building ontologies. 1999.

[141] Michael Uschold and Martin King. *Towards a methodology for building ontologies*. Artificial Intelligence Applications Institute, University of Edinburgh Edinburgh, 1995.

[142] Bill Swartout, Ramesh Patil, Kevin Knight, and Tom Russ. Toward distributed use of large-scale ontologies. In *Proc. of the Tenth Workshop on Knowledge Acquisition for Knowledge-Based Systems*, pages 138–148, 1996.

[143] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[144] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[145] V Maniraj and R Sivakumar. Ontology languages-a review. *International Journal of Computer Theory and Engineering*, 2(6):887, 2010.

[146] Patrick Hayes and Christopher Menzel. A semantics for the knowledge interchange format. In *IJCAI 2001 Workshop on the IEEE Standard Upper Ontology*, volume 1, page 145, 2001.

[147] William A Woods and James G Schmolze. The kl-one family. *Computers & Mathematics with Applications*, 23(2-5):133–177, 1992.

[148] Franz Baader. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[149] Grigoris Antoniou and Frank Van Harmelen. Web ontology language: Owl. In *Handbook on ontologies*, pages 67–92. Springer, 2004.

[150] J Hendler. Daml: The darpa agent markup language homepage. *Retrieved July*, 6, 2001.

[151] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[152] Dan Brickley and Ramanathan V Guha. Rdf vocabulary description language 1.0: Rdf schema. 2004.

[153] Chris Welty, Deborah L McGuinness, and Michael K Smith. Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www. w3. org/TR/2004/REC-owl-guide-20040210*, 2004.

[154] Ian Horrocks. Owl: A description logic based ontology language. In *ICLP*, volume 3668, pages 1–4. Springer, 2005.

[155] Marvin Minsky. A framework for representing knowledge. 1974.

[156] Hondjack Dehainsala, Guy Pierra, Ladjel Bellatreche, and Y Aït-Ameur. Conception de bases de données à partir d'ontologies de domaine: Application aux bases de données du domaine technique. *Actes des 1ère Journées Francophones sur les Ontologies (JFO'07)*, pages 215–230, 2007.

[157] Chimene Fankam. Ontodb2: support of multiple ontology models within ontology based database. In *Proceedings of the 2008 EDBT Ph. D. workshop*, pages 21–27. ACM, 2008.

[158] Bruno Bachimont. Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en ingénierie des connaissances. *Ingénierie des connaissances: évolutions récentes et nouveaux défis*, pages 305–323, 2000.

[159] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

[160] Duhwan Mun and Karthik Ramani. Knowledge-based part similarity measurement utilizing ontology and multi-criteria decision making technique. *Advanced Engineering Informatics*, 25(2):119–130, 2011.

[161] Visit Hirankitti and T Xuan. A meta-reasoning approach for reasoning with swrl ontologies. In *Internation Multiconference of Engineers*, 2011.

[162] Michael Leibman, Ruth A Bruer, and Bill R Maki. Succession management: The next generation of succession planning. *People and Strategy*, 19(3):16, 1996.

[163] Julia Tzu-Ya Weng, Li-Ching Wu, Wen-Chi Chang, Tzu-Hao Chang, Tatsuya Akutsu, and Tzong-Yi Lee. Novel bioinformatics approaches for analysis of high-throughput biological data. *BioMed research international*, 2014, 2014.

[164] Chaimaa Messaoudi, Rachida Fissoune, and Hassan Badir. A survey of semantic integration approaches in bioinformatics. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(12):2001–2006, 2016.

[165] Daniel L Rubin, Suzanna E Lewis, Chris J Mungall, Sima Misra, Monte Westerfield, Michael Ashburner, Ida Sim, Christopher G Chute, Margaret-Anne Storey, Barry Smith, et al. National center for biomedical ontology: advancing biomedicine through structured organization of scientific knowledge. *Omics: a journal of integrative biology*, 10(2):185–198, 2006.

[166] Ray W Fergerson, Paul R Alexander, Michael Dorf, Rafael S Gonçalves, Manuel Salvadores, Alex Skrenchuk, Jennifer Vendetti, and Mark A Musen. Ncbo bioportal version 4. 2015.

[167] John Hancock. Editorial: Biological ontologies and semantic biology. 5:18, 02 2014.

[168] Jonathan Bard, Seung Y Rhee, and Michael Ashburner. An ontology for cell types. *Genome biology*, 6(2):1, 2005.

[169] Stefanie Seltmann, Harald Stachelscheid, Alexander Damaschun, Ludger Jansen, Fritz Lekschas, Jean-Fred Fontaine, Throng Nghia Nguyen-Dobinsky, Ulf Leser, and Andreas Kurtz. Celda - an ontology for the comprehensive representation of cells in complex systems. *BMC bioinformatics*, 14(1):228, 2013.

[170] Midori A Harris, Jennifer I Deegan, Jane Lomax, Michael Ashburner, Susan Tweedie, Seth Carbon, Suzanna Lewis, Chris Mungall, John Day-Richter, Karen Eilbeck, et al. The gene ontology project in 2008. *Nucleic Acids Res*, 36:D440–D444, 2008.

[171] Purvesh Khatri and Sorin Drăghici. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595, 2005.

[172] Darren A Natale, Cecilia N Arighi, Winona C Barker, Judith A Blake, Carol J Bult, Michael Caudy, Harold J Drabkin, Peter D'Eustachio, Alexei V Evsikov, Hongzhan Huang, et al. The protein ontology: a structured representation of protein forms and complexes. *Nucleic acids research*, 39(suppl_1):D539–D545, 2010.

[173] Nick Juty and Nicolas le Novère. Systems biology ontology. *Encyclopedia of Systems Biology*, pages 2063–2063, 2013.

[174] Emek Demir, Michael P Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'eustachio, Carl Schaefer, Joanne Luciano, et al. The biopax community standard for pathway data sharing. *Nature biotechnology*, 28(9):935–942, 2010.

[175] Jonathan BL Bard. The aeo, an ontology of anatomical entities for classifying animal tissues and organs. *Frontiers in genetics*, 3, 2012.

[176] Cynthia L Smith and Janan T Eppig. The mammalian phenotype ontology: enabling robust annotation and comparative analysis. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 1(3):390–399, 2009.

[177] Christopher J Mungall, Georgios V Gkoutos, Cynthia L Smith, Melissa A Haendel, Suzanna E Lewis, and Michael Ashburner. Integrating phenotype ontologies across multiple species. *Genome biology*, 11(1):R2, 2010.

[178] Sebastian Köhler, Nicole A Vasilevsky, Mark Engelstad, Erin Foster, Julie McMurry, Ségolène Aymé, Gareth Baynam, Susan M Bello, Cornelius F Boerkoel, Kym M Boycott, et al. The human phenotype ontology in 2017. *Nucleic acids research*, 45(D1):D865–D876, 2017.

[179] James P Sluka, Abbas Shirinifard, Maciej Swat, Alin Cosmanescu, Randy W Heiland, and James A Glazier. The cell behavior ontology: describing the intrinsic biological behaviors of real and model cells seen as active agents. *Bioinformatics*, page btu210, 2014.

[180] Margaritis Voliotis, Philipp Thomas, Ramon Grima, and Clive G Bowsher. Stochastic simulation of biomolecular networks in dynamic environments. *PLoS computational biology*, 12(6):e1004923, 2016.

[181] Banks Jerry. *Discrete-event system simulation*. Pearson Education India, 1984.

[182] J Richard Harrison. The concept of simulation in organizational research, 1999.

[183] Robert Axelrod. Advancing the art of simulation in the social sciences. In *Simulating social phenomena*, pages 21–40. Springer, 1997.

[184] JP Lebacque. A finite acceleration scheme for first order macroscopic traffic flow models. *IFAC Proceedings Volumes*, 30(8):787–792, 1997.

[185] Ferdinando Semboloni, Jürgen Assfalg, Saverio Armeni, Roberto Gianassi, and Francesco Marsoni. Citydev, an interactive multi-agents urban model on the web. *Computers, Environment and Urban Systems*, 28(1-2):45–64, 2004.

[186] Kai Nagel, Martin Pieck, Patrice M Simon, and Marcus Rickert. Comparison between three different traffic micro-simulations and reality in dallas. Technical report, Los Alamos National Lab., NM (United States), 1998.

[187] Lawrence B Evans. Bioprocess simulation: a new tool for process development. *Nature Biotechnology*, 6(2):200–203, 1988.

[188] Demetri P Petrides. Biopro designer: an advanced computing environment for modeling and design of integrated biochemical processes. *Computers & chemical engineering*, 18:S621–S625, 1994.

[189] J Makinia, M Swinarski, and E Dobiegala. Experiences with computer simulation at two large wastewater treatment plants in northern poland. *Water science and technology*, 45(6):209–218, 2002.

[190] Bruce A Barshop, Richard F Wrenn, and Carl Frieden. Analysis of numerical methods for computer simulation of kinetic processes: development of kinsim—a flexible, portable system. *Analytical biochemistry*, 130(1):134–145, 1983.

[191] Athel Cornish-Bowden and Jan-Hendrik S Hofmeyr. Metamodel: a program for modelling and control analysis of metabolic pathways on the ibm pc and compatibles. *Bioinformatics*, 7(1):89–93, 1991.

[192] Herbert M Sauro. Scamp: a general-purpose simulator and metabolic control analysis program. *Bioinformatics*, 9(4):441–450, 1993.

[193] Magnus Ehlde and Guido Zacchi. Mist: a user-friendly metabolic simulator. *Bioinformatics*, 11(2):201–207, 1995.

[194] Pedro Mendes. Gepasi: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Bioinformatics*, 9(5):563–571, 1993.

[195] Igor Goryanin, T Charles Hodgman, and Evgeni Selkov. Mathematical simulation and analysis of cellular metabolism and regulation. *Bioinformatics (Oxford, England)*, 15(9):749–758, 1999.

[196] Martin Ginkel, Andreas Kremling, Torsten Nutsch, Robert Rehner, and Ernst Dieter Gilles. Modular modeling of cellular systems with promot/diva. *Bioinformatics*, 19(9):1169–1176, 2003.

[197] Thomas D Garvey, Patrick Lincoln, Charles John Pedersen, David Martin, and Mark Johnson. Biospice: access to the most current computational tools for biologists. *OMICS A Journal of Integrative Biology*, 7(4):411–420, 2003.

[198] Dong-Yup Lee, Hongsoek Yun, Sunwon Park, and Sang Yup Lee. Metafluxnet: the management of metabolic reaction information and quantitative metabolic flux analysis. *Bioinformatics*, 19(16):2144–2146, 2003.

[199] Susumu Goto, Yasushi Okuno, Masahiro Hattori, Takaaki Nishioka, and Minoru Kanehisa. Ligand: database of chemical compounds and reactions in biological pathways. *Nucleic acids research*, 30(1):402–404, 2002.

[200] Peter D Karp, Monica Riley, Milton Saier, Ian T Paulsen, Suzanne M Paley, and Alida Pellegrini-Toole. The ecocyc and metacyc databases. *Nucleic acids research*, 28(1):56–59, 2000.

[201] Andrzej M Kierzek. Stocks: Stochastic kinetic simulations of biochemical systems with gillespie algorithm. *Bioinformatics*, 18(3):470–481, 2002.

[202] Nicolas Le Novere and Thomas Simon Shimizu. Stochsim: modelling of stochastic biomolecular processes. *Bioinformatics*, 17(6):575–576, 2001.

[203] Thomas Simon Shimizu and Dennis Bray. Modelling the bacterial chemotaxis receptor complex. In *Novartis Found Symp*, volume 247, pages 162–77, 2003.

[204] G Bard Ermentrout and Leah Edelstein-Keshet. Cellular automata approaches to biological modeling. *Journal of theoretical Biology*, 160(1):97–133, 1993.

[205] Andreas Deutsch and Sabine Dormann. *Cellular automaton modeling of biological pattern formation: characterization, applications, and analysis*. Springer Science & Business Media, 2007.

[206] John Von Neumann, Arthur W Burks, et al. Theory of self-reproducing automata. *IEEE Transactions on Neural Networks*, 5(1):3–14, 1966.

[207] Martin Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970.

[208] Jean-Louis Giavitto and Olivier Michel. Modeling the topological organization of cellular processes. *BioSystems*, 70(2):149–163, 2003.

[209] Jean-Louis Giavitto, Grant Malcolm, and Olivier Michel. Rewriting systems and the modelling of biological systems. *Comparative and Functional Genomics*, 5(1):95–99, 2004.

[210] Nicholas J Savill and Paulien Hogeweg. Modelling morphogenesis: from single cells to crawling slugs. *Journal of Theoretical Biology*, 184(3):229–235, 1997.

[211] Athanasius FM Marée and Paulien Hogeweg. How amoeboids self-organize into a fruiting body: multicellular coordination in dictyostelium discoideum. *Proceedings of the National Academy of Sciences*, 98(7):3879–3883, 2001.

[212] P Hogeweg. Evolving mechanisms of morphogenesis: on the interplay between differential adhesion and cell differentiation. *Journal of Theoretical Biology*, 203(4):317–333, 2000.

[213] Mark Zajac, Gerald L Jones, and James A Glazier. Model of convergent extension in animal morphogenesis. *Physical Review Letters*, 85(9):2022, 2000.

[214] Volker Grimm. Ten years of individual-based modelling in ecology: what have we learned and what could we learn in the future? *Ecological modelling*, 115(2):129–148, 1999.

[215] Frédéric Amblard and Denis Phan. *Modélisation et simulation multi-agents: Applications pour les Sciences de l'Homme et de la Société*. Hermes science publ., 2006.

[216] Francesco Amigoni and Viola Schiaffonati. Multiagent-based simulation in biology. *Model-Based Reasoning in Science, Technology, and Medicine*, pages 179–191, 2007.

[217] Vincent Ginot and Christophe Le Page. Mobidyc, a generic multi-agents simulator for modeling populations dynamics. *Tasks and methods in applied artificial intelligence*, pages 805–814, 1998.

[218] Timothée Brochier, Jean Marc Ecoutin, Luis Tito de Morais, David M Kaplan, and Raymond Lae. A multi-agent ecosystem model for studying changes in a tropical estuarine fish assemblage within a marine protected area. *Aquatic Living Resources*, 26(2):147–158, 2013.

[219] Sorin Ilie and Costin Bădică. Multi-agent distributed framework for swarm intelligence. *Procedia Computer Science*, 18:611–620, 2013.

[220] Luca Bortolussi, Agostino Dovier, and Federico Fogolari. Multi-agent simulation of protein folding. In *Proceedings of the first international workshop on multi-agent systems for medicine, computational biology, and bioinformatics*, 2005.

[221] Stefania Bandini, Sara Manzoni, and Giuseppe Vizzari. Immune system modelling with situated cellular agents. In *First International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics*, page 78, 2005.

[222] Tibor Bosse, Catholijn M Jonker, and Jan Treur. Modelling the dynamics of intracellular processes as an organisation of multiple agents. *MAS-BIOMED*, 5:107–121, 2005.

[223] Renfrey Burnard Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press, 1952.

[224] James A Glazier and François Graner. Simulation of the differential adhesion driven rearrangement of biological cells. *Physical Review E*, 47(3):2128, 1993.

[225] J Hardy, O De Pazzis, and Yves Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical review A*, 13(5):1949, 1976.

[226] Uriel Frisch, Brosl Hasslacher, and Yves Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical review letters*, 56(14):1505, 1986.

[227] Dieter A Wolf-Gladrow. *Lattice-gas cellular automata and lattice Boltzmann models: an introduction*. Springer, 2004.

[228] Rafik Ouared, Bastien Chopard, Bernd Stahl, Daniel A Rüfenacht, Hasan Yilmaz, and Guy Courbebaisse. Thrombosis modeling in intracranial aneurysms: a lattice boltzmann numerical algorithm. *Computer Physics Communications*, 179(1):128–131, 2008.

[229] B Chopard, R Ouared, DA Ruefenacht, and H Yilmaz. Lattice boltzmann modeling of thrombosis in giant aneurysms. *International Journal of Modern Physics C*, 18(04):712–721, 2007.

[230] Haralambos Hatzikirou, Lutz Brusch, C Schaller, M Simon, and Andreas Deutsch. Prediction of traveling front behavior in a lattice-gas cellular automaton model for tumor invasion. *Computers & Mathematics with Applications*, 59(7):2326–2339, 2010.

[231] Michael Hucka, Andrew Finney, Herbert M Sauro, Hamid Bolouri, John C Doyle, Hiroaki Kitano, Adam P Arkin, Benjamin J Bornstein, Dennis Bray, Athel Cornish-Bowden, et al. The systems biology markup language (sbml): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[232] Michael Hucka, Frank T Bergmann, Stefan Hoops, Sarah M Keating, Sven Sahle, James C Schaff, Lucian P Smith, and Darren J Wilkinson. The systems biology markup language (sbml): language specification for level 3 version 1 core. *Journal of integrative bioinformatics*, 12(2):266, 2015.

[233] Warren Hedley, Poul Nielsen, and Peter Hunter. Xml languages for describing biological models and data. *Annals of Biomedical Engineering*, 28, 2000.

[234] Katsuyuki Yugi and Masaru Tomita. A general computational model of mitochondrial metabolism in a whole organelle scale. *Bioinformatics*, 20(11):1795–1796, 2004.

[235] Ayako Yachie-Kinoshita, Taiko Nishino, Hanae Shimo, Makoto Suematsu, and Masaru Tomita. A metabolic model of human erythrocytes: practical application of the e-cell simulation environment. *BioMed Research International*, 2010, 2010.

[236] Paul Smolen, Paul E Hardin, Brian S Lo, Douglas A Baxter, and John H Byrne. Simulation of drosophila circadian oscillations, mutations, and light responses by a model with vri, pdp-1, and clk. *Biophysical journal*, 86(5):2786–2802, 2004.

[237] Jesús A Izaguirre, Rajiv Chaturvedi, Chengbang Huang, Trevor Cickovski, J Coffland, G Thomas, Gabor Forgacs, M Alber, G Hentschel, Stuart A Newman, et al. Compucell, a multi-model framework for simulation of morphogenesis. *Bioinformatics*, 20(7):1129–1137, 2004.

[238] François Graner and James A Glazier. Simulation of biological cell sorting using a two-dimensional extended potts model. *Physical review letters*, 69(13):2013, 1992.

[239] Nikodem J Popławski, Abbas Shirinifard, Maciej Swat, and James A Glazier. Simulation of single-species bacterial-biofilm growth using the glazier-graner-hogeweg model and the compucell3d modeling environment. *Mathematical biosciences and engineering: MBE*, 5(2):355, 2008.

[240] Bakhtier Vasiev, Ariel Balter, Mark Chaplain, James A Glazier, and Cornelis J Weijer. Modeling gastrulation in the chick embryo: formation of the primitive streak. *PLoS One*, 5(5):e10571, 2010.

[241] Susan D Hester, Julio M Belmonte, J Scott Gens, Sherry G Clendenon, and James A Glazier. A multi-cell, multi-scale model of vertebrate segmentation and somite formation. *PLoS computational biology*, 7(10):e1002155, 2011.

[242] Ana S Dias, Irene de Almeida, Julio M Belmonte, James A Glazier, and Claudio D Stern. Somites without a clock. *Science*, 343(6172):791–795, 2014.

[243] Maciej H Swat, Gilberto L Thomas, Julio M Belmonte, Abbas Shirinifard, Dimitrij Hmeljak, and James A Glazier. Multi-scale modeling of tissues using compucell3d. *Methods in cell biology*, 110:325, 2012.

[244] David S Wishart, Robert Yang, David Arndt, Peter Tang, and Joseph Cruz. Dynamic cellular automata: an alternative approach to cellular simulation. *In silico biology*, 5(2):139–161, 2005.

[245] James Schaff, Charles C Fink, Boris Slepchenko, John H Carson, and Leslie M Loew. A general computational framework for modeling cellular structure and function. *Biophysical journal*, 73(3):1135–1146, 1997.

[246] Jonathon A Ditlev, Nathaniel M Vacanti, Igor L Novak, and Leslie M Loew. An open model of actin dendritic nucleation. *Biophysical journal*, 96(9):3529–3542, 2009.

[247] Sherry-Ann Brown, Ion I Moraru, James C Schaff, and Leslie M Loew. Virtual neuron: a strategy for merged biochemical and electrophysiological modeling. *Journal of computational neuroscience*, 31(2):385–400, 2011.

[248] Chungman Seo and Bernard P Zeigler. Interoperability between devs simulators using service oriented architecture and devs namespace. In *Proceedings of the 2009 Spring Simulation Multiconference*, page 157. Society for Computer Simulation International, 2009.

[249] J Jaime Caro and Jörgen Möller. Advantages and disadvantages of discrete-event simulation for health economic analyses, 2016.

[250] J Jaime Caro. Pharmacoeconomic analyses using discrete event simulation. *Pharmacoeconomics*, 23(4):323–332, 2005.

[251] Jonathan Karnon, James Stahl, Alan Brennan, J Jaime Caro, Javier Mar, and Jörgen Möller. Modeling using discrete event simulation: a report of the ispor-smdm modeling good research practices task force-4. *Value in Health*, 15(6):821–827, 2012.

[252] James G Xenakis, Elizabeth T Kinter, K Jack Ishak, Alexandra J Ward, Jenő P Marton, Richard J Willke, Simon Davies, and J Jaime Caro. A discrete-event simulation of smoking-cessation strategies based on varenicline pivotal trial data. *Pharmacoeconomics*, 29(6):497–510, 2011.

[253] Daniela Degenring, Mathias Röhl, and Adelinde M Uhrmacher. Discrete event, multi-level simulation of metabolite channeling. *BioSystems*, 75(1):29–41, 2004.

[254] Roxana Djafarzadeh, Gabriel Wainer, and Tofy Mussivand. Devs modeling and simulation of the cellular metabolism by mitochondria. In *Proceedings of the 2005 DEVS Integrative M&S Symposium*, pages 55–62, 2005.

[255] Imed Othmani. *Optimisation multicritère: fondements et concepts*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1998.

[256] Zesong Fei, Bin Li, Shaoshi Yang, Chengwen Xing, Hongbin Chen, and Lajos Hanzo. A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems. *IEEE Communications Surveys & Tutorials*, 19(1):550–586, 2017.

[257] Mohsen Ejday. *Optimisation Multi-Objectifs à base de Métamodèle pour les Procédés de Mise en Forme*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2011.

[258] MHA Bonte, Antonius H van den Boogaard, and J Huétink. A metamodel based optimisation algorithm for metal forming processes. In *Advanced Methods in Material Forming*, pages 55–72. Springer, 2007.

[259] Ilhem Boussaid. *Perfectionnement de métaheuristiques pour l'optimisation continue*. PhD thesis, Université Paris-Est, 2013.

[260] David A Van Veldhuizen. Multiobjective evolutionary algorithms: Classifications, analyzes, and new innovations. *Air Force Inst. Technol., Dayton, OH, Tech. Rep. AFIT/DS/ENG/99-01*, 1999.

[261] Ralph L Keeney and Howard Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.

[262] Zbigniew Michalewicz and David B Fogel. *How to solve it: modern heuristics*. Springer Science & Business Media, 2013.

[263] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

[264] Peter JM Van Laarhoven and Emile HL Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

[265] David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. *Evolutionary computation*, 8(2):125–147, 2000.

[266] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[267] Satya Prakash Sahoo, Manas Ranajan Kabat, and Asis Kumar Sahoo. Tabu search algorithm for core selection in multicast routing. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 17–21. IEEE, 2011.

[268] Alex S Fraser. Simulation of genetic systems by automatic digital computers i. introduction. *Australian Journal of Biological Sciences*, 10(4):484–491, 1957.

[269] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[270] ACMDV Maniezzo. Distributed optimization by ant colonies. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, page 134. Mit Press, 1992.

[271] Marco Dorigo and Mauro Birattari. Ant colony optimization. In *Encyclopedia of machine learning*, pages 36–39. Springer, 2011.

[272] Julio R Banga, Karina J Versyck, and Jan F Van Impe. Computation of optimal identification experiments for nonlinear dynamic process models: a stochastic global optimization approach. *Industrial & engineering chemistry research*, 41(10):2425–2430, 2002.

[273] Daniel Faller, Ursula Klingmüller, and Jens Timmer. Simulation methods for optimal experimental design in systems biology. *Simulation*, 79(12):717–725, 2003.

[274] Zoltan Kutalik, Kwang-Hyun Cho, and Olaf Wolkenhauer. Optimal sampling time selection for parameter estimation in dynamic pathway modeling. *Biosystems*, 75(1-3):43–55, 2004.

[275] Eva Balsa-Canto, Antonio A Alonso, and Julio R Banga. Optimal dynamic experimental design in systems biology: Applications in cell signaling. *IFAC Proceedings Volumes*, 40(4):73–78, 2007.

[276] Francisco J Romero-Campero, Hongqing Cao, Miguel Camara, and Natalio Krasnogor. Structure and parameter estimation for cell systems biology models. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 331–338. ACM, 2008.

[277] Maria Rodriguez-Fernandez, Markus Rehberg, Andreas Kremling, and Julio R Banga. Simultaneous model discrimination and parameter estimation in dynamic models of cellular systems. *BMC systems biology*, 7(1):76, 2013.

[278] Ali R Zomorrodi and Costas D Maranas. Optcom: a multi-level optimization framework for the metabolic modeling and analysis of microbial communities. *PLoS computational biology*, 8(2):e1002363, 2012.

[279] Marko Budinich, Jérémie Bourdon, Abdelhalim Larhlimi, and Damien Eveillard. A multi-objective constraint-based approach for modeling genome-scale microbial ecosystems. *PloS one*, 12(2):e0171744, 2017.

[280] Jialiang Yang, Jun Li, Stefan Grünewald, and Xiu-Feng Wan. Binaligner: a heuristic method to align biological networks. In *BMC bioinformatics*, volume 14, page S8. BioMed Central, 2013.

[281] Zhenping Li, Shihua Zhang, Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Alignment of molecular networks by integer quadratic programming. *Bioinformatics*, 23(13):1631–1639, 2007.

[282] Gunnar W Klau. A new graph-based method for pairwise global network alignment. *BMC bioinformatics*, 10(1):S59, 2009.

[283] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. Global alignment of protein–protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–1267, 2009.

[284] Reinhart Heinrich and Stefan Schuster. The modelling of metabolic systems. structure, control and optimality. *Biosystems*, 47(1-2):61–77, 1998.

[285] Reinhart Heinrich and Stefan Schuster. *The regulation of cellular systems*. Springer Science & Business Media, 2012.

[286] Néstor V Torres and Eberhard O Voit. *Pathway analysis and optimization in metabolic engineering*. Cambridge University Press, 2002.

[287] Wei Shi, Wanlei Zhou, and Yi-Ping Phoebe Chen. Biological sequence assembly and alignment. In *Bioinformatics Technologies*, pages 243–261. Springer, 2005.

[288] Masato Ishikawa, Tomoyuki Toya, Masaki Hoshida, Katsumi Nitta, Atushi Ogiwara, and Minoru Kanehisa. Multiple sequence alignment by parallel simulated annealing. *Bioinformatics*, 9(3):267–273, 1993.

[289] Makoto Hirosawa, Yasushi Totoki, Masaki Hoshida, and Masato Ishikawa. Comprehensive study on iterative algorithms of multiple sequence alignment. *Bioinformatics*, 11(1):13–18, 1995.

[290] Thomas D Schneider and David N Mastronarde. Fast multiple alignment of ungapped dna sequences using information theory and a relaxation method. *Discrete Applied Mathematics*, 71(1-3):259–268, 1996.

[291] Hung Dinh Nguyen, Ikuo Yoshihara, Kunihito Yamamori, and Moritoshi Yasunaga. Aligning multiple protein sequences by parallel hybrid genetic algorithm. *Genome Informatics*, 13:123–132, 2002.

[292] Tariq Riaz, Yi Wang, and Kuo-Bin Li. Multiple sequence alignment using tabu search. In *Proceedings of the second conference on Asia-Pacific bioinformatics-Volume 29*, pages 223–232. Australian Computer Society, Inc., 2004.

[293] Andrew F Neuwald and Jun S Liu. Gapped alignment of protein sequence motifs through monte carlo optimization of a hidden markov model. *BMC bioinformatics*, 5(1):157, 2004.

[294] Hsiao-Ping Hsu, Vishal Mehra, Walter Nadler, and Peter Grassberger. Growth algorithms for lattice heteropolymers at low temperatures. *The Journal of chemical physics*, 118(1):444–451, 2003.

[295] Jacek Błażewicz, Piotr Łukasiak, and Maciej Miłostan. Application of tabu search strategy for finding low energy structure of protein. *Artificial Intelligence in Medicine*, 35(1-2):135–145, 2005.

[296] Roberto Santana, Pedro Larranaga, and José A Lozano. Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In *International Symposium on Biological and Medical Data Analysis*, pages 388–398. Springer, 2004.

[297] JE Smith. The co-evolution of memetic algorithms for protein structure prediction. In *Recent advances in memetic algorithms*, pages 105–128. Springer, 2005.

[298] Rui-Sheng Wang, Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Inferring transcriptional regulatory networks from high-throughput data. *Bioinformatics*, 23(22):3056–3064, 2007.

[299] Reuben Thomas, Carlos J Paredes, Sanjay Mehrotra, Vassily Hatzimanikatis, and Eleftherios T Papoutsakis. A model-based optimization framework for the inference of regulatory interactions using time-course dna microarray expression data. *BMC bioinformatics*, 8(1):228, 2007.

[300] Xiaoxia Lin, Christodoulos A Floudas, Ying Wang, and James R Broach. Theoretical and computational studies of the glucose signaling pathways in yeast using global gene expression data. *Biotechnology and bioengineering*, 84(7):864–886, 2003.

[301] Soohee Han, Yeoin Yoon, and Kwang-Hyun Cho. Inferring biomolecular interaction networks based on convex optimization. *Computational Biology and Chemistry*, 31(5-6):347–354, 2007.

[302] Alejandro F Villaverde and Julio R Banga. Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of the Royal Society Interface*, 11(91):20130505, 2014.

[303] Anna Lombardi and Michael Hörnquist. Controllability analysis of networks. *Physical Review E*, 75(5):056110, 2007.

[304] Wen-Xu Wang, Xuan Ni, Ying-Cheng Lai, and Celso Grebogi. Optimizing controllability of complex networks by minimum structural perturbations. *Physical Review E*, 85(2):026115, 2012.

[305] Sean P Cornelius, William L Kath, and Adilson E Motter. Realistic control of network dynamics. *Nature communications*, 4:1942, 2013.

[306] Jianxi Gao, Yang-Yu Liu, Raissa M D'souza, and Albert-László Barabási. Target control of complex networks. *Nature communications*, 5:5415, 2014.

[307] Lin Wu, Yichao Shen, Min Li, and Fang-Xiang Wu. Drug target identification based on structural output controllability of complex networks. In *International Symposium on Bioinformatics Research and Applications*, pages 188–199. Springer, 2014.

[308] Lin Wu, Yichao Shen, Min Li, and Fang-Xiang Wu. Network output controllability-based method for drug target identification. *IEEE transactions on nanobioscience*, 14(2):184–191, 2015.

[309] Lin Wu, Min Li, Jianxin Wang, and Fang-Xiang Wu. Minimum steering node set of complex networks and its applications to biomolecular networks. *IET systems biology*, 10(3):116–123, 2016.

[310] Lin Wu, Lingkai Tang, Min Li, Jianxin Wang, and Fang-Xiang Wu. Biomolecular network controllability with drug binding information. *IEEE transactions on nanobioscience*, 16(5):326–332, 2017.

[311] Le-Zhi Wang, Yu-Zhong Chen, Wen-Xu Wang, and Ying-Cheng Lai. Physical controllability of complex networks. *Scientific reports*, 7, 2017.

[312] Harvey J Greenberg, William E Hart, and Giuseppe Lancia. Opportunities for combinatorial optimization in computational biology. *INFORMS Journal on Computing*, 16(3):211–231, 2004.

[313] Pedro Larranaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Inaki Inza, José A Lozano, Rubén Armananzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, pages 86–112, 2006.

[314] Loren L Looger and Homme W Hellinga. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics1. *Journal of molecular biology*, 307(1):429–445, 2001.

[315] Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Optimization meets systems biology, 2010.

[316] Paola Festa. Optimization problems in molecular biology: A survey and critical review. In *Int. Math. Forum*, volume 3, pages 269–289, 2008.

[317] Jin Kim, James R Cole, and Sakti Pramanik. Alignment of possible secondary structures in multiple rna sequences using simulated annealing. *Bioinformatics*, 12(4):259–267, 1996.

[318] Yousif Shamoo, Amy Tam, William H Konigsberg, and Kenneth R Williams. Translational repression by the bacteriophage t4 gene 32 protein involves specific recognition of an rna pseudoknot structure. *Journal of molecular biology*, 232(1):89–104, 1993.

[319] MARJORIE Russel, LARRY Gold, HOPE Morrissett, and PATRICIA Z O'Farrell. Translational, autogenous regulation of gene 32 expression during bacteriophage t4 infection. *Journal of Biological Chemistry*, 251(22):7263–7270, 1976.

[320] Jen-Ren Wu and Yun-Chi Yeh. Requirement of a functional gene 32 product of bacteriophage t4 in uv repair. *Journal of virology*, 12(4):758–765, 1973.

[321] Peter H Von Hippel, Stephen C Kowalczykowski, Nils Lonberg, John W Newport, Leland S Paul, Gary D Stormo, and Larry Gold. Autoregulation of gene expression: Quantitative evaluation of the expression and function of the bacteriophage t4 gene 32 (single-stranded dna binding) protein system. *Journal of molecular biology*, 162(4):795–818, 1982.

[322] T Kakegawa and S Hirose. Mode of inhibition of protein synthesis by metabolites of clarithromycin. *Chemotherapy*, 38:317–323, 1990.

[323] Laura Roa Castro and Julie Stal-Le Cardinal. Definition of the collaborative simulation system (cm&ss) from a systemic perspective in vehicle industry context. In *International Conference on Engineering Design (ICED 15)*, 2015.

[324] Ludwig Von Bertalanffy. General system theory. *New York*, 41973(1968):40, 1968.

[325] Jean-Louis Le Moigne. *La théorie du système général: théorie de la modélisation*. jeanlouis le moigne-ae mcx, 1977.

[326] Amir M Sharif and Zahir Irani. Applying a fuzzy-morphological approach to complexity within management decision making. *Management Decision*, 44(7):930–961, 2006.

[327] Sholom Glouberman and Brenda Zimmerman. Complicated and complex systems: what would successful reform of medicare look like? *Romanow Papers*, 2:21–53, 2002.

[328] Dominique Chu, Roger Strand, and Ragnar Fjelland. Theories of complexity. *Complexity*, 8(3):19–30, 2003.

[329] Barry Smith, Jennifer Williams, and Schulze-Kremer Steffen. The ontology of the gene ontology. In *AMIA Annual Symposium Proceedings*, volume 2003, page 609. American Medical Informatics Association, 2003.

[330] Willem Robert Van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011.

[331] Jerry R Hobbs and Feng Pan. Time ontology in owl. *W3C working draft*, 27:133, 2006.

[332] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[333] Saartje Brockmans, Raphael Volz, Andreas Eberhart, and Peter Löffler. Visual modeling of owl dl ontologies using uml. In *International Semantic Web Conference*, volume 3298, pages 198–213. Springer, 2004.

[334] Jānis Bārzdiņš, Guntis Bārzdiņš, Kārlis Čerāns, Renārs Liepiņš, and Artūrs Sproģis. Uml style graphical notation and editor for owl 2. *Perspectives in Business Informatics Research*, pages 102–114, 2010.

[335] Virginie Fortineau, Thomas Paviot, Ludovic Louis-Sidney, and Samir Lamouri. *SWRL as a Rule Language for Ontology-Based Models in Power Plant Design*, pages 588–597. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[336] Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[337] Johan De Kleer and John Seely Brown. A qualitative physics based on confluences. *Artificial intelligence*, 24(1-3):7–83, 1984.

[338] Kenneth D Forbus. Qualitative reasoning., 1997.

[339] Bernard P Zeigler, Herbert Praehofer, and Tag Gon Kim. *Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems*. Academic press, 2000.

[340] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[341] Ching-Lai Hwang and Kwangsun Yoon. Methods for multiple attribute decision making. In *Multiple attribute decision making*, pages 58–191. Springer, 1981.

[342] Eugene C Butcher, Ellen L Berg, and Eric J Kunkel. Systems biology in drug discovery. *Nature biotechnology*, 22(10):1253, 2004.

[343] Kun Yang, Hongjun Bai, Qi Ouyang, Luhua Lai, and Chao Tang. Finding multiple target optimal intervention in disease-related molecular network. *Molecular systems biology*, 4(1):228, 2008.

[344] Rollin M Gallagher and Lisa J Rosenthal. Chronic pain and opiates: balancing pain control and risks in long-term opioid treatment. *Archives of physical medicine and rehabilitation*, 89(3):S77–S82, 2008.

[345] Andrea M Trescot, Standiford Helm, Hans Hansen, Ramsin Benyamin, Scott E Glaser, Rajive Adlaka, Samir Patel, Laxmaiah Manchikanti, et al. Opioids in the management of chronic non-cancer pain: an update of american society of the interventional pain physicians'(asipp) guidelines. *Pain physician*, 11(2 Suppl):S5–S62, 2008.

[346] Pierre Kalfon, Olivier Mimoz, Pascal Auquier, Anderson Loundou, Rémy Gauzit, Alain Lepape, Jean Laurens, Bernard Garrigues, Thierry Pottecher, and Yannick Mallédant. Development and validation of a questionnaire for quantitative assessment of perceived discomforts in critically ill patients. *Intensive care medicine*, 36(10):1751–1758, 2010.

[347] Zahra Alizadeh Afrouzy, Mohammad Mahdi Paydar, Seyed Hadi Nasseri, and Iraj Mahdavi. A meta-heuristic approach supported by nsga-ii for the design and plan of supply chain networks considering new product development. *Journal of Industrial Engineering International*, 14(1):95–109, 2018.

[348] D Ozturk and F Batuk. Technique for order preference by similarity to ideal solution (topsis) for spatial decision problems. In *Proceedings ISPRS*, 2011.

[349] Xiaoping Jia, Tianzhu Zhang, Fang Wang, and Fangyu Han. Multi-objective modeling and optimization for cleaner production processes. *Journal of Cleaner Production*, 14(2):146–151, 2006.

[350] Adam Arkin, John Ross, and Harley H McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149(4):1633–1648, 1998.

[351] Ira Herskowitz and David Hagen. The lysis-lysogeny decision of phage lambda: explicit programming and responsiveness. *Annual review of genetics*, 14(1):399–445, 1980.

[352] Harrison Echols. Multiple dna-protein interactions governing high-precision dna transactions. *Science*, 233(4768):1050–1056, 1986.

[353] David I Friedman. Interaction between bacteriophage λ and its escherichia coli host. *Current opinion in genetics & development*, 2(5):727–738, 1992.

[354] Harley H McAdams and Lucy Shapiro. Circuit simulation of genetic networks. *Science*, 269(5224):650–656, 1995.

[355] Mark Ptashne and Alexander Gann. Transcriptional activation by recruitment. *Nature*, 386(6625):569, 1997.

[356] Xiao-Peng Zhang, Feng Liu, Zhang Cheng, and Wei Wang. Cell fate decision mediated by p53 pulses. *Proceedings of the National Academy of Sciences*, 106(30):12245–12250, 2009.

[357] René Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, 1973.

[358] Leon Glass. Classification of biological networks by their qualitative dynamics. *Journal of Theoretical Biology*, 54(1):85–107, 1975.

[359] Janez Brank and et al. Gold standard based ontology evaluation using instance assignment. In *IN: PROC. OF THE EON 2006 WORKSHOP*, 2006.

[360] Fabian Neuhaus, Amanda Vizedom, Ken Baclawski, Mike Bennett, Mike Dean, Michael Denny, Michael Grüninger, Ali Hashemi, Terry Longstreth, Leo Obrst, et al. Towards ontology evaluation across the life cycle. *Applied Ontology*, 8(3):179–194, 2013.

[361] Anusha Indika Walisadeera, Athula Ginige, and Gihan Nilendra Wikramanayake. Ontology evaluation approaches: a case study from agriculture domain. In *International Conference on Computational Science and Its Applications*, pages 318–333. Springer International Publishing, 2016.

[362] Robert Porzel and Rainer Malaka. A task-based approach for ontology evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, pages 1–6, 2004.

[363] Anni-Yasmin Turhan. Description logic reasoning for semantic web ontologies. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, pages 6:1–6:5, New York, NY, USA, 2011. ACM.

[364] Alexander Maedche and Steffen Staab. *Measuring Similarity between Ontologies*, pages 251–263. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[365] Denny Vrandečić. Ontology evaluation. In *Handbook on Ontologies*, pages 293–313. Springer, 2009.

[366] Christopher Brewster, Harith Alani, Srinandan Dasmahapatra, and Yorick Wilks. Data driven ontology evaluation. Event Dates: 24-30 May, 2004.

[367] Adolfo Lozano-Tello and Asunción Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of database management*, 2(15):1–18, 2004.

[368] Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Conceptual analysis of lexical taxonomies: The case of wordnet top-level. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, pages 285–296, New York, NY, USA, 2001. ACM.

[369] David Lira Nuñez and Milton Borsato. An ontology-based model for prognostics and health management of machines. *Journal of Industrial Information Integration*, 2017.

[370] Rosario Distefano, Nickolas Goncharenko, Franco Fummi, Rosalba Giugno, Gary D Badery, and Nicola Bombieri. Syqual: a platform for qualitative modelling and simulation of biological systems. In *High Level Design Validation and Test Workshop (HLDVT), 2016 IEEE International*, pages 155–161. IEEE, 2016.

[371] François Fages. From syntax to semantics in systems biology towards automated reasoning tools. In *Transactions on Computational Systems Biology IV*, pages 68–70. Springer, 2006.

Ali AYADI

Approches sémantiques pour la méta-optimisation des réseaux biomoléculaires complexes

Résumé :
Les modèles de la biologie des systèmes visent à comprendre le comportement d'une cellule à travers un réseau biomoléculaire complexe. Dans a littérature, la plupart des études ne se sont intéressés qu'à la modélisation des parties isolées du réseau biomoléculaire com les réseaux métaboliques, etc. Cependant, pour bien comprendre le comportement d'une cellule, nous devons modéliser et analyser le réseau biomoléculaire dans son ensemble. Les approches existantes ne répondent pas suffisamment à ces exigences. Dans ce projet de recherche, nous proposons une plate-forme qui permet aux biologistes de simuler les changements d'état des réseaux biomoléculaires dans le but de piloter leurs comportements et de les faire évoluer d'un état non désiré vers un état souhaitable. Cette plate-forme utilise des règles, des connaissances et de l'expérience, un peu comme celles que pourrait en tirer un biologiste expert. La plate-forme comprend quatre modules : un module de modélisation logique, un module de modélisation sémantique, un module de simulation qualitative à événements discrets et un module d'optimisation. Dans ce but, nous présentons d'abord une approche logique pour la modélisation des réseaux biomoléculaires complexes, incluant leurs aspects structurels, fonctionnels et comportementaux. Ensuite, nous proposons une approche sémantique basée sur quatre ontologies pour fournir une description riche des réseaux biomoléculaires et de leurs changements d'état. Ensuite, nous présentons une méthode de simulation qualitative à événements discrets pour simuler le comportement du réseau biomoléculaire dans le temps. Enfin, nous proposons une méthode d'optimisation multi-objectifs pour optimiser la transitabilité des réseaux biomoléculaires complexes dans laquelle nous prenons en compte différents critères tels que la minimisation du nombre de stimuli externes, la minimisation du coût de ces stimuli, la minimisation du nombre de nœuds cibles et la minimisation de l'inconfort du patient. En se fondant sur ces quatre contributions, un prototype appelé CBN-Simulateur a été développé. Nous décrivons nos approches et montrons leurs applications sur des études de cas réels, le bactériophage T4 gene 32, le phage lambda et le réseau de signalisation p53. Les résultats montrent que ces approches fournissent les éléments nécessaires pour modéliser, raisonner et analyser le comportement dynamique et les états de transition des réseaux biomoléculaires complexes.
Mots clés : Transittabilité, modélisation logique, raisonnement sémantique, simulation qualitative à événements discrets, comportement dynamique des réseaux biomoléculaires complexes

Semantic approaches for the meta-optimization of complex biomolecular networks

Abstract :
Systems biology models aim to understand the behaviour of a cell trough a complex biomolecular network. In the literature, most research focuses on modelling isolated parts of this network, such as metabolic networks. However, to fully understand the cell's behaviour we should analyze the biomolecular network as a whole. Available approaches do not address these requirements sufficiently. In this context, we aim at developing a platform that enables biologists to simulate the state changes of biomolecular networks with the goal of steering their behaviours. The platform employs rules, knowledge and experience, much like those that an expert biologist might derive. This platform consists of four modules: a logic-based modelling module, a semantic modelling module, a qualitative discrete-event simulation module and an optimization module. For this purpose, we first present a logic-based approach for modelling complex biomolecular networks including the structural, functional and behavioural aspects. Next, we propose a semantic approach based on four ontologies to provide a rich description of biomolecular networks and their state changes. Then, we present a method of qualitative discrete-event simulation to simulate the biomolecular network behaviour over time. Finally, we propose a multi-objective optimization method for optimizing the transittability of complex biomolecular networks in which we take into account various criteria such as minimizing the number of external stimuli, minimizing the cost of these stimuli, minimizing the number of target nodes and minimizing patient discomfort. Based on these four contributions, a prototype called the CBNSimulator was developed. We describe our approaches and show their applicability through real cases studies, the bacteriophage T4 gene 32, the phage lambda, and the p53 signaling network. Results demonstrate that these approaches provide the necessary elements to model, reason and analyse the dynamic behaviour and the transition states of complex biomolecular networks.
Keywords : Transittability, logical-based modelling, semantic reasoning, qualitative discrete-event simulation, dynamic behaviour of complex biomolecular networks

Laboratoire ICube (UMR7357) - 300 bd Sébastien Brant - CS 10413 - F-67412 Illkirch