Rosario Distefano

Modeling and Simulation of Biological Systems through Electronic Design Automation techniques

Ph.D. Thesis

May 25, 2017

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
Prof. Franco Fummi

Series N°:

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

To the most valuable thing in Life...friendship

Contents

Abbreviations

BioPAX Biological Pathway Exchange
CAC Colitis-associated Colon Cancer
EDA Electronic Design Automation
FSM Finite State Machine
GA General Asynchronous
IBD Inflammatory Bowel Disease
IEC Intestinal Epithelial Cell
ROA Random Order Asynchronous
SBGN Systems Biology Graphical Notation
SBML Systems Biology Markup Language

List of Figures

1

Introduction

Modeling and simulation of biological systems is a key requirement for integrating in-vitro and in-vivo experimental data. In-silico simulation allows testing different experimental conditions, thus helping in the discovery of the dynamics that regulate the system. These dynamics include errors in the cellular information processing that are responsible for diseases such as cancer, autoimmunity, and diabetes as well as drug effects to the system (Gonalves et al. [46]).

In this context, modeling approaches can be classified into two categories: *quantitative* and *qualitative* models. Quantitative modeling allows for a natural representation of molecular and gene networks and provides the most precise prediction. Nevertheless, the lack of kinetic data (and of quantitative data in general) hampers its use for many situations (Le Novere [72]). In contrast, qualitative models simplify the biological reality and are often able to reproduce the system behavior. They cannot describe actual concentration levels nor realistic time scales. As a consequence, they cannot be used to explain and predict the outcome of biological experiments that yield quantitative data. However, given a biological network consisting of input (e.g., receptors), intermediate, and output (e.g., transcription factors) signals, they allow studying the input-output relationships through discrete simulation (Samaga et al. [99]).

Boolean models are gaining an increasing interest in reproducing dynamic behaviors, understanding processes, and predicting emerging properties of cellular signaling networks through in-silico experiments. They are emerging as a valid alternative to the quantitative approaches (i.e., based on ordinary differential equations) for exploratory modeling when little is known about reaction kinetics or equilibrium constants in the context of gene expression or signaling. Even though several approaches and software have been recently proposed for logic modeling of biological systems, they are limited to specific contexts and they lack of automation in analyzing biological properties such as complex attractors, and molecule vulnerability.

1.1 Aims of the thesis

This thesis proposes a platform based on *Electronic Design Automation* (*EDA*) technologies for qualitative modeling and simulation of *Biological Systems*. It aims at overtaking limitations that affect the most recent qualitative tools. In particular, tools do not support

the simulation complexity of large networks, and lack of automation in analyzing biological properties such as complex attractors and molecule vulnerability. For example, in *BoolNet* (Müssel et al. [87]), the attractor identification using the non-heuristic analysis is limited to networks with 29 variables. For larger systems, they can be inferred heuristically. However, this approach does not guarantee that all attractors can be identified.

The proposed platform allows performing both automatics and efficient system simulation. Being based on languages and design tools well-established in the *EDA* field, it allows addressing high computational costs normally associated with the modeling and simulation of biological systems (Bombieri et al. [8], Distefano et al. [29]). The simulation core *BIODEA* relies on a discrete event-based framework developed in *SystemC*, which is the de-facto reference standard language in *EDA* for efficient and accurate simulations of systems at different levels of abstraction. Based on *SystemC*, *BIODEA* applies methodologies and languages well established in the *Electronic Design Automation (EDA)* field, such as assertion-based verification (ABV) and mutation analysis, to analyze

- Complex attractors (e.g., protein oscillations). In the context of discrete logic-based dynamical models, the attractors identification and analysis, in which stable cycles of states are represented, is a dominant task. As attractors comprise the states in which biological network dwells most of the time, they can be often linked to phenotypes (Kauffman [65], Li et al. [77]).
- Robustness/Sensitivity of signaling networks. Robustness analysis aims at investigating how biological systems keep their functionality under perturbation, without being disrupted or heavily modified. Sensitivity analysis represents the opposite of the Robustness analysis.

Chapter 4 describes the *BIODEA* framework in detail, while main characteristics and contributions are following summarized:

- It relies on a discrete event-based computational model, in which each network entity is modeled through Finite State Machines (FSMs).
- It uses *SystemC*, which is the standard, reference language in the *EDA* field for modeling and verifying complex systems at different abstraction levels. *SystemC*-based verification is the de-facto an alternative to static verification techniques (e.g., model checking) when such formal techniques cannot deal with the state-space complexity of the model.
- It automatically generates a *SystemC* model from a SBML description
- Dynamics of variables is explored through a hybrid approach: Deterministic to explore continuous units (e.g., activation delay times, protein lifetimes, etc.), and stochastic to explore molecular concentrations values.
- It relies on Assertion-based Verification (ABV) technique, in which any property is formally described through the *Property Specification Language* (*PSL*) and, then, automatically synthesized and integrated into the simulation system.
- The framework implements mutation analysis, a technique broadly adopted in the context of Software testing, to simulate the system behavior under perturbations. Given any variable configuration, deviations from standard behaviors are modeled and simulated by introducing syntactically changes to the system model (mutations). Examples are extra activation or inhibitions of any protein, delay time or lifetime modifications.

Figure 1.1 shows an overview of the *BIODEA* modeling and simulation flow, which consists of three phases:

1. *Modeling of Biological Entities in SystemC*.
2. *Network parameterization through Assertion-based Verification*. By using assertion-based verification, biological properties can be formally defined through the *property specification languages (PSL)* and, then, automatically synthesized and integrated as checkers into the simulation system.
3. *Robustness/Sensitivity Analysis through Assertion-based Verification and Mutation Analysis*.

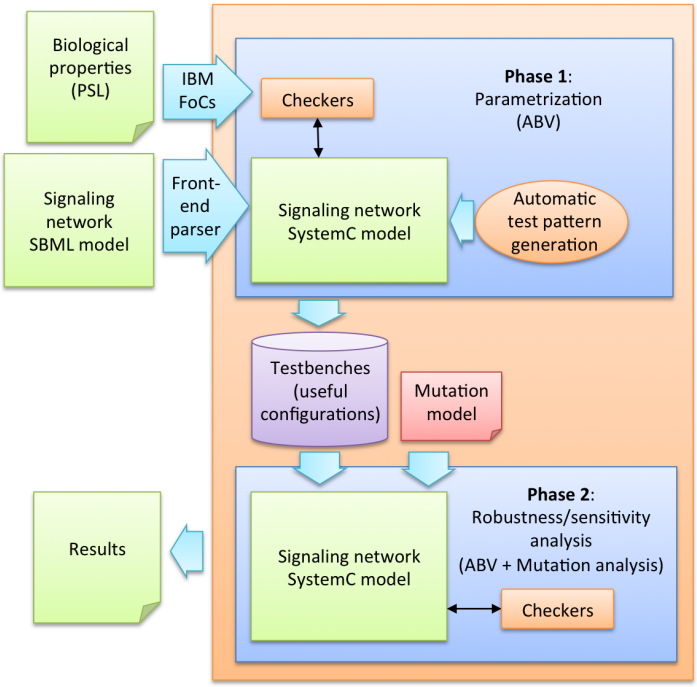


Fig. 1.1. The *BIODEA* framework overview.

In order to automate and provide an easy interface to execute the robustness/sensitivity analysis, especially the one based on drug targeting, it has been developed a web-oriented platform called *SyQUAL*. *SyQUAL* uses *BIODEA* as simulation core. The platform provides both synchronous and asynchronous updating policies for simulation, where the asynchronous method relies on a time-delayed updating policy controlled by topology-based constraints. *SyQUAL* provides support for *Systems Biology Markup Language* (*SBML*), the state of the art description model for Biological Systems data exchanges. SBML is supported according to its two most used declensions: SBML Level 2 for quantitative descriptions and SBML *qual* for qualitative descriptions.

It collects the information described in any SBML and uses it to generate *SystemC*-based simulators "on-the-fly". Simulation can be performed using a (*Zero* or *Drug*) *Knowledge-based Knock Out* approach, in which biological entities can be knocked out without using any prior knowledge or using drug targeting. Simulation relies on synchronous and asynchronous updating policies, in which a single global time step is counterposed to different local time steps, respectively. Chapter 5 provides more details regarding the *SyQUAL* platform.

1.2 Thesis overview

The thesis starts introducing notions on *Systems Biology* and *EDA*, focusing on languages and methodologies designed for modeling and simulation of systems, continuing with the description of the proposed platform, and closing with experimental results and conclusions.

Chapter 2 gives an overview of general molecular biology concepts, cell functionality, and the role of modeling in *Systems Biology*. Then, it discusses the *State of the Art* of modeling and simulation of Biological Systems. Section 2.1 describes the fundamental unit of life, the *Cell*, the basis for who is familiar with *EDA* but not with *Systems Biology*, and then illustrates the principal concepts regarding the biological elements involved into the main cell activities, such as *genes*, *proteins*, and *miRNAs*. Section concludes by describing main biological functionalities, such as *signaling*, *gene regulation*, and *metabolism*. Section 2.2 introduces the *Systems Biology*, what is a *model* in *Systems Biology*, and which motivations stand behind *modeling*. Section 2.3 discusses the two de facto standard description models used for biological data exchange. Section 2.4 focuses on the description of methodologies for modeling biological systems through a quantitative (Section 2.4.2), semi-quantitative (Section 2.4.2), and qualitative (Section 2.4.2) perspective.

Chapter 3 describes general concepts regarding embedded systems, used to introduce the context on which the thesis work is focused on. Section 3.1 describes basic concepts regarding the *Electronic Design Automation*, whose purpose is to design electronic products of all kinds. Finally, Section 3.2 presents *SystemC*, one of the most popular embedded system modeling description languages. It describes main *SystemC* components, especially for who is not familiar with these concepts.

Chapter 4 describes the *BIODEA* framework, a first generalizable approach for modeling and simulation of Biological Systems through *EDA* techniques and languages. Section 4.1 discusses how biological entities are modeled and then translated in a standalone *SystemC* simulator. Section 4.2 describes the *Assertion-based Verification* (*ABV*) techniques applied to investigate behaviors of interest through system parameters estimation. Finally, Section 4.3 presents how faults injection has been applied to alter the biological system behavior, in order to investigate the system robustness/sensitivity under perturbation.

Chapter 5 presents the *SyQUAL*, a platform for robustness/sensitivity analysis. Section 5.1 describes the platform architecture and its building blocks.

Chapter 6 shows how *BIODEA* and *SyQUAL* have been applied to formalize and verify biological hypotheses on given biological systems. Sections show experimental results obtained by modeling and simulation biological systems according to different cellular functionalities, such as signaling and gene regulation. Each biological case study

is correlated by a (i) introduction to the biological phenomenon, (ii) experimental conditions, (iii) biological hypotheses to be investigated, (iv) experimental results, and (v) conclusions. Section 6.1 describes the *Signaling Network Controlling LFA-1 beta2 integrin activation mediating Leukocyte recruitment from the blood into the tissues*, focused on uncover those dynamics behind the integrin periodic oscillation. Section 6.2 describes the *Colitis-associated Colon Cancer (CAC) Network*. It tries to uncover those dynamics that stand behind the inflammation-associated tumorigenesis.

Chapter 7 concludes the thesis discussing main results and future extensions.

2

Modeling and Simulation of Biological Systems

This chapter gives an overview of general molecular biology concepts, cell functionality, and the role of modeling in *Systems Biology*. Then, it discusses the *State of the Art* of modeling and simulation of biological systems.

Section 2.1 describes the fundamental unit of life, the *Cell*, the basis for who is familiar with *EDA* but not with *Systems Biology*, and then illustrates the principal concepts regarding the biological elements involved into the main cell activities, such as *genes*, *proteins*, and *miRNAs*. Section concludes by describing main biological functionalities, such as *signaling*, *gene regulation*, and *metabolism*. Section 2.2 introduces the *Systems Biology*, what is a *model* in *Systems Biology*, and which motivations stand behind *modeling*.

Section 2.3 discusses the two de facto standard description models used for biological data exchange.

Section 2.4 focuses on the description of methodologies for modeling biological systems through a quantitative (Section 2.4.2), semi-quantitative (Section 2.4.2), and qualitative (Section 2.4.2) perspective.

2.1 The Cell

Cells are the fundamental units of the living organisms. They interact with the environment and other cells by processing and exchanging environmental information. The way they interact is, essentially, carried out through biochemical reactions, exchanging the resulting molecular products. Each cell is autonomous and self-sustaining, carrying out its activities through its own set of instructions.

Essentially, cells can be divided into two main types, prokaryotic and eukaryotic cells, depending if the nucleus is enclosed within a membrane. Cells are composed of a cytoplasm enclosed by a plasma membrane. In some cases, plasma membrane can be enclosed within a cell wall, as observed in plants, fungi, and bacteria. The cytoplasm contains, in turn, the cytosol and small membrane-enclosed organelles, such as mitochondria, in which the biochemical processes of respiration and energy production occur. To better highlight the different level of complexity, the prokaryotic and eukaryotic cells are shown in Figure 2.1 and Figure 2.2, respectively.

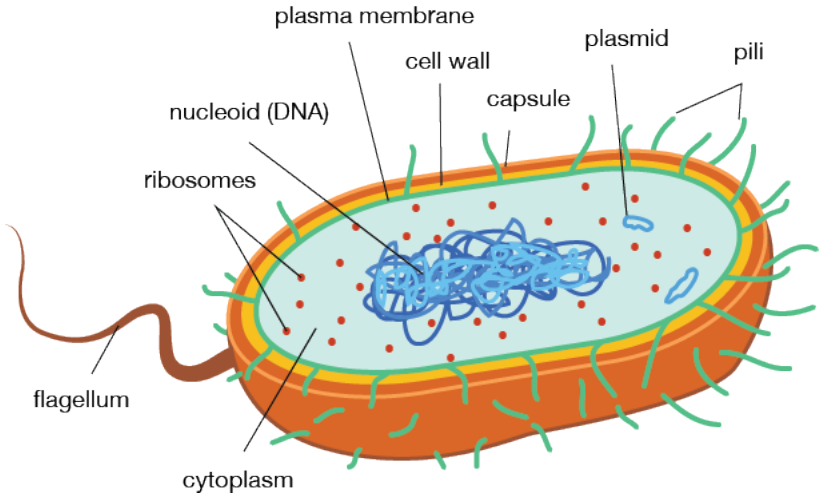


Fig. 2.1. The *prokaryotic* cell structure. (source: *http://www.shmoop.com/biology-cells/prokaryotic-cells.html*)

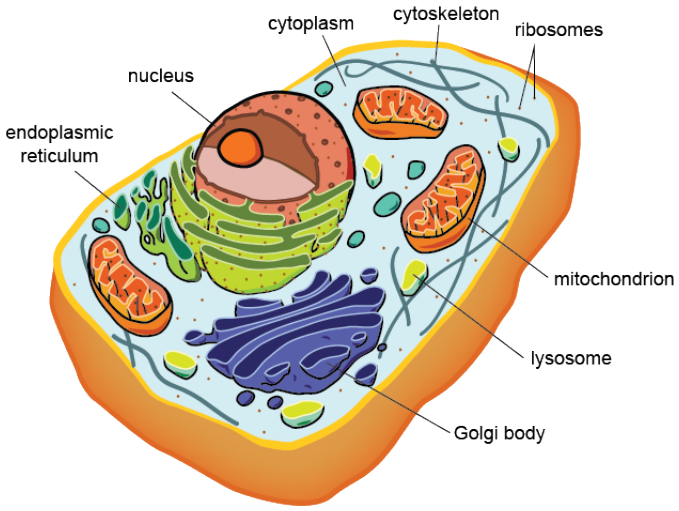


Fig. 2.2. The *eukaryotic* cell structure. (*http://www.shmoop.com/biology-cells/all-eukaryotic-cells.html*)

Cells are characterized by a wide range of sizes, as well as being tissues and organisms dependent. Despite these differences, cells store their information in the same way, in the form of double-stranded molecules, called *DNA* (*Deoxyribonucleic Acid*), composed of four types of chemical compounds, known as nucleotides. Nucleotides are identified using four letters - *A, T, C, G* - called *Adenine*, *Thymine*, *Cytosine*, and *Guanine*, respectively. The information is encoded in long linear sequences, in which these nucleotides are combined. This information, to be used, must be transcribed, and lastly translated; these processes are called *transcription* and *translation*, and they are involved into the *RNAs* (*Ribonucleic Acid*) and *Proteins* production, respectively. The *"instructions"* to produce

these molecules is associated to specific segments of DNA, called *genes*. Figure 2.3 shows the *information flow* from DNA to protein.

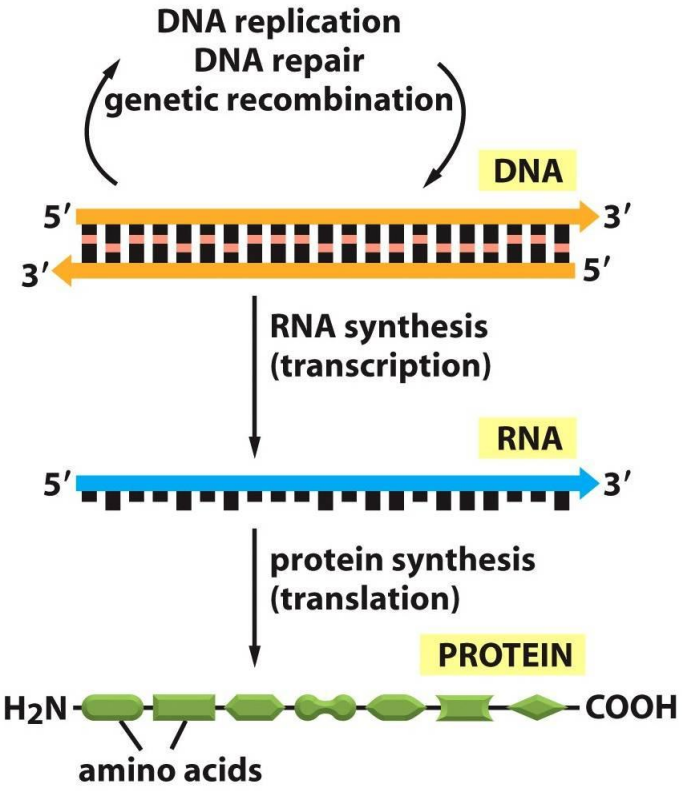


Fig. 2.3. From *DNA* to *protein*.

RNA molecules can be seen as a way to bring the DNA information out of the nucleus, in which it is stored. Protein molecules are involved in most of the cell activities. They allow to manage the function, structure, and regulation of organs and tissues.

The whole genetic information, known as *genome*, is not completely assigned to the proteins production. Special molecules, that act as *regulatory* elements, are produced as well. Inside cells, not all genes are expressed at the same time. They are expressed according to the cell necessities. Transcription and translation amount is regulated through regulatory elements, used to control the transcription rate. Since a cell can be considered as a dynamic biochemical system, different incoming environmental inputs produce different set of biochemical reactions, which can be interpreted as the cell *answer* to the environment. Those reactions depend on parameters, such as the reactants concentration, the chemical properties regulating the reaction speed, and they generate linear reaction pathways in turn organized in concurrent non-linear complex networks (Gilbert et al. [41]).

Usually, these networks are classified according to their biological function, such as signaling, gene regulatory and metabolic networks.

Signaling Networks

Any cell has to be sensible on incoming intracellular and extracellular changes (*signals*). Signals participate in crucial functions, such as *homeostasis*, where the inner-body physiological equilibrium must be controlled and preserved, as well as cell differentiation, proliferation, apoptosis (cell death), and forth.

Basically, the signaling process starts through the binding of an extracellular signaling molecule to a molecular receptor located on the cell surface. Signals can be of different nature, in the form of elements/small molecules, such as ions, organic or inorganic complexes, or physical changes, such as temperature, light, and pressure.

The bound receptor propagates the received message (the incoming signal) within the cell through a messenger, triggering a series of biochemical reactions resulting in a signaling cascade. Usually, several biochemical reactions can be mediated by a single signal. Finally, the cell gene expression represents the resulting signal transduction response, since the gene regulation system directly depends on the signaling cascade. For a precise set of signals, the complex of involved reactions, receptors, and final targets, are collected in a single description model called *signaling networ*k, or signaling pathway. Recently, studies (Gomperts et al. [44], Guo et al. [50]) shown that signaling networks can crosstalk, resulting in more complex scenarios. Figure 2.4 which is an example of the most studied signaling pathways, shows the MAPK signaling pathway. Mitogen-activated protein kinase (MAPK) signaling cascades plays a key role in transduction extracellular signals to cellular responses. MAPK pathway favorites cell activities, such as proliferation, differentiation, growth, inflammatory responses and apoptosis in mammalian cells.

Gene Regulatory Networks

Gene expression is (*time*, *tissue*, and *age*)-dependent, resulting in control of all cell functionalities. Some fundamental genes are maintained continuously expressed in all cells. The largest part is turned on (expressed) or off (inactive) depending on specific cell conditions.

The gene expression process is involved, most of the time, in proteins production. The process starts transcribing the gene information into a *Messenger RNA* (*mRNA*), and consequently, translates the resulting molecule into a protein. The first part of this process is mediated by special proteins, called *Transcription Factors* (*TFs*), which act as activator or inhibitor, resulting in control of the transcription rate. Transcription Factors are themself the result of other genes transcription and translation. Those gene regulatory networks can be highly complex (Lee et al. [75], Schlitt et al. [102], Karlebach et al. [63]).

Metabolic Networks

In cells, a specific set of biochemical reactions is involved in cell activities such as energy production and cell growth; these reactions produce and degrade organic molecules, and belong to *Metabolism*.

Metabolism can be classified into catabolic pathways, in which environmental imported molecules are broken down in small ones for energy production, and anabolic

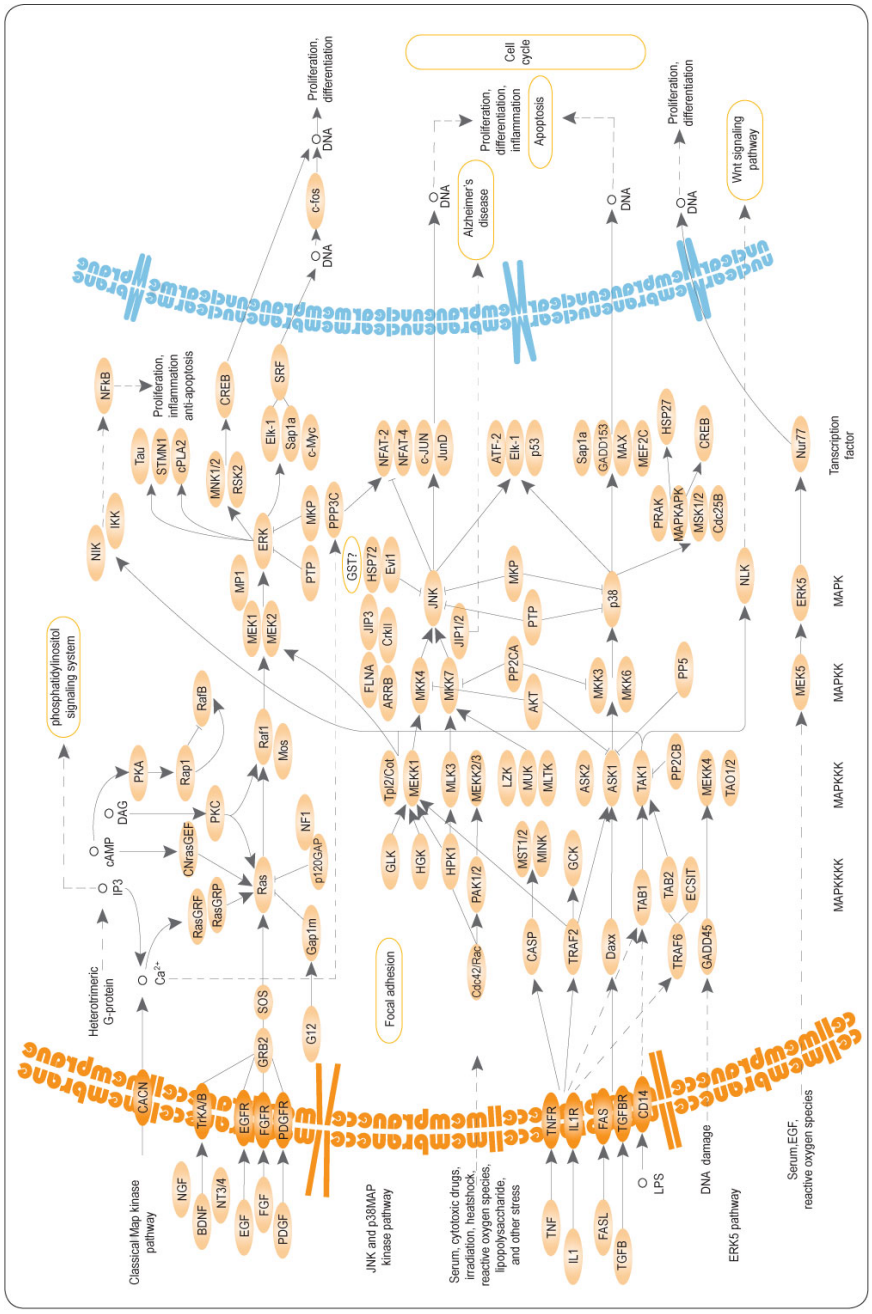


Fig. 2.4. The *MAPK* signaling pathway.

pathways, in which molecules produced by catabolic pathways are used, through enzyme-mediated reactions, as cell components building blocks. Gene regulation plays an important role for Metabolism, since it is directly involved into enzymes production (Orth

et al. [92]). Metabolism can be further classified as *primary* and *secondary*. In primary metabolism, processes involve fatty acids, carbohydrates, and other complexes, resulting in fundamental mechanism for the cell growth. In secondary metabolism, non-crucial substances, known as metabolites, are produced.

2.2 Systems Biology

In biology, two fundamental paradigms are adopted to study living organisms, both at different abstraction levels: the *reductionist* and *holistic* paradigm. The *reductionist* paradigm aims at understanding life considering single components and interactions. The *holistic* paradigm integrates components and interactions in such a way to form biological systems (*Complex Networks*) to better reflect the dynamic behaviors of life.

Complex networks are essentially focused on the depiction, analysis, modeling and simulation of complex systems characterized by several elements and connections. Complex networks are not only limited to unveil special patterns of connectivity. They can be used to explain the evolution of connectivity in a system, as well as its growth. Virtually, complex networks allow representing any system composed of discrete elements.

In *Systems Biology*, the dynamic modeling of biological systems aims at describing how such interactions, among defined elements, affect the time course of the *elements* state and, more in general of the whole system, under specific conditions.

2.2.1 What is a model in Systems Biology?

In *Systems Biology*, models denote a way to represent life, often in an abstract and simplified fashion, such as graphs, diagrams, mathematical equations, chemical formulas, and so forth. In general, models are not exclusively limited to represent known behaviors. They can also include hypotheses to be verified.

The *abstraction* term indicates a mental approach in which it is possible to replace a distinct set of elements by using a general concept, which describes them on the basis of their common properties. A model describes the state of a biological system and how it changes over time.

2.2.2 Why modeling in Systems Biology?

At the present time, the growing complexity of biological phenomena, driven by the availability of new and more accurate techniques and technologies, is increasing the requirement of new modeling tools to better interpret experimental data.
Molecular manipulation techniques, such as *gene knockout*, where a specific gene has been made inoperative, revealed to biologists that analyzed phenomena can lead to a greater complexity than they usually assume.
A validated dynamic model, which correctly captures experimentally observed behaviors, allows better tracking those changes due to perturbations, in order to unveil dynamics that regulate the system behavior. From this perspective, model development and analysis of biological systems is recognized as a key requirement for integrating in vitro and in vivo experimental data.

2.3 Representation formats for Biological Systems

The *Biological Pathway Exchange* (*BioPAX*, Demir et al. [27]) and the *Systems Biology Markup Language* (*SBML*, Hucka et al. [60]) represent the most used representation formats to exchange biological processes.

BioPAX provides visualization and qualitative analysis of biological pathways, represented as a *machine interpretable* content for the Web.

In contrast, SBML focuses on (i) quantitative modeling (SBML *Level 2* and *Level 3*) of such biological processes (i.e. reactions from a quantitative perspective) and (ii) their dynamic simulation. SBML is based on the *eXtensible Markup Language* (*XML*, Bray et al. [14]), for its portability, *human-readable* content, and the wide acceptance as language for computational biology.

Recently, SBML added a support to qualitative modeling (Chaouiya et al. [16]) through the qualitative (*qual*) extension package (SBML Level 3 Version 1 *qual*, commonly identified as SBML *qual*).

A valid alternative to describe biochemical systems is provided by rule-based languages, such as *BioNetGen language* (Faeder et al. [33]), named *BNGL*, and *Kappa* (Danos and Laneve [24]).

Rule-based formalism allows investigating the mechanistic aspects at the level of functional sites in biological molecules. Their use is especially involved to specify protein-protein interactions and to describe the variation in protein concentration.

In this context, BioNetGen language and Kappa have been designed for modeling of biochemical systems, such as mass-action kinetics models. They represent essentially the same language, except for a few small differences. These languages provide a way to enumerate all potentially existing molecular species and reactions in a biochemical system, which is characterized by a high combinatorial complexity. This is achieved by defining only a set of reactive sites/motifs in large molecular complexes, the interactions and transformations that can involve the sites themselves. As biological molecular interactions are encoded via formalized rules, these languages require specialized simulators.

An important lacking component in such a languages it is the absence of a rich annotation, unlike SBML (a reaction-based formalism). Metadata (semantic information) fulfills an important role to allow an automatic model interpretation and processing. Moreover, the process of translation of such rules into ODEs or reaction network is computationally demanding or occasionally not achievable.

The thesis work is focused on the modeling of biological systems to unveil system dynamics under perturbation, in which biological systems are treated as discrete event-based models. The thesis approach does not request any additional detail, such as kinetics rates, molecular motifs, and concentrations, except for the model provided through the SBML description, a standard de facto description model for biological data exchange. Additionally, the SBML annotation plays a key role for performing robustness/sensitivity analysis based on drug knock out. As the thesis platform provides support for SBML Level 3, it can be extended to include new functionalities. In the last few years, SBML Level 3 extended its functionalities by including a new package, called *multi* (Multi-state multicomponent species, interactions specific to particular domains and sites of proteins), to model biochemical systems using a rule-based formalism.

Biological Pathway Exchange (BioPAX)

BioPAX represents a standard format for exchanging pathways data, that can be described at cellular and molecular level. BioPAX is based on a RDF/OWL language. The *Resource Description Framework* (*RDF*, Lassila et al. [71]) is a family of *World Wide Web Consortium* (W3C) specifications initially designed as a web-oriented metadata-based data model. The *Web Ontology Language* (*OWL*, McGuinness et al. [82]) is a family of languages designed to represent complex knowledge about entities, group of entities, and so forth.

BioPAX covers mainly families of biological pathways, such as gene regulatory networks, signaling and metabolic pathways and, more in general, molecular interactions. BioPAX uses a class-based representation to group entities and their interactions. Classes are designed for (i) pathways, (ii) interactions, and (iii) physical entities and genes. Figure 2.5 shows the BioPAX top level structure. *Pathway* node contains a set of interactions describing, for example, Apoptosis and PI3K/AKT pathways. *Interaction* node represents a basic relationship among a set of entities, such as reaction and catalysis. Finally, the *Physical Entity* node represents simple entities, such as genes, proteins, transcription factors, and small molecules.

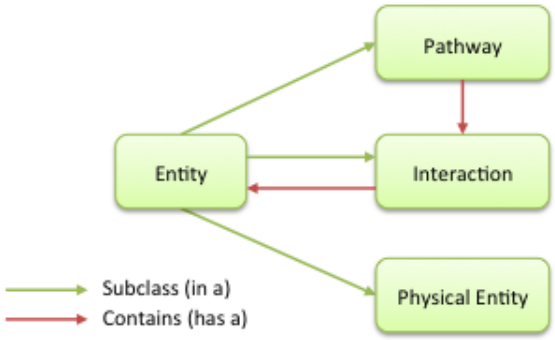


Fig. 2.5. The BioPAX top level structure.

PI3K/AKT pathway regulates different cell functionalities, such as cell surviving and cell growth. Studies show that PI3K/AKT pathway components are often mutated in human cancers. PI3K/AKT pathway mediates receptors mainly involved in survival signals.

Listing 2.1 presents a code snippet collected from the BioPAX description model regarding the PI3K/AKT pathway. In such a description, class names start with an uppercase character, while property terms start with a lowercase one. As example, the class *Pathway* (`bp:Pathway`) consists of different property terms (elements), in which each one describes a specific entity. An entity can be a:

- A reaction (`<bp:pathwayComponent rdf:resource="#BiochemicalReaction1" />`).
- A comment (`<bp:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Signaling by AKT... </bp:comment>`).

- A small molecule (<bp:SmallMoleculeReference rdf:ID="SmallMoleculeReference1">).
- A protein (<bp:ProteinReference rdf:ID="ProteinReference20">).
- etc...

Pathway components and their attributes are extensively and separately described, resulting in a quite long description model, not easy to read without specific tools and languages. All BioPAX specifications and documentation are available through the following URL: *http://www.biopax.org*.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax
    -ns#" xmlns:bp="http://www.biopax.org/release/biopax-
    level3.owl#" xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xml:base=
    "http://www.reactome.org/biopax/58/1257604#">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.biopax.org/release
       /biopax-level3.owl" />
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">BioPAX pathway converted from "
       PIP3 activates AKT signaling" in the Reactome
       database.</rdfs:comment>
  </owl:Ontology>
  <bp:BiochemicalReaction rdf:ID="BiochemicalReaction1">
    <bp:conversionDirection rdf:datatype="http://www.w3.org
       /2001/XMLSchema#string">LEFT-TO-RIGHT</bp:
       conversionDirection>
    <bp:left rdf:resource="#SmallMolecule1" />
    <bp:left rdf:resource="#SmallMolecule2" />
    <bp:right rdf:resource="#SmallMolecule3" />
    <bp:right rdf:resource="#SmallMolecule4" />
    <bp:eCNumber rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">2.7.1.153</bp:eCNumber>
    <bp:displayName rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">PI3K phosphorylates PIP2 to PIP3</
       bp:displayName>
    <bp:xref rdf:resource="#UnificationXref523" />
    <bp:xref rdf:resource="#UnificationXref524" />
    <bp:comment rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">A number of different
       extracellular signals converge on PI3K activation.
       PI3K can be activated downstream of receptor
       tyrosine kinases (RTKs) such as FGFR (Ong et al.
       2001, Eswarakumar et al. 2005), KIT (Chian et al.
       2001, Ronnstrand 2004, Reber et al. 2006), PDGF (
```

```
      Coughlin et al. 1989, Fantl et al. 1992, Heldin et
      al. 1998), insulin receptor IGF1R (Hadari et al.
      1992, Kooijman et al. 1995), and EGFR and ...</bp:
      comment>
   <bp:xref rdf:resource="#PublicationXref1" />
    ...
   <bp:xref rdf:resource="#PublicationXref21" />
   <bp:dataSource rdf:resource="#Provenance1" />
   <bp:comment rdf:datatype="http://www.w3.org/2001/
      XMLSchema#string">Authored: Orlic-Milacic, M,
      2012-07-18</bp:comment>
    ...
   <bp:comment rdf:datatype="http://www.w3.org/2001/
      XMLSchema#string">Edited: Matthews, L, 2012-08-03</
      bp:comment>
  </bp:BiochemicalReaction>
  <bp:ProteinReference rdf:ID="ProteinReference20">
   <bp:organism rdf:resource="#BioSource1" />
   <bp:name rdf:datatype="http://www.w3.org/2001/XMLSchema#
      string">UniProt:Q9Y4H2 IRS2</bp:name>
   <bp:name rdf:datatype="http://www.w3.org/2001/XMLSchema#
      string">IRS2</bp:name>
   <bp:xref rdf:resource="#UnificationXref129" />
   <bp:comment rdf:datatype="http://www.w3.org/2001/
      XMLSchema#string">FUNCTION May mediate the control
      of various cellular processes by insulin.</bp:
      comment>
  </bp:ProteinReference>
  ...
</rdf:RDF>
```

Listing 2.1. The *PI3K/AKT pathway* represented through a BioPAX description.

Systems Biology Markup Language (SBML)

Systems Biology Markup Language Level 2 and Level 3

Any biochemical reaction can be broken down into several elements, which include rate laws and their parameters, reactant and product species, reactions, and stoichiometries. These elements can be characterized by compartments and different units to describe quantities. Biochemical reactions networks, such as $Michaelis–Menten$ *kinetics* (Equation 2.1), can be described through SBML models, in which one or more of these entities lists are represented:

- *Compartments*.
- *Species*.
- *Reactions*.
- *Parameters*.

- *Units of definition*.
- *Rules*.

Compartments. Labeled as *listOfCompartments*, a list of compartments represents a set of containers, characterized by a finite volume, in which reactions take place. *Cytoplasm* and *Nucleus* are simple examples of compartments.

Species. Labeled as *listOfSpecies*, a list of species represents a set of biological entities, such as proteins, genes, transcription factors, involved in a particular biological phenomenon. Each biological entity is supplied with a set of attributes:

- *id*, a unique identifier for the specie.
- *name*, an optional name of type string.

Reactions. Labeled as *listOfReactions*, a list of reactions represents a set of statements, in which transformations, transports or binding processes can alter one or more species. Rate laws are associated with reactions, describing the way in which these ones take place.

Parameters. Labeled as *listOfParameters*, a list of parameters represents a set od symbols, characterized by a global (at model level) and local scope (at reaction level), used to describe quantities.

Unit of definitions. Labeled as *listOfParameters*, a list of unit definitions represents a set of names used in the expression of quantities.

Rules. Labeled as *listOfParameters*, a list of rules represents a set of mathematical expressions, created from the reactions set. Rules are typically used to set parameters value, or define constraints on quantities.

As an example, a complete model of the $Michaelis–Menten$ *kinetics* in the follow, one of the best-known models for enzyme kinetics, is shown through a SBML Level 2 description (Listing 2.2). The equation involves an enzyme *E*, binding to a substrate *S*, to form a complex *ES*, which in turn releases a product *P*, regenerating the original enzyme.

$$E + S \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} ES \rightarrow EP \overset{k_{cat}}{\rightarrow} E + P \tag{2.1}$$

Constants such as k_{on}, k_{off}, and k_{cat} denote the forward, reverse, and catalytic rate, respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version5"
    level="2" version="5">
  <model name="EnzymaticReaction">
    <listOfUnitDefinitions>
      <unitDefinition id="per_second">
        <listOfUnits>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
```

```
    </unitDefinition>
    <unitDefinition id="litre_per_mole_per_second">
      <listOfUnits>
        <unit kind="mole" exponent="-1"/>
        <unit kind="litre" exponent="1"/>
        <unit kind="second" exponent="-1"/>
      </listOfUnits>
    </unitDefinition>
  </listOfUnitDefinitions>
    <listOfCompartments>
      <compartment id="cytosol" size="1e-14"/>
    </listOfCompartments>
    <listOfSpecies>
      <species compartment="cytosol" id="ES" initialAmount=
          "0" name="ES"/>
      <species compartment="cytosol" id="P" initialAmount="
          0" name="P"/>
      <species compartment="cytosol" id="S" initialAmount="
          1e-20" name="S"/>
      <species compartment="cytosol" id="E" initialAmount="
          5e-21" name="E"/>
    </listOfSpecies>
    <listOfReactions>
      <reaction id="veq">
        <listOfReactants>
          <speciesReference species="E"/>
          <speciesReference species="S"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="ES"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci>cytosol</ci>
              <apply>
                <minus/>
                <apply>
                  <times/>
                  <ci>kon</ci>
                  <ci>E</ci>
                  <ci>S</ci>
                </apply>
                <apply>
                  <times/>
                  <ci>koff</ci>
```

```
            <ci>ES</ci>
           </apply>
          </apply>
         </apply>
        </math>
        <listOfParameters>
         <parameter id="kon" value="1000000" units="
            litre_per_mole_per_second"/>
         <parameter id="koff" value="0.2" units="
            per_second"/>
        </listOfParameters>
       </kineticLaw>
      </reaction>
      <reaction id="vcat" reversible="false">
       <listOfReactants>
        <speciesReference species="ES"/>
       </listOfReactants>
       <listOfProducts>
        <speciesReference species="E"/>
        <speciesReference species="P"/>
       </listOfProducts>
       <kineticLaw>
        <math xmlns="http://www.w3.org/1998/Math/MathML">
         <apply>
          <times/>
          <ci>cytosol</ci>
          <ci>kcat</ci>
          <ci>ES</ci>
         </apply>
        </math>
        <listOfParameters>
         <parameter id="kcat" value="0.1" units="
            per_second"/>
        </listOfParameters>
       </kineticLaw>
      </reaction>
     </listOfReactions>
    </model>
</sbml>
```

Listing 2.2. The *Michaelis-Menten kinetics equation* represented through a SBML Level 2 description.

Systems Biology Markup Language qual

Boolean Networks are described through SBML *qual* models, in which two main entity lists are provided: *Qualitative Species* and *Transitions*.

Qualitative Species A qualitative species list, labeled as *qualitativeSpecies*, represents a set of biological entities, such as proteins, genes, transcription factors, involved in a particular biological phenomenon. Each biological entity is supplied with a set of attributes:

- *level*, an integer value representing the current entity state. In logic networks, the $level \in [0, 1]$.
- *id*, a unique identifier for the qualitative specie.
- *name*, an optional name of type string.
- *initialLevel*, an optional starting level defined for the qualitative specie. Default is 0.
- *maxLevel*, an optional upper bound value reachable by the level. In logic network, the $maxLevel = 1$.

Transitions Transitions represent interactions among biological entities, in which output biological entities are modified according to a set of conditions. Each transition is supplied with a (i) list of *Input entities*, (ii) *Output entities*, and (iii) *Function Terms* representing boolean rules.

Any entity belonging to the *Input entities* list represents a *qualitative specie* (reported in *qualitativeSpecies*), supplied with specific attributes:

- *qualitativeSpecies*, the existing entity id, as reported into the *qualitativeSpecies* list.
- *transitionEffect*, describes how the input entity level can be affected by the transition. Default is none, representing no changes in the level value.
- *sign*, represents the entity contribution to the transition. The contribution can be positive, negative, both or unknown.

An entity belonging to the *Output entities* list still represents a *qualitative specie*, supplied with the same set of attributes seen above, except for the *sign* attribute. The entity level is transition-susceptible, where conditional transfer functions manage how the level value can be changed. Figure 2.6 shows a simple example of transition

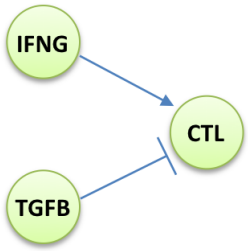


Fig. 2.6. An example of transition.

which represents the following boolean rule:

$$CTL = IFNG \wedge \neg TGFB \tag{2.2}$$

whereby *CTL* is active if and only if *IFNG* is active and *TGFB* is not active. Listing 2.3 reports the corresponding SBML *qual* description.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core"
     level="3" version="1" xmlns:qual="http://www.sbml.org/
    sbml/level3/version1/qual/version1" qual:required="true
    ">
  <model id="bio_network">
   <listOfCompartments>
    <compartment id="default" constant="true"/>
   </listOfCompartments>
   <qual:listOfQualitativeSpecies>
    <qual:qualitativeSpecies qual:compartment="default"
       qual:constant="false" qual:id="TGFB" qual:name="
       TGFB" qual:maxLevel="1"/>
    <qual:qualitativeSpecies qual:compartment="default"
       qual:constant="false" qual:id="IFNG" qual:name="
       IFNG" qual:maxLevel="1"/>
    <qual:qualitativeSpecies qual:compartment="default"
       qual:constant="false" qual:id="CTL" qual:name="CTL"
        qual:maxLevel="1"/>
   </qual:listOfQualitativeSpecies>
   <qual:listOfTransitions>
    <qual:transition qual:id="tr_CTL" qual:name="
       Interactions targeting CTL">
     <qual:listOfInputs>
      <qual:input qual:qualitativeSpecies="TGFB" qual:
         transitionEffect="none"/>
      <qual:input qual:qualitativeSpecies="IFNG" qual:
         transitionEffect="none"/>
     </qual:listOfInputs>
     <qual:listOfOutputs>
      <qual:output qual:qualitativeSpecies="CTL" qual:
         transitionEffect="assignmentLevel"/>
     </qual:listOfOutputs>
     <qual:listOfFunctionTerms>
      <qual:functionTerm qual:resultLevel="1">
       <math xmlns="http://www.w3.org/1998/Math/MathML">
        <apply>
         <and/>
         <apply>
          <eq/>
          <ci>IFNG</ci>
          <cn type="integer">1</cn>
         </apply>
         <apply>
          <eq/>
          <ci>TGFB</ci>
          <cn type="integer">0</cn>
```

```
            </apply>
          </apply>
        </math>
      </qual:functionTerm>
      <qual:defaultTerm qual:resultLevel="0"/>
    </qual:listOfFunctionTerms>
  </qual:transition>
 </qual:listOfTransitions>
</model>
</sbml>
```

Listing 2.3. The previously transition represented through a SBML *qual* description.

Systems Biology Graphical Notation (SBGN)

Systems Biology Graphical Notation (*SBGN, Le Novere et al. [74]*) provides a way to represent graphically any pathway. SBGN defines three orthogonal and complementary types of diagrams used to describe different types of pathway, as reported below:

- *Process Description* (*PD*) diagram is designed for the description of processes (e.g. , biochemical reactions) taking place in a biological system. *PD* is suitable for representing precisely metabolic networks, in which all the reactants and reactions are known. It allows to visualize chemical kinetic models.
- *Entity Relationship* (*ER*) diagram is designed for the description of all relations involving entities of a biological system. There is mainly one type of nodes, representing the objects considered, linked by relationships. These ones are independent, avoiding the problem of combinatorial explosion triggered by multi state entities. *ER* is suitable for visualizing rule-based models.
- *Activity Flow* (*AF*) diagram is designed for the description of the activity flow in a biological system. All nodes represent activities, linked by modulation arcs. *AF* is used to represent signaling pathways or gene regulatory networks, where mechanistic knowledge is missing or omitted. It is suitable for visualizing logical models.

Figure 2.7 shows all three different diagrams used to describe a specific biological pathway.

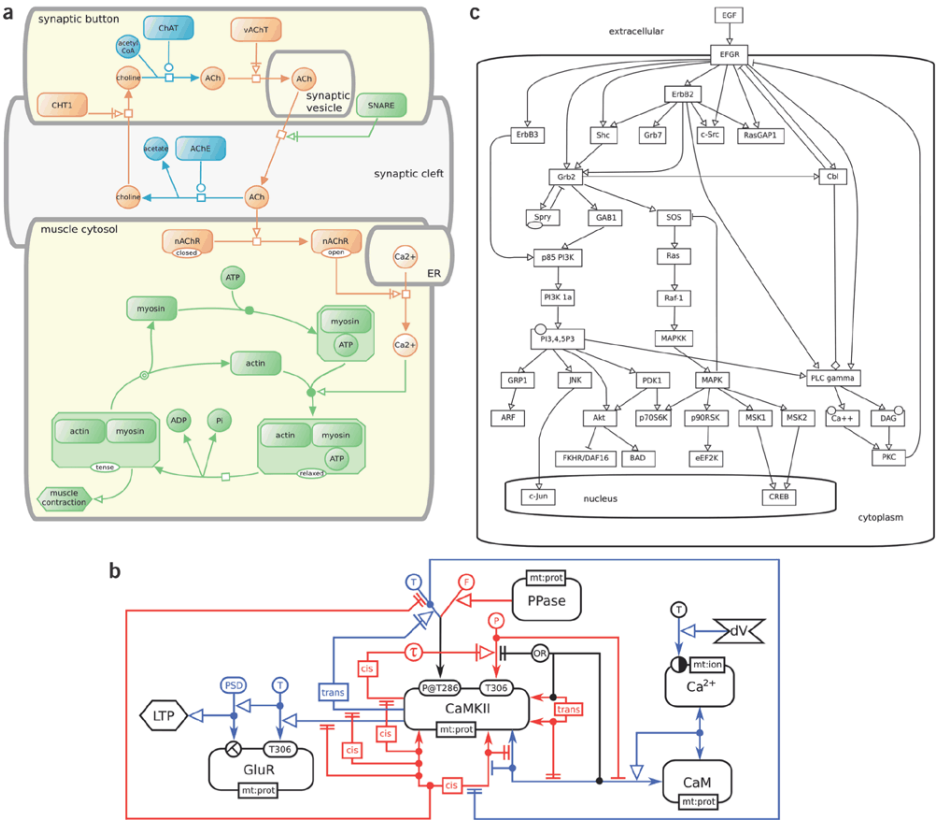


Fig. 2.7. Systems Biology Graphical Notation (SBGN) description diagrams. (a) A *Process Description* diagram representing the synthesis of the neurotransmitter acetylcholine in the synaptic button of a nerve terminal. (b) A *Entity Relationship* diagram representing the transduction, by calcium/calmodulin kinase II. (c) A *Activity Flow* diagram representing the cascade of signals triggered by the epidermal growth factor (source: *http://www.nature.com/nbt/journal/v27/n8/fig_tab/nbt.1558_F3.html*).

2.4 Models and Tools in Systems Biology

Systems modeling directly depends on the level of accuracy and details required, thus allowing to describe systems in numerous ways, such as (i) phenomenological vs. physical, (ii) discrete vs. continuous, and (iii) deterministic vs. stochastic, or even combining them in all possible combinations. Table 2.1 shows the most frequently used modeling techniques, with few examples of their application as well.

	Discrete	**Continuous**
Deterministic	Interacting particles	PDE
	Molecular dynamics	Diffusion
Stochastic	Random Events	SDE
	Population dynamics	Reaction diffusion

Table 2.1. Most common modeling techniques: (i) continuous vs. discrete models, deterministic vs. stochastic models.

As shown in Table 2.1, depending on the chosen modeling technique, different existing numerical and computational methods can be applied to model a given system.

Phenomenological vs. Physical Models. Phenomenological and physical modeling can be distinguished according to the detail level required to model a given system. In a phenomenological and physical model, an approximated description of the overall behavior is counterposed to an accurate description of the mechanistic functioning. Phenomenological modeling provides a way to analyze the system in response to known perturbations, without requiring any specific information on how the system response is brought about. In contrast, physical modeling, being strongly dependent on physic laws, it allows predicting unseen system behaviors, thus providing new insights about how the system works.

Discrete vs. Continuous Models. In discrete models, system components are represented through individual entities, in contrast with continuous models, in which quantities, such as molecular concentration and temperature, are represented in space and time. Considering a discrete perspective, in molecular dynamics simulations physic characteristics such as speed and positions are explicitly traced, and atoms are represented through a discrete entity.

Continuous deterministic modeling allows describing a wide range of phenomena, such as fluid dynamics, sound, heat, and forth. It is commonly based on partial differential equations (PDEs).

By contrast, *discrete deterministic modeling*, based on particle systems or automata, describes interacting finite entities over space and time, in agreement with deterministic rules. Cells in tissues, organisms in an environment, or molecule atoms, are modeled as interacting entities.

Stochastic vs. Deterministic Models. Biological Systems are characterized by a certain level of randomness, essentially due to (i) unpredictable environmental influences, (ii) fluctuations in molecular concentration, and (iii) noise in genes expression level. To deal with those issues, natural phenomena can be better modeled though stochastic modeling.

Basically, outputs are not entirely predetermined through the system state and its inputs, since they also depend on random fluctuations.

In *continuous stochastic modeling*, quantities evolve in space and time depending on certain probability densities. Generally formalized through stochastic differential equations (SDEs), they extend PDEs, in which stochastic terms can be used to model probabilistic processes, such as (i) epidemics spreading, (ii) evolutionary theory, and (iii) neuronal signal transduction.

In *discrete stochastic modeling*, probabilistic effects mostly refer to discrete random events, which are associated with probability density functions. Some examples of their application are population dynamics, and chemical reactions based on stochastic events.

2.4.1 Modeling Formalisms

Modeling formalisms can be classified into two main classes: mathematical models and computational models. This section provides a short overview of mathematical models, since the thesis work is focused on computational models, in particular on *Boolean Networks*.

Mathematical models

Mathematical models, such as those based on ordinary differential equations (ODE), can be used to represent a wide range of natural phenomena. They can be seen as a composition of *transfer functions*, where variation in biological quantities is correlated with each other. They model biological entities through quantitative relationships, such as molecular concentrations. Mathematical models provide the most accurate physical and biological representation for a given natural phenomenon. However, depending on the complexity and size of the modeled system, mathematical models can require high computational costs to be simulated.

Computational models

Computational procedures represent biological behavior through sequences of events. Unlike mathematical models, computational models are based on state machines, where qualitative states are related to each other. A state, typically, can be considered as the system response to certain conditions according to specific time scales. Changing in the state of a biological entity can be managed through a simple procedure that is sensible about particular events.

Despite mathematical models, computational models are characterized by a qualitative nature, allowing to deal with the absence of precise quantitative relationships between entities. In such a way, they provide many ways to describe the same biological system, even at different abstraction levels.

Even though the simplification of the biological reality, computational models are often able to reproduce the system behavior, even if they cannot describe (i) actual concentration levels nor realistic time scales, (ii) explain and predict the outcome of biological experiments that yield quantitative data. Although these limitations, given a biological network consisting of input (e.g., *receptors*), intermediate, and output (e.g., transcription factors) signals, they allow studying the input-output relationships through discrete simulations (Samaga et al. [100]).

In summary, since computational and mathematical models are based on different languages (Le et al. [73]), equations and computational procedures, respectively, they provide different kinds of insight, depending on which system aspects they are able to model.

In the last decade, different qualitative approaches have been successfully used to extrapolate insights from biological systems. Those approaches fall in the so-called *executable cell biology* (Fisher et al. [37]).

Many research efforts have been focused on redesigning well-established frameworks for modeling computational systems to be *"biology-compliant"*—notably *Process Algebra* (Ciocchetta and Hillston [20], Calder and Hillston [15]), *Ruled-based Systems* (William et al. [58], Danos et al. [23], Talcott and Dill [106], Talcott [7]), *Petri nets* (Goss and Peccoud [47], Heiner et al. [55] [54]), *Statecharts* (Harel et al. [51]), *Hybrid Systems* (Bortolussi et al. [12]), and *Boolean Networks* (Kauffman et al. [64], Thomas et al. [107]).

Process Algebra. As biological systems can be seen as high reactive concurrent systems, in which biological entities interact with each other in a concurrent way, Process Algebra can be used to formally describe those systems. It is well-suited for representing several aspects of concurrent systems, such as interactions, communications, and processes synchronizations. In general, Process Algebra acts as an intermediate model that can be further translated into other computational models, such as continuous differential equations (ODE).

Ruled-based Systems. Ruled-base modeling can be used to model variation in biochemical quantities among interacting molecular species, since it is based on notations close to chemical reactions representation. As an example, the basic *enzymatic reaction*, where the enzyme E binds a substrate S and generates the product P, can be easily represented through the following rules:

$$E + S \leftrightarrows ES \tag{2.3}$$

$$ES \rightarrow P \tag{2.4}$$

Rule-based models can be translated into (i) quantitative models, which model variation of quantities over time, and (ii) qualitative models, which do not consider the time factor, and focus only on the behavior investigation.

Petri Nets. Petri Nets have been initially designed for describing chemical processes, and widely applied to model concurrent and distributed systems. In a Petri Net, two sets of nodes are defined: (i) *transitions*, represented by a bar-based graphical notation (e.g., used to model biochemical reactions), and (ii) *places*, represented by a circle-based graphical notation (e.g., used to model biological entities, such as molecules). Nodes are combined in a directed graph, where arrows represent the flow of a described rule. In a rule, a single connection among a place and transition is established and vice versa. A black mark, called *token*, is used to represent the data (e.g., molecular species). Token is consumed in an input place and produced, through a transition, in a output place. Analyses can be done in a quantitative, by modeling the tokens evolution over the time, and qualitative, through the graph topology, perspective.

Statecharts. Statecharts is a formalism to represent a biological system in a natural fashion, in which a sequence of states describe its behavior. For example, transformations in a molecule state, from an inactive to phosphorylated condition, can be represented through a sequence of states, keeping the state assumed until the occurrence of specific events. The approach provides a way to model biological systems as an event-driven concurrent multi-scale system and at different levels (organs, tissues, cells, molecules).

Boolean Networks. In Boolean Networks, an entity can assume either *ON* (active) or *OFF* (inactive) value at a time, according to specific boolean rules (*transfer functions*). Boolean Networks are mainly involved in robustness and sensitivity analysis, for example, by perturbing a given system. Boolean Networks are characterized by a complexity delimited among static network models and continuous dynamic models. Such a complexity makes them a powerful and capable methodology to model wide biological networks. Essentially, they represent a collection of individual elements and regulatory interactions grouped all together in a rational network representation. In such a way, they can be used to depict the system qualitative behavior in a temporal way, as well as investigate how perturbations could affect its behavior.

A boolean rule maps a vector of *k* elements, in which each element can assume value $\{0, 1\}$, to a binary output, as shown below:

$$TF : \{0, 1\}^k \rightarrow \{0, 1\} \tag{2.5}$$

The boolean rule *TF* establishes how to calculate the output boolean value according to a set of *k* inputs.

In summary, a Boolean Network model represents a set of boolean entities (variables) $\{\phi_1, \phi_2, \ldots, \phi_n\}$ and a set of boolean rules $\{TF_1, TF_2, \ldots, TF_n\}$. Each ϕ_i variable changes its own boolean value according to the value of other variables (its inputs) mapped through the boolean rule TF_i.

Section 2.4.2 reports mainly used tools based on logic models.

A distinctive property of formal state-based models is their amenability to automatically derive properties from methods, using static analysis and *model checking* (Nielson et al. [89], Dubrova et al. [30]). Static analysis is a common and potentially very sophisticated automated technique for deriving properties of software programs without actually running them, through analysis of the way in which model descriptions are put together syntactically. Various kinds of static analysis can, for example, determine dependencies between parts, find range limits on variables representing state elements, or infer properties of dynamically created data structures in computer memory.

Model checking (Baier and Katoen [4]) is a powerful algorithmic technique for automatically proving certain properties of state-based models. Widely used in analysing hardware and software systems, model checking algorithms are capable of handling (some) system descriptions with very large sets of reachable states. In System Biology, the use of model checking methodology is justified by the opportunity to qualitatively or quantitatively infer model properties for all possible executions, in contrast with numerical simulation, where a single solution is available.

At different physical scales, experimentalists pose disparate scientific questions and make different observations; consequently, several disparate models can be explored. In particular, the choice of what counts as a *state* is conditioned by the available experimental techniques. The intrinsic complexity of biological systems is so high that all models result in a partial description. Achieving an executable model may therefore come at the cost of embodying unwarranted qualitative assumptions about system mechanisms.

In Fisher and Henzinger [38], design models through state machines *requires to think in terms of cause and effect rather than rates of change*; in Priami [95], model using algorithms forces to adopt a position on the causal mechanisms that generate trajectories. In Kohl et al. [67], the model prediction accuracy is questioned, since the observed behavior does not mean the described model mechanisms are responsible for the system under study.

2.4.2 Application tools

Quantitative Modeling

CoPaSI. COPASI (COmplex PAthway SImulator, Hoops et al. [59]) combines standards methodologies and widely used biological systems description, such as SBML (Hucka et al. [60]) and CellML (Lloyd et al. [79]), for the simulation and analysis of biochemical reaction networks. COPASI supports non-expert users by, for example, automatically converting reaction equations to the appropriate mathematical formalism (ODEs or reaction propensities). COPASI supports methodologies for the deterministic (integration of ordinary differential equations (ODEs)) and stochastic (e.g., using the Gillespie's algorithm (Gillespie et al. [42]) simulation of reaction networks, the computation of steady states and their stability, stoichiometric network analysis, e.g., computing elementary modes (Schuster et al. [103]), sensitivity analysis (metabolic control analysis (Fell et al. [35], Heinrich et al. [56]), optimization and parameter estimation.

CellDesigner. CellDesigner (Funahashi et al. [39]) is a modeling tool of gene-regulatory and biochemical networks. CellDesigner supports users to easily create such networks, using solidly defined and comprehensive graphical representation (SBGN). CellDesigner is systems biology markup language (SBML) compliant, and has Systems Biology Workbench(SBW)-enabled software so that it can import/export SBML-described documents and integrate with other SBW-enabled simulation/analysis software packages. CellDesigner also supports simulation and parameter search, which is supported by integration with SBML ordinary differential equation (ODE) Solver, enabling users to simulate through a sophisticated graphical user interface. SBML models can also be browsed and modified with references to existing databases.

SQUAD. SQUAD (Di Cara et al. [28]) is a software for the dynamic simulation of signaling networks using the standardized qualitative dynamical system approach. SQUAD converts the network into a discrete dynamical system, and it uses a binary decision diagram algorithm to identify all the steady states of the system. Then, the software creates a continuous dynamical system and localizes its steady states which are located near the steady states of the discrete system. The software permits to make simulations on the continuous system, in the form of a set of ODEs, allowing for the modification of several

parameters. SQUAD includes a framework for perturbing networks in a manner similar to what is performed in experimental laboratory protocols, for example by activating receptors or knocking out molecular components.

Advantage and Limitation of Quantitative Models and Tools

Quantitative modeling describes the most accurate and strongly knowledge-dependent methodologies, able to mimic precise system dynamics. However, considering the increase in complexity and quantity of experimental data, develop accurate and trustworthy quantitative models can become an arduous task. Experimental data quantity has to be evaluated objectively and unknown model parameters have to be estimated.

Qualitative Modeling

BoolNet. BoolNet (Müssel et al. [87]) is an R package for the generation, simulation, and analysis of synchronous, asynchronous, probabilistic and time-delayed networks. A boolean network can be constructed from a time series of measurements. Networks can also be created from a series of Boolean rules. BoolNet also features extensive tools for analysis of boolean networks. Random networks can be generated to compared the properties of real biological networks to random networks. BoolNet can also find one or all the attractors of a network and the transition states needed to reach the attractors. Beyond simple steady states, BoolNet can find complex, and asynchronous attractors. The transition state analyses can be plotted visually. SBML networks can be imported and exported. Only SBML *qual* description models are supported.

GINsim. GINsim (Gonzalez et al. [45]) is a Java software suite devoted for the qualitative modeling, analysis and simulation of genetic regulatory networks. The approach relies on discrete mathematical and graph-theoretical concepts. GINsim encompasses an intuitive graph editor, enabling the definition and the parameterization of a regulatory graph, as well as a simulation engine to compute the corresponding qualitative dynamical behavior. The approach for the modeling and analysis of regulatory networks relies in the logical formalism previously developed by Thomas and colleagues [108], and Thomas et al. [109]. It is based on the definitions of: (i) logical regulatory graphs to describe regulatory interactions between genes and (or via) their products, and (ii) state transition graphs to represent the qualitative dynamical behavior associated with a given regulatory graph, for given initial states.

CellNetAnalyzer. CellNetAnalyzer (Klamt et al. [66]) is a toolbox for MATLAB that provides a comprehensive structural analysis of metabolic, signaling and regulatory networks in an interactive and visual manner. The particular strengths of CellNetAnalyzer are methods for functional network analysis, i.e., for characterizing functional states, for detecting functional dependencies, for identifying intervention strategies, or for giving qualitative predictions on the effects of perturbations. CellNetAnalyzer extends its predecessor FluxAnalyzer (originally developed for metabolic network and pathway analysis) by a new modelling framework for examining signal-flow networks.

Genetic Network Analyzer. Genetic Network Analyzer (GNA, De Jong et al. [26]) is a computer tool for the qualitative simulation of genetic regulatory networks. GNA em-

ploys piecewise-linear (PL) differential equation models that have been well studied in mathematical biology (Glass and Kauffman [43], Snoussi [104], Mestl et al. [83]). While abstracting from the precise molecular mechanisms involved, the PL models capture essential aspects of gene regulation. Their simple mathematical form permits a qualitative analysis of the dynamics of the genetic regulatory systems to be carried out. Instead of numerical values for parameters and initial conditions, GNA asks the user to specify qualitative constraints on these values in the form of algebraic inequalities. Unlike precise numerical values, these constraints can usually be inferred from the experimental literature. GNA supports a qualitative simulation method that recasts the mathematical analysis of PL models of genetic regulatory networks in a computational form (De Jong et al. [25]).

BooleanNet. BooleanNet (Albert et al. [2]) is a software toolbox that greatly facilitates the implementation and study of Boolean dynamic models of biological systems. It is a tool that can simulate a Boolean model based on a very simple text based input describing the interactions and regulatory relationships in the system. The main distinguishing feature compared to previous efforts is that it aims to provide support for modeling the dynamic behavior of well defined biological sub-systems, rather than focusing on a larger scale network inference, analysis or modeling based on high throughput data. Once the rules are expressed the software can employ several simulation strategies: synchronous iterations, stochastic updates or hybrid modeling via a system of piecewise linear differential equations. More importantly the system allows the integration of non-boolean mechanisms into the simulation thus expanding its applicability to a wider domain. Every aspect of the simulation process may be customized: node states may be overridden at different stages of the operation, updating rules may be altered, and differential equations may be augmented or replaced.

Updating Policies

Updating policies can be synchronous or asynchronous. In a synchronous updating policy, all tools uniquely adopt a global updating strategy for nodes evaluation. In an asynchronous updating policy, most of the tools rely on random updating strategies, such as *Random Order Asynchronous (ROA)* (Harvey et al. [53]), *General Asynchronous (GA)* (Harvey et al. [53], Chaves et al. [17], Saadatpour et al. [97]), *Priority Class* (Thomas et al. [108], Fauré et al. [34]), and *Ranked Asynchronous(RA)* (Chaves et al. [17]).

- **Random Order Asynchronous (ROA).** Considering a network of N nodes. All nodes are updated at the same time step, but in a random order, such that no node is updated twice in the same time step. In the updating step, a random permutation $Q = Q_1, \ldots, Q_N$ is generated from the ordered set $\{1, \ldots, N\}$. Then, the state of node i, the Q_ith element of Q, at times $t + 1$ is calculated as follows:

$$X_i(t+1) = F_i(X_1(t_{1,i}), \ldots, X_N(t_{N,i})) \quad \forall\, i = 1, .., N$$

  where F_i is the boolean function that describes the state of node i at times $t + 1$, X_i is the state of the node i at a specific time:

$$t_{j,i} = \begin{cases} t & \text{if } Q_j > Q_i \\ t+1 & \text{if } Q_j < Q_i \end{cases}$$

This means that if the input node j has been updated at the $(t+1)$th time step, then $X_j(t+1)$ should be used in the right hand side of the equation. If an input node has not been updated (e.g., the last update was in the tth time step), then $X_j(t)$ should be used in the right hand side of the equation.

- **General Asynchronous (GA)** In this method, a randomly selected node is updated at each time step. In the updating step, a random element of the ordered set $\{1, \ldots, N\}$, i is selected. Then, the state of node i, at time $t+1$ is calculated as follows:

$$X_i(t+1) = F_i(X_1(t), \ldots, X_N(t))$$

where F_i is the boolean function describing the state of node i at time $t+1$, X_j is the state of a node at a given time step. Note that only node (i) is updated at a given time (e.g., this could lead to update the same node multiple times in a row).

- **Priority Class (PA)** The nodes are updated either synchronously or asynchronously (see *GA*) in a specific order. Each node belongs to one of the different priority classes $C_1, C_2, ..., C_p$, with $p \leq N$. Each class C_i has both a rank and a chosen updating method (synchronous or asynchronous). In the updating class, nodes with the highest ranked priority class are updated first, and are updated with the updating method chosen for the class. Classes of the same rank are updated independently and asynchronously and classes of lower rank occur after the highest ranked classes.
- **Ranked Asynchronous (RA)** It shares the same approach seen for *Priority Class*. However, a *Ranked Asynchronous* method adopts only the asynchronous updating method.

Each asynchronous updating method listed above has some limitation. Methods such as *GA* and *ROA*, can lead to the indiscriminate enumeration of all possible sequences of the node updating, which includes many incompatible or unrealistic pathways. This potentially leads to *biologically implausible* simulations of the qualitative networks. In contrast, methods of the PA or RA class are more realistic but, on the other hand, they rely on an a-priori knowledge (which is not always available) to categorize the network nodes in classes.

Advantage and Limitation of Qualitative Models and Tools

Qualitative modeling describes the simplest and basic knowledge-dependent methodologies, able to reproduce reasonable system dynamic properties. Proposed by Kauffman [64] and Thomas [107], the discrete logic-based dynamical models have been successfully applied for modeling biological systems, such as the cell cycle (Fauré et al. [34]), the gene regulatory system (Giacomantonio et al. [40]), and signaling networks (Schlatter et al. [101], Saez et al. [98]).

In the context of discrete logic-based dynamical models, the attractors identification and analysis, in which stable cycles of states are represented, is a dominant task. As attractors comprise the states in which biological network dwells most of the time, they can be often linked to phenotypes (Kauffman [65], Li et al. [77]).

Due to the increasing interest, different tools have been designed to investigate these properties. However, they still show important limitations: (i) they do not support the simulation complexity of large networks, (ii) the attractors analysis in asynchronous simulations is limited to networks of few tens of nodes and can not be exhaustive, (iii) they

are designed and customized for specific network type only, such as signaling, gene regulatory, or metabolic networks, and, finally, (iv) the absence of automatism for network drug-based perturbations limits their applicability to complex biological systems.

Last but not least, supported input description models, such as SBML, play an important role on how a biological system can be better modeled. Listed tools mostly support the SBML *qual* description model for modeling qualitative biological systems. Using SBML *qual* models for logic networks limits the set of available interaction types to the only activation and inhibition. In contrast, a SBML Level 2 reactions network model provides a more detailed description, including a wider set of interactions, such as stimuli and catalysts. SBML Level 2 models provide a way to perform simulations closer to real phenomena.

Semi-Quantitative Modeling

Odefy. Odefy (Krumsiek et al. [70]) is a MATLAB- and Octave-compatible toolbox for the automated transformation of Boolean models into systems of ordinary differential equations. Models can be created from sets of Boolean equations or graph representations of Boolean networks. The Boolean models are transformed into systems of ordinary differential equations by multivariate polynomial interpolation and optional application of sigmoidal Hill functions. Our toolbox contains basic simulation and visualization functionalities for both, the Boolean as well as the continuous models. For further analyses, models can be exported to SQUAD, GNA, MATLAB script files, the SB toolbox, SBML and R script files. Odefy contains a user-friendly graphical user interface for convenient access to the simulation and exporting functionalities. We illustrate the validity of our transformation approach as well as the usage and benefit of the Odefy toolbox for two biological systems: a mutual inhibitory switch known from stem cell differentiation and a regulatory network giving rise to a specific spatial expression pattern at the mid-hindbrain boundary.

The Cell Collective. The Cell Collective (Helikar et al. [57]) is a platform that allows the world-wide scientific community to create these models collectively. Its interface enables users to build and use models without specifying any mathematical equations or computer code - addressing one of the major hurdles with computational research. In addition, this platform allows scientists to simulate and analyze the models in real-time on the web, including the ability to simulate loss/gain of function and test what-if scenarios in real time.

Advantage and Limitation of Semi-Quantitative Models and Tools

Semi-quantitative modeling reports methodologies able to replicate system dynamics with a good level of accuracy, even if kinetic data are unknown or incomplete. Semi-Quantitative modeling combines qualitative and quantitative models, such as those based on discrete states and ODEs, to capture the behavior of those systems that show a switching nature. An example is given by those systems that show a sigmoid trend, such as Hill-based kinetics systems.

3

Modeling and Simulation of Embedded Systems

This chapter describes general concepts regarding embedded systems, used to introduce the context on which the thesis work is focused on.

Section 3.1 describes basic concepts regarding the *Electronic Design Automation*, whose purpose is to design electronic products of all kinds.

Finally, Section 3.2 presents *SystemC*, one of the most popular embedded system modeling description languages. It describes main *SystemC* components, especially for who is not familiar with these concepts.

3.1 Electronic Design Automation

The design of embedded systems usually follows an approach based on different abstract levels. In general, an engineer uses a top-down methodology, in which levels are described from a high level (behavioral level) to a low one (geometrical level). Translations from an abstract level to another is called *synthesis*, usually performed through automatic tools. Depending on the complexity level of systems, they can even integrate a mixing of hardware and software components, as *System-on-Chip* (*SoC*) and embedded systems. The higher level of complexity requires a higher level of synthesis, usually called *system level*, in which there is no difference between hardware and embedded systems. At this level, a system describes interconnected components as independent subunits that communicate with each other through messages, representing logic values. In this context, *SystemC* represents a suitable language for system level modeling. It relies on the flexibility of *C++* and its typical features.

3.2 SystemC

In last years, *SystemC* has become a relevant alternative to the conventional hardware and systems description languages *VHDL* and *Verilog*. In industry, *SystemC* is widely used, being supported by several tools for synthesis and design automation.

SystemC represents a language to describe hardware and software components as parts of the same system. In detail, *SystemC* is not properly a language, but rather a *C++* class library. It provides structures for modeling hardware components and their interactions,

positioning *SystemC* as an equivalent hardware description language like *VHDL* and *Verilog*. All these languages share a common aspect: they have a simulation kernel that allows to evaluate the system behavior through simulation. Despite *VHDL* and *Verilog*, *SystemC* provides a richer set of port and signal types, as well as data types. This represents an important aspect when different abstract levels must be achieved. A *SystemC* description consists of:

- *Modules*. They represent the *SystemC* basic containers. Being hierarchical, a module can contain other modules and processes.
- *Processes*. They provide a way to describe functionality.
- *Ports*. They allow modules to communicate with each other.
- *Signals*. They provide a way to exchange data.

Time Component

In *SystemC*, modeling of time represents one of the most important components, and it is provided through the data type (a class) `sc_time`. There is a minimum representable amount of time (*time resolution*) and all times under this "resolution" are rounded to zero. Time resolution restricts the maximum amount of simulated time, since time is represented using a 64-bit integer value. The `sc_time` class provides a constructor that uses the following form:

```
sc_time(double, sc_time_unit)
```

where `double` represents the amount of time and `sc_time_unit` depicts an enumerated type (equivalent `enum` in C/C++). `sc_time_unit` provides attributes as follows:

- `SC_FS` for femtosecond.
- `SC_PS` for picosecond.
- `SC_NS` for nanosecond.
- `SC_US` for microsecond.
- `SC_MS` for millisecond.
- `SC_SEC` for second.

Listing 3.1 shows a C++ statement that creates a *time1* instance of type `sc_time`, representing an amount of time equal to 60 seconds.

```
sc_time time1 = sc_time(60, SC_SEC);
```

Listing 3.1. A *time1* instance of type sc_time representing 60 seconds.

Module Component

Modules represent the *SystemC* fundamental building blocks. Essentially, a module consists of concurrent processes, ports, channels, and internal data structures. Described through the macro `SC_MODULE`, they extend `sc_module`, a base class for all *SystemC* modules. Each module constructor, declared through the macro `SC_CTOR`, requires a mandatory argument. This argument represents the *name* that has to be associated with the generated instance. Listing 3.2 shows a simple "toy" module template, in which *InstanceT1* represents the name associated to the instance.

```
SC_MODULE(ToyModule) {
  /*
   Internal data structures, processes, ports, ...
   */
  SC_CTOR(ToyModule){
    /*
     Processes declaration, ...
     */
  }
};

// ToyModule instace declaration
ToyModule instanceToy1("InstanceT1");
```

Listing 3.2. A "toy" module template supplied with the relative constructor.

Sometimes it is necessary pass more parameters to the constructor. In this case, constructors must be declared using the `SC_HAS_PROCESS` macro. Listing 3.3 extends the previous template using the `SC_HAS_PROCESS` macro and passing two parameters (integer type).

```
SC_MODULE(ToyModule) {
  /*
   Internal data structures, processes, ports, ...
   */
  SC_HAS_PROCESS(ToyModule);
  ToyModule(sc_module_name name, int parameter1, int
      parameter2) : sc_module(name) {
    /*
     Processes declaration, ...
     */
  }
};

// ToyModule instace declaration
ToyModule instanceToy1("InstanceT1", 2, 3);
```

Listing 3.3. A "toy" module template supplied with the relative constructor.

Processes

In *SystemC*, functionality is carried out through processes. *SystemC* module member functions, processes are used to provide a mechanism to simulate concurrency. Declared inside the constructor, processes use two macros to be defined: `SC_METHOD` and `SC_THREAD`. Processes declared through each macro differ as follows:

- **`SC_METHOD`**. A `SC_METHOD` process can not contain any `wait` statement. It executes its task up to the end, without any interruption.

- **SC_THREAD**. Run only once at the beginning of a simulation, a SC_THREAD process can suspend its activity through the statement wait, letting another process to continue the simulation. For example, a given thread suspends itself for 10 seconds, as shown below:

```
wait(10, SC_SEC);
```

SC_METHOD and SC_THREAD processes can be triggered by specific set of *events*, making these processes ready to be executed. A *sensitivity list* provides a way to set processes able to be triggered through a set of events (Listing 3.4).

```
// A SC_METHOD sensibles to two events
SC_METHOD(MyMethod);
sensitive << event1 << event2;
```

Listing 3.4. Sensitivity list declaration.

Listing 3.5 reports the syntax how to use an event *ev* (sc_event type):

```
// Event declaration
sc_event ev;

// Event immediately triggered through the "notify"
   function
ev.notify();

// Event triggered after 10 seconds through the "notify"
   function
ev.notify(10, SC_SEC);

// Event removed
ev.cancel();


// A SC_THREAD process waits until the ev event is
   triggered , instead of use a timed waiting
wait(ev);
```

Listing 3.5. How to declare, trigger, cancel and wait an event.

Only waits and events notification push forward the simulated time.

Channels and Ports

SystemC provides different type of channels. sc_signal represents one of the basic channels, and any change in the channel value is effectively available after the end of a delta cycle, in order to achieve concurrency. Delta cycle does not represent a clock cycle and no time is advanced. Delta cycle is used to simulate new updates and event, triggered processes to be simulated from current execution phase. *SystemC* provides the following syntax to declare a signal:

```
sc_signal<T> signal_name;
```

where `T` represents the signal data type. Listing 3.6 reports the signal declaration, writing, and reading.

```
// Declares a signal sig1 of type integer
sc_signal<int> sig1;

// Writes the integer 2 on sig1
sig1.write(2);

/*
 Moves forward the simulation of a delta cycle in order to
 update the sig1 value with the integer 2
 */
wait(SC_ZERO_TIME);

// Reads the sig1 value
int signal_value = sig1.read();
```

Listing 3.6. Signal declaration, writing, and reading.

Channels provide the way how modules can exchange data. Modules use ports in order to communicate with each other. *SystemC* provides ports, which act as inputs and outputs of a module. `sc_in<T>` and `sc_out<T>` are the basic most important ports available in *SystemC*, where `T` represents the port data type. Basically, a port can be interpreted as a pointer to a channel. This is achieved by connecting a port to a channel through an interface.

4

BIODEA: An Electronic Design Automation-based Framework

This chapter presents the *BIODEA* framework, a first generalizable approach for modeling and simulation of biological systems through Electronic Design Automation techniques and languages.

Section 4.1 discusses how biological entities are modeled and then translated in a standalone *SystemC* simulator. Section 4.2 is devoted to the description of *Assertion-based Verification* (*ABV*) techniques applied to investigate behaviors of interest through system parameters estimation. Finally, Section 4.3 presents how faults injection has been applied to alter the biological system behavior, in order to investigate the system robustness/sensitivity under perturbation.

4.1 Modeling of Biological Entities in SystemC

The *BIODEA* framework relies on two main concepts: (i) Biological Boolean Networks modeling and (ii) Finite State Machines (FSMs).

*FSM*s provide a way to formally represent the biological entity logic behavior in terms of (i) states (e.g., *inactive* or *active*), (ii) transitions between states, and (iii) guard conditions (i.e., boolean conditions).

Figure 4.1 shows the *FSM* template designed to model the biological entity behavior. A biological entity changes its state (i.e., a transition occurs) when the guard condition is evaluated to be true. The condition may be set on a particular reaction *event* (e.g., activation via phosphorylation, steric binding, auto-phosphorylation, cofactor synthesis or inhibition via phosphatase) generated by any upstream biological entity or on any environment status. Such a transition occurs after a *delay time*, which represents the time spent by a biological entity to encounter its targets. The parameter t represents the time elapsed, which is constantly updated during simulation, while *lifetime* represents the maximum time from the activation instant in which the biological entity carries out its biological function.

The *FSM* template includes (i) two sets of input data that can affect the biological entity behavior and its output, and (ii) one set of generated output:

- *Upstream inputs (U_S)*: They are *inputs* whose values are generated during simulation and depend on topological interactions that occur in upstream biological entities.

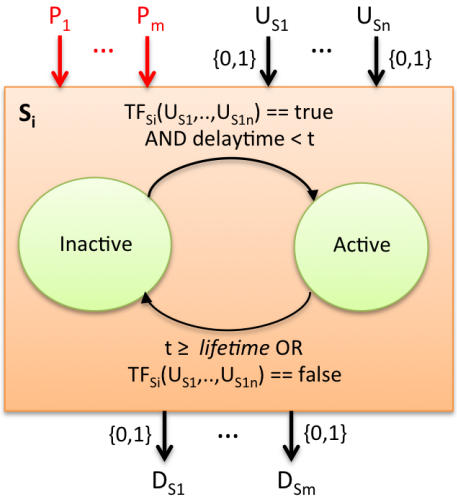


Fig. 4.1. The *FSM* template for the biological entity modeling.

Each biological entity changes its state value from *inactive* to *active* and vice versa, according to a guard condition called *Transfer Function (TF)*. A *Transfer Function* is used to evaluate the current biological entity state value, depending on the values of its input signals (U_S). Some examples are the activation via phosphorylation, steric binding, cofactor synthesis, or inhibition.

- *Parameters (P)*: They are *inputs* whose values depend on the environment characteristics and status, which are unknown at modeling time. Some examples are the *delay time* , the biological entity *lifetime*, the downstream biological entities concentrations which affect the delay time, the biological entity initial state, and so forth. An expired lifetime will force the biological entity state to be false, changing the entity state from *active* to *inactive*. For each parameter, the simulation platform generates different values with the aim of observing how such values affect the system dynamics (i.e., *parametrization*).
- *Downstream outputs (D_S)*: They are *outputs* whose values are calculated at simulation time and depend on the role of the biological entity toward downstream biological entities.

Each biological entity is implemented through a *SystemC module*, as *C++ template class*, with (i) upstream *inputs* and downstream *outputs* modeled as *SystemC ports*, and (ii) *parameters* implemented as simple state variables. The biological entity behavior represented by *FSM*, as shown in Figure 4.1, is implemented through a *SystemC process*, which is sensitive to any *event* on the input signals. An activation/inhibition from an upstream biological entity is represented by an input (boolean) signal set to *true*. Being event-driven, the process *wakes up* and updates both the internal state and the output signals whenever a new event on inputs occurs. Each network node is implemented as a *SystemC* module through processes. The element modules are finally connected and simulated at system level.

In the simplest version of the *SystemC* module, the C++ template class representing the biological entity is characterized by four parameters:

- *A*, the number of input activation signals.
- *I*, the number of input inhibition signals.
- *O*, the number of output signals.
- *F*, the biological entity *transfer function*; this parameter has type `function`, an enumeration containing all nodes transfer functions.

The *SystemC* module interface consists of:

- *inputActivations*, a `sc_vector` containing *A* ports of type `sc_in< bool >`.
- *inputInhibitions*, a `sc_vector` containing *I* ports of type `sc_in< bool >`.
- *outputs*, a `sc_vector` containing *O* ports of type `sc_out< bool >`.

Finally, the biological entity state is managed through a `sc_signal< bool >`.

Starting with the simplest version of the *SystemC* module, additional versions have been developed for qualitative and semi-quantitative simulations, each one characterized by an increasing level of modeling details. Listing 4.1 shows the C++ template declaration of the reaction reported in Figure 4.2.

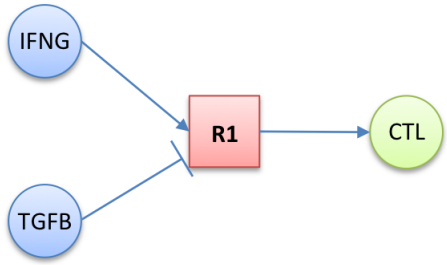


Fig. 4.2. An example of biological reaction, in which *IFNG* acts as activator for *CTL* and *TGFB* acts as inhibitor for *CTL*. *R1* represents the reaction process.

```
// IFGN declaration
entities[0] = new entity< 1, 0, 1, sIFGN > ("sIFGN");
entities[0]->in_activations[0](sig_sIFGN_sIFGN);
entities[0]->outputs[0](sig_sIFGN_pR1);

// TGFB declaration
entities[1] = new entity< 1, 0, 1, sTGFB > ("sTGFB");
entities[1]->in_activations[0](sig_sTGFB_sTGFB);
entities[1]->outputs[0](sig_sTGFB_pR1);
```

```
// R1 declaration
entities[2] = new entity< 1, 1, 1, sTGFB > ("sTGFB");
entities[2]->in_activations[0](sig_sIFGN_pR1);
entities[2]->in_inhibitions[0](sig_sTGFB_pR1);
entities[2]->outputs[0](sig_pR1_sCTL);

// CTL declaration
entities[3] = new entity< 1, 0, 0, sCTL > ("sCTL");
entities[3]->in_activations[0](sig_pR1_sCTL);
```

Listing 4.1. An example of C++ template declaration regarding the reaction reported in Figure 4.2.

BIODEA provides a set of models classified on their own features, allowing to deal with problems at different levels of complexity. Below, models are listed from the simplest to the most complex.

The *FSM* model, which is shared by each network entity, allows the corresponding *SystemC* implementation to be automatically generated from an SBML description. The automatic SBML-*SystemC* translation is achieved through a front-end parser, as described in Section 5.1.

4.2 Network Parametrization through Assertion-based Verification

In *EDA*, functional verification based on assertions represents one of the main applied and investigated techniques that combines dynamic and static verification (Bombieri et al. [10]). Assertions are a formal description of what behavior is expected during the model simulation. It allows bugs detection and it drives the test pattern generation (Boulé et al. [13]).

BIODEA applies simulation-based ABV, by which assertions are defined in *PSL*, automatically synthesized into *checkers* through the *IBM FoCs* synthesizer (Abarbanel et al. [1]), and plugged to the *SystemC* model (Bombieri et al. [10]). The checkers monitor observable signals of the model under verification during simulation and raise a failure signal when a failure is detected. In the context of signaling networks, they aim at monitoring the biological entity states, whose temporal activity is a key to understand crucial biological properties such as (i) steady states and (ii) oscillations (Samaga et al. [99]).

Figure 4.3 shows an example of state dynamics of a hypothetical biological entity to be observed and for which an assertion has be defined. The assertion describes a periodic oscillations activity (*positive edge* and *negative edge* are constant in every *active* and *inactive* state, respectively, at each oscillation) and that considers a percentage of natural tolerance (δ) in such a periodicity as follows:

```
define pos_edge:=(ta-delta) <= t & t <= (ta+delta);

define neg_edge:=(ti-delta)<= t & t <= (ti+delta);

assert (G((state=ACT)->X((state=INACT)&(pos_edge=true))) &
G((state=INACT)->X((state=ACT)&(neg_edge=true))));
```

where *ta* and *ti* are temporal counters initialized at the first oscillation, and that hold the time elapsed from the first state transition ($inactive \rightarrow active$ and $active \rightarrow inctive$, respectively).

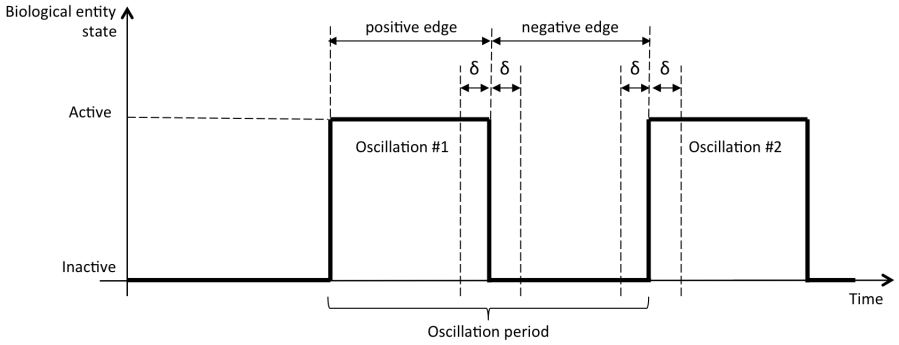


Fig. 4.3. The periodic oscillation state of a hypothetical biological entity as example of complex attractor.

They are used to measure the positive and negative edge values, respectively. The parameter t is the counter set, from the second oscillation on, at each state transition, and it is used to measure the period of subsequent oscillations to be compared with the first ones.

The assertion-based verification is applied for the *parametrization phase*, as shown in Figure 1.1, which aims at identifying the parameter settings, as described in Section 4.1, that lead the network to satisfy the property described in the assertions. The biological entity is connected to an *automatic test pattern generator (ATPG)*, which generates the parameter values. The set of all parameter values, for all biological entities, represents a *configuration*. The ATPG generates a configuration and runs (i.e., executes) a dynamic simulation of the network behavior for such a set of input values for a given simulation time. Then, the ATPG generates a new different configuration for a new simulation. The run ends when all the possible configurations have been simulated and, as a result, it generates a set of *useful configurations*, i.e., all and only configurations that lead the network to satisfy the given properties.

The useful configurations are then applied to analyses the network robustness and sensitivity, as described in the next section.

4.3 Robustness/Sensitivity Analysis through Assertion-based Verification and Mutation Analysis

In Systems Biology, *robustness* and *sensitivity* analysis represents a systematic evaluation of the network response, whenever compared with *failures*. Such an analysis goal is twofold aiming at (i) understanding the network behaviors, complexity, and its response to internal/external failures, and (ii) validating the simulation model on in-vitro/in-vivo experimental results.

Generally, robustness represents a predominant system behavior under perturbations or uncertainty conditions. Biological systems robustness, such as stability, encompasses a relative, not an absolute, property since no system can maintain stability for all its functions under perturbation. Considering simple systems, like the *BIODEA* case study described in Section 6.1, robustness is often equivalent to a dynamical regime. Genetic oscillators investigation mainly focused on the persistence of a regular periodic solution, does not preclude quantitative changes, in oscillation period or amplitude, to occur (Barkai et al. [5]). *BIODEA* provides a way to specify (i) which characteristic behavior or property should remain unchanged under perturbation through assertion-based verification, and (ii) for which type of perturbations, in a mutation model, this invariance property is held.

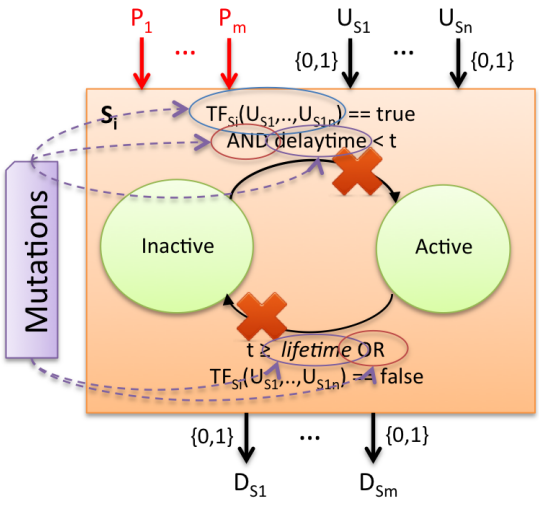


Fig. 4.4. The *FSM Mutation Model* representation and its available mutations.

Figure 4.4 shows an overview of the designed mutation model, which relies on fault injection in the *FSM* model [19] for the network biological entities. Each fault represents the real effects (mutations) of the biological entity behavior due to natural diseases, such as gene transcription alterations, cellular environmental fluctuations, and so forth. For the sake of clarity, the figure shows how the In this first stage, the mutation model implements only some of the well-known biological entities alterations, such as (i) total inactivity, (ii) alternation in the transcription function, (iii) variations in delay time and (iv) lifetime.

Figure 4.5 shows an overview of the *BIODEA* robustness/sensitivity analysis flow.

Starting with (i) all useful generated configurations during the *phase 1* and (ii) the observed properties, *BIODEA* uses this acquired knowledge as stimuli for the *Golden Model Network*, lacking in mutations, and the mutated model, in which a single mutation is activated at a time. The assertion-based verification (ABV) results are then matched to identify which mutation has generated the highest result divergence compared with the golden model. The divergence is measured by comparing the number of configurations that still lead to oscillations and how much the number of them is preserved. The generated ranking, representing the mutation sensitivity results, is analyzed to identify, among the most sensitive mutations, which one has to be experimentally reproduced (in-vitro or in-vivo).

The flow continues to analyze mutations, discriminating sensitivity results experimentally confirmed from those that do not fully represent the signaling network and, as a consequence, that have to be additionally refined.

As shown in Figure 4.5, the *BIODEA* robustness/sensitivity analysis flow is essentially divided into two main sub flows:

- an automatic flow, composed of (i) an extension version of Lisherness et al. [78] for the mutation injection through *SystemC* code, and (ii) an adapted version of Bombieri et al. [9] for automatic assertion-based verification and mutation sensitivity ranking generation.
- an in-vitro and in-vivo-based experimental flow, as highlighted in Figure 4.5.

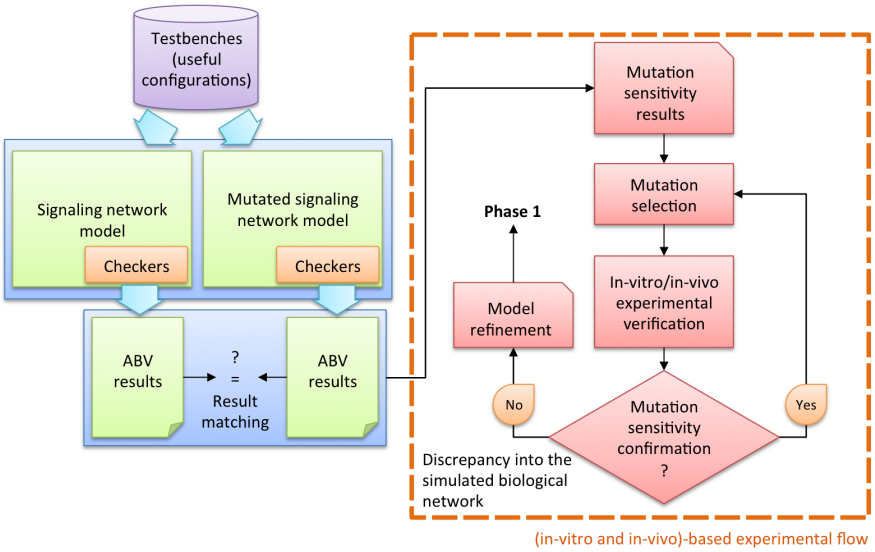


Fig. 4.5. The *BIODEA* robustness/sensitivity analysis flow.

4.4 Updating Policies

BIODEA relies on (i) a *Deterministic Asynchronous (DA)* updating method (Chaves et al. [18], Saadatpour et al. [97]) for asynchronous simulations, in which each node has a pre-selected delay time τ_i, (ii) a full support for SBML Level 2 and SBML *qual* description models, (iii) a discrete event-based framework developed in SystemC, for efficient and accurate simulations. Recent studies (Monk [84], Novák et al. [91]) have shown that the time-delayed approaches can play an important role in Biological Systems. For example, in gene regulatory networks, a changing in protein levels can be time-delayed dependent, as biochemical reactions can occur from milliseconds up to few seconds. Figure 4.6 shows an example of network used to better clarify how *BIODEA* uses the time-delayed approach in synchronous and asynchronous updating policies.

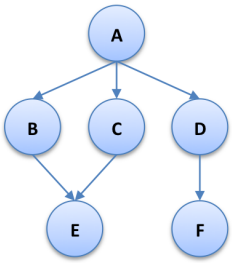


Fig. 4.6. An example of network. Node *A* activates node *B*, *C* and *D*. Node *B* and *C* activate together node *E*. Node *D* activates node *F*.

Depending on the required level of details, *BIODEA* provides (i) different temporal information, (ii) resolution (molecular concentration), and (iii) two synchronous and asynchronous updating policies:

- In a *synchronous* updating policy, the state value x of the i-th biological entity, called node, at time $t+1$, is the resulting evaluation of the logical function associated to its input nodes $i_1, \ldots, i_n$ at time *t*, as follow:

$$x_i^{t+1} = f(x_{i_1}^t, \ldots, x_{i_n}^t) \tag{4.1}$$

  All nodes states are evaluated and updated simultaneously, based on the assumption that all biological processes have the same delay time.
- In a *asynchronous* updating policy, the state value x of the i-th node is evaluated whenever an event, a change in the signal value, occurs in one or more of its inputs.

Figure 4.7 shows how entities are activated according to a specific updating policy. *Synchronous* updating policy uses a unique delay time equal for all outputs of all entities.

- **Step 1.** At time $t = 0$, node *A* gets activated.
- **Step 2.** At time $t = 1$, nodes *B*, *C* and *D* get activated.
- **Step 3.** At time $t = 2$, node *A E* and *F* get activated.

In contrast, the *asynchronous* updating policy assigns a specific delay time to each output ($source \rightarrow target$) of each entity. At each step, a specific node gets activated, depending on the specific delay time (temporal annotation). Assuming a specific delay time for each entity

Given the following list of delay times:

- $delay\_time_{A \rightarrow B} = 1$
- $delay\_time_{A \rightarrow C} = 3$
- $delay\_time_{A \rightarrow D} = 5$
- $delay\_time_{B \rightarrow E} = 8$
- $delay\_time_{C \rightarrow E} = 8$
- $delay\_time_{D \rightarrow F} = 2$

Nodes are chronologically evaluated as follows:

- **Step 1.** At time $t = 0$ node *A* gets activated.
- **Step 2.** At time $t = 1$ node *B* gets activated.
- **Step 3.** At time $t = 3$ node *C* gets activated.
- **Step 4.** At time $t = 5$ node *D* gets activated.
- **Step 5.** At time $t = 7$ node *F* gets activated.
- **Step 6.** At time $t = 9$ node *E* receives an activation signal from node *B*. However, node *E* needs an addition activation signal (from *C*) to be activated.
- **Step 7.** At time $t = 11$ node *E* gets finally activated.

4.5 Abstract Levels

BIODEA provides a set of models distinguished by (i) the available features, (ii) the supported updating policy, and (iii) the qualitative or semi-quantitative modeling approach. The *Model v1* represents the simplest one, and successive models extend its basic functionalities, in order to increase the modeling accuracy.

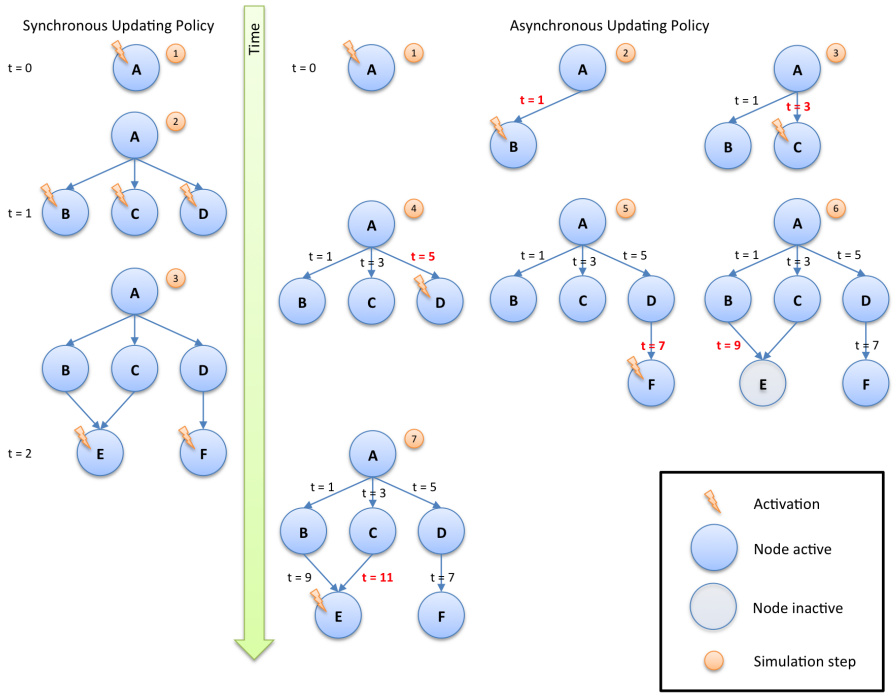


Fig. 4.7. *BIODEA*: Synchronous vs. Asynchronous Updating Policy.

Model v1

Model v1 represents the simplest representation of a model. In this model, features such as delay time and lifetime are not provided, while the *FSM* is implemented using a single `SC_METHOD` which is sensible to a global clock. Whenever the `SC_METHOD` is woken up, it evaluates its transfer function `F`, changing the node state value according to the evaluation result, resulting in the activation or deactivation of the biological entity. As a consequence, all entity outputs, defined by the D_s set, are activated or deactivated at the same time, accordingly to the node state value. The model aims at investigating, by using a zero knowledge-based approach, simple biological properties, such as (i) *stable states* and (ii) simple complex attractors, whenever feedback loops are available within the modeled network.

Model v2

Model v2 introduces the concept of delay time. Still relying on a synchronous updating policy, the model groups biological entities in classes, adding to each one a different delay time. As a results, biological entities belonging to the same class have the same delay time. Biologically speaking, this differentiation can be interpreted as different timings in the reactions execution, since biological processes take different times to be carried out.

Model v3, Model v4, and Model v5

In contrast with previous models, in *Model v3*, *Model v4* and *Model v5*, biological entities are no longer sensible to a global clock. Rather, they change their state value according to events that occur

on input signals. Biological entities use a specific delay time for each one of their inputs through (i) a user specified value, (ii) a random value chosen within a reasonable range, and (iii) a value exhaustively chosen within a reasonable range, respectively provided by *Model v3*, *Model v4* and *Model v5*. To keep down the computational cost of handling several events triggered at different times, the *SystemC* module does not use the `SC_THREAD`, subjected to an overheads in term of computational costs, but relies on two `SC_METHOD`. A first `SC_METHOD`, called `stateMethod`, is devoted to manage incoming signals defined by the U_s set, while a second `SC_METHOD`, called `delayMethod`, is devoted to manage the biological entity state and all `outputEvents`, one for each output signal. In this way, in an activated biological entity, the `stateMethod` stores the current activation timestamp, and it calls the function `notify` on each `outputEvent`. Events will trigger the activation of downstream biological entities. Any deactivation of the biological entity results in a suppression of all events (`outputEvents`) associated with its outputs. As a result of the activation, `delayMethod` verifies, for each output *i*, whether the current timestamp results equal to `lastActivationTime + delayTime[i]`; if so, the corresponding output signal will be activated.

Model v6

Model v6 introduces the concept of *lifetime*, representing the maximum amount of time after activation, in which a biological entity carries out its own biological function. Introducing the lifetime required an additional `SC_METHOD` to be managed, called `lifetimeMethod`. This method is sensible on an event, `lifetimeEndEvent`, that is triggered after the biological entity activation. The `lifetimeMethod` is in charge to manage the biological entity lifetime; as direct consequence, an expired lifetime, results in the deactivation of (i) all outputs, and (ii) the biological entity itself. Lifetime represents an important feature, allowing complex attractors observation, even in absence of feedback loops. Biological speaking, the lifetime can be considered as the amount of time before molecular degradation.

Model v7

Model v7 provides the same features seen with *Model v6*; however, it differs in how it selects the output signals to be activated. In this model, each biological entity activates only one of its outputs at a time, choosing a delay time and an output to be activated in a stochastic fashion. The stochastic approach tries to take into account molecular concentrations, used to weigh provabilities, in order to give a first simple quantitative insight.

Model v8

Model v8 represents the closest way to model reactions network. It expands the amount of available interactions such as *activation* and *inhibition*, by adding the *cofactor synthesis*. In this new type of interaction, any activation of downstream outputs will not be affected by any further upstream deactivation, allowing the outputs to be deactivated only by the lifetime expiration. Biologically speaking, biochemical reactions contribute not only to form molecular complexes (molecular binding), but also to create new biological entities (synthesis process), that are free to move around the cell space.

Model v9

As previously described, in boolean networks, the entity state can assume the *active* or *inactive* value according to its transfer function, without considering any molecular concentration. This simply

the natural phenomenon allowing the reproduction of qualitative behaviors, without requiring high computational costs. Antithetically, in a semi-quantitative model, different instances of a biological entity can establish (individually) concurrent interactions at a time, mimicking a behavior more close to the the real one. Unlike boolean models, where a *"single"* instance of a biological entity is available to establish potential interactions at a time , in a semi-quantitative model, it is possible to observe concurrency among molecular sets of biological entities. In *Model v9*, the molecular concentration of a biological entity is split into its downstream targets, according to a probability distribution, in which targets molecular concentrations are taken into account. This model provides a semi-quantitative parametric approach, adopting most of the features shown by previously models. Through the parameter *K*, *Model v9* sets the minimum *packet* size, representing the *"minimum amount"* of molecules of the same entity that can participate to any interaction. With a deeper level of detail required, a different *SystemC* interface is provided, if compared with the one developed for the qualitative modeling. In particular, the C++ template class is characterized by:

- *I*, the number of input signals, comprehending activation and inhibition signals.
- *O*, the number of output signals.
- *F*, the biological entity *transfer function*; this parameter has type `function`, an enumeration containing all nodes transfer functions.

The *SystemC* module interface is constituted by:

- *inputActivations*, *I* ports of type `sc_in< signalType >`.
- *inputDeactivation*, *I* ports of type `sc_in< signalType >`.
- *inputLinkInfo*, *I* ports of type `sc_in< signalInfo >`.
- *outputActivation*, *O* ports of type `sc_out< signalType >`.
- *outputDeactivation*, *O* ports of type `sc_out< signalType >`.
- *outputLinkInfo*, *O* ports of type `sc_out< signalType >`.

where (i) *ports* are of type `sc_vector`, (ii) `signalType` and `signalInfo` represent custom type of *SystemC* signals.

The `signalType` represents an array of messages, with type `messageType`, in which each message is characterized by the following parameters:

- *id*, a unique progressive integer number designed to identify a specific message, chosen by the sender (upstream entity).
- *size*, the amount of molecules, as a multiple of *K*, within a *packet*, that can be activated/inhibited.
- *isPartial*, a boolean flag, available for inhibition signals, used to establish if only a subset of packets, associated to a message id, are inhibited.

The `signalInfo` contains the number of the downstream entity inactive molecules and the interaction type, as shown below:

- *ACTIVATION*, steric and cofactor binding, phosphorylation.
- *INHIBITION*.
- *SYNTHESIS*.

Starting from a higher complexity, *Model v9* relies on different methods to accomplish its tasks, as described below.

The activation signal received method

Replicated for each model input signal, the method is activated whenever a changing in value occurs on the corresponding `inputActivation` signal. All messages generated by the upstream entity are stored into an activation messages queue; as direct consequence, the method notifies the received messages, by activating the *activation scheduler method*. The queue is used whenever all messages can not be processed at the same time step, for example, when all molecules of the biological entity are activated.

The deactivation signal received method

Based on the same logics described in Section 4.5, the method, however, is sensible to the corresponding `inputDeactivation` signal, and manages the activation of the *deactivation scheduler method*.

The activation scheduler method

Based on the activation messages queue and the entity transfer function `F`, the method decides which downstream target has to be activated. It decides how to split its molecules amount among its downstream targets using a random distribution, weighted on their current concentration (memorized in `outputLinkInfo`). For each split of activated molecules, a structure is stored with a set of information, as follows:

- *id*, a unique identifier number used by sent messages to the downstream target.
- *size*, the amount of molecules within the *packet*, as multiple of *K*.
- *activationTime*, the activation timestamp.
- *activators*, a list of downstream targets activated by the current split, according to the transfer function `F`.
- *target*, the downstream target that will receive the current split through a message.
- *delayEvent*, a scheduled event, associated to a delay time value, used to notify the activation of the *delay time output* method.
- *lifetimeEvent*, a scheduled event, associated to a lifetime value, used to notify the activation of the *lifetime ended* method.
- *state*, it establishes if the split is active, inhibited, or must be removed.

Other tasks accomplished by the method are: (i) updating the number of the upstream entity inactive molecules, through the `inputLinkInfo`, and (ii) checking whenever a message of *INHIBITION* type requires the deactivation of active molecules, through the activation of the *deactivation scheduler* method.

The deactivation scheduler method

As introduced in Section 4.5, a *split* can be also deactivated, according to specific conditions:

- receiving a deactivation message from any of its *upstream entities*.
- receiving an inhibition message from the *activation scheduler* method.
- receiving a lifetime expiration message from the *lifetime ended* method.

As a consequence, any received deactivation message results in the deletion of the corresponding splits, setting all molecules within those splits back to the *inactive* state. Lastly, the method sends (i) deactivation messages, through the `outputDeactivation`, to its downstream targets, and (ii) the updated count of inactive molecules to its upstream entities, through the `inputLinkInfo`.

The delay time output method

The method verifies, for each split, whether its (i) delay time is expired, and (ii) the state is still *active*; whenever those conditions are verified, the method creates a message to be sent to its downstream targets through the `outputActivation`, setting the message *id*, *size* and *target* using the information stored structure.

The lifetime ended method

The method verifies, for each split, whether its (i) lifetime is expired, and (ii) the state is still *active*; whenever those conditions are verified (`activationTime + lifetime`), the method activates the *deactivation scheduler* method.

5

SyQUAL: A Web-oriented Platform for Robustness and Sensitivity Analysis

This chapter presents the *SyQUAL*, a platform for robustness/sensitivity analysis.

Section 5.1 describes the platform architecture and its building blocks.

5.1 Platform Architecture

As briefly introduced in Chapter 1, *SyQUAL* is a web-oriented platform designed to provide a set of facilities for modeling and simulation of biological systems described through a Systems Biology Markup Language (SBML) model, and to perform automatized robustness/sensitivity analysis under specific conditions. Given its web-oriented nature, *SyQUAL* does not require any particular library or adjustment to be used, allowing the *user* to focus only on experiment setting and execution.

As shown in Figure 5.1, the *SyQUAL* platform is organized into two main blocks: *front-end* and *back-end*.

Each block consists of components that perform specific tasks, as follows:

- Front-end block
  - *SBML Importer*
  - *Pathway Detail and Initial Conditions Setting*
  - *Simulation Setting*
  - *Results*
- Back-end block
  - *SBML Validator and Data Enrichment*
  - *SBML-to-SystemC Translator*

The platform follows a modular-oriented structure. Each component prepares, separately and autonomously, all necessary support data for each other. In such a way, it is possible to keep a good level of maintenance and expandability, simplifying the inclusion of new features.

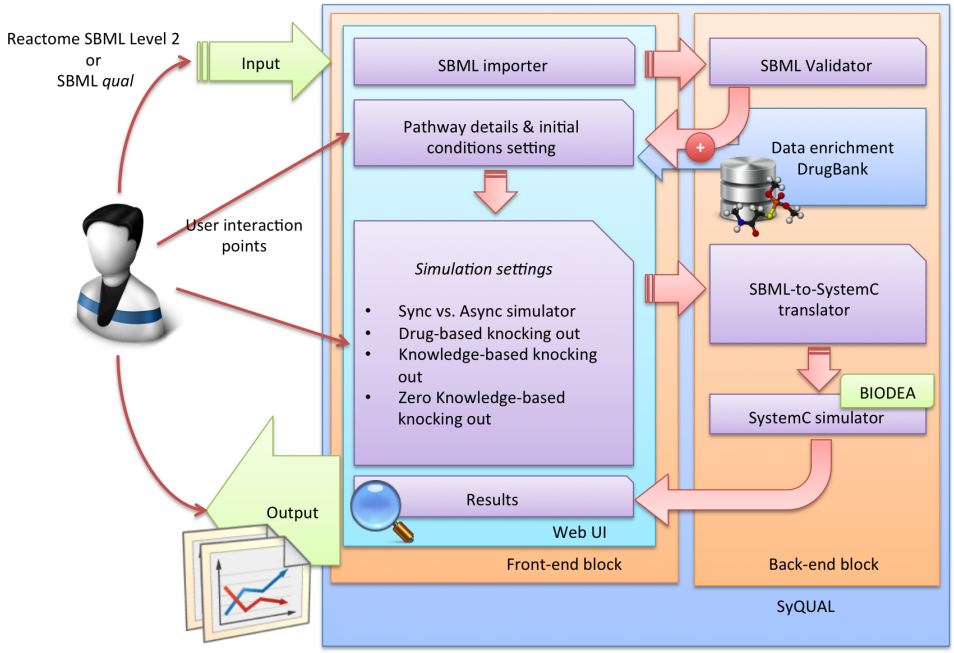


Fig. 5.1. The *SyQUAL* Platform.

Front-end Block

Basically, the *front-end block* (Figure 5.2) represents the interaction point (*user interface*) between the user and the *back-end block*. Next sections describe the user interface components.

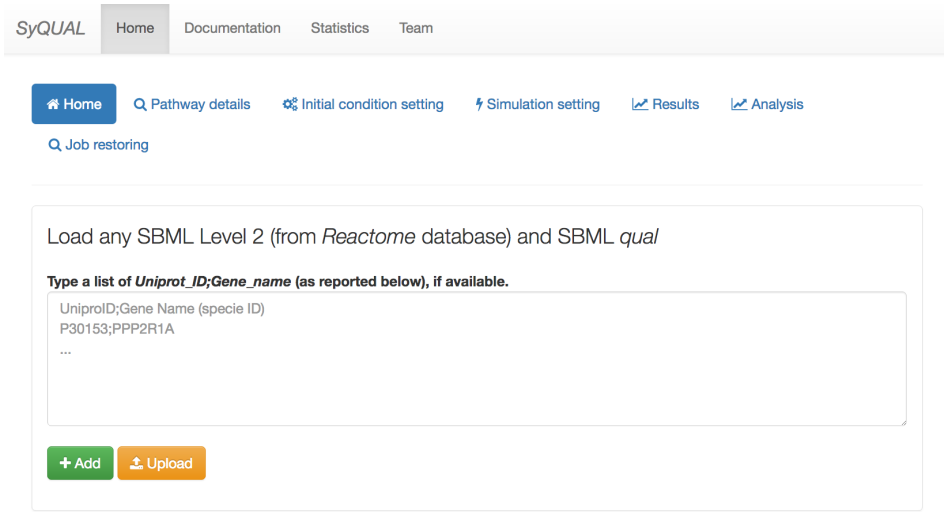


Fig. 5.2. The *SyQUAL* User Interface.

SBML Importer

This component represents the starting point in any new user-defined project. The *SBML import* component is designed to facilitate the uploading of any SBML-based model. It provides a support for the SBML *Level 2*, available through the *Reactome* database (Joshi-Tope et al. [61]), one of the most used databases for biological processes, and SBML *qual*.

Reactome is a hand-curated peer-reviewed database of biological processes, mostly (Homo Sapiens)-based. All biochemical reactions that take place in an organism are grouped under the term "reactome". These biochemical reactions represent the Reactome basic building blocks, and they are organized into networks (pathways), each one associated with a specific biological process, such as:

- Pathways of intermediary metabolism.
- Regulatory pathways.
- Signal transduction.
- High-level processes (e.g. the cell cycle).
- Neural function.
- Immune function.

In Reactome, pathways are visually represented in a full mechanistic fashion, and provided in a computational format. Pathway reactions are species-specific and experimentally verified through literature citations, otherwise they are manually inferred using non-human experiments. Reactome was not the only database taken into account. The *Kyoto Encyclopedia of Genes and Genomes* (*KEGG*) represents a pathway database designed for methodical analysis of gene functions. KEGG provides information on metabolic and signaling pathways, available for different species (from bacteria to complex organisms), represented as manually drawn pathway maps. In these maps, interacting molecules and reactions are represented through links that connect genes and their products, mostly proteins. However, KEGG showed important limitations, as reported below:

- It uses different data models to represent signaling and metabolic pathways, in which a semantic graph representation is counterposed to a chemical reaction one. This semantic graph representation takes into account only positive/negative contributions, without reporting more complex interactions, such as catalysis, and so on.
- KEGG relies on Enzyme Commission (EC) numbers used to associate metabolic reactions to physical polypeptides (gene/protein databases), leading to ambiguous and potentially erroneous assignments.

In *SyQUAL*, the SBML description can be further enriched by adding *Uniprot IDs* to the corresponding biological entities. The *Universal Protein Resource* (*Uniprot*, [21]) is a database of information on proteins. *SyQUAL* uses these *ID*s to identify potential drug targets, fundamental for the drug-based knockout (Robustness/Sensitivity) analysis.

A *user* can type any *Uniprot ID;Entity ID* list into a specific *textarea*, as reported in Figure 5.3. In particular, this list refers to the Colitis-associated Colon Cancer Pathway, as described in Section 6.2 .

To build this list, any *Uniprot ID* must be associated with the biological entity *ID*; depending on the SBML chosen, Level 2 or *qual*, the biological entity *ID* (highlighted in red) is expressed as follow:

- SBML Level 2

```
<species compartment="default" id="TGFB" initialAmount="0"
name="TGFB"/>
```

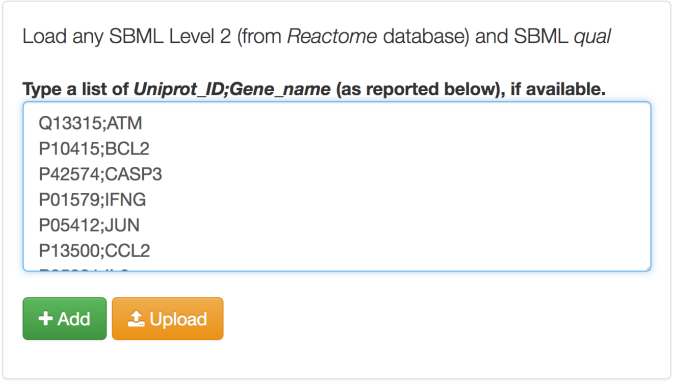


Fig. 5.3. An example of *Uniprot ID;Entity ID* list.

- SBML *qual*

```
<qual:qualitativeSpecies qual:compartment="default"
qual:constant="false" qual:id="TGFB" qual:name="TGFB"
qual:maxLevel="1"/>
```

and, the resulting list is:

```
P01137;TGFB
P35354;COX2
...
```

SBML Importer can be used to convert any SBML Level 2 description into an SBML *qual* one. Currently, this represents a unique feature.

Any uploaded SBML model generates a new job, in which the (i) *JOB_ID* (a unique job identifier), (ii) SBML *level*, and (iii) *version* are visualized, as shown in Figure 5.4.

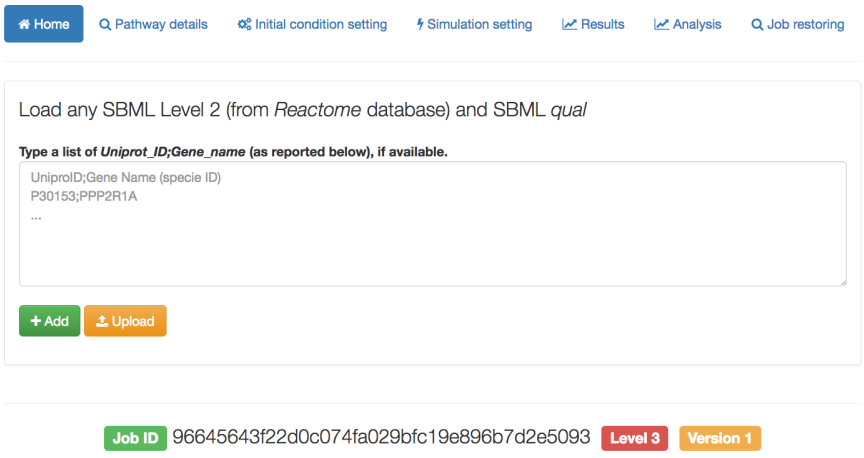


Fig. 5.4. A job generated by the submission of an SBML *qual* description.

Pathway Detail and Setting of Initial Conditions

This component shows summary information retrieved from the submitted SBML model and enriched through the *SBML Validator* component. Moreover, it provides a topological visualization of the loaded pathway and an interface to set all initial conditions, used as base for any performed experiment.

SyQUAL reports a complete list (Figure 5.5) of biological entities (e.g., genes, proteins, miRNAs) supplied by their *Uniprot IDs* if available, and a list of DrugBank (Wishart et al. [111]) drugs associated with each biological entity if available. Eventually, it reports a list of identified issues generated by the *SBML Validator*.

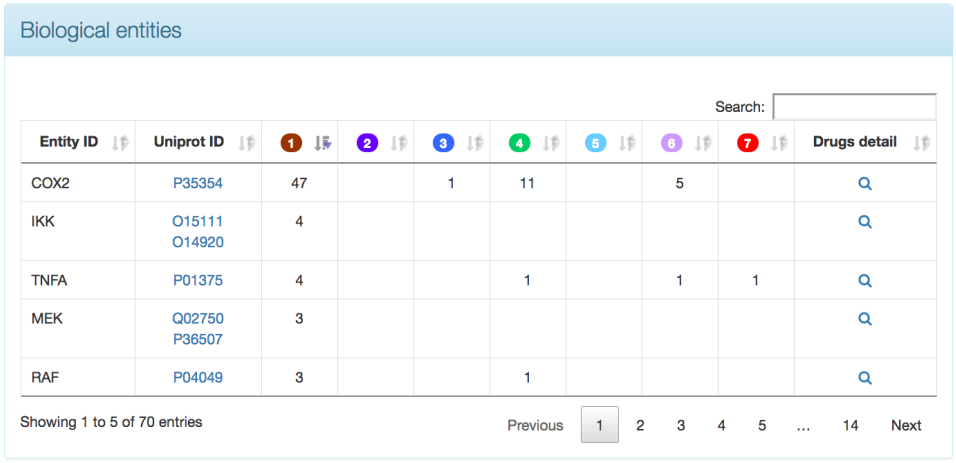


Fig. 5.5. Biological insight retrieved from an SBML description and enriched using *SyQUAL*. List of biological entities supplied with their *Uniprot IDs* and drugs. For a given biological entity, drugs are shown if and only if the biological entity represents their target.

The DrugBank database provides details of drugs (i.e., chemical, pharmacological and pharmaceutical), comprehending their targets (i.e., structure, sequence, and pathway). The provided drugs are (DrugBank version 5.0):

- 8250 drug entries including 2016 FDA-approved small molecule drugs;
- 229 FDA-approved biotech (protein/peptide) drugs;
- 94 nutraceuticals;
- Over 6000 experimental drugs.

Drugs are grouped into seven different drug categories (Figure 5.6), as follows:

- Approved drugs (according to the “Food and Drug Administration” - FDA).
- Small molecular drugs.
- Experimental drugs.
- Investigational drugs.
- Illicit drugs.
- Withdraw drugs.
- Biotech drugs.

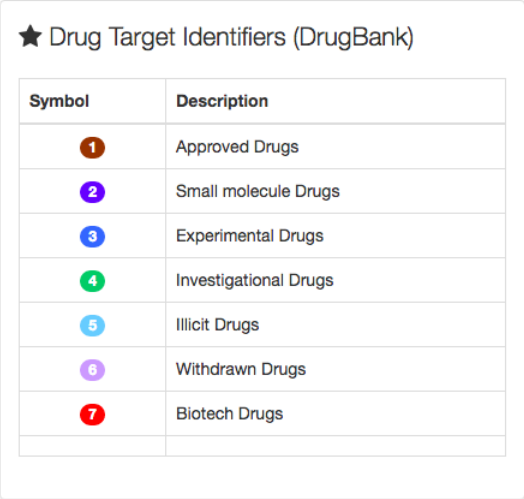


Fig. 5.6. Drugs categories and their corresponding labels, supported by the *SyQUAL* platform.

In *SyQUAL*, drugs categories are characterized by a specific color and number label. The main aim is to help users to quickly identify drug categories shown in specific parts of the *SyQUAL* platform. Each drug reports, if available, a comprehensive set of information, such as:

- Pharmacological actions (yes, no, unknown).
- Actions (inhibitor, activator, etc.).
- Drug ID, Drug name, and so on.

Drugs can be used to identify potential targets. In this first stage, *SyQUAL* carries out the drug-based Robustness/Sensitivity analysis only through those drugs that are characterized by an inhibitor biological action. Moreover, since drugs are divided into categories, *SyQUAL* helps users to perform more accurate drug-dependent experiments. Details of drugs associated with a specific target (Figure 5.7) can be retrieved by clicking on the *magnifier* icon, as shown in Figure 5.5.

Search:

DrugBank ID	Drug name	Pharm. action	Actions	General Func.	Specific Func.	Pathway's targets
DB00159	Icosapent	yes	inhibitor	Prostaglandin-endoperoxide synthase activity	Converts arachidonate to prostaglandin H2 (PGH2), a committed step in prostanoid synthesis. Constitutively expressed in some tissues in physiological conditions, such as the endothelium, kidney and brain, and in pathological conditions, such as in cancer. PTGS2 is responsible for production of inflammatory prostaglandins. Up-regulation of PTGS2 is also associated with increased cell adhesion, p...	COX2
DB00244	Mesalazine	yes	inhibitor	Prostaglandin-endoperoxide synthase activity	Converts arachidonate to prostaglandin H2 (PGH2), a committed step in prostanoid synthesis. Constitutively expressed in some tissues in physiological conditions, such as the endothelium, kidney and brain, and in pathological conditions, such as in cancer. PTGS2 is responsible for production of inflammatory prostaglandins. Up-regulation of PTGS2 is also associated with increased cell adhesion, p...	COX2 IKK
DB00316	Acetaminophen	yes	inhibitor	Prostaglandin-endoperoxide synthase activity	Converts arachidonate to prostaglandin H2 (PGH2), a committed step in prostanoid synthesis. Constitutively expressed in some tissues in physiological conditions, such as the endothelium, kidney and brain, and in pathological conditions, such as in cancer. PTGS2 is responsible for production of inflammatory prostaglandins. Up-regulation of PTGS2 is also associated with increased cell adhesion, p...	COX2
DB00328	Indomethacin	yes	inhibitor	Prostaglandin-endoperoxide synthase activity	Converts arachidonate to prostaglandin H2 (PGH2), a committed step in prostanoid synthesis. Constitutively expressed in some tissues in physiological conditions, such as the endothelium, kidney and brain, and in pathological conditions, such as in cancer. PTGS2 is responsible for production of inflammatory prostaglandins. Up-regulation of PTGS2 is also associated with increased cell adhesion, p...	COX2
DB00461	Nabumetone	yes	inhibitor	Prostaglandin-endoperoxide synthase activity	Converts arachidonate to prostaglandin H2 (PGH2), a committed step in prostanoid synthesis. Constitutively expressed in some tissues in physiological conditions, such as the endothelium, kidney and brain, and in pathological conditions, such as in cancer. PTGS2 is responsible for production of inflammatory prostaglandins. Up-regulation of PTGS2 is also associated with increased cell adhesion, p...	COX2

Showing 1 to 5 of 47 entries

Previous 1 2 3 4 5 ... 10 Next

Fig. 5.7. Drugs details, such as drug id, drug name, and general function, associated to the *COX2* target.

SyQUAL collects such acquired and enriched knowledge to provide an easy way to set experimental initial conditions. These conditions can be set through a *initial condition setting interface*, as shown in Figure 5.8.

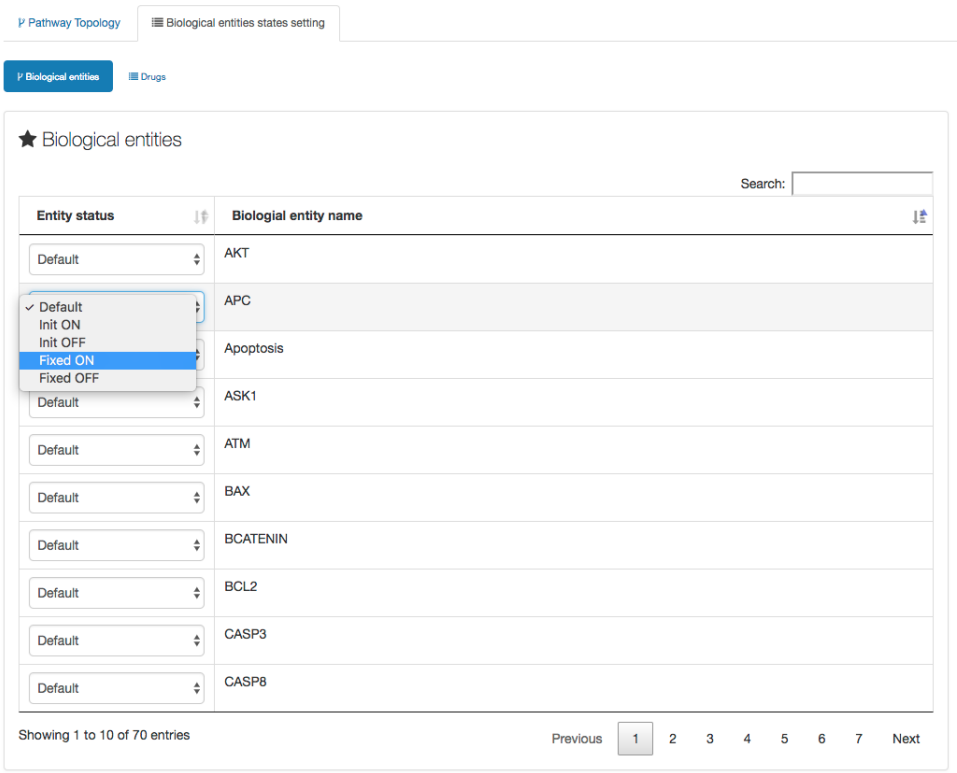
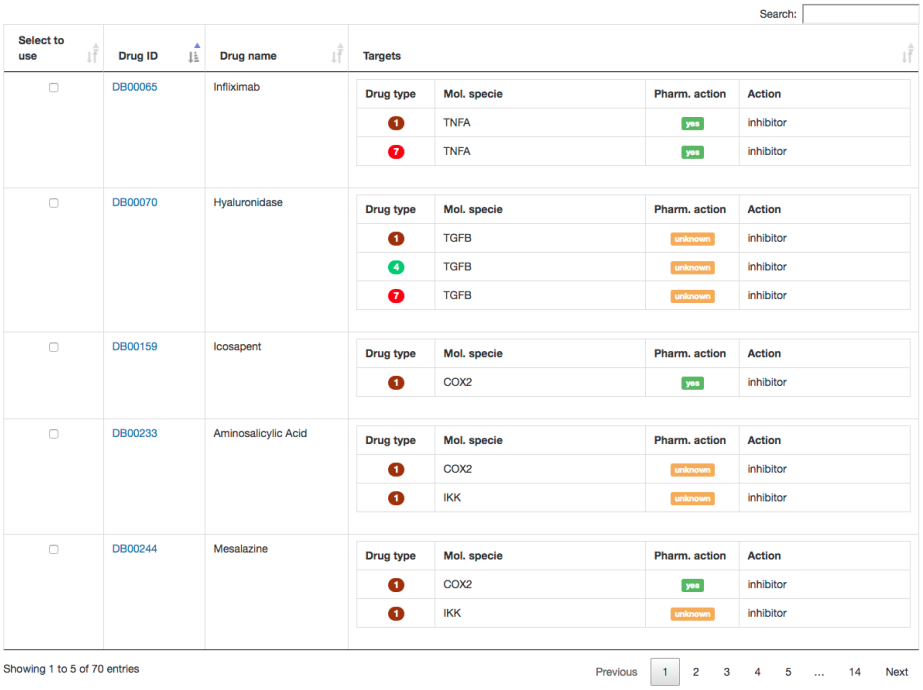


Fig. 5.8. Experimental initial conditions *setting interface*.

Experimental initial conditions can be grouped as:

- A set of stimuli to be activated. Stimuli are biological entities that act as starting points of a simulation. Each stimulus can be (i) activated only during the initial (option *Init ON*) part of a simulation, or (ii) kept always active (option *Fixed ON*) during the whole simulation. As default, no stimulus is selected.
- A set of drugs to be used (Figure 5.9). As default, no drug is used.
- A set of biological entities to be knocked out. In this case, a user can force the suppression of one or more entities. *SyQUAL* simulates the presence of hypothetical drugs, even if no drug is available. Suppression can influence the biological entity during a simulation according to two temporal points: the initial part (option *Init OFF*) or the whole time (option *Fixed OFF*).



Search:

Select to use	Drug ID	Drug name	Targets
☐	DB00065	Infliximab	Drug type: 1 — Mol. specie: TNFA — Pharm. action: yes — Action: inhibitor Drug type: 7 — Mol. specie: TNFA — Pharm. action: yes — Action: inhibitor
☐	DB00070	Hyaluronidase	Drug type: 1 — Mol. specie: TGFB — Pharm. action: unknown — Action: inhibitor Drug type: 4 — Mol. specie: TGFB — Pharm. action: unknown — Action: inhibitor Drug type: 7 — Mol. specie: TGFB — Pharm. action: unknown — Action: inhibitor
☐	DB00159	Icosapent	Drug type: 1 — Mol. specie: COX2 — Pharm. action: yes — Action: inhibitor
☐	DB00233	Aminosalicylic Acid	Drug type: 1 — Mol. specie: COX2 — Pharm. action: unknown — Action: inhibitor Drug type: 1 — Mol. specie: IKK — Pharm. action: unknown — Action: inhibitor
☐	DB00244	Mesalazine	Drug type: 1 — Mol. specie: COX2 — Pharm. action: yes — Action: inhibitor Drug type: 1 — Mol. specie: IKK — Pharm. action: unknown — Action: inhibitor

Showing 1 to 5 of 70 entries

Previous 1 2 3 4 5 … 14 Next

Fig. 5.9. Example of drug list available for the Colitis-associated Colon Cancer Network (Section 6.2).

To verify if a given submitted pathway has been correctly imported, *SyQUAL* provides its graphical representation, as shown in the example of Figure 5.10. In this first stage, *SyQUAL* uses this visualization to show the pathway topology. A future extension will provide all previously described options (*Init ON*, and so forth) through the direct interaction with the pathway graphical representation.

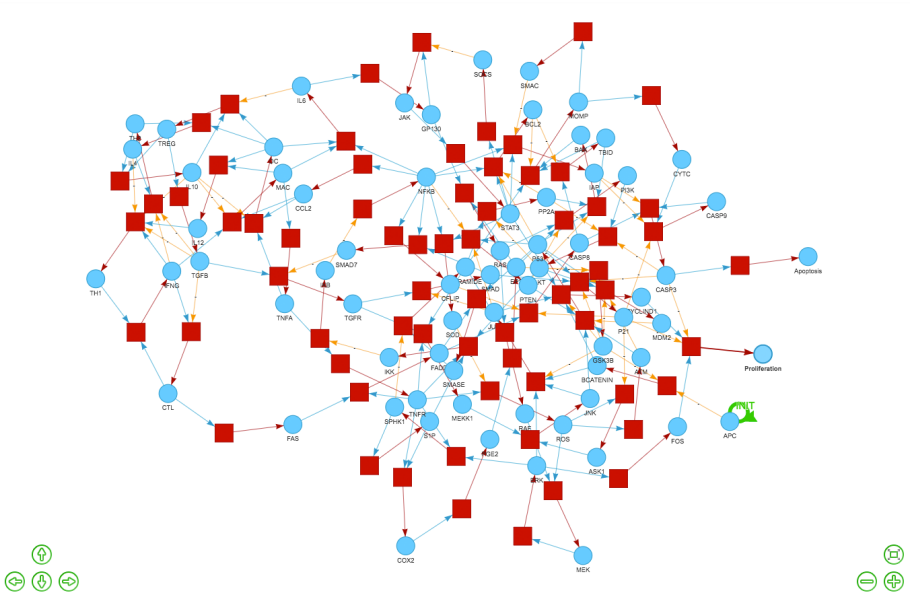


Fig. 5.10. The *SyQUAL* pathway graphical representation.

Simulation Setting

All specified experimental initial conditions are used as base of any performed simulation. *SyQUAL* provides a set of options to perform the Robustness/Sensitivity analysis, according to:

- *Zero Knowledge-based Knock Out Approach.*
- *Drug Knowledge-based Knock Out Approach.*

Zero Knowledge-based Knock Out

Any experiment performed using this approach does not require any particular information about the pathway to be analyzed. For a given pathway composed of N unique biological entities, *SyQUAL* performs N distinct simulations, in which a specific biological entity is knocked out. This approach is used when there is no knowledge about the pathway, and the user wants to investigate how its behavior can change under different knockout conditions. All simulations are performed automatically. A final report shows the normal condition compared with the perturbed ones. The complexity is linear, as it is proportional to the number of unique biological entities to be knocked out. Figure 5.11 shows the *Zero Knowledge-based Knock Out* interface.

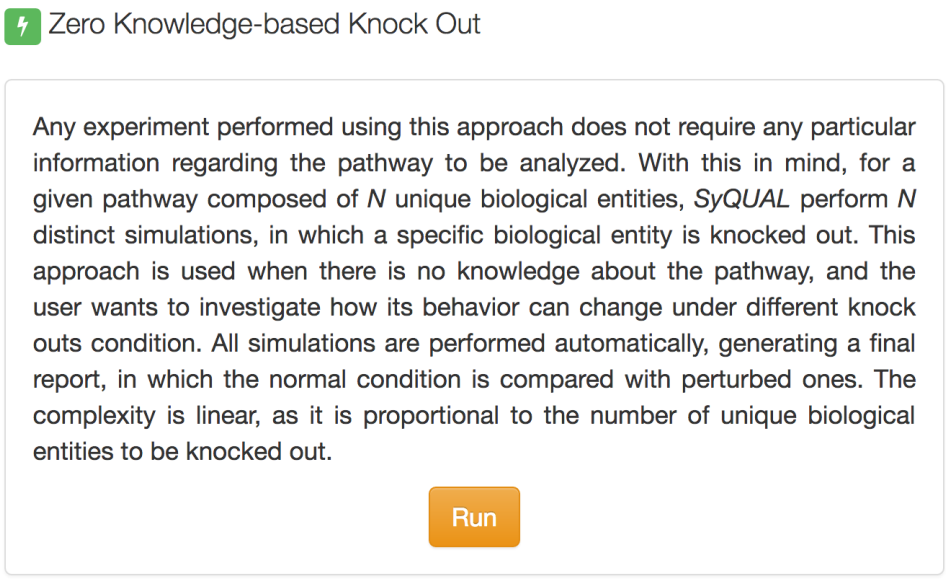


Fig. 5.11. The *Zero Knowledge-based Knock Out* interface.

Drug Knowledge-based Knock Out

Experiments performed through this approach use drugs to identify which biological entities have to be knocked out. These drugs are selected through the *Uniprot IDs*, and grouped per target. There can be one or more drugs that target the same biological entity. *SyQUAL* provides three ways to knock out biological entities using drugs: *Systematic Knock Out*, *Precise Knock Out*, and *Combinatorial Knock Out* (Synthetic Lethality, Kaelin et al. [62]).

Systematic Knock Out. The number of performed simulations is proportional to the amount of unique targets, *K*. For a given pathway of *N* biological entities, the number of simulations is:

$$K \leq N \tag{5.1}$$

Given a hypothetical pathway in which

- Drug D_1 targets biological entities E_1 and E_2.
- Drug D_2 targets biological entities E_1 and E_2.
- Drug D_3 targets biological entity E_3.
- Drug D_4 targets biological entity E_1.

the number of unique simulations *K* is equal to *3*, since $\{D_1, D_2\}$ target the same set of biological entities $\{E_1, E_2\}$, leading to the same behavior (in this stage all drugs act as inhibitors). In such a way, *SyQUAL* avoids repeating the same simulations, reducing the computational time. In particular:

- A single drug can target more than one biological entity. In this case, a single simulation knocks out all biological entities that are target for the drug.
- Multiple drugs can target the same biological entity. As a consequence, a single simulation knocks out only the targeted biological entity.

Figure 5.12 shows the *Systematic Knock Out* interface.

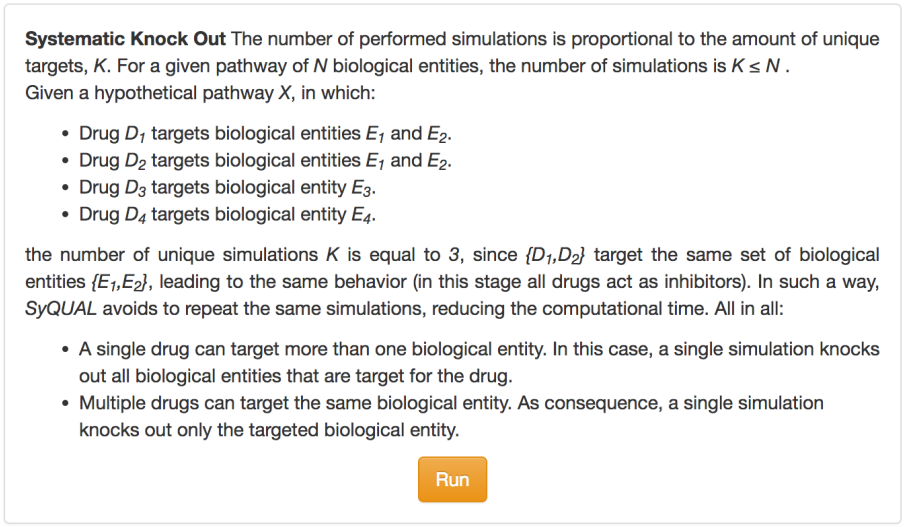


Fig. 5.12. The *Systematic Knock Out* interface.

Precise Knock Out. In its basic version, a single simulation is performed. The user manually choses a set of drugs to be used. In this simulation, a user can observe the system behavior as result of combining one or more drugs, for example, by simulating a therapeutic drug cocktail. *Systematic* and *Precise knock outs* are essentially distinguished by their unsupervised and supervised approaches, respectively. In the first one, the *SyQUAL* platform decides which biological entity must be knocked out on the i-*th* experiment, in contrast with the second one, where the user choses which biological entity must be knocked out.

In the advanced version, *Precise knock out* allows to use the same set of drugs (used to knock out biological entities) as initial condition for a set of simulations, *M*, in which a set of user-defined stimuli are combined. The number of simulations *M* is expressed as follows:

$$M = \sum_{k=1}^{h=n} C_n^k \tag{5.2}$$

where *n* represents the number of selected stimuli. Figure 5.13 shows the *Precise Knock Out* interface.

Combinatorial Knock Out. This approach performs a combination of biological entities knocked out in order to investigate the *Synthetic Lethality* on a given pathway. As reported in literature (Kaelin et al. [62]), Synthetic lethality appears when a combination of mutations in two or more genes leads to cell death, whereas a single mutated gene does not. Synthetic Lethality shows functional relationships between genes. The amount of performed simulations, *M*, is expressed as follows:

$$M = \frac{N_k!}{2(N_k - 2)} \tag{5.3}$$

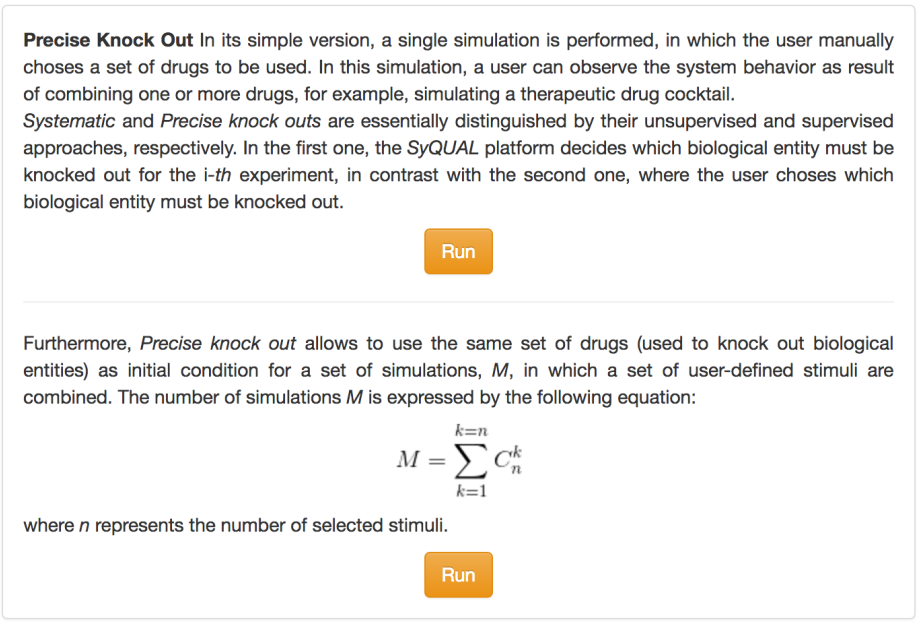


Fig. 5.13. The *Precise Knock Out* interface.

where N_k represents the number of unique biological entities knocked out. N_k depends on the set of drugs chosen, comprehending *all* of them or a *used*-specified set. Figure 5.14 shows the *Combinatorial Knock Out* interface.

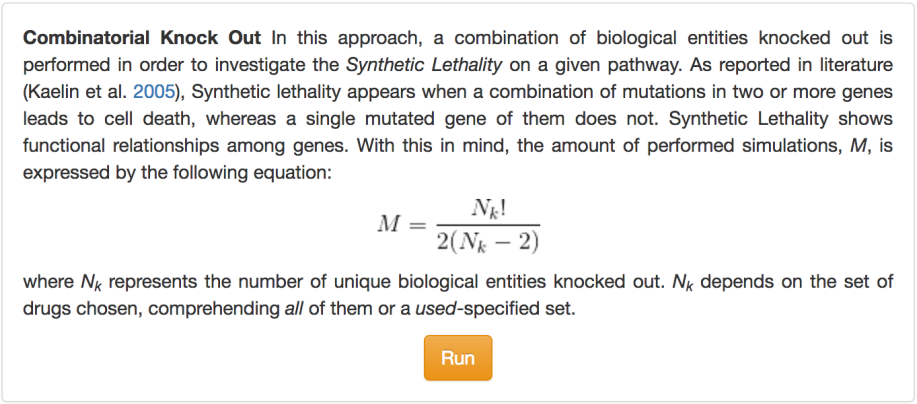


Fig. 5.14. The *Combinatorial Knock Out* interface.

Updating Policies

Zero and *Drug knowledge-based knock out* approaches (Figure 5.15) can be used according to: a *synchronous updating policy* and an *asynchronous updating policy*, based on an modified version of the *BIODEA* (Chapter 4) *Model v1* and *Model v8*, respectively. Briefly:

- **Synchronous Updating Policy**. This approach represents the simplest one. Given a boolean representation of a biological pathway, each element is evaluated by using a global clock. In this case, the time step, (hereafter called *delay time*) is equal for all biological entities. Interactions among entities are performed in a synchronous manner. For each time step, all entities states are evaluated, according to their own boolean rules (*transfer function*). Such a policy provides a fast way to investigate some basic behaviors, such as feedback loops, particular signals paths, and simple attractors.
- **Asynchronous Updating Policy**. This approach provides a different delay time for each system element output. Interactions among entities are performed in an asynchronous manner. Moreover, this policy introduces the *lifetime* concept, that is, the maximum time in which an entity can execute its biological function. As default, each element uses the same lifetime. Such a policy allows investigating more complex behaviors.

Figure 5.15 shows the synchronous and asynchronous updating policies interface.

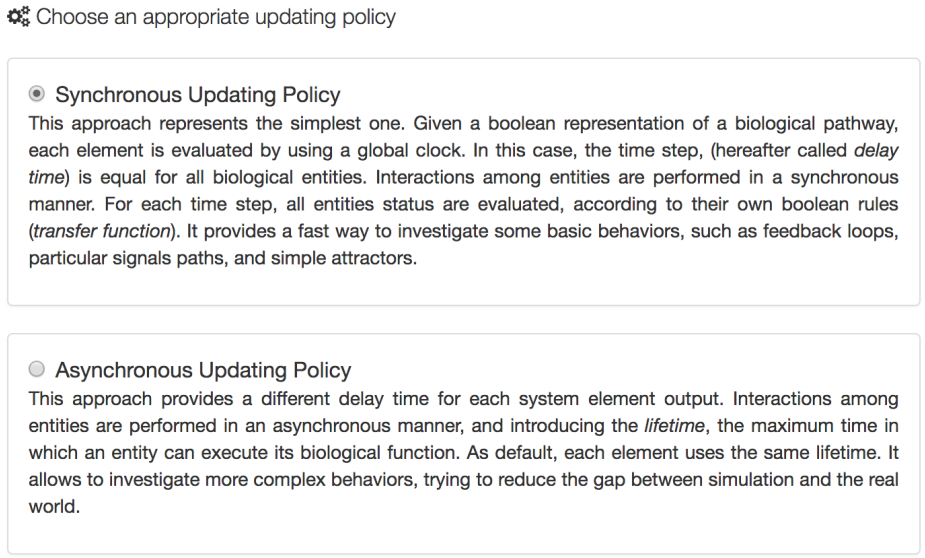


Fig. 5.15. Synchronous and asynchronous updating policies interface.

Results

SyQUAL generates a set of simulations according to a specific updating policy and knock out approach, in which one or more biological entities have been knocked out. In each simulation, biological entities change their state value from active (*ON*) to inactive (*OFF*). The term *activity level* indicates the changes in the state value over the time. The activity level is nothing more than the sequence of activations and inactivations (e.g., *OFF, ON, OFF, ...*) for a specific biological entity. For each simulation, *SyQUAL* produces:

- A *global activity level*, GAL_S, generated by summing all biological entities activity levels (*EAL*s), as follows:

$$GAL_S = \sum_{i=1}^{i=N} EAL_i \tag{5.4}$$

SyQUAL provides GAL_S expressed as percentage, P_{GAL}, as shown below:

$$P_{GAL} = \frac{GAL_S}{GAL_N} \tag{5.5}$$

where GAL_N represents the *global activity level* under normal conditions. A user can quickly identify the perturbations (Figure 5.16) that could lead to deep change (under or over expression) into the pathway behavior.

Search:

Sim. ID	Biological entity knocked out	Global Activity level	Heatmap
59	GP130	56.85%	See more
29	RAS	69.55%	See more
16	PI3K	77.19%	See more
8	AKT	79.08%	See more
15	RAF	79.31%	See more

Showing 1 to 5 of 70 entries Previous 1 2 3 4 5 ... 14 Next

Fig. 5.16. Figure shows P_{GAL} values under perturbed conditions. Each simulation is performed according to a specific biological entity knocked out (in this figure, the *Zero Knowledge-based knock out*).

- A heatmap reporting the activity levels of all biological entities. An example of heatmap according to the *GP130* knock out is shown in Figure 5.17.

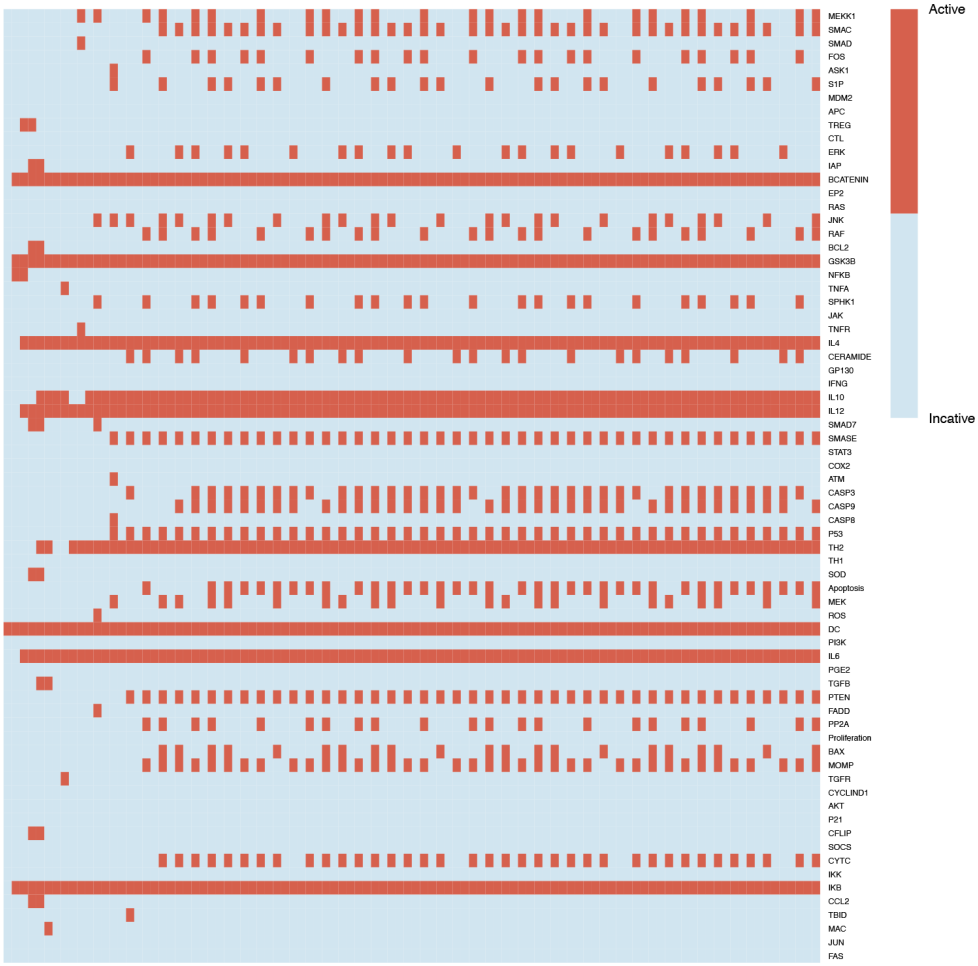


Fig. 5.17. An example of heatmap, in which the *GP130* biological entity has been knocked out. For each biological entity the state can be active (*red*) or inactive (*cyan*). Time progression from left to right.

- A set of identified attractors if available.

As additional resources, any experiment is supplied with:

- An expression profile heatmap, whether available, created by merging all simulation activity levels and clustered them per column according to the Pearson's correlation coefficient, as shown in Figure 5.18. Rows represent biological entities, while columns represent distinct experiment, in which one or more biological entities are knocked out.

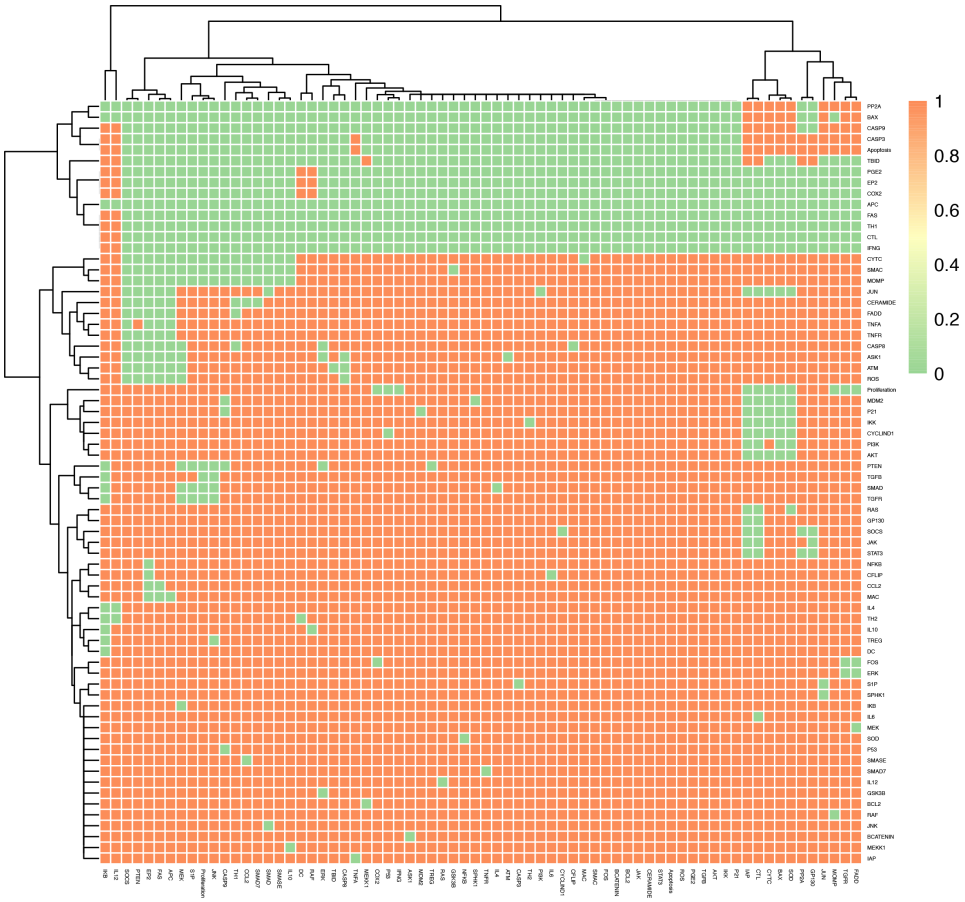


Fig. 5.18. An example of expression profile heatmap. It merges all simulation activity levels (each heatmap associated to each biological entity knocked out) and clusters them per column according to the Pearson's correlation coefficient. Rows represent biological entities, while columns represent distinct experiment, in which one or more biological entities are knocked out.

- A heatmap reporting the activity levels of all biological entities under normal conditions, as shown in Figure 5.19.

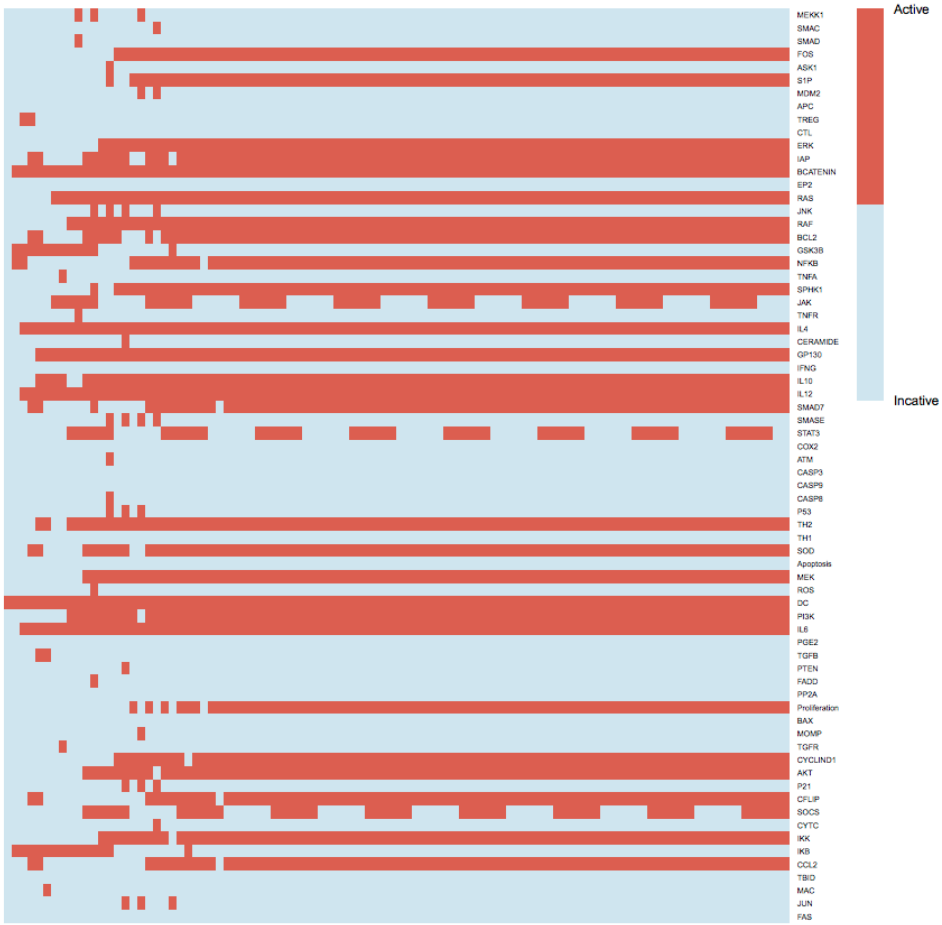


Fig. 5.19. An example of heatmap generated under normal conditions. For each biological entity the state can be active (*red*) or inactive (*cyan*). Time progression from left to right.

Each heatmap is generated through *pheatmap* (Kolde et al. [69]), a *R* package.

As additional feature, a user can examine biological entities activity levels without any graphical representation. Activity levels are expressed through a normalized (in the range [0,1]) numerical representation.

A user can investigate activity levels according to a specific (i) knock out approach, (ii) updating policy, and (iii) a set of biological entities (Figure 5.20) of interest.

Fig. 5.20. An example of selected biological entities to be investigated.

It is possible to investigate and quantify how the entity activity level is related to each other, under normal (Figure 5.21) and perturbed condition (Figure 5.22).

Normal condition (no perturbation)

BCATENIN	STAT3	IL6	IL4	IL12	IL10	Proliferation	Apoptosis
0.99	0.48	0.98	0.98	0.98	0.94	0.8	0

Fig. 5.21. An example of biological entities activity levels under normal condition.

Search:

BCATENIN	STAT3	IL6	IL4	IL12	IL10	Proliferation	Apoptosis	Biological entity knocked out
×	0.48	0.98	0.98	0.98	0.94	0.4	0	BCATENIN
0.99	0.48	0.98	0.98	0.98	0.94	0.82	0	CCL2
0.99	0.48	0.98	0.98	0.98	0.94	0.44	0.32	SPHK1
0.99	0.48	0.98	0.98	0.98	0.94	0.8	0	CYTC
0.99	0.48	0.98	0.98	0.98	0.94	0.81	0	ATM
0.99	0.48	0.98	0.98	0.98	0.94	0.8	0	SMASE
0.99	0.44	0.85	0	0.81	0	0.32	0.06	DC
0.99	0.48	0.98	0.98	0.98	0.94	0.8	0	BAX
0.99	0.48	0.98	0.98	0.98	0.94	0	0.01	AKT

Fig. 5.22. An example of biological entities activity levels under perturbed condition. Each row represents a specific simulation, in which a biological entity has been knocked out (rightmost column).

Back-end Block

SBML Validator

This component plays a key role for retrieving information from a given biological system described through an SBML model. *SyQUAL* supports the SBML *Level 2*, which is provided by the *Reactome* database, and the SBML *qual*, in order to generate a set of fundamental information for the "SBML-to-*SystemC* Translator" component. Reactome provides a well-defined knowledge about reactions for a specific biological process, compared to *KEGG* or other resources.
Supporting Reactome SBML Level 2, it is possible to construct a fully-comprehensive qualitative reactions boolean network, by keeping important details related to the molecular interaction nature, such as inhibition, stimulation, and catalysis. Any submitted Reactome pathway is analyzed to extract details such as:

- Biological entity names
- Stoichiometries
- Uniprot IDs
- Go terms
- Reactome IDs
- Kegg IDs
- Chebi IDs
- Reactions
- Reaction effects (compound, stimulation, inhibition, catalysis, production)
- Reaction signs (positive, negative)

Differently from SBML Level 2, the SBML *qual* preserves a subset of these details, by limiting the number and the quality of observable behaviors. Moreover, Reactome SBML Level 2 model reports:

- an SBML annotation, describing all involved elements, reactions and reactants.
- an SBGN (*Systems Biology Graphical Notation*) annotation, whether available, describing the system through a graphical representation.

Supporting Reactome SBML Level 2 pathways, *SyQUAL* embraces the *Process Description* (*PD*) SBGN diagram notation, for the following reasons:

- *AF* diagrams can be ambiguous.
- An *AF* diagram should be associated with either a *PD* or *ER* diagram, whether possible.
- Automatic conversion between *PD* and/or *ER* to *AF*.

The SBGN annotation plays a fundamental role to construct a fully-comprehensive network, since there are no details, for example, about inhibition and stimulation reported by the SBML annotation. *SyQUAL* creates a fully-comprehensive network by merging the SBML and SBGN annotations, when this last one is available. This represents a unique and useful feature to model all potential interactions in a given pathway.

SyQUAL tries to deal with some issues concerning the use of Reactome pathways. Since there is no a well-defined correspondence between the SBML and SBGN annotations, *SyQUAL* tries to match reactions among SBML and SBGN. Any identified issue is visualized to the user, by using the web interface.

In order to manage any loaded SBML, *SyQUAL* relies on the *LibSBML API* (Bornstein et al. [11]), in particular the Python Library version, to acquire and manipulate a SBML model. LibSBML API is available for several languages and provided with a complete documentation.

LibSBML is a free, open-source programming library to read, write, manipulate, translate, and validate SBML files and data streams.

SBML-to-SystemC Translator

SBML Validator component generates a set of resources used to facilitate the SBML annotation mapping into a complete standalone *SystemC*-based simulator. Biological entities and interactions are translated into processes and signals. Processes are the central building blocks in a *SystemC* description. A *SystemC* description can be seen as a set of concurrent processes that communicate with each other using (clock or event)-dependent signals. As discussed in Chapter 4, *SyQUAL* maps each aspect of a SBML-based biological entity as follows:

- The entity behavior is modeled through a *FSM*. *FSM* is used to formally model the entity boolean representation in order to manage the entity state (e.g., active, inactive), state transitions and guard conditions, such as the boolean ones. The *FSM* implementation is accomplished through a *SystemC* process, which is sensible to any event coming as input signals.
- Each biological entity (e.g., gene, protein, etc.) is implemented through a *SystemC module*, in which both inputs (*P* and U_s) and outputs (D_s) are *SystemC ports*.

Any new event (e.g., signal value variation) which occurs to a specific entity input, leads to a new evaluation of its guard conditions, bringing to a potential updating of its state and output signals.

Lastly, *SyQUAL* core simulator relies on a discrete event-based framework developed in *SystemC*, providing efficient simulations of wide networks.

Reactions Boolean Network

In a boolean network, the node state value depends on a combination of its inputs, which in turn depends on their inputs combination, and so forth. A single change in a node state value implies a cascade evaluation of each directly reachable downstream node state. Biologically speaking, this behavior can be seen as a binding reaction process (creation of a molecular complex), in contrast with the catalysis one, in which the output nodes are not bound to the fate of their inputs.

Through the *asynchronous updating policy*, *SyQUAL* provides a way to simulate both interaction typologies (Figure 5.25), introducing the concept of *Reactions Boolean Network*, as extension of observable behaviors in a boolean network. In a *Reactions Boolean Network*, the node state value depends on both the combination of its inputs and their interaction types. To better explain such concepts, Figure 5.23 introduces a hypothetical Reactome catalysis reaction, while Figure 5.24 shows the corresponding *SyQUAL* representation. Reaction elements are defined as follows:

- *A* and *B* are molecular compounds.
- *C* is a molecular catalyzer.
- *F* is a molecular inhibitor.
- *D* and *E* molecular products.
- *R* catalysis reaction.

As shown in Figure 5.23, the state value of the reaction products *D* and *E* can assume value OFF (*inactive*) or ON (*active*), according to the following transfer function:

$$state(D) = state(E) = A \;\&\; B \;\&\; C \;\&\; !F \tag{5.6}$$

where

$$state(D) = \begin{cases} ON, & \text{if } A, B, C \text{ are ON and } F \text{ is OFF} \\ OFF, & \text{otherwise} \end{cases} \tag{5.7}$$

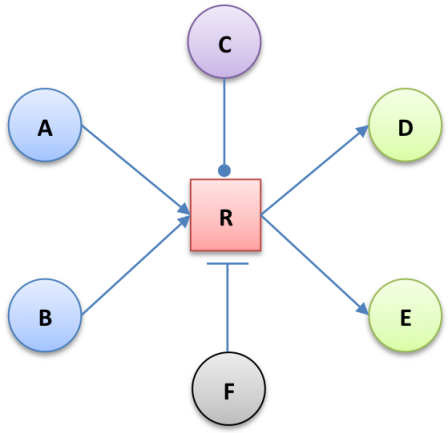


Fig. 5.23. An example of a hypothetical Reactome catalysis reaction.

In contrast, in Figure 5.24, the state value of the reaction products *D* and *E* can assume value ON (*active*), if and only if the transfer function result is equal to *ON*, otherwise no update occurs. Biologically speaking, this behavior can be assimilated to the lipid biosynthesis.

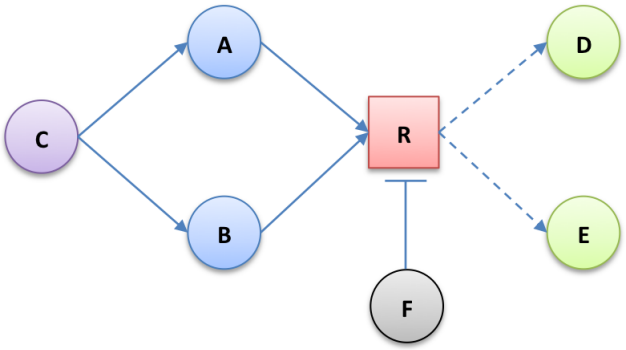


Fig. 5.24. The corresponding *SyQUAL* representation of the reaction shown in Figure 5.24.

The catalysis reaction shown in Figure 5.23 represent the canonical Reactome representation for such reaction type.

In this reaction, the catalyzer *C*, through the molecular compounds *A* and *B*, accomplishes the catalysis process, in order to create products *D* and *E*. *SyQUAL* redefines the Reactome catalysis topology to model this kind of reaction in a reasonable logic representation, as shown in Figure 5.24.

SyQUAL provides a way to model and simulate more complex behaviors in contrast with classic boolean networks, in which all reactions are modeled as binding processes, by limiting the number of observable behaviors.

SyQUAL models catalysis and binding reactions as shown in Figure 5.25, highlighting how these reactions react under activation and deactivation stimuli. Below, some useful notation:

- Stimulus (thunder).
- Biological entity (circle).
- Reaction process (square).
- Interaction (arrow).

Each biological entity can be activated (displayed as filled node) or deactivated (displayed as opaque entity). Under activation stimuli, each reaction type reacts producing its products, as explained below:

- The *binding process* produces a single molecular complex *C*. This complex is the result of the combination of *A* and *B*. The fate of this complex *C* is tied to its own constituents. The molecular complex represents the reaction process itself.
- The *catalysis process* produces the elements *G* and *H*. These ones are the result of the transformation of *E* and *F*. Despite the binding process, *G* and *H* are no more dependent on the reaction process *R*.

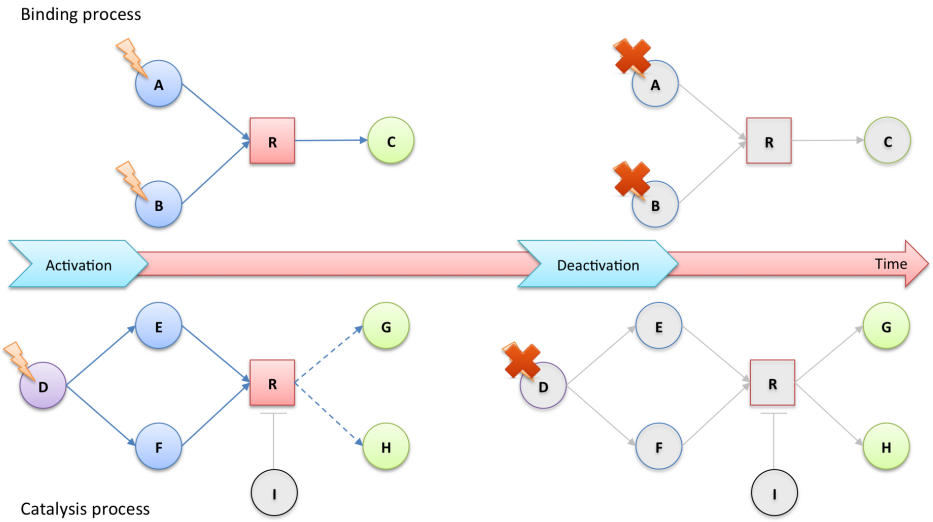


Fig. 5.25. The *SyQUAL* representation of *binding* and *catalysis* reaction behavior under activation and deactivation.

The main difference is observable under deactivation stimuli, in which each reaction type reacts as follows:

- In the *binding process*, if one or more biological entities of the complex turn to be deactivated (e.g., degraded), as direct consequence the complex C is no longer active.
- In the *catalysis process*, if one or more biological entities of the reaction turn to be deactivated, no reaction product is affected by any deactivation.

As a consequence, it allows observing more complex behaviors.

6

Case Studies

This chapter shows how *BIODEA* and *SyQUAL* have been applied to formalize and verify biological hypotheses on given biological systems.

Sections show experimental results obtained by modeling and simulation biological systems according to different cellular functionalities, such as signaling and gene regulation. Each biological case study is correlated by an (i) introduction to the biological phenomenon, (ii) biological hypotheses to be investigated, (iii) experimental conditions, (iv) experimental results, and (v) conclusions.

Section 6.1 describes the *Signaling Network Controlling LFA-1 beta2 integrin activation mediating Leukocyte recruitment from the blood into the tissues*, focused on uncover those dynamics behind the integrin periodic oscillation.

Section 6.2 describes the *Colitis-associated Colon Cancer (CAC) Network*. It uncovers those dynamics that stand behind the inflammation-associated tumorigenesis.

6.1 The Signaling Network Controlling the Leukocyte Recruitment from the Blood into the Tissues

Introduction

Signaling network controlling LFA-1 beta2 integrin activation mediates leukocyte recruitment from blood into tissues. The mechanism of leukocyte recruitment is a fundamental homeostatic process of the immune response. It is modeled as a concurrent ensemble of cellular events consisting of a stereotyped sequence of leukocyte behaviors on the vascular endothelium and including tethering, rolling, integrin activation, arrest and diapedesis. In this context, a critical event is integrin activation since it mediates cell arrest underflow and diapedesis (Ley et al. [76]).

In order to investigate dynamics underlying the leukocyte recruitment, Figure 6.1 reports a complex signal transduction network involved in controlling integrin activation (Constantin et al. [22]) developed at *Laboratory of Cell Trafficking and Signal Transduction*, University of Verona (*http://dp.univr.it/~laudanna/LCTST/*). Mainly generated by chemotactic factors, it involves at least 67 different intracellular molecules, in which (i) JAK protein tyrosine kinases, (ii) RHO and RAP small GTPases, (iii) lipid kinases and (iv) a number of cytoskeletal proteins cover an important role (Montresor et al. [85]).

The concerted action of these signaling proteins generates a concurrent modular mechanism of regulation of integrin activation characterized both by topological and dynamic properties. It includes the generation of specific emergent properties, such oscillators and hysteresis. Although a qualitative characterization of such a complex mechanisms is, at least partially, available.

Due to the lacking of a complete quantitative description, both the dynamical modeling of the leukocyte recruitment process and the identification of emergent properties involved in the leukocyte regulation is still limited.

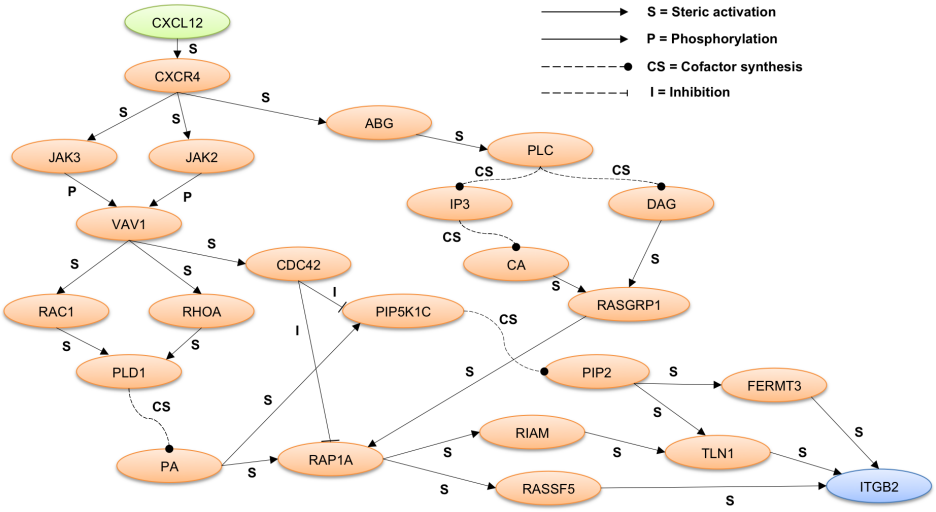


Fig. 6.1. The leukocyte signaling network. The chemokine *CXCL12* (green node) acts as triggering signal for the integrin (*ITBG2*) cascade activation, where orange nodes take part of the integrin activation. The integrin *ITBG2* (cyan node) represents the system response to the chemokine signal, resulting in the integrin activation (adhesion).

Figure 6.1 shows the leukocyte recruitment signal transduction network. It is characterized by different interaction types grouped as follows:

- *Steric activation* (*S*), *inhibition* (*I*), and *phosphorylation* (*P*) are treated as simple activation/deactivation signals (molecular complex creation). After their activation, biological entities are sensible about the fate of their activators.
- *Cofactor synthesis* (*CS*) is treated as synthesis signal. After their activation, biological entities are no longer sensible about the fate of their activators.

These interactions are biologically interpreted as follows:

- *Steric activation*. In this interaction, molecules bind each other in order to create a molecular complex. This binding generates a spatial arrangement of atoms in involved molecules.
- *Inhibition*. In this interaction, a molecule *A* binds a molecule *B*, resulting in a decrease of *B* activity.
- *Phosphorylation*. In this interaction, a protein kinase adds a “phosphate group” to an amino acid residue (proteins building blocks) of another protein, resulting in a strong energization of the phosphorylated protein. A protein kinase is an enzyme that catalyzes the transfer of a phosphate group from the *Adenosine triphosphate* (*ATP*) to a specified molecule.
- *Cofactor synthesis*. In this interaction, a chemical synthesis generates a complex organic molecules from simple components. Cofactors are non-protein chemical compounds or metallic ions. They are involved in protein biological activities.

Behaviors of Interest

The analysis of the leukocyte recruitment signal transduction network aimed at investigating the *on-off* dynamics of integrin triggering, representing the *ITGB2* oscillation. Oscillations are the result of activation states alternated to inactivation ones, and vice versa. Among these oscillations, the analysis mainly involved the investigation into those ones which are characterized by a periodic trend, since the integrin oscillation phenomenon shows a regular behavior.

Integrins are proteins characterized by mechanical and biochemical functions. In mechanical context, they physically attach the cell cytoskeleton (a complex network of interlinking filaments and tubules enclosed among the cell nucleus and the plasma membrane) to extracellular matrix (the cell physical scaffolding). In biochemical context, integrins act as sensor to identify if adhesion occurs.

Figure 6.2 shows how integrins are involved in leukocyte recruitment from blood vessels into inflamed tissues.

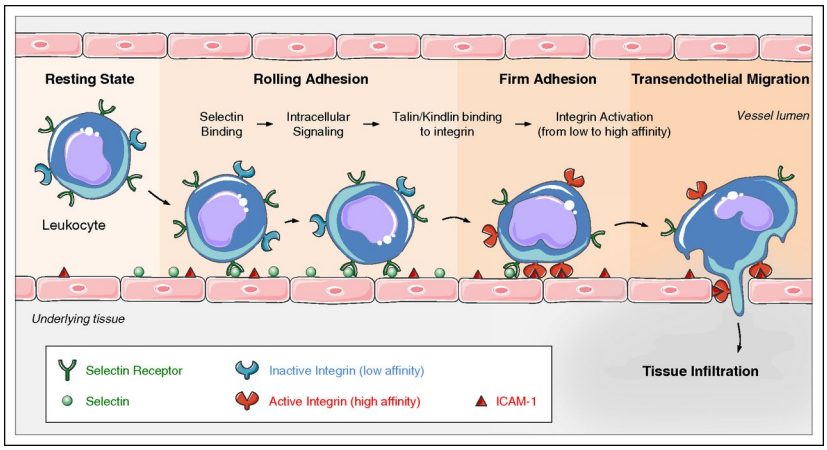


Fig. 6.2. Integrin role in leukocyte recruitment from blood vessels into inflamed tissues. Figure shows how integrin is involved in the leukocyte adhesion. The *"Selectin"* works as signal used to advise the leukocyte about an inflamed area. Activated integrins work as anchor letting the leukocyte to proceed toward to the inflamed area (source: *http://www.bloodjournal.org/content/128/4/479?sso-checked=true*).

In order to identify those configurations (a set of input values used for each simulation) that lead to periodic oscillations, analysis relied on *ABV*, as described in Section 4.2. Figure 6.3 shows two types of assertions, *P1* and *P2*, used to differentiate oscillations. Each assertion is designed to identify periodic oscillations with specific *positive* and *negative* edges. The positive edge represents the time frame in which the integrin is active, corresponding to the leukocyte adhesion. Differently, the negative edge represents the time frame in which the integrin results inactive.

Assertion *P1* considers those configurations that lead to periodic oscillations, in which positive and negative edges are comparable up to an additive/subtractive tolerance δ.

Assertion *P2* results less constrained, since it considers those configurations in which only positive edges must be comparable up to an additive/subtractive tolerance δ.

All identified useful configurations have been used as input data for the robustness/sensitivity analysis phase (Section 4.3). These configurations have been simulated using a mutated model, in which mutated entities have been kept inactive. In this analysis, mutated entities have been con-

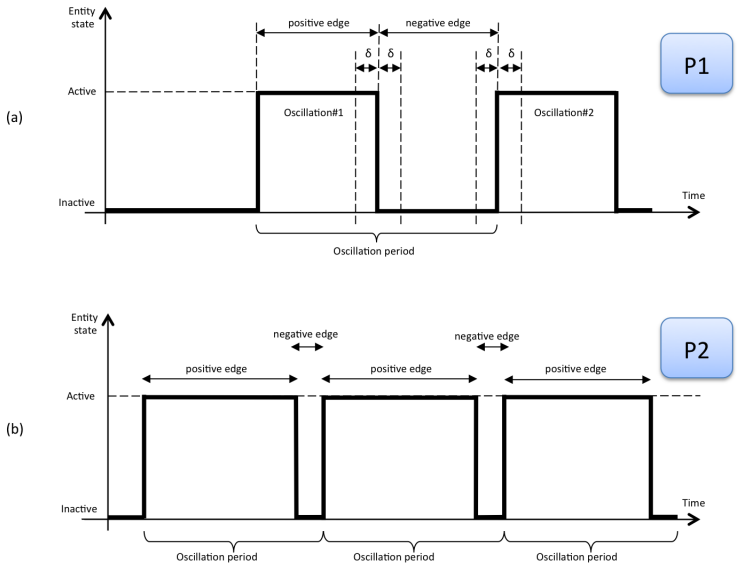


Fig. 6.3. Periodic oscillation identified according to assertion *P1* and *P2*.

sidered inactivated as result of loss of function (alteration due to disease) or drug action (through inhibition).

Experimental Conditions

As described in Section 4.1, the *delay time* represents the time spent by a biological entity to encounter its targets, while *lifetime* represents the maximum time from the activation instant in which the biological entity carries out its biological function.

The leukocyte recruitment signal transduction network has been simulated considering delay times in range *[2-8]* ms (with a step of *6* ms) and lifetimes equal to *50* ms, since these values represent a reasonable approximation of experimental data available in literature.

Through the automatic test pattern generator (Section 4.2), simulations have been performed exploring the whole delay time solution space. The amount of generated configurations (a unique configuration for each simulation) is equal to $\simeq 2.14 \cdot 10^9$, as result of *2* different delay times (*2* and *8* ms) and *31* biological interactions. Each simulation has been performed within a total simulated time of 200 ms, where the periodicity has been set with a tolerance of $\pm 10\%$. Such a period represents the average stopping time of a cell when it interacts with the blood vessel epithelium. Notably, although accurate experimental measurement of on-off dynamics of integrin triggering is, at the present, unavailable, the extremely rapid kinetics of leukocyte arrest under flow conditions, occurring in the experimentally-determined range of few milliseconds clearly suggest that it is reasonable to consider this rapid time-frame as a correct reference time to simulate on-off dynamics of integrin triggering. Furthermore, since directional leukocyte motility (chemotaixs) appears to maintain constant speed, at least in the context of a chemotactic gradient, it is reasonable to observe regular oscillatory dynamics of signaling mechanisms controlling integrin triggering.

Experimental Results

Models Comparison over Execution Time

As described in Section 4.5, the availability of different models provided a way to deal with large and complex networks of different levels of accuracy. In order to reduce the computational cost, models (*v1*, *v2*, . . ., *v9*) have been chosen according to required properties to be observed.

Model	Number of configurations	Approximate execution time	Properties
Synchronous (v1)	1	0.01 s	Attractors
Synchronous with classes (v2)	4,194,304	195 s	Attractors
Asynchronous (v5)	2.147.483.648	19 h	Attractors
Asynchronous with lifetime (v6)	2,147,483,648	80 h	Attractors Number/type of oscillations
Random (v7)	random(2,147,483,648)	25 h	Attractors Number/type of oscillations
Asynchronous with lifetime and cofactor synthesis (v8)	2,147,483,648	150 h	Attractors Number/type of oscillations
Asynchronous with lifetime, cofactor synthesis, molecular concentration (v9)	2,147,483,648	715 h	Attractors Number/type of oscillations

Table 6.1. Models comparison.

The Role of Lifetime and Cofactor Synthesis

In this first stage, experiments aimed at investigating whether the cofactor synthesis represented an essential condition to observe *ITGB2* periodic oscillations. Model *v6* and *v8* have been compared considering the assertion *P1*, classifying the resulting periodic oscillations according to a *period* threshold equal to the entities lifetime (50 *ms*).

- **Model v6**. Table 6.2 shows the number of configurations that led *ITGB2* to oscillate. Oscillations are grouped as periodic and aperiodic oscillations. Table 6.3 provides more details on observed periodic oscillations.

No. of configurations with oscillation of ITGB2	19,157,278
No. of periodic oscillation	19,157,278
No. of aperiodic oscillation	0

Table 6.2. Assertion *P1*. Number of periodic and aperiodic oscillations according to a tolerance $\delta = 10\%$.

No. of oscillations	No. of configurations	
	$Period = 50ms$	$Period > 50ms$
3	19,157,278	0

Table 6.3. Assertion *P1*. Periodic oscillations according to a tolerance $\delta = 10\%$.

- **Model v8**. Table 6.4 shows the number of configurations that led *ITGB2* to oscillate. Oscillations are grouped as periodic and aperiodic oscillations. Table 6.5 provides more details on observed periodic oscillations.

No. of configurations with oscillation of ITGB2	1,996,974,016
No. of periodic oscillation	2,008,188
No. of aperiodic oscillation	1,994,965,828

Table 6.4. Assertion *P1*. Number of periodic and aperiodic oscillations according to a tolerance $\delta = 10\%$.

No. of oscillations	No. of configurations	
	$Period = 50ms$	$Period > 50ms$
2	0	2,008,188

Table 6.5. Assertion *P1*. Periodic oscillations according to a tolerance $\delta = 10\%$.

Experiments showed a significant divergence. Considering Model *v6*, all configurations associated with the *ITGB2* oscillation are characterized by a periodic trend, even without considering any tolerance δ. Remaining configurations (majority) did not lead to any *ITGB2* activation. In contrast, Model *v8* showed a different behavior. The configurations majority led to aperiodic oscillations, with a small part associated to periodic ones. By comparing these results, it is clear that Model *v6* is not sufficient to correct reproduce the biological phenomenon, since it cannot discern any aperiodic configuration. In such a way, Model *v8* allowed to better represent the leukocyte recruitment phenomenon in a more realistic way.

Model v8: Robustness/Sensitivity Analysis (Assertion P1)

As Model *v8* provided a better representation of the dynamics of the leukocyte recruitment network, robustness/sensitivity analysis mainly focused on such a model.

Table 6.6 (first row - *Golden model*[1]) reports the mutation analysis results obtained using the assertion *P1*.

The table reports the number of configurations, among all the generated ones, that led to aperiodic and periodic oscillations (column *Useful conf. (periodic)*), and the relative percentage.

[1] The golden model represents the network with no mutation injected.

Mutated biological entity	**Aperiodic config.**	**Useful config. (periodic)**	
		No. of config.	Percentage (%)
Golden model	-	2,008,188	100
CXCR4	0	0	0
JAK3	2,008,188	0	0
JAK2	2,008,188	0	0
ABG	0	2,008,188	100
VAV1	0	0	0
PLC	0	2,008,188	100
RAC1	860,652	1,147,536	57
RHOA	860,652	1,147,536	57
CDC42	2,008,188	0	0
IP3	0	2,008,188	100
DAG	17,640	1,990,548	99
PLD1	0	0	0
PIP5K1C	0	0	0
CA	0	2,008,188	100
RASGRP1	0	2,008,188	100
PA	0	0	0
RAP1A	0	2,008,188	100
PIP2	0	0	0
RIAM	0	2,008,188	100
RASSF5	0	2,008,188	100
TLN1	792,036	1,216,152	61
FERMT3	952,140	1,056,048	53

Table 6.6. Assertion *P1*. Mutation analysis experimental results according to a tolerance $\delta = 10\%$.

Model v8: Robustness/Sensitivity Analysis (Assertion P2)

As analyzed in "The Role of Lifetime and Cofactor Synthesis", Model *v8* has been simulated according to assertion *P1*. Starting from this analysis, Table 6.7 shows the number of configurations that led *ITGB2* to oscillate according to assertion *P2*. Oscillations are grouped as periodic and aperiodic oscillations.

No. of configurations with oscillation of ITGB2	1,996,974,016
No. of periodic oscillation	1,382,405,153
No. of aperiodic oscillation	614,568,863

Table 6.7. Assertion *P2*. Number of periodic and aperiodic oscillations according to a tolerance $\delta = 10\%$.

Table 6.8 provides more details on observed periodic oscillations.

No. of oscillations	No. of configurations	
	$Period = 50ms$	$Period > 50ms$
2	1,231,107,615	4,864,608
3	146,414,786	0
4	18,144	0

Table 6.8. Assertion *P2*. Periodic oscillations according to a tolerance $\delta = 10\%$.

Table 6.9 (first row - *Golden model*[2]) reports the mutation analysis results obtained using the assertion *P2*. The table shows the number of configurations, among all the generated ones, that led to periodic oscillations (column *Useful conf. (periodic)*), the number of oscillations, and the relative percentage.

Mutated biological entity	Aperiodic config.	Useful config. (periodic)	
		No. of config.	Percentage (%)
Golden model	-	1,382,405,153	100
CXCR4	0	0	0
JAK3	77,947,813	1,303,361,596	94
JAK2	77,947,813	1,303,361,596	94
ABG	0	1,264,395,681	91
VAV1	0	0	0
PLC	0	1,264,385,681	91
RAC1	22,201,983	1,360,203,170	98
RHOA	22,201,983	1,360,203,170	98
CDC42	4,628,087	1,377,777,066	99
IP3	162,947,837	1,200,594,788	87
DAG	27,692,815	1,352,551,442	98
PLD1	0	0	0
PIP5K1C	17,151,838	717,698,739	52
CA	162,947,837	1,200,594,788	87
RASGRP1	0	1,264,385,681	91
PA	0	0	0
RAP1A	0	1,264,385,681	91
PIP2	17,151,838	717,698,739	52
RIAM	27,347,365	1,335,484,156	97
RASSF5	0	1,306,696,545	95
TLN1	155,848,165	1,206,983,356	87
FERMT3	114,441,433	1,267,963,720	92

Table 6.9. Assertion *P2*. Mutation analysis experimental results according to a tolerance $\delta = 10\%$.

Results highlighted a significant difference between assertions *P1* and *P2*. In assertion *P1*, only $\approx 0.1\%$ of all configurations (Table 6.4) led to periodic oscillations, characterized by *2* peaks (Table 6.5). Mutation analysis (Table 6.6) underlined that the network is (i) extremely sensitive to the

[2] The golden model represents the network with no mutation injected.

path involving *PIP5K1C* (Figure 6.4) and (ii) robust to the path involving *RASGRP1* (Figure 6.5). Results also showed that *JAK3/JAK2* and *CDC42* (Figure 6.6) are essential to guarantee the oscillation periodicity.

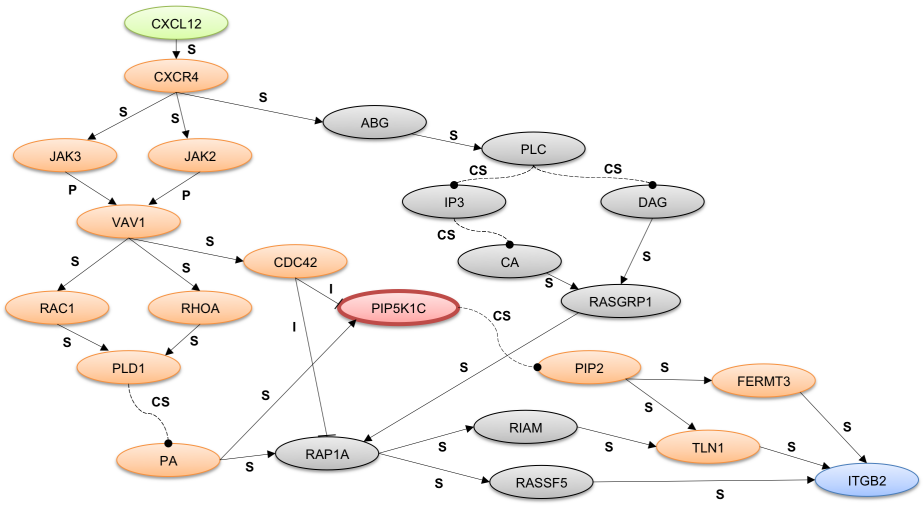


Fig. 6.4. Network shows an extremely sensitivity in the path involving *PIP5K1C*.

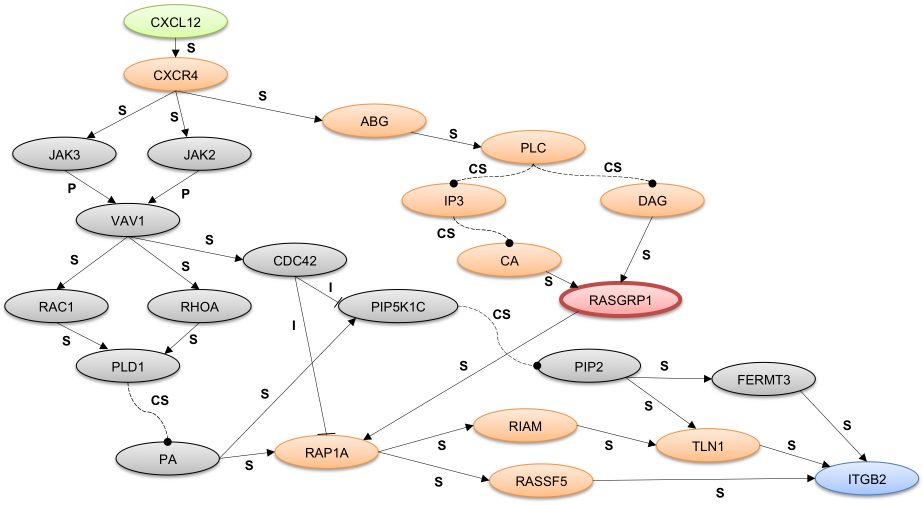


Fig. 6.5. Network shows a robustness in the path involving *RASGRP1*.

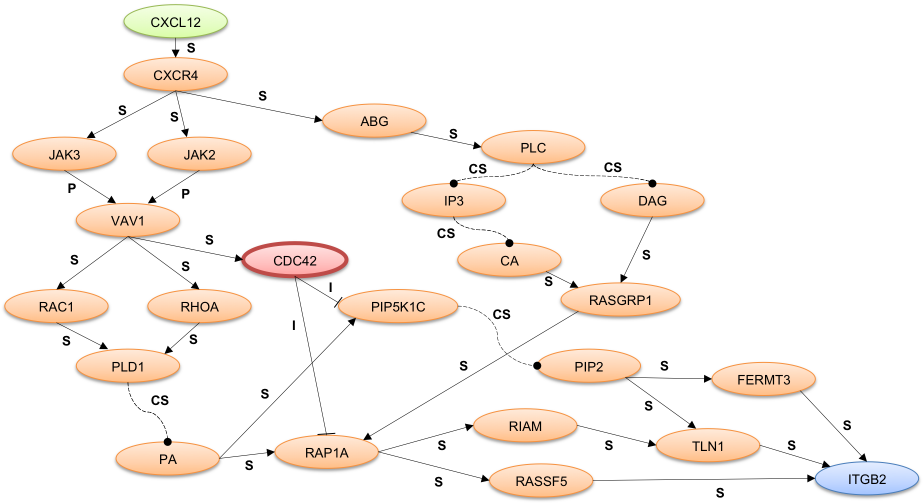


Fig. 6.6. *ITGB2* periodicity shows a strong dependence on *CDC42* activity.

In assertion *P2*, Table 6.7 showed a higher amount of configurations that led to periodic oscillations whether compared with Table 6.4. The amount of useful configurations rose to $\approx 42\%$, showing a number of oscillations between *2* and *4* peaks (Table 6.8). Mutation analysis (Table 6.8) showed a smaller impact if compared with assertion *P1* (Table 6.5). In this case, network showed a sensitivity to the path involving *RASGRP1*.

Finally, considering those configurations characterized by *2* oscillations (peaks) and a $period >$ 50 ms, assertion *P2* showed a higher amount of oscillations if compared with assertion *P1*. It is clear that oscillations identified with the assertion *P1* represent a subset of those identified through assertion *P2*.

Model v9: Robustness/Sensitivity Analysis and Molecular Concentration

As described in Section 4.5, Model *v9* provides a way to model several instances of each biological entity. This model considers molecular concentrations, providing an addition level of accuracy compared with previous models. Biological entities are represented through *packets* (instances). Each packet (a set of molecules) is associated with a specific granularity. For instance, if a biological entity is available with a molecular concentration equal to *100* molecules, it corresponds to:

- A single packet if the granularity *G* is equal to *100*.
- Five packets if the granularity *G* is equal to *20*.

Packets interact with an independent and concurrent way. Model *v9* results are not directly comparable with previous models, even considering its stochastic nature. Downstream targets are chosen using a weighted random distribution. Since the distribution depends on the number of inactive downstream molecules, same configurations can lead to different results.

Despite earlier models, Model *v9* used assertions *P1* and *P2* to verify the periodic oscillation of each packet which belongs to *ITGB2*. A configuration has been considered useful if at least *2* packets show a periodic oscillation.

Through the less restrictive assertion *P2*, the analysis showed a higher amount of configurations (compared with Model *v8*) that led to periodic oscillations (Table 6.10), according to a granularity $G = 100$. However, Table 6.11 showed that the major number of configurations is characterized by a periodic trend with a single oscillation (peak), even varying the tolerance δ. An acceptable explanation has been found in the fact that a single packet associated with each biological entity was not sufficient to obtain a realistic simulation.

No. of configurations with oscillation of ITGB2	995,157,334
No. of periodic oscillation	913,670,077
No. of aperiodic oscillation	81,487,257

Table 6.10. Assertion *P2*. Number of periodic and aperiodic oscillations according to a tolerance $\delta = 10\%$ and a granularity $G = 100$.

No. of oscillations	**No. of configurations**	
	$Period = 50ms$	$Period > 50ms$
1	46,422,844	769,731,464
2	144,168	97,370,486
3	1,115	0

Table 6.11. Assertion *P2*. Periodic oscillations according to a tolerance $\delta = 10\%$ and a granularity $G = 100$.

Nevertheless, mutation analysis (Table 6.12) confirmed the "sensibility" of the path involving *PIP5K1C*.

Mutated biological entity	Aperiodic config.	Useful config. (periodic)	
		No. of config.	Percentage (%)
Golden model	-	913,670,077	100
CXCR4	0	0	0
JAK3	12,846,246	298,207,210	33
JAK2	13,110,362	304,220,538	33
ABG	25,073,246	604,179,764	66
VAV1	0	0	0
PLC	17,090,388	406,236,965	44
RAC1	1,418,725	347,135,259	38
RHOA	1,418,221	347,137,534	38
CDC42	35,135,294	495,975,345	54
IP3	17,677,682	406,768,267	45
DAG	16,703,595	404,592,428	44
PLD1	0	0	0
PIP5K1C	0	0	0
CA	16,701,333	404,549,236	44
RASGRP1	16,704,734	404,589,369	44
PA	0	0	0
RAP1A	276,590,584	310,332,911	34
PIP2	0	0	0
RIAM	17,249,492	406,312,354	44
RASSF5	17,262,927	406,295,330	44
TLN1	17,256,921	406,309,417	44
FERMT3	14,835,463	405,267,383	44

Table 6.12. Assertion *P2*. Mutation analysis experimental results according to a tolerance $\delta = 10\%$.

Considering a lower granularity, with $G = 20$, the distribution of periodic oscillations over peaks showed a more interesting trend. By comparing distributions of assertions *P1* (Table 6.13) and *P2* (Table 6.14), this latter one showed a higher amount due to its less restrictive nature.

No. of packets oscillation periodically	No. of configurations
1	114,547,929
2	6,230,568
3	179,006
4	2,110
5	5
No. of useful configurations	6,411,689

Table 6.13. Assertion *P1*. Packets grouped per number of periodic oscillations according to a tolerance $\delta = 30\%$ and a granularity $G = 20$.

Even mutation analysis among assertions *P1* (Table 6.15) and P2 (Table 6.16) showed a higher discrepancy. It highlighted a strong divergence relative to the path involving *RASGRP1*.

No. of packets oscillation periodically	No. of configurations
1	308,833,171
2	44,528,655
3	3,489,149
4	84,958
5	272
No. of useful configurations	48,103,034

Table 6.14. Assertion *P2*. Packets grouped per number of periodic oscillations according to a tolerance $\delta = 30\%$ and a granularity $G = 20$.

Mutated biological entity	Aperiodic config.	Useful config. (periodic)	
		No. of config.	Percentage (%)
Golden model	-	6,411,689	100
CXCR4	0	0	0
JAK3	6,139,431	10,931	0.17
JAK2	6,160,843	11,397	0.18
ABG	6,222,250	117,081	1.83
VAV1	0	0	0
PLC	5,998,363	63,497	0.99
RAC1	6,258,202	13,518	0.21
RHOA	6,257,591	13,260	0.21
CDC42	6,343,332	44,702	0.70
IP3	6,335,018	28,866	0.45
DAG	6,333,241	31,122	0.49
PLD1	0	0	0
PIP5K1C	5,476,751	1	0
CA	6,314,705	37,502	0.58
RASGRP1	5,997,093	64,006	1.00
PA	0	0	0
RAP1A	6,277,650	78,231	1.22
PIP2	5,292,903	3	0
RIAM	6,331,749	33,417	0.52
RASSF5	6,335,909	27,607	0.43
TLN1	6,302,617	30,303	0.47
FERMT3	6,341,841	2,1851	0.34

Table 6.15. Assertion *P1*. Mutation analysis experimental results according to a tolerance $\delta = 30\%$.

Most relevant differences have been found considering the number of periodic oscillations related to those mutations (inhibition) involving *CDC42* and *RASGRP1* paths, which lead to *RAP1A* activation. Complete inhibition of such molecules caused a higher number of periodic oscillations especially if compared with other mutations, such as those involving the *PIP5K1C* path.

RAP1A and *ABG* represented the most *robust* biological entities. Even *ABG* and *PLC* are involved in the same path, their inhibition caused different results. *PLC* generated less useful configurations than *ABG*. This discrepancy was due to the possibility to address all available upstream

Mutated biological entity	Aperiodic config.	Useful config. (periodic)	
		No. of config.	Percentage (%)
Golden model	-	48,103,034	100
CXCR4	0	0	0
JAK3	45,772,250	350,076	0.73
JAK2	45,930,522	360,866	0.75
ABG	44,622,829	2,816,161	5.85
VAV1	0	0	0
PLC	43,896,648	1,405,068	2.92
RAC1	46,595,823	353,638	0.74
RHOA	46,594,446	354,192	0.74
CDC42	45,793,907	2,130,312	4.43
IP3	46,981,010	744,240	1.55
DAG	46,870,618	856,471	1.78
PLD1	0	0	0
PIP5K1C	41,013,525	89,317	0.19
CA	46,744,142	893,742	1.86
RASGRP1	43,882,176	1,405,567	2.92
PA	0	0	0
RAP1A	40,284,893	7,335,303	15.25
PIP2	39,659,171	31,674	0.07
RIAM	46,941,059	790,573	1.64
RASSF5	46,893,502	827,345	1.72
TLN1	46,708,041	792,511	1.65
FERMT3	47,071,579	649,062	1.35

Table 6.16. Assertion *P2*. Mutation analysis experimental results according to a tolerance $\delta = 30\%$.

packets (*CXCR4* molecules) on *JAK2/JAK3* when *ABG* is inhibited. Whenever *ABG* is active and *PLC* is inhibited, *ABG* packets had to wait their lifetime expiration, since *ABG* packets cannot change their target (*PLC* is not available).

Conclusions

According to the investigation of the periodic oscillation property, the mutation analysis underlined that the leukocyte signaling network is (i) extremely sensitive to the path involving *PIP5K1C* and (ii) robust in the path involving *RASGRP1*. Results also showed that biological entities *JAK2/JAK3* and *CDC42* are essential to guarantee the oscillation periodicity.

The analysis confirmed some biological insight. *PIP5K1C* has been found responsible for the final states associated to affinity. *RAP1A* has been confirmed to be involved in leukocyte adhesion process and not in affinity one.

CDC42, which acts as negative regulator in this network, has been computationally found involved in *ITGB2* periodic oscillation. However, as observed in literature, *CDC42* has been found involved in *neutrophil chemotaxis*, a biological process not directly associated with *ITGB2*. This fact represent an interesting behavior to be further experimentally investigated.

6.2 Colitis-associated Colon Cancer (CAC) Network

Introduction

Unlike previous section, which has mainly focused on investigating system dynamics from a reductionist perspective, the present section provides an analysis based on a more holistic approach, investigating connections between inflammation and tumorigenesis. Inflammation plays a relevant role in the development and evolution of several cancer types, including colon cancer. In colon cancer, the administration of anti-inflammatory drugs showed a decreasing in its expression, in those patients affected by familiar adenoma polyposis, acknowledging the connection between inflammation and cancer (Phillips et al. [93]). Due to inflammation, chronic inflamed tissues facilitate events such as genetic mutations and alterations in apoptosis mechanism leading to cell survival and proliferation. Therefore, *inflammatory bowel disease* (*IBD*) represents an important risk factor associated with the development of *colitis-associated colon cancer* (*CAC*). Mainly produced by epithelial and immune cells, cytokines revealed a role in the development of *CAC*.

In the context of *CAC*, studies showed that *tumor necrosis factor alpha* (*TNF-*α) (i) promotes tumor initialization and progression in experimental colitis (Popivanova et a. [94]), and (ii) stimulates the production of those molecules that mediate oxidative stress (Noguchi et al. [90], Szlosarek et al. [105]). It is clear that strategies in blocking the *TNF-*α action represent basis therapies for the *IBD*. In regard to transition from inflammatory microenvironment to malignant condition (Figure 6.7), the transcription factor *NF-*κ*B* plays an important role in regulating immunological and inflammatory responses (Ben-Neriah and Karin [6]).

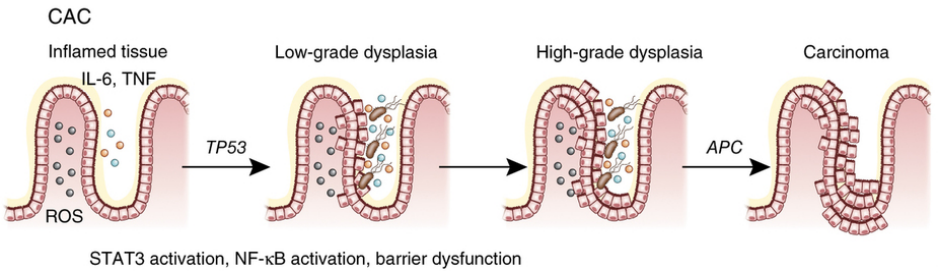


Fig. 6.7. Under *CAC*, intestinal tissue is characterized by a continue expression of inflammatory cytokines (*IL-6* and *TNF*), as well as early mutations of gene encoding *p53* (*TP53*) which favorite a continuous expression of *NF-*κ*B* and increase in inflammation. This deterioration leads to a barrier dysfunction (*dysplasia*, that is, presence of cells of an abnormal type) promoting bacteria infiltration, and an additional increasing in inflammatory response. An inflamed microenvironment is also characterized by the expression of *ROS* involved in DNA damaging. As result, mutation in *APC* gene encoding favorites the development of carcinoma (source: *http://www.nature.com/ni/journal/v17/n3/full/ni.3384.html*).

Interleukin 6 (*IL-6*), a *NF-*κ*B* induced cytokine, is another promoter of tumor growth (Grivennikov et al. [49]). Its action is regulated by *transforming growth factor beta* (*TGF-*β), acting as inhibitor for *IL-6*. *TGF-*β is an anti-tumor immune-response attenuator, promoting T cells (a subtype of white blood cells) activity. Members of the *interleukin 12* (*IL-12*) family, such as *interleukin 12*, *interleukin 23*, and *interleukin 27*, show implications in development of colitis (Duerr et al. [31]). Since all these cytokines participate in both inflammation and tumorigenesis, it is difficult to establish cytokines contribution to each step during the development of *CAC*. However, cytokines targeting could represent a promising therapy in *CAC*.

In order to investigate the role of inflammation in tumorigenesis, the analysis in this section has been based on the *CAC* network developed by Lu et al. [80]. This work describes a complex *CAC* network (Figure 6.8) analyzed both in-silico and in-vitro. It aimed at understanding dynamics behind inflammation-associated tumorigenesis and identifying potential novel therapies. Provided insight aimed at clarifying how molecular mechanisms lead to *CAC* using a refined boolean network model related to the growth and survival of preneoplastic epithelial cells. Such a network model can be decomposed into two main parts: the *intestinal epithelial cells* (*IEC*) part, which contains entities related to the intracellular signaling, and a second part associated to the immune micro-environment, which contains elements such as immune cells, cytokines and chemokines.

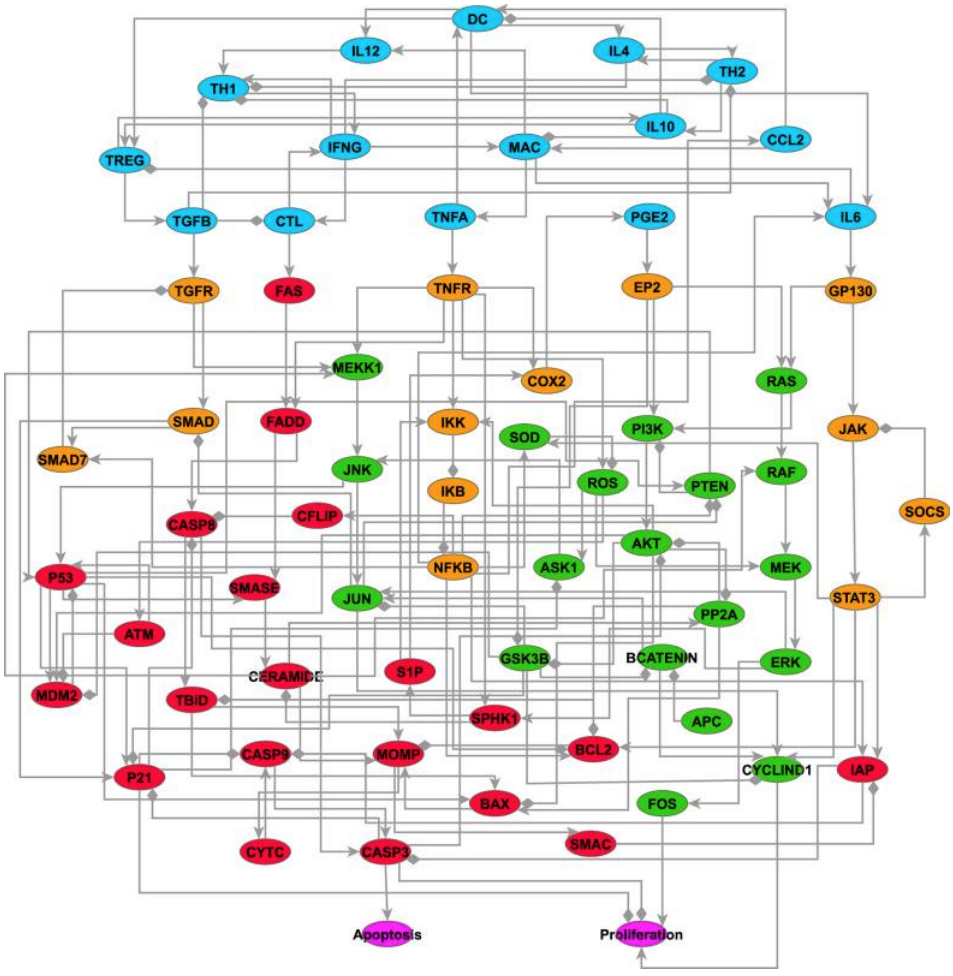


Fig. 6.8. The *CAC* Network. Each color represents a specific biological function. Cyan nodes belong to the extracellular immune microenvironment. Yellow nodes mainly participate in inflammatory signaling. Green nodes mainly mediate cell proliferation, while red nodes regulate cell survival. Finally, purple nodes represent the output effects of the network model (proliferation and apoptosis). Arrows represent activation (arrowhead) and inhibition (diamond).

Table 6.17 shows the full name and the gene symbol associated to each node in the *CAC* network.

Node name	Gene symbol	Full name
AKT	AKT1/AKT2	v-akt murine thymoma viral oncogene homolog 1 / v-akt murine thymoma viral oncogene homolog 2
APC	APC	adenomatous polyposis coli
ASK1	MAP3K5	apoptosis signal-regulating kinase 1 (also known as mitogen-activated protein kinase kinase kinase 5)
ATM	ATM	ataxia telangiectasia mutated
BAX	BAX	BCL2-associated X protein
BCATENIN	CTNNB1	beta-catenin
BCL2	BCL2	B-cell CLL/lymphoma 2
CASP3	CASP3	caspase 3, apoptosis-related cysteine peptidase
CASP8	CASP8	caspase 8, apoptosis-related cysteine peptidase
CASP9	CASP9	caspase 9, apoptosis-related cysteine peptidase
CCL2	CCL2	chemokine (C-C motif) ligand 2
CERAMIDE	N/A	ceramide
CFLIP	CFLAR	CASP8 and FADD-like apoptosis regulator
COX2	PTGS2	prostaglandin-endoperoxide synthase 2 (prostaglandin G/H synthase and cyclooxygenase)
CTL	N/A	CD8+ cytotoxtic lymphocyte
CYCLIND1	CCND1	cyclin D1
CYTC	CYCS	cytochrome c, somatic
DC	N/A	dendritic cells
EP2	PTGER2	prostaglandin E receptor 2 (subtype EP2), 53kDa
ERK	MAPK1/MAPK3	mitogen-activated protein kinase 1/ mitogen-activated protein kinase 3 (also known as ERK1 and ERK2)
FADD	FADD	Fas (TNFRSF6)-associated via death domain
FAS	FAS	Fas (TNF receptor superfamily, member 6)
FOS	FOS	FBJ murine osteosarcoma viral oncogene homolog
GP130	IL6ST	interleukin 6 signal transducer (gp130, oncostatin M receptor)
GSK3B	GSK3B	glycogen synthase kinase 3 beta
IAP	XIAP/BIRC3/BIRC2	X-linked inhibitor of apoptosis / baculoviral IAP repeat containing 3 / baculoviral IAP repeat containing 2
IFNG	IFNG	interferon, gamma
IKB	NFKBIA/NFKBIB	nuclear factor of kappa light polypeptide gene enhancer in B-cells inhibitor, alpha/beta
IKK	CHUK/IKBKB	conserved helixloophelix ubiquitous kinase / inhibitor of kappa light polypeptide gene enhancer in B-cells, kinase beta (also known as IKK-alpha and IKK-beta)
IL10	IL10	interleukin 10
IL12	IL12A	interleukin 12A
IL4	IL4	interleukin 4
IL6	IL6	interleukin 6 (interferon, beta 2)
JAK	JAK2	Janus kinase 2
JNK	MAPK8	mitogen-activated protein kinase 8
JUN	JUN	jun proto-oncogene
MAC	N/A	macrophages
MDM2	MDM2	Mdm2, p53 E3 ubiquitin protein ligase homolog (mouse)

continued . . .

...continued

Node name	Gene symbol	Full name
MEK	MAP2K1/MAP2K2	mitogen-activated protein kinase kinase 1 / mitogen-activated protein kinase kinase 2 (also known as MEK1 and MEK2)
MEKK1	MAP3K1	mitogen-activated protein kinase kinase kinase 1, E3 ubiquitin protein ligase
MOMP	N/A	mitochondrial outer membrane permiabilization
NFKB	NFKB1/NFKB2/RELA	nuclear factor of kappa light polypeptide gene enhancer in B-cells 1 (p105) / nuclear factor of kappa light polypeptide gene enhancer in B-cells 2 (p49/p100) / v-rel reticuloendotheliosis viral oncogene homolog A, nuclear factor of kappa light polypeptide gene enhancer in B-cells 3, p65 (avian)
P21	CDKN1A	cyclin-dependent kinase inhibitor 1A (p21, Cip1)
P53	TP53	tumor protein p53
PGE2	N/A	prostaglandin E2
PI3K	PIK3CA/PIK3CB	phosphoinositide-3-kinase, catalytic, alpha polypeptide / phosphoinositide-3-kinase, catalytic, beta polypeptide
PP2A	PPP2CA/PPP2CB	protein phosphatase 2, catalytic subunit, alpha isoform / protein phosphatase 2, catalytic subunit, beta isozyme
PTEN	PTEN	phosphatase and tensin homolog
RAF	RAF1	v-raf-1 murine leukemia viral oncogene homolog 1
RAS	KRAS/HRAS	v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog / v-Ha-ras Harvey rat sarcoma viral oncogene homolog
ROS	N/A	reactive oxygen species
S1P	N/A	Sphingosine-1-phosphate
SMAC	DIABLO	diablo, IAP-binding mitochondrial protein
SMAD	SMAD2/SMAD3/SMAD4	SMAD family member 2/3/4
SMAD7	SMAD7	SMAD family member 7
SMASE	SMPD1/SMPD2/SMPD4	sphingomyelin phosphodiesterase 1, acid lysosomal / sphingomyelin phosphodiesterase 2, neutral membrane (neutral sphingomyelinase) / sphingomyelin phosphodiesterase 4, neutral membrane (neutral sphingomyelinase-3)
SOCS	SOCS1	suppressor of cytokine signaling 1
SOD	SOD1/SOD2	superoxide dismutase 1, soluble / superoxide dismutase 2, mitochondrial (also known as Cu/ZN-SOD and Mn-SOD)
SPHK1	SPHK1	sphingosine kinase 1
STAT3	STAT3	signal transducer and activator of transcription 3 (acute-phase response factor)
TBID	BID	BH3 interacting domain death agonist
TGFB	TGFB1/TGFB2	transforming growth factor, beta 1 / transforming growth factor, beta 2
TGFR	TGFBR2	transforming growth factor, beta receptor II (70/80kDa)
TH1	N/A	type 1 T helper cells
TH2	N/A	type 2 T helper cells
TNFA	TNF	tumor necrosis factor
TNFR	TNFRSF1A	tumor necrosis factor receptor superfamily, member 1A
TREG	N/A	regulatory T cells

Table 6.17: *CAC* network nodes supplied with their gene symbol and full name.

Behaviors of Interest

Starting from Lu et al. [80], analysis involved the identification of which factors can influence *Proliferation* and *Apoptosis* activities. Cell proliferation represents the process which results in an increase in the number of cells, especially during tumorigenesis. Proliferation is defined by the balance among cells division and cells loss (through death or differentiation).

In contrast, apoptosis represents the process which directs the cell self-destruction. It plays a fundamental role in preventing uncontrolled cell growth and tumor development.

The investigation of *Proliferation* and *Apoptosis* activities involved four microenvironments, as shown in Table 6.18, that is, *non-inflammatory microenvironment* (1), *normal inflammation response* (2), *pro-tumor microenvironment* (3), *pro-tumor microenvironment and P53 inactivation* (4).

Each microenvironment has been analyzed according to synchronous and asynchronous updating policies. In general, in a model based on a synchronous updating policy, all entities change their state value simultaneously at each time point, making tractable the analysis of very large networks. However, the resulting model is characterized by a lower accuracy as in nature biological entities take different times to perform their activities. In contrast, models based on asynchronous updating policies provide a more accurate modeling of the natural phenomenon at the expense of a higher complexity.

It is clear that the analysis of *CAC* network involved the investigation of new potential pharmacological therapies, in order to identify which drugs could be used to reduce proliferation and/or increase the apoptosis activity of cells in a pro-tumor microenvironment.

Experimental Conditions

According to Lu et al. [80], analysis and experimental observations involved four main microenvironments (conditions). Each microenvironment (Table 6.18) is characterized by a set of input stimuli and a list of nodes knocked out, in particular:

Condition	Initially On	Initially Off	Fixed On	Fixed Off
Non-inflammatory microenvironment		Prolifetation, Apoptosis	APC	IL6, IL12, IL4, TH1, TH2, IL10, TREG, IFNG, MAC, CCL2, TGFB, CTL, TNFA, PGE2, DC
Normal inflammation response	DC	Prolifetation, Apoptosis	APC	IL6, IL12, IL4, TH1, TH2, IL10, TREG, IFNG, MAC, CCL2, TGFB, CTL, TNFA, PGE2
Pro-tumor microenvironment		Prolifetation, Apoptosis	DC	
Pro-tumor microenvironment and P53 inactivation		Prolifetation, Apoptosis	DC	P53

Table 6.18. List of main microenvironments.

- A set of activated entities (stimuli) at the beginning of the simulation (*Initially On*).
- A set of deactivated entities at the beginning of the simulation (*Initially Off*).
- A set of entities (stimuli) always kept active during the whole simulation (*Fixed On*).
- A set of entities always kept deactivated during the whole simulation (*Fixed Off*).

Non-inflammatory Microenvironment

A non-inflammatory microenvironment is characterized by the absence of inflammatory factors. In order to simulate a non-inflammatory microenvironment, the state of all immune microenvironment entities (cyan nodes) are fixed at OFF and the adenomatous polyposis coli (*APC*) is fixed at ON. *APC* represents the precancerous intestinal epithelial cells, and its continuous expression suppresses the β-catenin signaling (Morin et al. [86]).

Normal inflammation response

In a normal inflammatory response, dendritic cells (*DC*s) act as strong initializer and maintainer of the immunity response (Rescigno and Di Sabatino. [96]). To simulate a normal inflammatory response, the transitory activation of dendritic cells is performed through the initial activation of the *DC* node (*Init ON*). The remaining settings are the same seen for the non-inflammatory microenvironment.

Pro-tumor microenvironment

To simulate a pro-tumor inflammatory microenvironment, the strongest tumor initializer *DC* is fixed at ON. In an inflammatory microenvironment, it could be biologically explained as growth-controlling factor of the precancerous *IEC*s.

Experimental Results

Experiments involved the attractors analysis according to each microenvironment. In literature, discrete logic-based dynamical models are often analyzed in order to identify stable cycles of states. These stable cycles, called attractors, represent the states in which biological networks dwell most of the time. According to Kauffman [65] and Li et al. [77], attractors can be often linked to phenotypes. Phenotypes represent a set of observable characteristics (i.e., physiological properties and morphology) associated to an organism and its development stage.

Depending on the adopted updating policy, the attractors analysis can show different results. Usually, asynchronous updating policies show a larger set of states, resulting in more complex attractors. This complexity can be easily recognized by observing the number of states and their interconnections which compose an attractor. An attractor can be composed by a single state (*self loop*, also called *steady state*), or by a set of states (*simple loop* or *complex loop*). In a *simple loop* each state has a single successor state. In a *complex loop* each state can have two or more successor states.

Figure 6.9 shows the different attractors complexity achievable through synchronous and asynchronous updating policies.

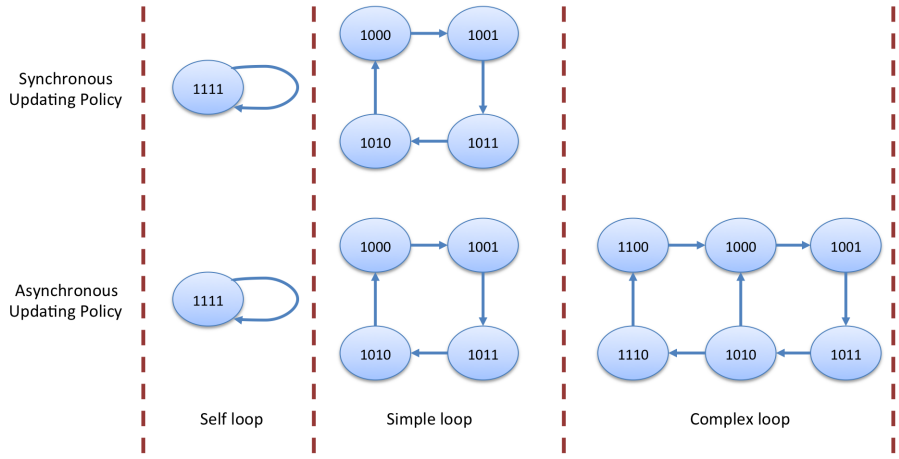


Fig. 6.9. Attractors complexity differentiation among synchronous and asynchronous updating policies.

As defined, attractors represent the system *response* under specific conditions. In this context, *SyQUAL* has been compared with the most representative tools for qualitative modeling and simulation of biological systems in literature, such as *BoolNet*, *BooleanNet*, and *GINsim*. These tools share common features with *SyQUAL*. They support SBML *qual* as input model and both synchronous and asynchronous updating policies. *BoolNet*, *BooleanNet*, and *GINsim* rely on the *GA* and *ROA* updating policies which do not require any prior information (Section 2.4.2 - *Updating Policies*). *The Cell Collective*, *ChemChains*, and *CellNetAnalyzer* (CNA) have been excluded for the following reasons:

- *The Cell Collective* does not support any SBML description as input format and it does not provide any asynchronous updating policy. Moreover, some issues have been identified during simulations, leading to a wrong modeling of the *CAC* network.
- *CellNetAnalyzer* requires specific input formats, based on complex internal rules. They turned up to be hard to prepare, especially for large networks.

Each tool feature in summarized in Table 6.19.

	Boolean Model		Analysis			Knock out	
	Boolean Network	Reaction Boolean Network	Attractor Identification	Robustness Analysis	Model Checking	Specie-based	Drug-based
BoolNet	•		•	M		•	
GINsim	•		•	M	•	•	
BooleanNet	•		•	M		•	
SyQUAL	•	•	•	A		•	•

Table 6.19. Table shows a comparison in term of features and analysis between SyQUAL and the tools previously listed. The marker • represents the support for a specific feature/analysis. Robustness analysis can be performed manually (*M*) or automatically (*A*).

Synchronous Updating Policy

Attractors analysis has been performed according to each microenvironment, that is, *non-inflammatory microenvironment* (1), *normal inflammation response* (2), *pro-tumor microenvironment* (3), *pro-tumor microenvironment and P53 inactivation* (4).

Table 6.20 shows results obtained through the synchronous updating policy in terms of simulation time, the corresponding standard deviation (for a batch of one hundred executions), and the number of identified attractors, according to each microenvironment and tool.

Tool	**Microenv.**	**Timing in ms**		**No. of attractors**
		Exec. time	*Standard Dev.*	
BoolNet	1	1.16	0.51	**1**
	2	1.02	0.47	**1**
	3	1.10	0.56	**6**
	4	1.22	0.66	**6**
GINsim	1	3,000.00	1,000.00	**1**
	2	3,000.00	1,000.00	1
	3	Out of memory		
	4	Out of memory		
BooleanNet	1	11.59	1.64	**1**
	2	11.58	1.64	**1**
	3	15.07	0.23	**6**
	4	16.340	4.79	**6**
SyQUAL	1	11.46	1.69	**1**
	2	10.97	1.56	**1**
	3	13.03	2.04	**6**
	4	12.21	1.87	**6**

Table 6.20. A quantitative overview of attractors analysis according to the synchronous updating policy. Table reports the execution time and number of identified attractors in agreement with each microenvironment and tool.

Table 6.20 underlined that all tools (except for *GINsim* which does not support the network complexity) led to the identification of the same set of attractors for each microenvironment. This was expected as all tools are based on the same strategy (synchronous updating policy). It must be clarified that the synchronous analysis has been used as fundamental *testbench* to evaluate the quality of *SyQUAL* results, since all compared tools used the same updating method.

Results showed that *SyQUAL* reached the same accuracy of *BoolNet* (a reference tool for qualitative attractor investigation), identifying the same set of attractors (Table 6.21), confirming the validity of modeling and simulation of Biological Systems through *EDA* technologies.

Identified attractors (Table 6.20) are listed in Table 6.21.

Each microenvironment showed a different response and attractors complexity, due to the peculiar experimental conditions.

Microenvironment	Attactors
Non-inflammatory microenvironment	1. IKB, GSK3B, APC
Normal inflammation response	1. IKB, GSK3B, APC
Pro-tumor microenvironment	1. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 2. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 3. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 4. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 5. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 6. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS
Pro-tumor microenvironment and P53 inactivation	1. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 2. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 3. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 4. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 5. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 6. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS

Table 6.21. A qualitative overview of attractors analysis according to the synchronous updating policy. Table reports the list of identified attractors for each microenvironment.

Next paragraphs report the quantitative analysis of the activity levels of a set of biological entities of interest, according to each microenvironment.

Non-Inflammatory Microenvironment. Attractor (Table 6.21) showed a suppression of *β-catenin* (*BCATENIN*), as result of the constant expression of *APC*. Transcription factors such as *STAT3* (*STAT3*) and *NF-κB* (*NFKB*) are considered, with *β-catenin* (*BCATENIN*), distinctive characteristics of *CAC* (Morin et al. [86], Kojima et al. [68], Atreya and Neurath [3], Yu, Pardoll and Jove [113]). It is clear that without any inflammatory signal, the *Proliferation* activity level (Table 6.22) suggests that *IEC*s tend to stay in a resting state (no proliferation).

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
1	0.02	0.02	0	0	0	0	0	0	0	0	0	**0**	0

Table 6.22. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). *Proliferation* showed no activity, suggesting a resting state for *IEC*s.

Normal Inflammation Response. Despite the non-inflammatory microenvironment, the normal inflammation response is characterized by an initial activation of dendritic cells (*DC*). Even considering an initial presence of dendritic cells, attractor showed the same behavior observed under the non-inflammatory microenvironment, suggesting a resting state for *IEC*s. Due to the initial activation of *DC*, Table 6.23 showed a feeble expression of *DC*.

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
1	0.02	0.02	0	0	0	0	0	0	0	0	0.01	**0**	0

Table 6.23. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). Even with an initial activation of dendritic cells (*DC*), *Proliferation* continued to show no particular activity.

As known in literature, the inflammatory microenvironment is characterized by the expression of different types of immune cells. Investigating how different immune microenvironments can influence *IEC*s represents an interesting analysis. As a result of this analysis, Table 6.24 shows how a mixing of expressed immune cells affected *IEC*s and the activity levels of *Proliferation* and *Apoptosis*.

Stimuli	TH2	CTL	DC	MAC	TH1	TREG	APC	BCATENIN	NFKB	STAT3	TNFA	TGFB	IL6	IL4	IL12	IL10	Proliferation	Apoptosis	
	•						1	0.04	0.04	0.04	0	0.9	0.04	0.98	0	0.98	0	**0.49**	**1**
		•					1	0.79	0.82	0.47	0.92	0.02	0.95	0.89	0.92	0.02	**0.32**	0.22	**2**
			•				1	0.87	0.85	0.48	0.01	0.02	0.98	0.98	0.98	0.94	**0.8**	0	**3**
				•			1	0.86	0.88	0.48	0.98	0	0.98	0.94	0.98	0	**0.36**	0.02	**4**
					•		1	0.83	0.85	0.47	0.92	0.02	0.95	0.89	0.92	0.02	**0.38**	0.02	**5**
						•	1	0.05	0.05	0.05	0	0.98	0.05	0	0	0.98	0	**0.48**	**6**
		•	•				1	0.82	0.84	0.48	0.92	0.02	0.98	0.98	0.98	0.02	**0.37**	0.14	**7**
		•		•			1	0.85	0.88	0.48	0.98	0	0.98	0.94	0.98	0	**0.36**	0.04	**8**
	•	•					1	0.02	0.02	0.02	0	0.92	0.02	0.98	0	0.98	0	**0.87**	**9**
			•		•		1	0.87	0.86	0.48	0.92	0.02	0.98	0.98	0.98	0.02	**0.42**	0.02	**10**
	•		•				1	0.87	0.85	0.48	0	0.02	0.98	0.98	0.98	0.98	**0.82**	0	**11**
			•			•	1	0.86	0.84	0.48	0	0.98	0.98	0.98	0.98	0.98	**0.72**	0	**12**
	•				•		1	0.04	0.04	0.04	0	0.9	0.04	0.98	0	0.98	0	**0.64**	**13**
	•	•	•				1	0.34	0.33	0.48	0	0.02	0.98	0.98	0.98	0.98	0.21	**0.53**	**14**
		•	•			•	1	0.22	0.21	0.48	0	0.98	0.98	0.98	0.98	0.98	0.01	**0.49**	**15**
	•		•		•		1	0.86	0.84	0.48	0	0.02	0.98	0.98	0.98	0.98	**0.79**	0.01	**16**
	•					•	1	0.02	0.02	0.02	0	0.98	0.02	0	0	0.98	0	**0.87**	**17**

Table 6.24. Effects of different immune cells expression on *IEC*s. Table reports activity levels of cytokines, as well as the most relevant activity levels of *Apoptosis* and *Proliferation*. The *bullet notation* (•) represents an expressed immune cell.

By analyzing the influence of mixed expressed immune cells on *IEC*s, results (Table 6.24) confirmed experimental observations.

The continuous expression of *DC*, which can be biologically interpreted as a constant activation of dendritic cells, contributed to produce one of the most pro-proliferation microenvironments (rows *3* and *11*), even combined with the activation of *TH2*. Experimental observation showed that transient activation of dendritic cells "triggered" controlled inflammatory reactions (Fiocchi [36]). A constant activation of dendritic cells led to chronic inflammation in *IBD* (Hart et al. [52]), increasing the growth and survival of *IEC*s (MacDonald [81]).

The constant expression of *MAC* or *TH1* led *Proliferation* to rise its activity at a disadvantage of *Apoptosis* (rows *4* and *5*).

The stable expression of *TH2* or *TREG* produced an important increment of the *Apoptosis* activity (rows *1* and *6*). Considering the *TREG* expression, *IL10* and *TGFB* showed an increasing activity, in contrast with *IL6* and *TNFA*. Experimental observations proved that the T Cell regulatory activity reduces the tumor development through (i) the production of immune suppressive cytokines, such as *TGF-*β and *IL-10*, and (ii) the reduction of pro-inflammatory cytokines, such as

IL-6 and *TNF-*α (Erdman et al. [32]). The combined expression of *TREG-CTL* or *TH2-CTL* led to the most anti-tumorigenic response (row *9* and *17*). Clinical evidences showed that *CTL* plays a role in intestinal inflammation and promotion of tumor growth (Waldner and Neurath [110]), in contrast with previous studies in which *CTL* is usually associated with an encouraging prognosis in sporadic colon cancer (Naito et al. [88]). These combinations indicated a restoring of the cytotoxic activity of *CTL*, leading to an increasing of the immune surveillance activity.

Pro-tumor Microenvironment. Characterized by a constant expression of dendritic cells (*DC*s), attractors showed a *Proliferation* predominance and a complete *Apoptosis* suppression (Table 6.25).

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
0	0.99	0.85	0.48	0.01	0.94	0	0.98	0.98	0.98	0.94	1	**0.8**	0

Table 6.25. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). With a constant activation of dendritic cells (*DC*), *Proliferation* showed a high activity.

In a pro-tumor microenvironment, the activity of transcription factors *STAT3* and *NFKB* was somewhat higher than in non-inflammatory microenvironment and normal inflammatory response. Experimental observations showed that *STAT3* and *NFKB* are high expressed under inflammatory stimuli (Greten et al. [48], Grivennikov et al. [49]).

Pro-tumor Microenvironment and P53 Inactivation. As described in literature, *P53*, known as *TP53*, acts as tumor suppressor, playing a fundamental role in the *Apopotosis* process (Zilfou et al. [114]).

As result of *P53* deactivation, attractors highlighted once more a *Proliferation* predominance and a complete *Apoptosis* suppression (Table 6.26). However, *Proliferation* slightly rose its activity if compared with the simple pro-tumor microenvironment.

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
0	0.99	0.85	0.48	0.01	0.94	0	0.98	0.98	0.98	0.94	1	**0.84**	0

Table 6.26. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). With a constant activation of dendritic cells (*DC*) and the deactivation of *P53*, *Proliferation* showed a higher activity compared to Table 6.25.

Influencing Proliferation and Apoptosis using Drugs in a Pro-tumor Microenvironment. As observed in pro-tumor microenvironment, the continue expression of dendritic cells (*DC*s) produced a predominance of *Proliferation*, with a complete deactivation of the *Apoptosis* activity (Table 6.25).

In general, drugs play an important role to restore a normal condition. Consequently, this paragraph investigated how drugs could be used to restore a normal microenvironment or at least to reduce the *Proliferation* activity.

Table 6.27 reported how the *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies.

	APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
1	0	0.99	0.85	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0	BCL2
2	0	0.99	0.85	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0	COX2
3	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0	COX2, IKK
4	0	0.99	0.85	0.48	0	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.82	0	COX2, TNFA
5	0	0.99	0.5	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.32**	ERK
6	0	0.99	0.85	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	0	FOS
7	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.81	0	GSK3B
8	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0	IKK
9	0	0.99	0.84	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0.01	JAK
10	0	0.99	0.5	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.32**	MEK
11	0	0.99	0.83	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	0.01	RAF
12	0	0.99	0.85	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	0	SMASE
13	0	0.99	0.85	0.48	-	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.82	0	TNFA
14	0	0.99	0.86	0.48	0.01	0.96	0	-	0.98	0.98	0.98	0.96	1	0.82	0	TGFB
15	0	0.99	0.85	0.48	0	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.82	0	CCL2

Table 6.27. Alteration of entities activity levels through *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Results in Table 6.27 showed a restoring of the *Apoptosis* activity through the inhibition of *ERK* (row *5*) or *MEK* (row *10*). The *ERK* and *MEK* inhibition has been accomplished using the following drugs:

- *ERK* inhibitors:
  - **Purvalanol** experimental drug [*https://www.drugbank.ca/drugs/DB02733*].
- *MEK* inhibitors:
  - **Cobimetinib**, approved drug [*https://www.drugbank.ca/drugs/DB05239*].
    Cobimetinib is a potent and highly selective small molecule inhibitor of mitogen-activated protein kinase kinase 1 (MAP2K1 or MEK1), and central components of the RAS/RAF/MEK/ERK signal transduction pathway. It is used in combination with vemurafenib for the treatment of patients with unresectable or metastatic BRAF V600 mutation-positive melanoma.
  - **Bosutinib**, approved drug [*https://www.drugbank.ca/drugs/DB06616*].
    Bosutinib is a Bcr-Abl kinase inhibitor for the treatment of Philadelphia chromosome-positive (Ph+) chronic myelogenous leukemia (CML).
  - **Trametinib**, approved drug [*https://www.drugbank.ca/drugs/DB08911*].
    Trametinib dimethyl sulfoxide is a kinase inhibitor.

Influencing Proliferation and Apoptosis using a Mix of Drugs in a Pro-tumor Microenvironment. Despite the previous analysis, this paragraph explored the effects of combining drugs in order to restore a normal microenvironment or at least to reduce the *Proliferation* activity.

Table 6.28 showed how the *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies. Table reported only those results with an additive influence (compared with the *Systematic Knock Out*) of *Apoptosis* activity.

	APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
1	0	0.99	0.47	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.38**	ERK, JAK
2	0	0.99	0.47	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.38**	JAK, MEK
3	0	0.99	0.49	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.01	**0.35**	JAK, RAF
4	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.34**	IKK, ERK
5	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	**0.34**	IKK, MEK

Table 6.28. Alteration of entities activity levels through *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Results in Table 6.28 showed a restoring and increasing of the *Apoptosis* activity compared with Table 6.27 (row *5* and *10*). This has been accomplished through the combined inhibition of *ERK-JAK*, *ERK-IKK*, *MEK-JAK*, *MEK-IKK*, or *RAF-JAK*.

Inhibitors listed below are reported only for *JAK*, *IKK*, and *RAF*, since *MEK* and *ERK* inhibitors have been already reported in the previous paragraph (*Influencing Proliferation and Apoptosis using Drugs in a Pro-tumor Microenvironment*).

- *JAK* inhibitors:
  - **Ruxolitinib** approved drug [*https://www.drugbank.ca/drugs/DB08877*].
    Ruxolitinib is a janus-associated kinase inhibitor indicated to treat bone marrow cancer, specifically intermediate or high-risk myelofibrosis.
- *IKK* inhibitors:
  - **Acetylcysteine** approved drug [*https://www.drugbank.ca/drugs/DB06151*].
    Acetylcysteine is commonly used in individuals with renal impairment and in treating mild to moderate traumatic brain injury including ischemic brain injury.
- *RAF* inhibitors:
  - **Sorafenib**, approved and investigational drug [*https://www.drugbank.ca/drugs/DB00398*].
    Sorafenib (rINN) is a drug approved for the treatment of advanced renal cell carcinoma (primary kidney cancer). Sorafenib is a small molecular inhibitor of Raf kinase, PDGF (platelet-derived growth factor), VEGF receptor 2 & 3 kinases and c Kit the receptor for Stem cell factor.
  - **Regorafenib**, approved drug [*https://www.drugbank.ca/drugs/DB08896*].
    Regorafenib is an inhibitor of multiple kinases. It is used for the treatment of metastatic colorectal cancer and advanced gastrointestinal stromal tumors.
  - **Dabrafenib**, approved drug [*https://www.drugbank.ca/drugs/DB08912*].
    Dabrafenib mesylate is a reversible ATP-competitive kinase inhibitor and targets the MAPK pathway.

Influencing Proliferation and Apoptosis using Drugs in a Pro-tumor Microenvironment and P53 Inactivation. As observed in pro-tumor microenvironments with the deactivation of *P53*, the continue expression of dendritic cells (*DC*s) produced a predominance of *Proliferation*, with a complete deactivation of the *Apoptosis* activity (Table 6.26).

In this case, the analysis investigated how drugs could be used to restore a normal microenvironment or at least to reduce the *Proliferation* activity under a complete deactivation of *P53*.

Table 6.29 reported how the *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies.

	APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
1	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	BCL2
2	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	COX2
3	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	COX2, IKK
4	0	0.99	0.86	0.48	0	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	COX2, TNFA
5	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	0	ERK
6	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	0	FOS
7	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	GSK3B
8	0	0.99	0.02	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	IKK
9	0	0.99	0.85	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.82	**0.01**	JAK
10	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0	0	MEK
11	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.01	0	RAF
12	0	0.99	0.86	0.48	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	SMASE
13	0	0.99	0.86	0.48	-	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	TNFA
14	0	0.99	0.86	0.48	0.01	0.96	0	-	0.98	0.98	0.98	0.96	1	0.84	0	TGFB
15	0	0.99	0.86	0.48	0	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.84	0	CCL2

Table 6.29. Alteration of entities activity levels through *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Results in Table 6.29 highlighted the fundamental role of *P53* in cell survival (Zilfou et al. [114]). Considering the complete deactivation of *P53*, no inhibition led to a notable restoring of the *Apoptosis* activity. Only *JAK* inhibition showed a slightly increment of the *Apoptosis* activity.

Influencing Proliferation and Apoptosis using a Mix of Drugs in a Pro-tumor Microenvironment and P53 Inactivation. This paragraph extended the analysis performed in *"Influencing Proliferation and Apoptosis using a Mix of Drugs in a Pro-tumor Microenvironment"* by considering the complete deactivation of *P53*. In this case, analysis aimed at investigating potential combination of drugs to improve obtained showed in Table 6.29.

Table 6.30 showed how the *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies. Table reported only those results with an additive influence (compared with the *Systematic Knock Out*) of *Apoptosis* activity.

Even through a combination of drugs, results in Table 6.30 showed no significant increment in *Apoptosis* activity, confirming yet again the crucial role of *P53* in cell survival. Only the combined inhibition of *IKK-JAK* led to a slightly increment of the *Apoptosis* activity compared with Table 6.29 (row *9*).

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
0	0.99	0.02	0	0.01	0.94	0	0.02	0.98	0.98	0.98	0.94	1	0.8	**0.02**	IKK, JAK

Table 6.30. Alteration of entities activity levels through *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Asynchronous Updating Policy.

Attractors analysis has been performed according to each microenvironment, that is, *non-inflammatory microenvironment* (1), *normal inflammation response* (2), *pro-tumor microenvironment* (3), *pro-tumor microenvironment and P53 inactivation* (4).

All previous analyses reported in *"Synchronous Updating Policy"* are now performed according to the asynchronous updating policy. Table 6.31 showed results obtained through the asynchronous updating policy in terms of simulation time, the corresponding standard deviation (for a batch of one hundred executions), and the number of identified attractors, according to each microenvironment and tool.

Tool	**Microenv.**	**Updating policy**	**Timing in ms**		**No. of attractors**
			Exec. time	*Standard Dev.*	
BoolNet	1	GA	5.45	1.37	**1**
	2	GA	5.02	0.80	**1**
	3	GA	15.41	31.73	**6**
	4	GA	4.67	0.95	**6**
GINsim	1	GA	3,000.00	1,000.00	**1**
	2	GA	3,000.00	1,000.00	**1**
	3	GA	Out of memory		
	4	GA	Out of memory		
BooleanNet	1	GA	11,499.89	493.73	**1**
	2	GA	10,560.72	207.89	**1**
	3	GA	12,023.21	227.93	**6**
	4	GA	11,957.15	216.84	**6**
SyQUAL	1	Time-delayed	10.43	1.87	**1**
	2	Time-delayed	11.60	1.92	**1**
	3	Time-delayed	12.38	1.49	**6**
	4	Time-delayed	13.82	1.71	**6**

Table 6.31. A quantitative overview of attractors analysis according to the asynchronous updating policy. Table reports the execution time and number of identified attractors in agreement with each microenvironment and tool. Despite synchronous updating policy, asynchronous analysis involved different updating policies. In *ROA* (Chapter 2 - *Updating Policies*), all nodes are updated at the same time step, but in a random order, such that no node is updated twice in the same time step. In *GA* (Chapter 2 - *Updating Policies*), a randomly selected node is updated at each time step. In *Time-delayed* (Section 4.4), each node has a pre-selected delay time, being activated at a specific time.

Table 6.31 underlined that all tools (except for *GINsim*) led to the identification of same attractors. Even based on different asynchronous updating policies, *SyQUAL* and *BoolNet* showed (for each microenvironment) the same set of attractors.

Even analyzing the required amount of execution time, *BoolNet* and *SyQUAL* resulted comparable. In contrast, *GINsim* and *BooleanNet* showed a higher request of execution time.

Identified attractors using *SyQUAL* (Table 6.31) are listed in Table 6.32. Each microenvironment showed a different response and attractors complexity, due to the peculiar experimental conditions.

Microenvironment	Attactors
Non-inflammatory microenvironment	1. IKB, GSK3B, APC
Normal inflammation response	1. IKB, GSK3B, APC
Pro-tumor microenvironment	1. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 2. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 3. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 4. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 5. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 6. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS
Pro-tumor microenvironment and P53 inactivation	1. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 2. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 3. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 4. CCL2, IKK, SOCS, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 5. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS 6. CCL2, IKK, CFLIP, AKT, CYCLIND1, Proliferation, IL6, PI3K, DC, MEK, SOD, TH2, STAT3, SMAD7, IL12, IL10, GP130, IL4, JAK, SPHK1, NFKB, BCL2, RAF, RAS, BCATENIN, IAP, ERK, S1P, FOS

Table 6.32. A qualitative overview of attractors analysis according to the asynchronous updating policy. Table reports the list of identified attractors for each microenvironment.

Next paragraphs report the quantitative analysis of the activity level of a set of biological entities of interest, according to each microenvironment.

Non-Inflammatory Microenvironment. Results confirmed what observed during the synchronous analysis. Attractor showed a suppression of *β-catenin* (*BCATENIN*), as result of the constant expression of *APC*. Transcription factors *STAT3* (*STAT3*) and *NF-κB* (*NFKB*) are suppressed as well, confirming what observed experimentally.

It is clear that without any inflammatory signal, the *Proliferation* activity level (Table 6.33) suggests that *IEC*s tend to stay in a resting state (no proliferation).

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
1	0	0	0	0	0	0	0	0	0	0	0	**0**	0

Table 6.33. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). *Proliferation* showed no activity, suggesting a resting state for *IEC*s.

Normal Inflammation Response. Even through an initial activation of dendritic cells (*DC*), attractor showed the same behavior as observed under the non-inflammatory microenvironment. In this case, Table 6.34 showed a greater activation of *DC* compared with the equivalent synchronous due to the different strategy used to activate/deactivate biological entities.

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
1	0	0	0	0	0	0	0	0	0	0	0.75	**0**	0

Table 6.34. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). Even with an initial activation of dendritic cells (*DC*), *Proliferation* continued to show no particular activity.

As proposed for the synchronous analysis, Table 6.35 shows how a mixing of expressed immune cells affected *IEC*s and the activity levels of *Proliferation* and *Apoptosis*, according to the asynchronous updating policy. Results confirmed the same trends observed during the synchronous analysis.

The continuous expression of *DC*, which can be biologically interpreted as a constant activation of dendritic cells, contributed to produce one of the most pro-proliferation microenvironments (rows *3* and *12*), even combined with the activation of *TH2*. Experimental observation showed that transient activation of dendritic cells "triggered" controlled inflammatory reactions (Fiocchi [36]). A constant activation of dendritic cells led to chronic inflammation in *IBD* (Hart et al. [52]), increasing the growth and survival of *IEC*s (MacDonald [81]).

Stimuli	TH2	CTL	DC	MAC	TH1	TREG	APC	BCATENIN	NFKB	STAT3	TNFA	TGFB	IL6	IL4	IL12	IL10	Proliferation	Apoptosis	
	•						1	0.04	0.04	0.04	0	0.9	0.04	0.98	0	0.98	0	**0.49**	**1**
		•					1	0.79	0.82	0.47	0.92	0.02	0.95	0.89	0.92	0.02	**0.32**	0.22	**2**
			•				1	0.87	0.85	0.48	0.01	0.02	0.98	0.98	0.98	0.94	**0.8**	0	**3**
				•			1	0.86	0.88	0.48	0.98	0	0.98	0.94	0.98	0	**0.36**	0.02	**4**
					•		1	0.83	0.85	0.47	0.92	0.02	0.95	0.89	0.92	0.02	**0.38**	0.02	**5**
						•	1	0.05	0.05	0.05	0	0.98	0.05	0	0	0.98	0	**0.48**	**6**
		•	•				1	0.82	0.84	0.48	0.92	0.02	0.98	0.98	0.98	0.02	**0.37**	0.14	**7**
		•		•			1	0.85	0.88	0.48	0.98	0	0.98	0.94	0.98	0	**0.36**	0.04	**8**
	•	•					1	0.02	0.02	0.02	0	0.92	0.02	0.98	0	0.98	0	**0.87**	**9**
		•				•	1	0.02	0.02	0.02	0	0.98	0.02	0	0	0.98	0	**0.87**	**10**
			•		•		1	0.87	0.86	0.48	0.92	0.02	0.98	0.98	0.98	0.02	**0.42**	0.02	**11**
	•		•				1	0.87	0.85	0.48	0	0.02	0.98	0.98	0.98	0.98	**0.82**	0	**12**
			•			•	1	0.86	0.84	0.48	0	0.98	0.98	0.98	0.98	0.98	**0.72**	0	**13**
	•	•	•				1	0.34	0.33	0.48	0	0.02	0.98	0.98	0.98	0.98	0.21	**0.53**	**14**
		•	•			•	1	0.22	0.21	0.48	0	0.98	0.98	0.98	0.98	0.98	0.01	**0.49**	**15**
	•		•		•		1	0.86	0.84	0.48	0	0.02	0.98	0.98	0.98	0.98	**0.79**	0.01	**16**

Table 6.35. Effects of different immune cells expression on *IEC*s. Table reports activity levels of cytokines, as well as the most relevant activity levels of *Apoptosis* and *Proliferation*. The *bullet notation* (•) represents an expressed immune cell.

The constant expression of *MAC* or *TH1* led *Proliferation* to rise its activity at a disadvantage of *Apoptosis* (rows *4* and *5*).

The stable expression of *TH2* or *TREG* produced an important increment of the *Apoptosis* activity (rows *1* and *6*). Considering the *TREG* expression, *IL10* and *TGFB* showed an increasing activity, in contrast with *IL6* and *TNFA*. Experimental observations proved that the T Cell regulatory activity reduced the tumor development through (i) the production of immune suppressive cytokines, such as *TGF-*β and *IL-10*, and (ii) the reduction of pro-inflammatory cytokines, such as *IL-6* and *TNF-*α (Erdman et al. [32]). The combined expression of *TREG-CTL* or *TH2-CTL* led to the most anti-tumorigenic response (row *9* and *10*). Clinical evidences showed that *CTL* plays a role in intestinal inflammation and promotion of tumor growth (Waldner and Neurath [110]), in contrast with previous studies in which *CTL* is usually associated with an encouraging prognosis in sporadic colon cancer (Naito et al. [88]). These combinations indicated a restoring of the cytotoxic activity of *CTL*, leading to an increasing of the immune surveillance activity.

Pro-tumor Microenvironment. Characterized by a constant expression of dendritic cells (*DC*s), attractors showed a *Proliferation* predominance with a complete *Apoptosis* suppression (Table 6.36).

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
0	0.97	0.75	0.44	0	0	0.91	0.01	0.94	0.94	0.97	0.89	**0.73**	0

Table 6.36. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). With a constant activation of dendritic cells (*DC*), *Proliferation* showed a high activity.

In a pro-tumor microenvironment, the activity of transcription factors *STAT3* and *NFKB* was somewhat higher than in non-inflammatory microenvironment and normal inflammatory response. Experimental observations showed that *STAT3* and *NFKB* are high expressed under inflammatory stimuli (Greten et al. [48], Grivennikov et al. [49]).

Pro-tumor Microenvironment and P53 Inactivation. As described in literature, *P53*, known as *TP53*, acts as tumor suppressor, playing a fundamental role in the *Apopotosis* process (Zilfou et al. [114]).

As result of *P53* deactivation, attractors showed once more a *Proliferation* predominance and a complete *Apoptosis* suppression (Table 6.37). In this case, *Proliferation* did not change its activity if compared with the simple pro-tumor microenvironment.

APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis
0	0.97	0.75	0.44	0	0	0.91	0.01	0.94	0.94	0.97	0.89	**0.73**	0

Table 6.37. Table reports the contribution of the entities expressed as activity levels (Section 5.1 - *Results*). With a constant activation of dendritic cells (*DC*) and the deactivation of *P53*, *Proliferation* showed the same activity compared to Table 6.36.

Influencing Proliferation and Apoptosis using Drugs in a Pro-tumor Microenvironment. As observed in pro-tumor microenvironment, the continue expression of dendritic cells (*DC*s) produced a predominance of *Proliferation*, with a complete deactivation of the *Apoptosis* activity (Table 6.25).

This paragraph proposes once more the investigation of how to restore a normal microenvironment or at least to reduce the *Proliferation* activity using drugs. However, the analysis has been performed according to the asynchronous updating policy.

Table 6.38 reported how the *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies.

	APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
1	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	BCL2
2	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	COX2
3	0	0.97	0.01	0.5	0	0.92	0.01	0	0.94	0.94	0.97	0.9	0.99	0.77	0	COX2, IKK
4	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	COX2, TNFA
5	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	ERK
6	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	FOS
7	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	GSK3B
8	0	0.97	0.01	0.5	0	0.92	0.01	0	0.94	0.94	0.97	0.9	0.99	0.77	0	IKK
9	0	0.88	0.21	0	0	0.65	0.03	0	0.76	0.76	0.88	0.59	0.99	0.15	0	JAK
10	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	MEK
11	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	RAF
12	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	SMASE
13	0	0.97	0.75	0.44	-	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	TNFA
14	0	0.97	0.75	0.44	0	0.91	0.01	-	0.94	0.94	0.97	0.89	0.99	0.73	0	TGFB
15	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	CCL2

Table 6.38. Alteration of entities activity levels through *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Results in Table 6.38 showed a suppression of the *Proliferation* activity through the inhibition of *ERK* (row *5*), *FOS* (row *6*), *MEK* (row *10*), and *RAF* (row *11*). The *ERK*, *MEK*, *FOS*, and *RAF* inhibition has been accomplished using the following drugs:

- *ERK* inhibitors:
  - **Purvalanol** experimental drug [*https://www.drugbank.ca/drugs/DB02733*].
- *FOS* inhibitors:
  - **Nadroparin** approved drug [*https://www.drugbank.ca/drugs/DB08813*].
    Nadroparin is a low molecular weight heparin (LMWH) which, when bound to antithrombin III (ATIII), accelerates the inactivation of factor II and factor Xa.
- *MEK* inhibitors:
  - **Cobimetinib**, approved drug [*https://www.drugbank.ca/drugs/DB05239*].
    Cobimetinib is a potent and highly selective small molecule inhibitor of mitogen-activated protein kinase kinase 1 (MAP2K1 or MEK1), and central components of the RAS/RAF/MEK/ERK signal transduction pathway. It is used in combination with vemurafenib for the treatment of patients with unresectable or metastatic BRAF V600 mutation-positive melanoma.
  - **Bosutinib**, approved drug [*https://www.drugbank.ca/drugs/DB06616*].
    Bosutinib is a Bcr-Abl kinase inhibitor for the treatment of Philadelphia chromosome-positive (Ph+) chronic myelogenous leukemia (CML).
  - **Trametinib**, approved drug [*https://www.drugbank.ca/drugs/DB08911*].
    Trametinib dimethyl sulfoxide is a kinase inhibitor.
- *RAF* inhibitors:
  - **Sorafenib** approved and investigational drug [*https://www.drugbank.ca/drugs/DB00398*].
    Sorafenib (rINN) is a drug approved for the treatment of advanced renal cell carcinoma (primary kidney cancer). Sorafenib is a small molecular inhibitor of Raf kinase, PDGF

(platelet-derived growth factor), VEGF receptor 2 & 3 kinases and c Kit the receptor for Stem cell factor.

– **Regorafenib** approved drug [*https://www.drugbank.ca/drugs/DB08896*].
Regorafenib is an inhibitor of multiple kinases. It is used for the treatment of metastatic colorectal cancer and advanced gastrointestinal stromal tumors.
– **Dabrafenib** approved drug [*https://www.drugbank.ca/drugs/DB08912*].
abrafenib mesylate is a reversible ATP-competitive kinase inhibitor and targets the MAPK pathway.

Influencing Proliferation and Apoptosis using a Mix of Drugs in a Pro-tumor Microenvironment. Despite the use of the *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*), the analysis showed no notable advantages in using a combination of drugs to restore a normal microenvironment or at least to reduce the *Proliferation* activity

Influencing Proliferation and Apoptosis using Drugs in a Pro-tumor Microenvironment and P53 Inactivation. As observed in pro-tumor microenvironments with the deactivation of *P53*, the continue expression of dendritic cells (*DC*s) produced a predominance of *Proliferation*, with a complete deactivation of the *Apoptosis* activity (Table 6.37).

Again, the analysis explored how drugs could be used to restore a normal microenvironment or at least to reduce the *Proliferation* activity considering a complete deactivation of *P53*.

Table 6.39 showed how the *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*) has been used to explore rational new immune therapies.

	APC	BCATENIN	NFKB	STAT3	TNFA	TH2	TH1	TGFB	IL6	IL4	IL12	IL10	DC	Proliferation	Apoptosis	Biol. entity knocked out
1	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	BCL2
2	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	COX2
3	0	0.97	0.01	0.5	0	0.92	0.01	0	0.94	0.94	0.97	0.9	0.99	0.77	0	COX2, IKK
4	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	COX2, TNFA
5	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	ERK
6	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	FOS
7	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	GSK3B
8	0	0.97	0.01	0.5	0	0.92	0.01	0	0.94	0.94	0.97	0.9	0.99	0.77	0	IKK
9	0	0.88	0.21	0	0	0.65	0.03	0	0.76	0.76	0.88	0.59	0.97	0.15	0	JAK
10	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	MEK
11	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	**0**	0	RAF
12	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	SMASE
13	0	0.97	0.75	0.44	-	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	TNFA
14	0	0.97	0.75	0.44	0	0.91	0.01	-	0.94	0.94	0.97	0.89	0.99	0.73	0	TGFB
15	0	0.97	0.75	0.44	0	0.91	0.01	0	0.94	0.94	0.97	0.89	0.99	0.73	0	CCL2

Table 6.39. Alteration of entities activity levels through *Systematic Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*). On the right side, biological entities knocked out.

Results emphasized once again the fundamental role of *P53* in the cell survival. However, even considering the complete deactivation of *P53*, Table 6.39 showed the same trends observed in Table 6.38.

Influencing Proliferation and Apoptosis using a Mix of Drugs in a Pro-tumor Microenvironment and P53 Inactivation. Despite the use of the *Combinatorial Knock Out* (Section 5.1 - *Drug Knowledge-based Knock Out*), the analysis showed no notable advantages in using a combination of drugs to restore a normal microenvironment or at least to reduce the *Proliferation* activity

Conclusions

Through the modeling and simulation of the *CAC* network, *SyQUAL* has been compared with the most representative qualitative tools (state of the art) for attractors identification. *SyQUAL* results underlined a best trade-off considering the results accuracy, simulation performance, and usability. *SyQUAL* allowed to reproduce experimental observations, confirming biological trends.

In the context of *non-inflammatory microenvironment* (Table 6.22 and Table 6.33), β*-catenin* showed a complete suppression (as result of the constant expression of *APC*), as well as transcription factors *STAT3* (*STAT3*) and *NF-*κ*B* (*NFKB*), considered with β*-catenin* (*BCATENIN*) distinctive characteristics of *CAC* (Morin et al. [86], Kojima et al. [68], Atreya and Neurath [3], Yu, Pardoll and Jove [113]). Without any inflammatory signal results confirmed that *IEC*s tend to stay in a resting state (no proliferation).

In the context of *normal inflammatory response* (Table 6.23 and Table 6.34), the expression of different types of immune cells influenced the inflammatory microenvironment in different ways (Table 6.24 and Table 6.35). The continuous expression of *DC* contributed to produce one of the most pro-proliferation microenvironments, even combined with the activation of *TH2*. Experimental observation showed that transient activation of dendritic cells "triggered" controlled inflammatory reactions (Fiocchi [36]). Furthermore, a constant activation of dendritic cells led to chronic inflammation in *IBD* (Hart et al. [52]), increasing the growth and survival of *IEC*s (MacDonald [81]). The constant expression of *MAC* or *TH1* led *Proliferation* to rise its activity at a disadvantage of *Apoptosis*. The stable expression of *TH2* or *TREG* produced an important increment of the *Apoptosis* activity. Considering the *TREG* expression, *IL10* and *TGFB* showed an increased activity, in contrast with *IL6* and *TNFA*. Experimental observations proved that the T Cell regulatory activity reduces the tumor development through (i) the production of immune suppressive cytokines, such as *TGF-*β and *IL-10*, and (ii) the reduction of pro-inflammatory cytokines, such as *IL-6* and *TNF-*α (Erdman et al. [32]). The combined expression of *TREG-CTL* or *TH2-CTL* led to the most anti-tumorigenic response. Clinical evidences showed that *CTL* plays a role in intestinal inflammation and promotion of tumor growth (Waldner and Neurath [110]), in contrast with previous studies in which *CTL* is usually associated with an encouraging prognosis in sporadic colon cancer (Naito et al. [88]). These combinations indicate a restoring of the cytotoxic activity of *CTL*, leading to an increasing of the immune surveillance activity.

In *pro-tumor microenvironment* (Table 6.25 and Table 6.36), transcription factors *STAT3* and *NFKB* showed an higher activity when compared with non-inflammatory microenvironment and normal inflammatory response. Experimental observations showed that *STAT3* and *NFKB* are high expressed under inflammatory stimuli (Greten et al. [48], Grivennikov et al. [49]).

Finally, in a *pro-tumor microenvironment* characterized by a *complete deactivation of P53* (Table 6.26 and Table 6.37), results highlighted the fundamental role of *P53* in the cell survival (Zilfou et al. [114]), since no (notable) Apoptosis activity was detected.

7

Conclusion and Future Works

This thesis has provided an efficient platform for modeling and simulation of Biological Systems through *Electronic Design Automation* (*EDA*) techniques. The need of such a platform is due to notable limitations that still affect the most recent qualitative tools. Current tools do not support the simulation complexity of large networks and lack of automation in analyzing biological properties such as complex attractors and molecule vulnerability. Moreover, such tools are designed and customized for specific network type only, such as signaling, gene regulatory, or metabolic networks.

Last but not least, most of these tools only support the SBML *qual* description model for modeling qualitative Biological Systems, in which logic networks report only interactions such as activation and inhibition.

In light of this, the main contribution of this thesis is summarized as follows:

- The proposed platform allows performing both automatic and efficient simulation of Biological Systems, such as signaling and gene regulatory networks. Being based on languages and design tools well-established in the *EDA* field, it allows addressing high computational costs normally associated with the modeling and simulation of biological systems. Performed simulations are not strictly dependent on the size and/or complexity of the network.
- The platform provides both synchronous and asynchronous updating policies for simulation, where the asynchronous method relies on a time-delayed updating policy controlled by topology-based constraints, in contrast with those policies based on random entity evaluation which could lead to unrealistic behaviors.
- Additionally to the support of SBML *qual*, the proposed platform allows modeling and simulation of networks described through SBML Level 2. A SBML Level 2 reactions network model provides a more detailed description of the natural phenomenon, providing a wider set of interactions, by considering stimuli and catalysts.
- Availability of automatisms for network perturbation through *zero* (systematic knockout of all biological entities without any particular criterion) and *drug* knowledge-based knockout approaches. Through drug targeting, perturbed network (under pathological conditions) can be automatically analyzed in order to investigate novel potential pharmacological therapies, using individual or combination of drugs. In such a way, biologists can test their systems according to drug availability.
- Robustness/sensitivity analysis available through a user-friendly interface. Platform provides its functionalities and generated results through a web-oriented interface. Simulation initial conditions, updating policies and other options can be interactively set without requiring coding or specific skills.

Although the achieved results are encouraging, there are still potential interesting extensions and future works:

- Extends the analysis to metabolic networks. Despite signaling and gene regulatory networks in which biological entities are mainly involved in activation/inhibition reactions, metabolic networks describe complex networks of biochemical reactions designed to explain flows of substances.
  Last decades seen the rising of *Petri Nets* (*PNs*) as valid formalism for modeling metabolic networks, in alternative to quantitative models based on *ODE*s (strongly dependent on precise kinetics and quantitative data). Available in various extensions, *PNs* allow the definition of both qualitative and quantitative models. *PNs* are characterized by a graph-based structure, allowing a logic representation of biological phenomena. However, the graph-based structure can be enriched by considering quantitative information. For example, by adding stochastic reaction rates, continuous concentration levels, and so forth. The growing interest in applying *PNs* to model metabolic networks is reflected by the high number of works based on them.
- Provides functionalities to merge and model together different biological networks in order to investigate the resulting system from a holistic perspective. In such a way, user could investigate cross-talking networks, and how networks could influence each other.
  Thus, pathological conditions could be recovered (restoring a normal condition) by targeting (using drugs) biological networks directly involved in the activity of damaged networks. In this way, even considering the absence of drug targets for damaged biological network, acting on its neighborhood could be useful to identify indirected potential pharmacological therapies.
- Provides an integration of (in-vitro and in-vivo) experimental data and biological databases, such as SABIO-Reaction Kinetics Database (Wittig et al. [112]), to better drive the delay times definition through the use of kinetics rates.
- Extends the analysis to new biological systems scenarios, such as cell-to-cell communication. Cell-to-cell communication provides a way to understand complex biological mechanisms underlying phenotypes and how the evolutionary machinery affects them. As direct consequence, the variation of complex biological phenomena is studied under different growth stages, such as development, differentiation, aging, and regeneration, in which biological processes make living beings to be resilient to natural fluctuations (i.e., DNA damaging). A deeper understanding of mechanisms involved in cell-to-cell communication can play a fundamental role to better explain cancers evolution, and how the microenvironment affects its progression and differentiation (e.g., metastasis). Cell-to-cell communication plays an important role in the pathogenesis of different cancer processes. By modeling those biological systems, it could be possible to discover novel therapeutic approaches. At single-cell level, modeling biological systems can improve the understanding of cancers biology. Last technological improvements in single-cell measurements have provided a way to analyze with high accuracy the genetic and proteomic variation, fundamental to better characterize signaling pathways in tumor cells. Modeling the combination of cells interaction, their microenvironment, and tumor single-cell data in a complete model will require high performance systems able to scale, as well as parallel implementations, issues already addressed in the EDA filed.

Part II

Articles

A

Published Contribution

A.1 International Conferences

Distefano, Rosario, 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, pp. 155-161. IEEE, 2016.

Coati, Daniele, Rosario Distefano, Nicola Bombieri, Franco Fummi, Michela Mirenda, Carlo Laudanna, and Rosalba Giugno.
A SystemC-based platform for assertion-based verification and mutation analysis in systems biology.
In 2016 17th Latin-American Test Symposium (LATS), pp. 159-164. IEEE, 2016.

Distefano, Rosario, Franco Fummi, Carlo Laudanna, Nicola Bombieri, and Rosalba Giugno. *A SystemC Platform for Signal Transduction Modelling and Simulation in Systems Biology.*
In Proceedings of the 25th edition on Great Lakes Symposium on VLSI, pp. 233-236. ACM, 2015.

Bombieri, Nicola, Rosario Distefano, Giovanni Scardoni, Franco Fummi, Carlo Laudanna, and Rosalba Giugno.
Dynamic modeling and simulation of leukocyte integrin activation through an electronic design automation framework.
In International Conference on Computational Methods in Systems Biology, pp. 143-154. Springer International Publishing, 2014.

References

1. Yael Abarbanel, Ilan Beer, Leonid Glushovsky, Sharon Keidar, and Yaron Wolfsthal. FoCs: Automatic generation of simulation checkers from formal specifications. In *Proc. of CAV*, pages 538–542, 2000.
2. István Albert, Juilee Thakar, Song Li, Ranran Zhang, and Réka Albert. Boolean network simulations for life scientists. *Source code for biology and medicine*, 3(1):1, 2008.
3. Raja Atreya and Markus F Neurath. Involvement of il-6 in the pathogenesis of inflammatory bowel disease and colon cancer. *Clinical reviews in allergy & immunology*, 28(3):187–195, 2005.
4. Christel Baier, Joost-Pieter Katoen, et al. *Principles of model checking*, volume 26202649. MIT press Cambridge, 2008.
5. N. Barkai and S. Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636):913–917, 1997.
6. Yinon Ben-Neriah and Michael Karin. Inflammation meets cancer, with nf-[kappa] b as the matchmaker. *Nature immunology*, 12(8):715–723, 2011.
7. Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro. *Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008*, volume 5016. Springer, 2008.
8. N. Bombieri, R. Distefano, G. Scardoni, F. Fummi, C. Laudanna, and R. Giugno. Dynamic modeling and simulation of leukocyte integrin activation through an electronic design automation framework. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8859:143–154, 2014.
9. N. Bombieri, F. Fummi, V. Guarnieri, and G. Pravadelli. Testbench qualification of systemc tlm protocols through mutation analysis. *IEEE Transactions on Computers*, 63(5):1248–1261, 2014.
10. N. Bombieri, F. Fummi, V. Guarnieri, G. Pravadelli, F. Stefanni, T. Ghasempouri, M. Lora, G. Auditore, and M.N. Marcigaglia. Reusing rtl assertion checkers for verification of systemc tlm models. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 31(2):167–180, 2015.
11. Benjamin J Bornstein, Sarah M Keating, Akiya Jouraku, and Michael Hucka. Libsbml: an api library for sbml. *Bioinformatics*, 24(6):880–881, 2008.
12. Luca Bortolussi and Alberto Policriti. Hybrid systems and biology. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pages 424–448. Springer, 2008.
13. M. Boulé and Z. Zilic. *Generating hardware assertion checkers: for hardware verification, emulation, post-fabrication debugging and on-line monitoring*. Springer, 2008.

14. Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, 16:16, 1998.
15. Muffy Calder and Jane Hillston. Process algebra modelling styles for biomolecular processes. In *Transactions on computational systems biology XI*, pages 1–25. Springer, 2009.
16. Claudine Chaouiya, Duncan Bérenguier, Sarah M Keating, Aurélien Naldi, Martijn P Van Iersel, Nicolas Rodriguez, Andreas Dräger, Finja Büchel, Thomas Cokelaer, Bryan Kowal, et al. Sbml qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC systems biology*, 7(1):1, 2013.
17. Madalena Chaves, Reka Albert, and Eduardo D Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *Journal of theoretical biology*, 235(3):431–449, 2005.
18. Madalena Chaves, Eduardo D Sontag, and Réka Albert. Methods of robustness analysis for boolean models of gene control networks. *arXiv preprint q-bio/0605004*, 2006.
19. K.-T. Cheng and J.-Y Jou. A single-state-transition fault model for sequential machines. In *Proc. of IEEE ICCAD*, pages 226–229, 1990.
20. Federica Ciocchetta and Jane Hillston. Process algebras in systems biology. In *Formal methods for computational systems biology*, pages 265–312. Springer, 2008.
21. UniProt Consortium et al. The universal protein resource (uniprot). *Nucleic acids research*, 36(suppl 1):D190–D195, 2008.
22. G. Constantin, M. Majeed, C. Giagulli, L. Piccio, J.Y. Kim, E.C. Butcher, and C. Laudanna. Chemokines trigger immediate beta2 integrin affinity and mobility changes: differential regulation and roles in lymphocyte arrest under flow. *Immunity*, 13:759–769, 2000.
23. Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *CONCUR 2007–Concurrency Theory*, pages 17–41. Springer, 2007.
24. Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.
25. Hidde De Jong. Modeling and simulation of genetic regulatory systems: a literature review. *Journal of computational biology*, 9(1):67–103, 2002.
26. Hidde De Jong, Johannes Geiselmann, Céline Hernandez, and Michel Page. Genetic network analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics*, 19(3):336–344, 2003.
27. 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.
28. Alessandro Di Cara, Abhishek Garg, Giovanni De Micheli, Ioannis Xenarios, and Luis Mendoza. Dynamic simulation of regulatory networks using squad. *BMC bioinformatics*, 8(1):1, 2007.
29. Rosario Distefano, Franco Fummi, Carlo Laudanna, Nicola Bombieri, and Rosalba Giugno. A systemc platform for signal transduction modelling and simulation in systems biology. In *Proceedings of the 25th Edition on Great Lakes Symposium on VLSI*, GLSVLSI '15, pages 233–236, 2015.
30. Elena Dubrova and Maxim Teslenko. A sat-based algorithm for computing attractors in synchronous boolean networks. *arXiv preprint arXiv:0901.4448*, 2009.
31. Richard H Duerr, Kent D Taylor, Steven R Brant, John D Rioux, Mark S Silverberg, Mark J Daly, A Hillary Steinhart, Clara Abraham, Miguel Regueiro, Anne Griffiths, et al. A genome-wide association study identifies il23r as an inflammatory bowel disease gene. *science*, 314(5804):1461–1463, 2006.
32. Susan E Erdman, Jane J Sohn, Varada P Rao, Prashant R Nambiar, Zhongming Ge, James G Fox, and David B Schauer. Cd4+ cd25+ regulatory lymphocytes induce regression of intestinal tumors in apcmin/+ mice. *Cancer research*, 65(10):3998–4004, 2005.

33. James R Faeder, Michael L Blinov, and William S Hlavacek. Rule-based modeling of biochemical systems with bionetgen. *Systems biology*, pages 113–167, 2009.
34. Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.
35. David Fell and Athel Cornish-Bowden. *Understanding the control of metabolism*, volume 2. Portland press London, 1997.
36. Claudio Fiocchi. The normal intestinal mucosa: a state of controlled inflammation. In *Inflammatory bowel disease: from bench to bedside*, pages 101–120. Springer, 2003.
37. Jasmin Fisher and Thomas A Henzinger. Executable cell biology. *Nature biotechnology*, 25(11):1239–1249, 2007.
38. Jasmin Fisher and Thomas A Henzinger. Executable cell biology. *Nature Biotechnology*, 25:1239 – 1249, 2007.
39. Akira Funahashi, Mineo Morohashi, Hiroaki Kitano, and Naoki Tanimura. Celldesigner: a process diagram editor for gene-regulatory and biochemical networks. *Biosilico*, 1(5):159–162, 2003.
40. Clare E Giacomantonio and Geoffrey J Goodhill. A boolean model of the gene regulatory network underlying mammalian cortical area development. *PLoS Comput Biol*, 6(9):e1000936, 2010.
41. D Gilbert, H Fuss, X Gu, R Orton, S Robinson, V Vyshemirsky, M J Kurth, C S Downes, and W Dubitzky. Computational methodologies for modelling, analysis and simulation of signalling networks. *Briefings in Bioinformatics*, 7(4):339–353, 2006.
42. 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.
43. Leon Glass and Stuart A Kauffman. The logical analysis of continuous, non-linear biochemical control networks. *Journal of theoretical Biology*, 39(1):103–129, 1973.
44. Bastien D Gomperts, Ijsbrand M Kramer, and Peter ER Tatham. *Signal transduction*. Academic Press, 2009.
45. A Gonzalez Gonzalez, Aurélien Naldi, Lucas Sanchez, Denis Thieffry, and Claudine Chaouiya. Ginsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *Biosystems*, 84(2):91–100, 2006.
46. E. Gonalves, J. Bucher, A. Ryll, J. Niklas, K. Mauch, S. Klamt, M. Rocha, and J. Saez-Rodriguez. Bridging the layers: Towards integration of signal transduction, regulation and metabolism into mathematical models. *Molecular BioSystems*, 9(7):1576–1583, 2013.
47. Peter JE Goss and Jean Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic petri nets. *Proceedings of the National Academy of Sciences*, 95(12):6750–6755, 1998.
48. Florian R Greten, Lars Eckmann, Tim F Greten, Jin Mo Park, Zhi-Wei Li, Laurence J Egan, Martin F Kagnoff, and Michael Karin. Ikkβ links inflammation and tumorigenesis in a mouse model of colitis-associated cancer. *Cell*, 118(3):285–296, 2004.
49. Sergei Grivennikov, Eliad Karin, Janos Terzic, Daniel Mucida, Guann-Yi Yu, Sivakumar Vallabhapurapu, Jürgen Scheller, Stefan Rose-John, Hilde Cheroutre, Lars Eckmann, et al. Il-6 and stat3 are required for survival of intestinal epithelial cells and development of colitis-associated cancer. *Cancer cell*, 15(2):103–113, 2009.
50. Xing Guo and Xiao-Fan Wang. Signaling cross-talk between tgf-β/bmp and other pathways. *Cell research*, 19(1):71–88, 2009.
51. David Harel. Statecharts: A visual formalism for complex systems. *Science of computer programming*, 8(3):231–274, 1987.
52. Ailsa L Hart, Hafid Omar Al-Hassi, Rachael J Rigby, Sally J Bell, Anton V Emmanuel, Stella C Knight, Michael A Kamm, and Andrew J Stagg. Characteristics of intestinal dendritic cells in inflammatory bowel diseases. *Gastroenterology*, 129(1):50–65, 2005.

53. Inman Harvey and Terry Bossomaier. Time out of joint: Attractors in asynchronous random boolean networks. In *Proc. of European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge, 1997.
54. Monika Heiner, David Gilbert, and Robin Donaldson. Petri nets for systems and synthetic biology. In *Formal methods for computational systems biology*, pages 215–264. Springer, 2008.
55. Monika Heiner, Ina Koch, and Jürgen Will. Model validation of biological pathways using petri nets?demonstrated for apoptosis. *Biosystems*, 75(1):15–28, 2004.
56. Reinhart Heinrich and Stefan Schuster. *The regulation of cellular systems*. Springer Science & Business Media, 2012.
57. Tomáš Helikar, Bryan Kowal, Sean McClenathan, Mitchell Bruckner, Thaine Rowley, Alex Madrahimov, Ben Wicks, Manish Shrestha, Kahani Limbu, and Jim A Rogers. The cell collective: toward an open and collaborative approach to systems biology. *BMC systems biology*, 6(1):96, 2012.
58. William S Hlavacek, James R Faeder, Michael L Blinov, Richard G Posner, Michael Hucka, and Walter Fontana. Rules for modeling signal-transduction systems. *Sci STKE*, 2006(344):re6, 2006.
59. Stefan Hoops, Sven Sahle, Ralph Gauges, Christine Lee, Jürgen Pahle, Natalia Simus, Mudita Singhal, Liang Xu, Pedro Mendes, and Ursula Kummer. Copasia complex pathway simulator. *Bioinformatics*, 22(24):3067–3074, 2006.
60. 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.
61. G Joshi-Tope, Marc Gillespie, Imre Vastrik, Peter D'Eustachio, Esther Schmidt, Bernard de Bono, Bijay Jassal, GR Gopinath, GR Wu, Lisa Matthews, et al. Reactome: a knowledgebase of biological pathways. *Nucleic acids research*, 33(suppl 1):D428–D432, 2005.
62. William G Kaelin. The concept of synthetic lethality in the context of anticancer therapy. *Nature reviews cancer*, 5(9):689–698, 2005.
63. Guy Karlebach and Ron Shamir. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10):770–780, 2008.
64. Stuart Kauffman. Homeostasis and differentiation in random genetic control networks. *Nature*, 224:177–178, 1969.
65. Stuart A Kauffman. *The origins of order: Self organization and selection in evolution*. Oxford University Press, USA, 1993.
66. Steffen Klamt, Julio Saez-Rodriguez, and Ernst D Gilles. Structural and functional analysis of cellular networks with cellnetanalyzer. *BMC systems biology*, 1(1):1, 2007.
67. P Kohl, EJ Crampin, TA Quinn, and D Noble. Systems biology: an approach. *Clinical Pharmacology & Therapeutics*, 88(1):25–33, 2010.
68. MASAYUKI KOJIMA, TAKASHI MORISAKI, NOBUHIKO SASAKI, KENJI NAKANO, RYUICHI MIBU, MASAO TANAKA, and MITSUO KATANO. Increased nuclear factor-kb activation in human colorectal carcinoma and its correlation with tumor progression. *Anticancer research*, 24(2B):675–682, 2004.
69. Raivo Kolde. Pheatmap: pretty heatmaps. *R package version*, 61, 2012.
70. Jan Krumsiek, Sebastian Pölsterl, Dominik M Wittmann, and Fabian J Theis. Odefy-from discrete to continuous models. *BMC bioinformatics*, 11(1):1, 2010.
71. Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification. 1999.
72. Nicolas Le Novere. Quantitative and logic modelling of molecular and gene networks. *Nat Rev Genet*, 16(3):146–158, March 2015.
73. Nicolas Le Novère. Quantitative and logic modelling of molecular and gene networks. *Nature Reviews Genetics*, 16(3):146–158, 2015.

74. Nicolas Le Novere, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, Sarala M Wimalaratne, et al. The systems biology graphical notation. *Nature biotechnology*, 27(8):735–741, 2009.
75. Tong Ihn Lee, Nicola J Rinaldi, François Robert, Duncan T Odom, Ziv Bar-Joseph, Georg K Gerber, Nancy M Hannett, Christopher T Harbison, Craig M Thompson, Itamar Simon, et al. Transcriptional regulatory networks in saccharomyces cerevisiae. *science*, 298(5594):799–804, 2002.
76. Klaus Ley, Carlo Laudanna, Myron I Cybulsky, and Sussan Nourshargh. Getting to the site of inflammation: the leukocyte adhesion cascade updated. *Nature Reviews Immunology*, 7(9):678–689, 2007.
77. Fangting Li, Tao Long, Ying Lu, Qi Ouyang, and Chao Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, 2004.
78. Peter Lisherness and Kwang-Ting (Tim) Cheng. SCEMIT: A SystemC error and mutation injection tool. In *Proc. of ACM/IEEE DAC*, pages 228 –233, 2010.
79. Catherine M Lloyd, Matt DB Halstead, and Poul F Nielsen. Cellml: its future, present and past. *Progress in biophysics and molecular biology*, 85(2):433–450, 2004.
80. Junyan Lu, Hanlin Zeng, Zhongjie Liang, Limin Chen, Liyi Zhang, Hao Zhang, Hong Liu, Hualiang Jiang, Bairong Shen, Ming Huang, et al. Network modelling reveals the mechanism underlying colitis-associated colon cancer and identifies novel combinatorial anti-cancer targets. *Scientific reports*, 5, 2015.
81. Thomas T MacDonald. Cytokine regulation of intestinal epithelial cell proliferation. *Expert review of clinical immunology*, 6(4):531–535, 2010.
82. Deborah L McGuinness, Frank Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.
83. Thomas Mestl, Erik Plahte, and Stig W Omholt. A mathematical framework for describing and analysing gene regulatory networks. *Journal of Theoretical Biology*, 176(2):291–300, 1995.
84. Nicholas AM Monk. Oscillatory expression of hes1, p53, and nf-κb driven by transcriptional time delays. *Current Biology*, 13(16):1409–1413, 2003.
85. Alessio Montresor, Matteo Bolomini-Vittori, Lara Toffali, Barbara Rossi, Gabriela Constantin, and Carlo Laudanna. Jak tyrosine kinases promote hierarchical activation of rho and rap modules of integrin activation. *The Journal of cell biology*, 203(6):1003–1019, 2013.
86. Patrice J Morin, Andrew B Sparks, Vladimir Korinek, Nick Barker, Hans Clevers, Bert Vogelstein, and Kenneth W Kinzler. Activation of β-catenin-tcf signaling in colon cancer by mutations in β-catenin or apc. *Science*, 275(5307):1787–1790, 1997.
87. Christoph Müssel, Martin Hopfensitz, and Hans A Kestler. Boolnetan r package for generation, reconstruction and analysis of boolean networks. *Bioinformatics*, 26(10):1378–1380, 2010.
88. Yoshitaka Naito, Kazuya Saito, Kenichi Shiiba, Akio Ohuchi, Katsunori Saigenji, Hiroshi Nagura, and Haruo Ohtani. Cd8+ t cells infiltrated within cancer cell nests as a prognostic factor in human colorectal cancer. *Cancer research*, 58(16):3491–3494, 1998.
89. Flemming Nielson, Hanne R Nielson, and Chris Hankin. *Principles of program analysis*. Springer, 1999.
90. M Noguchi, N Hiwatashi, Z Liu, and T Toyota. Secretion imbalance between tumour necrosis factor and its inhibitor in inflammatory bowel disease. *Gut*, 43(2):203–209, 1998.
91. Béla Novák and John J Tyson. Design principles of biochemical oscillators. *Nature reviews Molecular cell biology*, 9(12):981–991, 2008.
92. Jeffrey D Orth, Tom M Conrad, Jessica Na, Joshua A Lerman, Hojung Nam, Adam M Feist, and Bernhard Ø Palsson. A comprehensive genome-scale reconstruction of escherichia coli metabolism2011. *Molecular systems biology*, 7(1):535, 2011.

93. RKS Phillips, MH Wallace, PM Lynch, E Hawk, GB Gordon, BP Saunders, N Wakabayashi, Y Shen, S Zimmerman, L Godio, et al. A randomised, double blind, placebo controlled study of celecoxib, a selective cyclooxygenase 2 inhibitor, on duodenal polyposis in familial adenomatous polyposis. *Gut*, 50(6):857–860, 2002.
94. Boryana K Popivanova, Kazuya Kitamura, Yu Wu, Toshikazu Kondo, Takashi Kagaya, Shiuchi Kaneko, Masanobu Oshima, Chifumi Fujii, and Naofumi Mukaida. Blocking tnf-α in mice reduces colorectal carcinogenesis associated with chronic colitis. *The Journal of clinical investigation*, 118(2):560–570, 2008.
95. Corrado Priami. Algorithmic systems biology. *Communications of the ACM*, 52(5):80–88, 2009.
96. Maria Rescigno and Antonio Di Sabatino. Dendritic cells in intestinal homeostasis and disease. *The Journal of clinical investigation*, 119(9):2441–2450, 2009.
97. Assieh Saadatpour, István Albert, and Réka Albert. Attractor analysis of asynchronous boolean models of signal transduction networks. *Journal of theoretical biology*, 266(4):641–656, 2010.
98. 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 Comput Biol*, 3(8):e163, 2007.
99. Regina Samaga and Steffen Klamt. Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell Communication and Signaling*, 11(1), 2013.
100. Regina Samaga and Steffen Klamt. Modeling approaches for qualitative and semi-quantitative analysis of cellular signaling networks. *Cell communication and signaling*, 11(1):1, 2013.
101. 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 Comput Biol*, 5(12):e1000595, 2009.
102. Thomas Schlitt and Alvis Brazma. Current approaches to gene regulatory network modelling. *BMC bioinformatics*, 8(6):1, 2007.
103. Stefan Schuster, Thomas Dandekar, and David A Fell. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in biotechnology*, 17(2):53–60, 1999.
104. El Houssine Snoussi. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. *Dynamics and stability of Systems*, 4(3-4):565–583, 1989.
105. Peter Szlosarek, Kellie A Charles, and Frances R Balkwill. Tumour necrosis factor-α as a tumour promoter. *European journal of cancer*, 42(6):745–750, 2006.
106. Carolyn Talcott and David L Dill. Multiple representations of biological processes. In *Transactions on Computational Systems Biology VI*, pages 221–245. Springer, 2006.
107. René Thomas. Boolean formalization of genetic control circuits. *Journal of theoretical biology*, 42(3):563–585, 1973.
108. René Thomas. Regulatory networks seen as asynchronous automata: a logical description. *Journal of theoretical biology*, 153(1):1–23, 1991.
109. René Thomas, Denis Thieffry, and Marcelle Kaufman. Dynamical behaviour of biological regulatory networksi. biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of mathematical biology*, 57(2):247–276, 1995.
110. Maximilian J Waldner and Markus F Neurath. Colitis-associated cancer: the role of t cells in tumor development. In *Seminars in immunopathology*, volume 31, pages 249–256. Springer, 2009.
111. David S Wishart, Craig Knox, An Chi Guo, Savita Shrivastava, Murtaza Hassanali, Paul Stothard, Zhan Chang, and Jennifer Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic acids research*, 34(suppl 1):D668–D672, 2006.
112. Ulrike Wittig, Renate Kania, Martin Golebiewski, Maja Rey, Lei Shi, Lenneke Jong, Enkhjargal Algaa, Andreas Weidemann, Heidrun Sauer-Danzwith, Saqib Mir, et al. Sabio-rkdatabase for biochemical reaction kinetics. *Nucleic acids research*, 40(D1):D790–D796, 2012.

113. Hua Yu, Drew Pardoll, and Richard Jove. Stats in cancer inflammation and immunity: a leading role for stat3. *Nature Reviews Cancer*, 9(11):798–809, 2009.
114. Jack T Zilfou and Scott W Lowe. Tumor suppressive functions of p53. *Cold Spring Harbor perspectives in biology*, 1(5):a001883, 2009.