MASTER IN COMPUTER SCIENCE

Foundations for an Expert System Managing Gene Regulatory Networks in the Context of Cancer Traetmaent

Tonus, Vincent

Award date:
2018

Awarding institution:
University of Namur

University of Namur
Faculty of Computer Science
Academic Year 2017–2018

Foundations for an Expert System Managing Gene Regulatory Networks in the Context of Cancer Treatment

Vincent Tonus

Supervisor: _____ (Signed for Release Approval - Study Rules art. 40)
Jean-Marie Jacquet

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Computer Science at the University of Namur

ABSTRACT

English

The progress made during the last 20 years in terms of gene sequencing and editing have allowed cancer treatment techniques to take a bold step forward. Indeed, the study of cell signaling and gene regulations mechanisms now allows to better comprehend various cancers development and thus, propose more efficient and targeted treatments than in the past. But the path between the biology and medical worlds requires the analysis, by specialists, of a large amount of data to be able to provide doctors with precise and easily exploitable information. This thesis tries to lay down the foundation of an expert system able to automatically treat this information and help specialists in their analysis.

Français

Les avancées faites ces 20 dernires années en termes de séquencement et d'édition génétique ont permis une progression spectaculaire des méthodes de traitement du cancer. En effet, l'étude de réseaux de signalisation cellulaire et des mécanismes de régulation génétique permet maintenant de mieux appréhender le développement de nombreux cancers et ainsi de proposer des traitements plus efficaces et plus ciblés qu'auparavant. Mais le passage entre le monde biologique et le monde médical nécessite l'analyse, par des spécialistes, d'une grande quantité de données afin de pouvoir fournir aux médecins des informations précises et facilement exploitables. Ce mémoire tente de poser les bases d'un système expert capable de traiter automatiquement ces informations afin d'aider les spécialistes dans leur travail d'analyse.

Keywords

Biological Networks, Cell Signaling, Signaling Pathways, Gene Regulatory Networks, Cancer, Expert Systems, Boolean Networks, Simplification, Drugs

ACKNOWLEDGMENT

I would like to thank my supervisor, Prof. Jean-Marie Jacquet, for his help and support throughout this work.

I would also like to express my gratitude towards Dr. Denis Burton, Julien Nocco, Julien Tonka and Dr. Céline Tonus for their input and constructive criticism of this manuscript.

And finally, thank you to all the friends and family that have encouraged and supported me throughout my years of study and the writing of this thesis.

Vincent Tonus

CONTENTS

GLOSSARY

A

Aerobic Characterized by the presence of oxygen. 17

Allele Specific variation of a gene. 4

Amino acid Organic molecule considered as the "building blocks" of the proteins in the human body. 5

Anaerobic Characterized by the lack of oxygen. 17

Antigen Molecule capable of inducing an immune response. 18

Apoptosis Form of programmed cell death. 11

Apoptotic Referring to apoptosis. 11

C

Chromosome Long strand of DNA containing multiple genes. 4

D

Differentiation Mechanism through which a cell changes from one cell type to another. 6

DNA **D**eoxyribo**N**ucleic **A**cid is present in nearly all living organisms as the main constituent of chromosomes. It is the carrier of genetic information. 4

E

Epithelial Located in the outer surface of organs or delimiting the inner surface of a cavity. 13

Extravasation Leakage of a content from inside a vessel to the extravascular tissue. 14

G

Gene Sequence of DNA encoding proteins. 3

Gene pool Set of all the genetic information of a species. 4

Genome The entire repertoire of an organism's genetic information. 3

Genotype Type and arrangements of the genes of a living organism. 3

I

Intravasation Process through which cancer cells enter blood or lymphatic vessels. 14

M

Macroscopic Large enough to be seen without optical magnifying instruments. 13

N

Neoplastic Related to the uncontrolled growth of abnormal tissue. 15

Nucleotide Base element of nucleic acids like DNA. The four types of nucleotides that compose DNA are Adnine (A), Cytosine (C), Guanine (G) and Thymine (T). 12

O

Oncogenic That causes the formation of tumors. 3

P

Parenchyma Ensemble of cells that constitute the functional part of a tissue, in constrast with the stroma. 13

Phagocyte Any cell that ingests and destroys foreign particles, bacteria, and cell debris. 11

Phenotype Set of visible, physical characteristics of a living organism. 3

Proteins Large biomolecules composed of one or more chain of amino-acids. They are responsible for a vast array of actions in the human body. 5

S

Stroma In a tissue, environment of the cells with a structural or connective role (blood vessels, connective tissue...). 10

CHAPTER

1

INTRODUCTION

The last 20 years have seen the apparition and development of revolutionary technologies in the field of genetics. Two of the most important ones are high-throughput sequencing, that allows faster and cheaper DNA sequencing [1], and a gene editing tool named CRISPR, allowing highly efficient DNA editing [2]. The combination of these technologies provided cancer researchers with the tools to analyze thousands of cancer genomes and generate precise genetically modified systems, allowing them to validate the consequences of genetic mutations on cancer phenotypes.

These technologies and the tremendous progress they have fostered in understanding cells signaling pathways and gene regulatory networks have also opened the road for a new approach on cancer treatment. The sequencing of cancer cells' DNA now allows biologists to map these cells' gene regulatory networks, and the analysis of these networks gives precious insights on treatment prescriptions and their potential effects.

But to infer treatment options from these networks and generate relevant data for oncologists, a lot of work is necessary. Biologists need to find out which types of drugs could be used to activate or inhibit the right genes expression in cancer cells to stop the tumor progression without compromising the integrity of the other healthy cells in the body. They then need to run simulations on models of these gene regulatory networks, including various combinations of these drugs, to find out which combinations have the best chance to work. Also, because some aspects are still not well understood, the output of these analyses is refined based on statistics of the efficiency of previous similar treatments.

This study lays out the foundation of an expert system able to help biologists in this work so that the time cost of these analysis can be reduced. The focus is put on two specific use cases: the simplification of gene regulatory networks, so that they can be more human readable, and the inference of the drugs that could have a positive effect on the analyzed cancer cells. The result is a portable .NET Core library with the ability to perform these two tasks based on qualitative models of the networks and the usage of logical regulatory networks. A test application to run this library is also provided.

This document is structured as follows. Chapter 2 introduces the necessary biological background to understand the way cancer behaves and the development of targeted therapy. Chapter 3 then goes more into details over the goal of the system developed during this work. Chapter

4 gives an insight on the current state of the art in terms of software, data encoding standards and data availability. Chapter 6 then briefly introduces the various modeling and simulation techniques used to work with gene regulatory networks. Chapter 5 details the structure, architecture and technologies of the application while Chapters 7 and 8 then describe the process and techniques used to implement both use cases of the system. Finally, Chapter 9 discusses the strengths, weaknesses and future perspectives of the system while Chapter 10 exposes the conclusions of this work.

CHAPTER

2

BIOLOGICAL BACKGROUND

This chapter introduces the necessary biological background to understand the aim of this work. The beginning of the chapter is a reminder of basic principles of cell and organism genetics and biology. Then, cell signaling, signaling pathways and gene regulatory networks are explained, as well as their role in cancer development. The reader will then come to understand how gene mutations can impact these networks and induce oncogenic behaviors. Finally, cancer treatments will be discussed with a focus on targeted treatments based on patients' genetic information analysis.

Note that this chapter is not an exhaustive explanation of all the biological concepts behind cancer as they are known. It aims to provide the inexperienced reader with enough information to understand the concepts that drive this work.

2.1 Introduction to cell and organism biology

A basic understanding of cells biology and genetics is important to understand how normal cells work and how they can evolve and grow to finally become a tumor. This section introduces the necessary prerequisites before going further into details.

2.1.1 Basics of genetic

In the 1860s, Gregor Mendel's work on the breeding of pea plants discovered many of the basic rules of genetics. Of course, some of these rules have been somehow updated by later researches to fit all living cells, but most of his work still matches today's understanding [3].

Amongst other things, he observed that the transfer of genetic information from an organism to its offspring could be explained by a set of rules, suggesting that the entire genetic properties of living organisms (its genome) was organized as a collection of discrete, separable information packets called *genes* [3].

His work also brought to light the notions of *phenotype* and *genotype*, the former being the set of visual characteristics of an organism, and the later, the type and arrangements of its genes (e.g. blue eyes is a phenotype, while the specific sequences in the genes encoding eyes color is the genotype). He discovered that the genotype of an organism could be divided in a set of

independent genes and that the *chromosomes* actually carry two sets of the same gene (except for the sex chromosomes) [3].

These two sets, called *alleles*, can carry different interpretations of the gene. In this case the organism would be called heterozygous for that gene, in opposition to an homozygous organism carrying two identical alleles of the gene [3]. In case of heterozygous organisms, the phenotype encoded by one allele will be dominant and the other one recessive.[1]

Here is a practical example of dominance and recessiveness in cancer development (Figure 2.1). Some people carry a defective allele of the gene encoding proteins involved in DNA repair. This allele is relatively rare and behaves recessively so its phenotype is not apparent. But if two heterozygous people, carrying the defective allele mate, one fourth of their offspring, on average, will inherit two defective alleles. These people, now homozygous for mutant allele, will then lack the DNA repair function the normal gene should express and be more propitious to develop certain kinds of cancers [3].

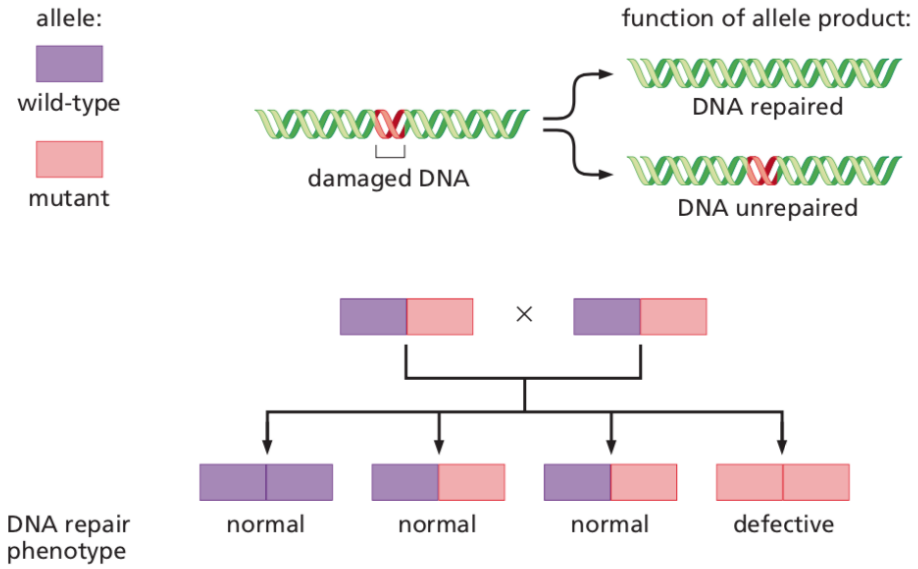

Figure 2.1: Genotype, phenotype and heredity. Some individuals can carry a dominant allele, encoding the DNA repair function, and a recessive mutant allele, whose DNA repair function is impaired. If two of these individuals mate, one fourth of their offspring will end up with both mutant alleles and express a deficient DNA repair phenotype [3].

2.1.2 Genes mutations and Darwinian evolution

Something Mendel's research did not explain is how multiple alleles of a gene could appear. They seemed to just be present in the gene pool of a species. But in the 1920s and 1930s, it appeared that the alteration of genes and creation of new alleles was due to mutations. Genetic mutations occur throughout the lifespan of a species, continuously increasing the number of different alleles in the genome of its members [3].

This implies that the older a species is, the more heterogeneous its gene pool is and the more variety of alleles in its genome it has. Humans for example, being a relatively young species of <150,000 years old, have an alleles count three time lower than chimpanzees. It can then be

[1] This is not completely accurate. Later research will demonstrate that the alleles of some genes can be co-dominant, their phenotype thus being a blend of the two alleles expressions. They can also be dominant but not expressed because their expression depends on other genes in the organism [3].

inferred that chimpanzees have been around for $\sim$ 450,000 years [3].

Although, while this is theoretically true, the reality is slightly more complex. These continuous genetic mutations are somehow "regulated" by the rules of natural selection described by Charles Darwin. Indeed, some alleles may confer upon individuals carrying them better survivability, while others will be expressed as some sort of handicap. Their carriers will then have more difficulty surviving (if they can survive) and the allele will probably disappear completely from the gene pool [3].

Also, that does not mean that these mutations necessarily change the individuals, because not all mutations imply a change in the organism phenotype. Research has shown that, in the $\sim$ 21,000 genes that compose the human genome, only $\sim$ 3.5% carry biologically relevant sequences, impacting our phenotype. This means that only mutations occurring in these 3.5% are subject to natural selection while countless mutations in the so-called "junk DNA" survive in our gene pool and have absolutely no impact on individuals' phenotypes (see Figure 2.2) [3].

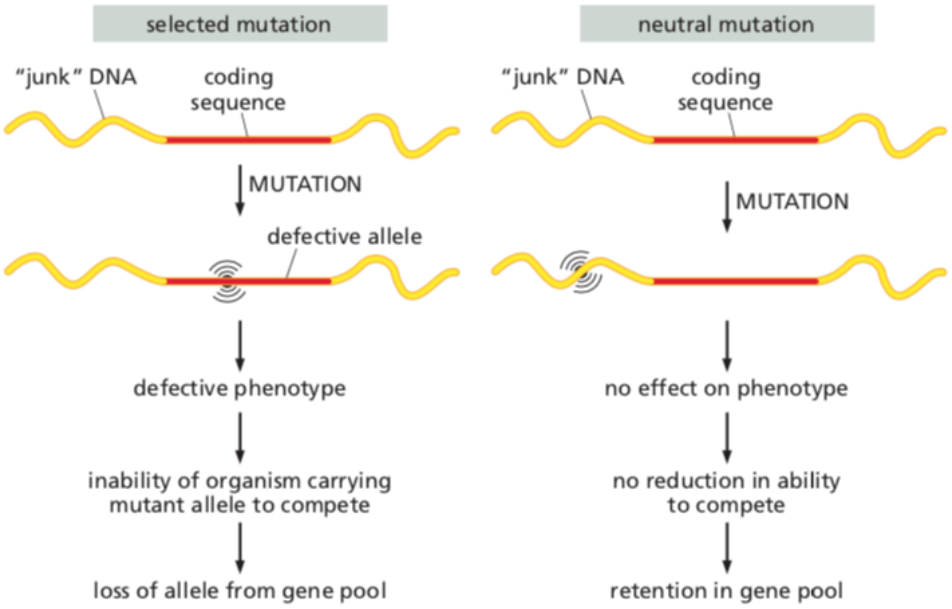

Figure 2.2: Neutral mutations and evolution. Mutations that occur on a coding sequence (red) of the DNA (left) can result in a defective phenotype and compromise the organism ability to survive. The defective allele would then be lost from the species gene pool. On the contrary, mutations occurring on the non-coding part (yellow) of the DNA do not express any phenotype. Therefore, most of the time, they are preserved in the species gene pool [3].

2.1.3 The cellular and organismic phenotypes

What also lacks in Medelian genetics, is the explanation of how genes create cellular and organismic phenotypes. In other words, how can the information stored in genes influence the way cells look and behave.

The basic concepts to answer that question where first introduced in 1944, when DNA was proven to be the chemical entity in which genetic information is stored. In the following twenty years, Watson and Crick elucidated the double-helical structure of DNA [4] and it became clear that the sequence of amino acids in proteins are determined by the sequences of bases in the DNA [3].

These proteins, once synthesized, create phenotype in multiple ways. The ones within cells will determine the behavior of these cells as an entity, while the ones secreted in the space between

cells will form the extracellular matrix (ECM) that ties cells together to form complex tissues. Proteins can, for example, act as enzyme, catalyzing chemical reactions inside cells, or they can contract and create cellular movement or muscle contraction [3].

2.1.4 Cell Signaling

Communication inside and between cells can be seen as an electronic integrated circuit where transistors are replaced by proteins [5]. The term *cell signaling* is used to talk about the different intra- and extra-cellular communication channels. In this work, focus will be put on intra-cellular signaling and the way signals from cell surface receptors are transmitted and interpreted inside the cell.

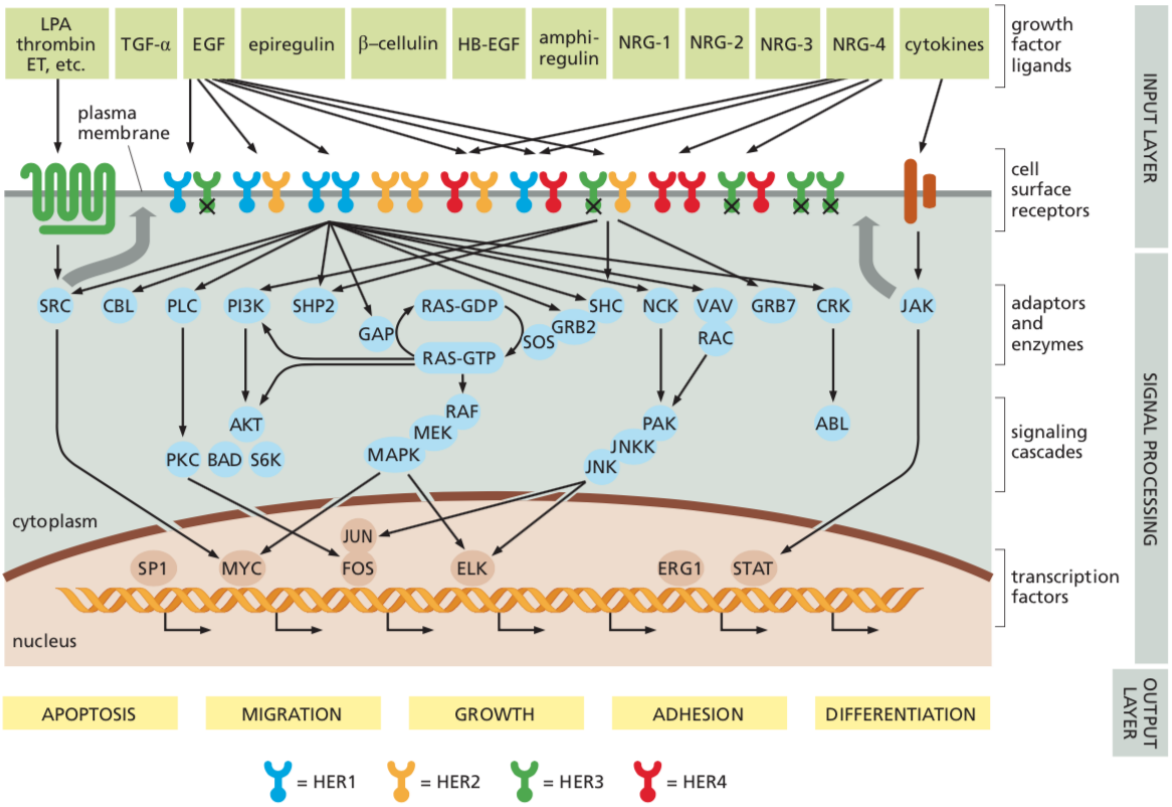

Figure 2.3: The human EGF receptor (HER) signaling network. Growth factors interacts with cell surface receptors that transmit signals inside the cell. These signals are then processed by a complex protein network until they reach the transmission factors in the nucleus. Transmission factors then express their associated genes resulting in a variety phenotypes [3, 6].

Cells in our body all carry the same, complete, genetic information in their chromosomes. But that does not mean all cells are the same. Muscle cells do not behave the same way as skin cells or bone cells. Only the relevant part of their genes is expressed so that they work properly in their environment. This is achieved by gene expression regulation. But regulation does not only induce cell differentiation. It is also how cells can react to their environment, determining whether they need to grow, die or produce a specific protein.

The mechanism usually works as follow:

1. A cell receives an external signal via its cell surface receptors.

2. This signal triggers a chain reaction in a complex protein network, each protein either activating or inhibiting others.

3. This signal finally activates *Transcription Factors (TFs)*, a special kind of proteins that have the ability to express or repress a particular gene according to their type.

4. The expressed gene will then result in a particular phenotype or it may produce another protein that will trigger the expression of other genes.

Cellular communication is usually divided into two main types of networks:

Signaling Pathways, representing the protein networks that handle the transmission of signals from extra-cellular receptors to the TFs on the surface of the nucleus.

Gene Regulatory Networks, representing the complex interactions between TFs, the genes they express or repress and the resulting phenotypes.

Both these networks are usually inferred from the analysis of gene sequences and validated via experimentations [7]. The complexity of the interactions in both networks has resulted in the separated study of signaling pathways and gene regulatory networks. But only the combination of both can fully explain, which and how extra-cellular signals can impact a specific gene expression [8, 9]. An example of such signaling mechanism is illustrated in Figure 2.3

2.1.5 The cell cycle

The cell cycle is the succession of stages through which a cell passes from one cell division to the next. It consists of five phases named G_0, G_1, S, G_2 and M (Figure 2.4).

The G_0 phase is a resting phase, usually called quiescent or senescent state[2]. In this phase, the cell is just in stand by, until a signal forces it to enter G_1 and start a new cycle, or maybe differentiate [3].

The G_1 phase is the first growth phase. During this phase, cellular content, excluding the chromosomes, is duplicated. At the end, a first checkpoint can prevent the cell from entering the S phase, for example if DNA damage has been detected, thus blocking until DNA is repaired or if the cell does not have enough nutrients to complete the cycle, in which case it will also block until these nutrient levels are high enough [3].

The S phase (synthesis phase) is the phase during which the DNA is replicated. During this phase, a second checkpoint ensures that the DNA has been properly replicated [3].

The G_2 phase is the second growth phase. The cell gets ready to enter the M phase and a third checkpoint ensures that the S phase was properly completed [3].

The M phase contains the **mitosis** and the **cytokinesis**. The mitosis is itself divided in the following four sub-phases:

1. The **prophase**, during which the chromosomes condense so they become thicker [3].
2. The **metaphase**, during which they are aligned along the cell central axis and the nuclear membrane disappears [3].
3. The **anaphase**, where the two halves of each chromosome (the chromatides) are split and pulled apart to the two opposite poles of the cell. During this phase, a last checkpoint blocks the progression if all the chromosomes are not properly split and divided across the cell [3].

[2]While quiescence and senescence are sometimes used indistinctly in the literature, they are in fact two different cell states, the main difference being that senescence is irreversible while quiescence is not [10, 11].

4. The **telophase**, where the new chromosomes de-condense and a new nuclear membrane forms around each set [3].

Finally, when the cell enters **cytokinesis**, it is actually divided in two daughter cells that will either enter the G_0 state or start a new growth and division cycle themselves [3].

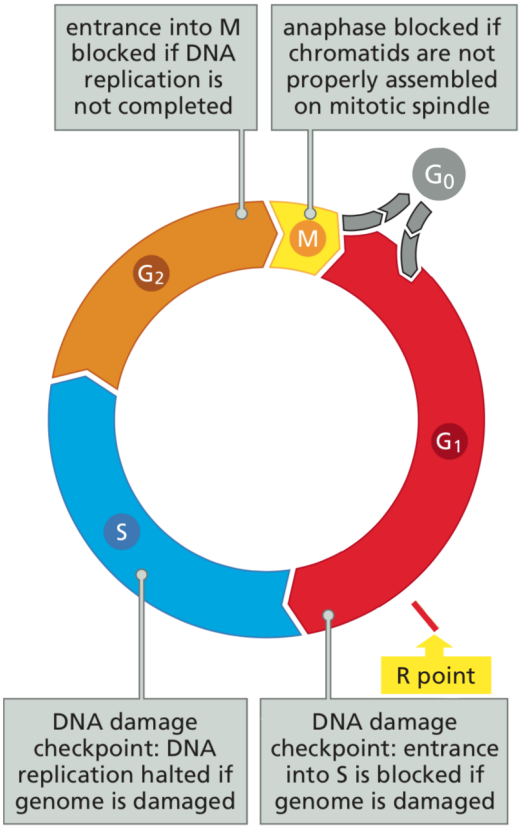

Figure 2.4: The cell active cycle starts at G_1, the first growth phase, through which the cell will advance based on external signals until it reaches the restriction (R) point. After this point, the cycle will continue solely based on internal signals. If it passes the first checkpoint, the cell will continue its way through the S phase during which another DNA damage checkpoint will occur. Once that checkpoint passed, it will enter the second growth phase G_2, at the end of which a third checkpoint will test its ability to enter the M phase. During this last phase, once the fourth checkpoint is passed, the cell will go through mitosis and divide. After the division, the daughter cells will either enter the inactive state G_0 or start a new cycle [3].

2.1.6 The cell cycle clock

The cell cycle clock is a network of proteins that receives multiple signals, originating both from inside and outside the cell, computes them and regulates whether the cell will enter the active cell cycle phases or become quiescent (Figure 2.5).It also provides the necessary information for the cell to pass the multiple checkpoints in the active cell cycle and complete its growth and division. If the cell is in G_0, the cell cycle clock will also determines if it will differentiate or not[3] [3]). Note that, while the quiescent state is reversible, some cells leave the active cell cycle irreversibly, giving up all chance to go back to the G_1 phase and start a new active cycle. This state is then called post-mitotic [3].

[3]Cellular differentiation is the process during which unspecialized cells specialize, changing their genes expression to endorse different roles.

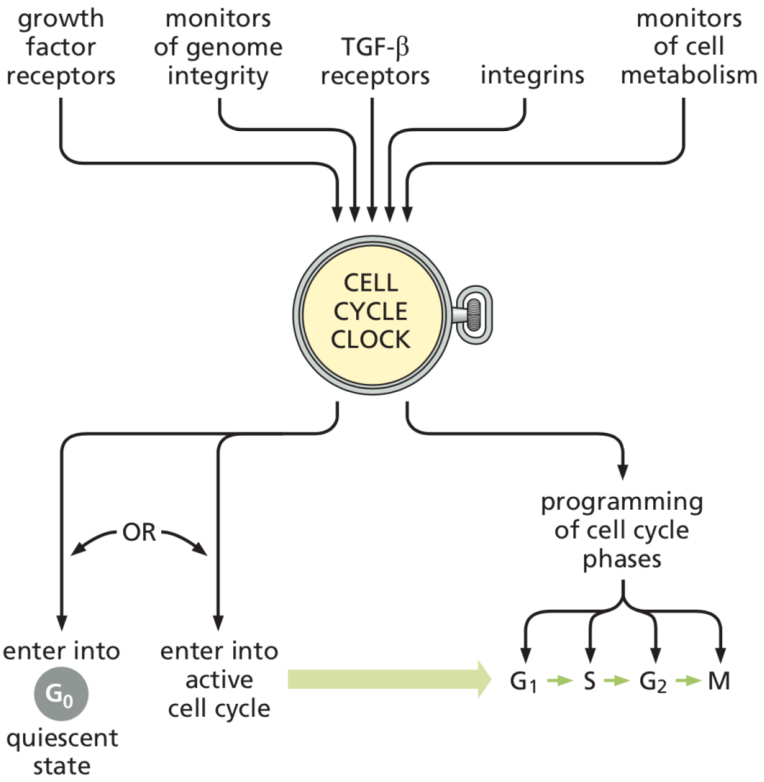

Figure 2.5: The Cell Cycle Clock. A network of proteins regulates cells behavior. Based on intra- and extra-cellular signals, it forces the cell to go in a quiescent state or to enter the active cell cycle. It also controls the progression of the cell throughout this cycle [3].

The only part where the cell cycle clock is influenced by external signals is from the start of G_1 until nearly the end of G_1. During this time, the clock is sensible to growth and anti-growth signals, thus allowing it to either continue to the S phase or enter a quiescent state. After this point, called restriction (R) point (see Figure 2.4), the cell is committed to complete the rest of the cycle autonomously. The clock will only keep working based on intracellular signals to pass all the breakpoints between the end of G_1 and the cytokinesis. If the cell is not able to go through the R-point, it will either remain in G_1 or go back to G_0 [3].

2.2 The hallmarks of cancer

In 2000, in an article entitled "The Hallmarks of Cancer" [5], Douglas Hanahan and Robert A. Weiberg first introduced a set of rules that, according to them, governs the transformation of any human cell into a malignant tumor. Their theory is that the genotype of every cancer cell is the result of six alterations: **self-sufficiency in growth signals**, **insensitivity to growth-inhibitory signals**, **evasion of apoptosis**, **limitless replicative potential**, **sustained angiogenesis**, and **tissue invasion and metastasis** (Figure 2.6).

Roughly ten years later, Hanahan and Weinberg published an updated version of their famous review entitled "The Hallmarks of Cancer: Next Generation" [11]. In this article, they explain how the six hallmarks described a decade ago are now confirmed, adding nonetheless new details and new discoveries about them in the past decade. They also introduce two new emerging hallmarks: **evasion of immune destruction** and **cellular energetics deregulation**; and two enabling capabilities: **genome instability and mutation** and **tumor-promoting inflam-**

mation (Figure 2.10).

This section explains each of these hallmarks and enabling capabilities, how they behave compared to normal cells, how they can be acquired by cancer cells and, when relevant, how they are related to signaling pathways and gene regulatory networks.

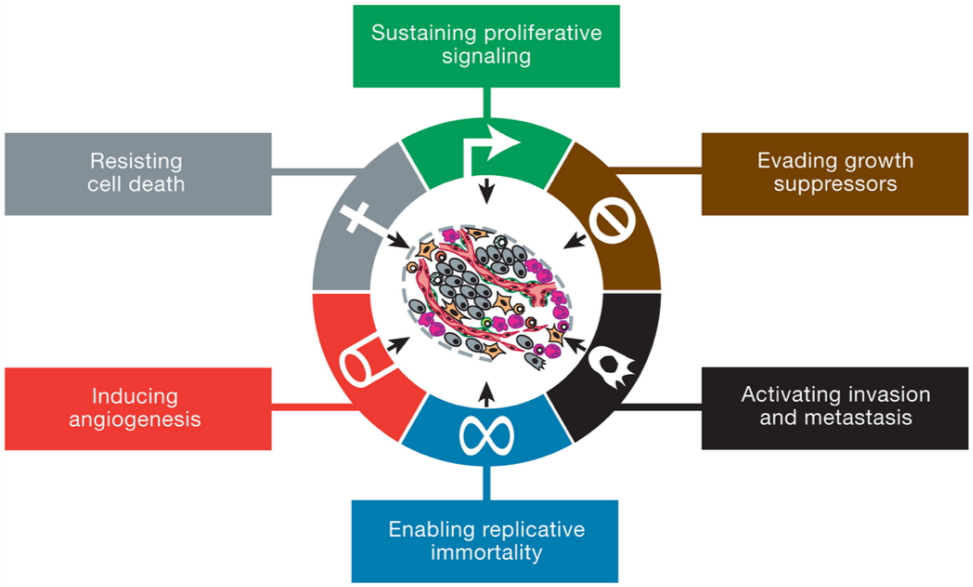

Figure 2.6: The six basic hallmarks of cancer. In 2000, Douglas Hanahan and Robert A. Weiberg the first hallmarks of cancer, a set of rules governing the transformation of any human cell into a malignant tumor. [5]

2.2.1 Self-sufficiency in growth signals

To be able to start proliferating, normal cells require external signals. These signals, materialized by growth factors, are usually generated by other cells and transmitted into the cell via ECM receptors. But somehow, cancerous cells acquire the ability to mimic these signals in one way or another [5]. There are four simple strategies for achieving this autonomy:

Alteration of extracellular growth signals. As stated previously, most growth factors are produced by one cell type to stimulate the proliferation of another. But many cancer cells acquire the ability to produce their own growth factors, thus creating a positive feedback signaling loop [5].

Alteration of the transcellular transducers of these signals. This can be achieved in two ways. Either cancer cells can change the ECM receptors they express to favor progrowth signals. Or, receptors overexpression can make them become reactive to lower levels of growth factors than usual to trigger proliferation [5].

Alteration of the intracellular circuits that translate those signals into action. The alteration of signaling pathways within cancer cells can cause other receptors information to be misinterpreted as growth signals [5].

Stimulation of normal cells within the stroma. Research also determined that cancer cells can stimulate normal cells within the tumor-associated stroma to supply them with various growth factors [11] [12].

No matter which one of these 4 strategies is used by a given tumor, they all imply changes in cell signaling and genes expression regulation.

2.2.2 Insensitivity to anti-growth signals

In normal tissues, multiple growth-inhibitory signals can be transmitted to a cell so that the cell proliferation cycle is stopped. These signals are transmitted via pathways that interact with the cell cycle clock, during the G1 phase of the cycle when cells monitor their external environment to regulate their progression toward growth, quiescence, or to a postmitotic state [5]. In order to thrive, cancer cells must evade these signals to be able to keep growing.

Many antiproliferative signals rely on the actions of tumor suppressor genes. The two main tumor suppressors encode the RB (retinoblastoma-associated) and P53 proteins. These proteins play a major role in two cellular regulatory pathways that can trigger cells proliferation or activate senescence and apoptotic (see Section 2.2.3) programs [11].

Disruptions in the RB and P53 pathways can then render cells insensitive to antigrowth signals, blocking the progression of the G1 phase of the cell cycle and preventing the P53 protein to trigger apoptosis, thus allowing the cells to multiply endlessly (see Figure 2.7) [5, 11].

2.2.3 Evading cell death

Normal cells are subject to *apoptosis*. Basically, it's a programmed cell death that can be triggered by a variety of signals. It causes the cell to be progressively disassembled and consumed by its neighbors and phagocyte cells [5, 11]. The cell apoptosis actors can be divided in two classes: the sensors and the effectors. The sensors monitor the extra and intracellular environment for conditions that influence the cell fate, generating signals accordingly. Extracellular receptors bind survival or death factors while intracellular sensors monitor the cell's well-being by detecting abnormalities like DNA damage, signaling imbalance, survival factor insufficiency, hypoxia...These signals then regulate the second class of components, potentially triggering the cell apoptotic death [5].

In normal cells, two of the most common apoptosis-inducing stresses are elevated levels of oncogene signaling and DNA damage associated with hyper-proliferation (see Section 2.2.4) [11]. Resistance to apoptosis can then be acquired by cancer through multiple strategies:

Loss of proteins functions. Most common is the loss of P53 tumor suppressor function, eliminating this critical damage-sensor from the apoptosis pathways [11].

Changes in signals expression. By increasing expression of anti-apoptotic regulators or survival signals, or by down-regulating pro-apoptotic factors, cancer cells can acquire the ability to evade apoptosis.

Alteration of circuitry. Cell death evasion can also simply result from a "short-circuit" in cell death related signaling pathways.

Also, while apoptosis and the way cancer cells acquire the ability to avoid it were already well understood in the early 2000s, new conceptual advances involving other forms of cell death have been discovered [11].

Autophagy. Like apoptosis, autophagy is a reaction that can be induced in certain states of cellular stress, the most obvious being nutrient deficiency usually experienced by cancer cells. It enables cells to break down some of its components (such as ribosomes and mitochondria) and recycle them so they can be used for energy metabolism. Like apoptosis, autophagy relies on regulatory and effectors components. This is then another barrier to break down for cancer cells to proliferate [11].

Necrosis. Unlike during apoptosis or autophagy, necrotic cells become bloated and explode, releasing their contents in their environment. Amongst other things, they also release pro-inflammatory signals, giving them the ability to alert and attract inflammatory cells. The

function of these cells is to detect tissue damage and remove associated necrotic debris. But evidence suggests that inflammatory cells can foster angiogenesis, cancer proliferation, and invasiveness (see Section 2.2.8). Additionally, necrotic cells can release bioactive regulatory factors that stimulate surrounding viable cells to proliferate [11]. As cell death by necrosis is clearly under genetic control in some circumstances, cancer cells may gain advantage in tolerating some degree of necrotic cell death. It would allow them to attract inflammatory cells that bring growth factors to the surviving neighbor cells [11].

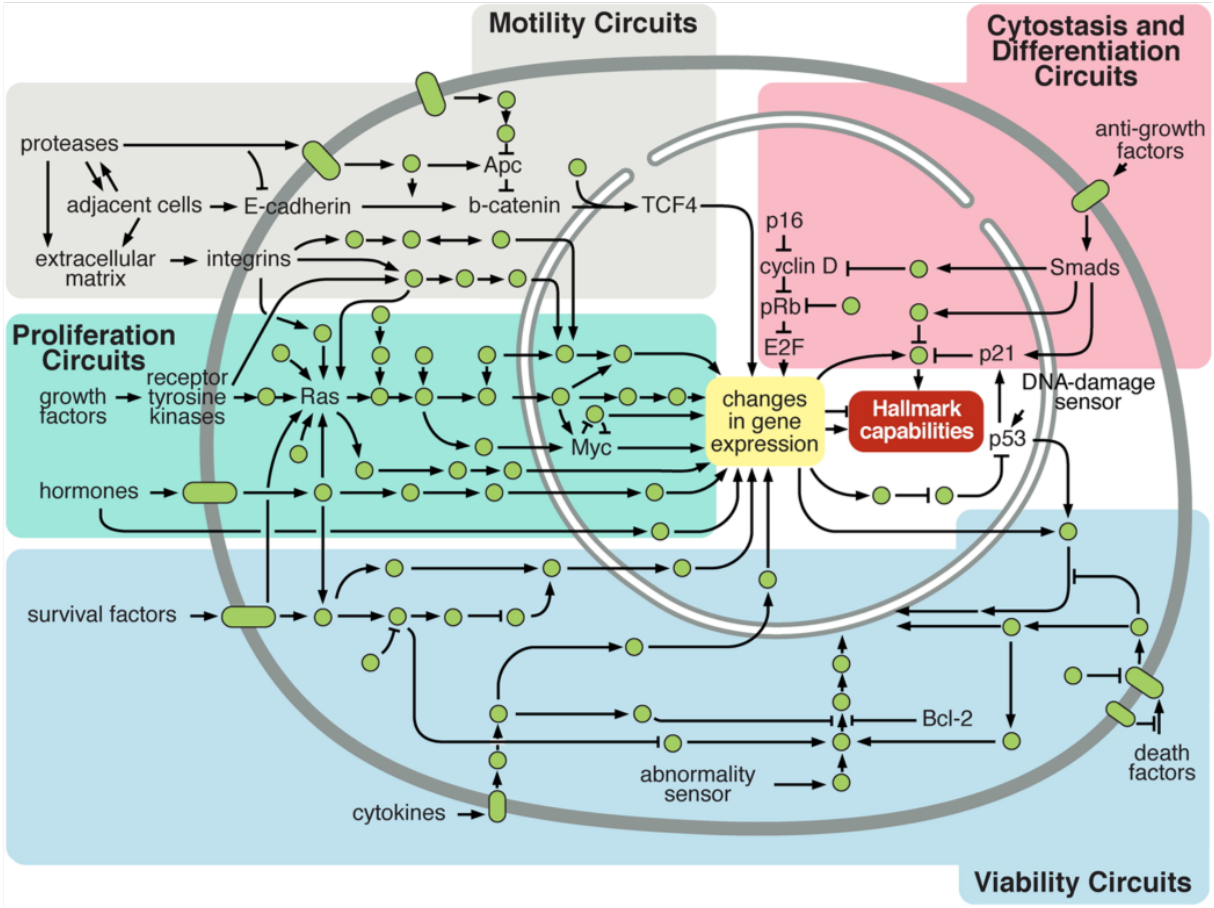

Figure 2.7: Intracellular signaling networks regulate the operations of cancer cells. They can be seen as an elaborate integrated circuit that has been reprogrammed within cancer cells to regulate the hallmark capabilities. Separate sub-circuits, here represented in different color fields, are specialized to regulate each capability. This is a simplistic view because there is considerable crosstalk between these sub-circuits, and also because these sub-circuits are all responsive to signals emitted by other cells and the tumor micro-environment [11].

2.2.4 Limitless replicative potential

At the ends of chromosomes, *telomeres*, composed of multiple repetitions the same six nucleotides chain (TTAGGG) [13], protect these chromosomes from end-to-end fusion that could render them unstable and threaten the viability of the cell [11]. Indeed, during DNA replication, normal cells are not able to completely duplicate the end of DNA [13]. This implies that, based on the length of its telomeric DNA, a cell can only go through a limited number of divisions before its telomeres are too eroded to play their protective role [11].

Cells propagation in cultures show that repeated cycles of cell division usually end up inducing senescence, an irreversible quiescent-like state. The few cells that manage to circumvent this

barrier end up in a crisis state resulting in their death via apoptosis[4] [5].

But in cancer cells, both these mechanisms must be evaded. They need to acquire replicative immortality. This ability is achieved by maintaining a telomeric DNA long enough to avoid triggering senescence or apoptosis. Two mechanisms allow this:

Telomerase expression upregulation. Telomerase is a specific protein whose function is to add new telomere repeat segments to the end of DNA. It is usually absent in normal cells (except for stem cells) but it has been proven to be expressed in the vast majority (around 85%) of cancer cells [5]. By continuously extending the length of telomeres, telomerase can counter natural telomeres erosion and grant cancer cells the limitless replicative potential they need.

Telomere maintenance mechanism. The other 15% of cancer cells seem to have acquired the ability to maintain their telomere length by a mechanism called ALT (Alternative Lengthening of Telomeres) [14, 5]. This mechanism is still poorly understood but it seems to involve homologous-directed DNA recombination; a mechanism through which broken DNA is repaired using another homologue piece of DNA [15].

2.2.5 Sustained angiogenesis

Like all tissues, tumors need nutrients, oxygen and the ability to evacuate metabolic wastes and carbon dioxide [11]. This requires that all cells be located within 100 μm of a capillary blood vessel [5]. In normal tissues, this is insured by a coordinated growth of vessels and parenchyma. This dependence would let presume that proliferating cells within a tissue would have the ability to encourage blood vessels growth. But studies show otherwise. Tumors initially lack this ability an need to acquire it in order to reach a larger size.

Angiogenesis, the sprouting of new vessels from existing ones, is the process that addresses this need. This process is usually active during embryogenesis and development but, once normal vasculature is in place, it becomes mostly quiescent. Only exceptions in the adult are wound healing and female reproductive cycle, two physiological processes where angiogenesis is transiently turned on. In tumor tissues though, this process is almost permanently activated, sustaining constant expansion and allowing cancer cells to form macroscopic tumors [11].

Of course, this mechanism does not only depend on cancer cells. To grow new vessels, the tumor-associated stroma needs to recruit other cell types from the body, In the same way as it occurs during wounded tissues healing. Therefore, the whole process actually relies on the same procedures implied in wound healing [3]: he release of multiple factors to express and regulate genes responsible for orchestrating new blood vessels growth and repress angiogenesis inhibitors.

2.2.6 Tissue invasion and metastasis

The vast majority of malicious cancers take place in epithelial tissues. They are called carcinomas. These tissues are composed of thin sheets of epithelial cells, atop complex layers of stroma. These two environments are separated by a specialized type of ECM called the basement membrane, composed by proteins secreted by both epithelial and stromal cells. Tumors begin on the epithelial side of the basement membrane and are considered benign as long as they remain on this side. But eventually, carcinomas manage to breach the basement membrane, starting invasion and becoming malignant. This is the first step of what is called the invasion-metastasis cascade [3].

[4]Recent experiments have shown that senescence as a result of excessive duplications can be delayed and possibly eliminated by improved cell culture conditions, leaving the cells proliferate until crisis state. This suggests that senescence might in fact not be a barrier to limitless replicative potential but only the apoptosis triggered by the crisis state [11]

This multi-step process usually happens in late stages of tumors development and involves a succession of biological changes (see Figure 2.8) [11].

1. **Local invasion**. First, benign carcinomas breach the basement membrane and start invading the nearby stroma [3].
2. **Intravasation**. Once present in the stromal side of the membrane, cancer cells access blood and lymphatic vessels and move through their walls [3].
3. **Transport**. Then, once in the vessels, cancer cells travel through blood or lymph to other areas in the body. To do so, they need to be surrounded and escorted by platelets that protect them from being teared apart by the blood flow [3].
4. **Arrest**: Due to their important size, especially if they are covered by platelets, cancer cells rapidly find themselves trapped in small blood vessels, most of the time in the lungs [3].
5. **Extravasation**. Using various techniques, cancer cells then have to find a way to go through the vessels walls again and arrive in the parenchyma of a tissue [3].
6. **Formation of micro-metastasis**. Once there, if the conditions are favorable, cancer cells will form new microscopic metastases [3].
7. **Colonization**. Finally, some of these metastases will manage to adapt to their new environment and start to grow uncontrollably until they reach a macroscopic size [3].

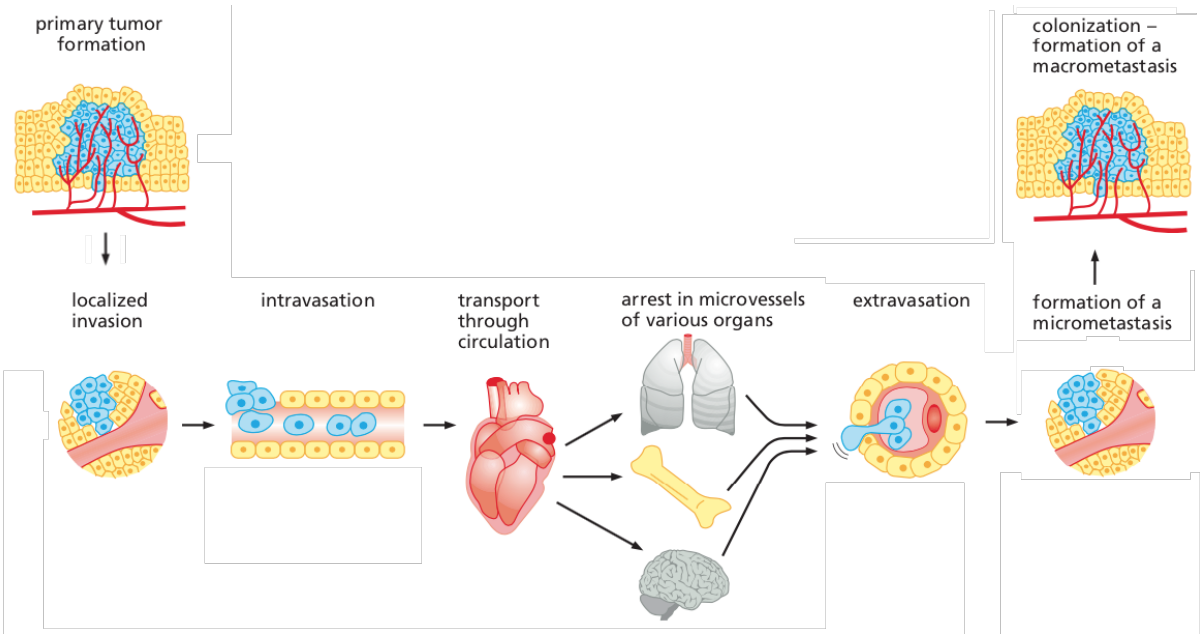

Figure 2.8: Invasion-metastasis cascade. First, the carcinoma breaches the basement membrane and invades the nearby stroma, then some of it cells access blood and lymphatic vessels (intravasation). They are then transported through thee vessels until they get stuck in micro-vessels of other organs. Finally, they get out of these vessels (extravasation) and grow to become metastasis [3].

In 2000, the mechanisms behind these processes where poorly understood. But in the last two decades, even though some behaviors still remain unexplained, research has greatly improved the understanding of this complex hallmark capability mechanisms [11]. Here is a brief explanation of some of these mechanisms:

Key roles of cell adhesion molecules. A characteristic alteration of carcinoma cells is the loss of E-cadherin, a key cell-to-cell adhesion molecule. In healthy tissues, E-cadherin helps assemble epithelial cell sheets and maintains the quiescence of the cells within these sheets [11]. Studies have shown that downregulation and, sometimes, inactivation of E-cadherin due to mutations was frequently observed in human carcinomas, supporting the theory that it is an important barrier to invasion and metastasis. Additionally, expression of other cell-to-cell and cell-to-ECM adhesion molecules has been proven altered in some highly aggressive carcinomas. Though, inversely, adhesion molecules normally associated with the cell migrations that occur during embryogenesis and inflammation, like N-cadherin for example, are often upregulated [11].

The epithelial-mesenchymal transition (EMT) program regulation. EMT is a development regulatory program. It is known to be implicated in the epithelial cells transformation giving them the ability to invade. This program is regulated by a set of transcription factors (TFs) that are also related to migratory processes during embryogenesis. Evidence indicates that the expression of these TFs in cells destined to pass through EMT is triggered by signals sent by neighboring cells [11].

Contribution of stromal cells. It becomes evident that crosstalk between cancer cells and surrounding stromal cells is essential for cancer cells to acquire invasion and metastasis capabilities. For example, some cells present in the tumor stroma have been found to secrete CCL5 (a protein also known as RANTES) in response to a signal released by cancer cells. This protein then activates invasive behavior on the cancer cells. This kind of behavior supports the theory that cancer phenotypes cannot be understood just by studying the genome of tumor cells. These observations also suggest that cancer cells, once they have invaded new tissues, might be able to revert to a non-invasive state as they do not benefit anymore from the invasion/EMT-inducing signals provided by their previous stroma [11].

Other types of invasion. Two other types of invasion, different from EMT, have been identified and implicated in the invasion-metastasis cascade. The first one, collective invasions, involves cancer cells moving as a group. The second one, less clear, is a form of invasion where cancer cells show morphological plasticity, enabling them to go through existing interstices in the basement membrane instead of clearing a path for themselves [11].

2.2.7 Genome instability and mutation

At this point, it is now clear that tumors development depends largely on a succession of mutations in the cancer cells genome. In many ways, it can be likened to Darwin's theory of evolution. Every genetic mutation has a chance to confer neoplastic cells a selective advantage enabling, step by step, their dominance in a local tissue environment [11]. But there is not one path to cancer. As illustrated in Figure 2.9, many different combinations of selective mutations enabling one of the hallmarks of cancer can lead to malignant tumor development.

Nevertheless, the common denominator of all cancers is mutation. The more genetic mutations a cell undergoes in a generation, the higher is the probability to unlock enough hallmarks to become a malignant tumor. But, usually, genome maintenance systems included in our cells, responsible of detecting and fixing deficient DNA ensures that the rate of spontaneous mutations are very low. It means that, one of the first necessary steps to give a cell a chance to become cancerous, is to undergo some specific mutations; mutations that break down its genomic maintenance and integrity monitoring systems, forcing genetically damaged cells into senescence or apoptosis, or mutations on genes responsible for intercepting mutagenic molecules before they have damaged the DNA [11]. For example, mutations affecting the TP53 gene, responsible for the expression of the protein P53.

More recently discovered, another major source of genomic instability is the loss of telomeric DNA. As stated in section 2.2.4, once the amount of telomeric DNA is insufficient to protect the chromosomes, it can lead to instabilities, amplification or deletion of some genes. This means, quite paradoxically, that telomerase can be viewed, not only as an enabler of limitless replicative potential, but also as a caretaker, maintaining genome integrity.

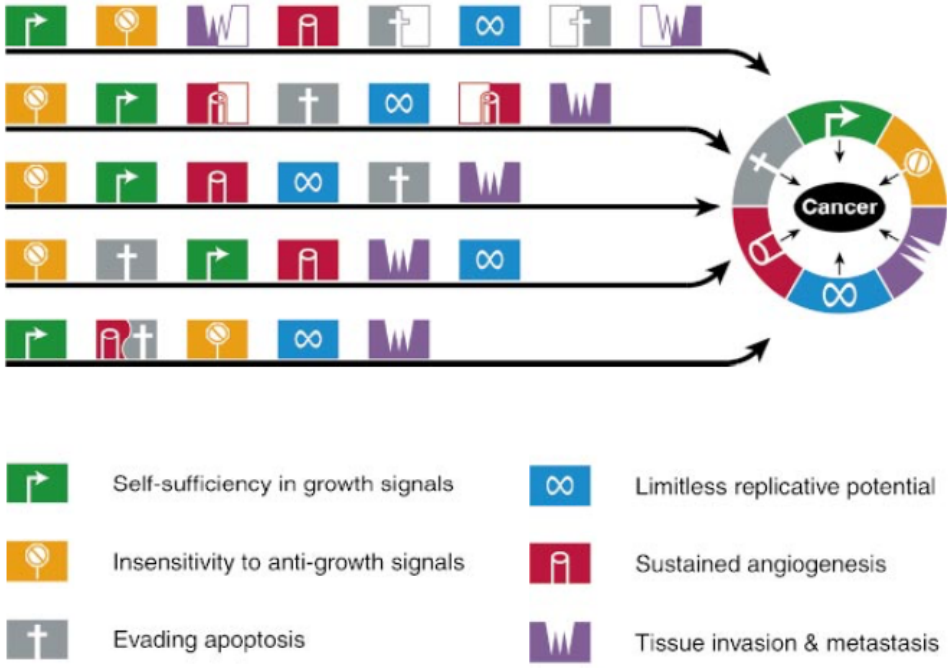

Figure 2.9: There is more than one pathway to cancer. Although some hallmarks are usually acquired before others, there is an infinity of mutations combinations that can lead a cell to become a tumor [5].

2.2.8 Tumor-promoting inflammation

It is recognized that tumors contain a lot of cells coming from the immune system, thereby triggering inflammatory conditions of various intensity depending on the type of cancer and the amount of immune cells present in the neoplastic lesion. Historically, the presence of these cells was seen as an attempt by the immune system to fight tumors, and indeed, there is more and more evidence of antitumoral response from the immune system to many types of cancers, implying that these cancers have to find a way to avoid immune detection and destruction [11].

But, paradoxically, inflammation was also proved to be necessary for the acquisition of multiple hallmarks by providing the presence of specific types of cells in the tumor stroma. These cells provide, amongst other things, growth factors to support proliferation signaling, survival factors to prevent cells death, enzymes that facilitate angiogenesis, invasion and inductive signals activating EMT [11]. This tends to classify inflammation as an enabling characteristic for the other hallmarks even though it might be triggered by the immune system trying to fight the cancer.

2.2.9 Deregulating cellular energetics

To enable cell growth and limitless division, not only do cancer cells need to have their genes expression deregulated, but they also need to adapt their energy metabolism [11]. Indeed, fueling this much activity requires more energy than usual, thus forcing the cell to exploit more sources than usual.

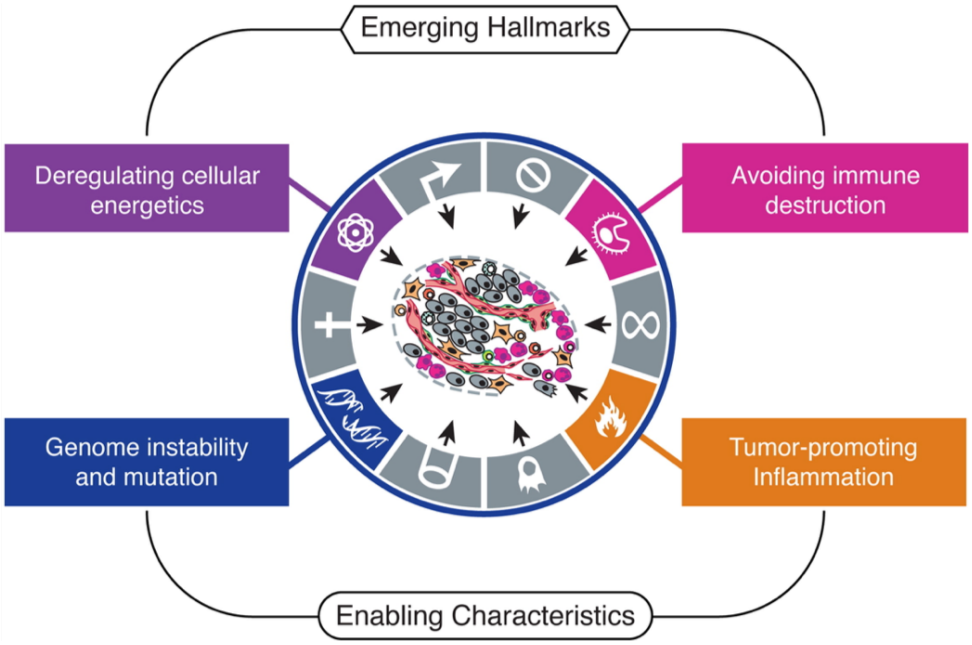

Figure 2.10: Ten years of research suggest two new emerging hallmarks: evasion of immune destruction and cellular energetics deregulation; and two enabling capabilities: genome instability and mutation and tumor-promoting inflammation [11]

In healthy tissue, under aerobic conditions, normal cells process glucose via glycolysis, producing pyruvate and a little ATP, the actual energy "currency" of the cells. Then, pyruvate is processed by the mitochondria using oxygen, producing much more ATP and releasing carbon dioxide (this is basically why humans breathe). Under anaerobic conditions, when the cell lacks oxygen, it produces much more ATP via glycolysis and very few pyruvate is sent to the mitochondria that needs oxygen to process it. In the early 1900s, Otto Warburg observed that even in the presence of oxygen, cancer cells can change their energy production metabolism to use only glycolysis. This looks fairly counterintuitive as to compensate for the lower efficiency of glycolysis for ATP production, cancer cells need to increase their glucose import by upregulating glucose transporters. [11]

Observations have shown that some tumors use multiple energy production processes, creating a perfect symbiosis. In these tumors, one part of the cells use the so-called "Warburg-effect" to create ATP via glycolysis, thus secreting lactate, while the other part imports and uses the lactate as their main energy source to create ATP via the mitochondria using oxygen [11].

As the mechanisms redirecting the energy metabolism are largely triggered by proteins involved in programming other hallmarks of cancer, it is not clear if cellular energy metabolism deregulation should be considered as another core hallmark or if it's just another side-effect of proliferation-inducing oncogenes. But, its evident importance suggests it probably is a fundamental hallmark as well [11].

2.2.10 Avoiding immune destruction

Some persisting theories state that cells are constantly monitored by the immune system and that this surveillance system is responsible for recognizing and eradicating the vast majority of cancer cells, thus preventing tumor formation. With this rational, the appearance of macroscopic tumors suggests that their cells have somehow found the ability to evade detection and/or destruction by the immune system, thus making it a potential new hallmark [11].

The immune system has the ability to react to both antigens expressed by normal tissues and those expressed by foreign elements. But it also has the ability, via various mechanisms, to develop a tolerance towards normal tissues antigens and avoid reacting to them. It means that the immune system is able to recognize and attack cancer cells, but it might be thrown off because their antigens are usually part of normal cells proteins. Although, some of these cancer cells' antigens might still trigger it, because they are usually expressed in early stages of development, expressed at smaller levels, or in parts of the body where tolerance does not develop [3].

There are also other ways for cancer cells to avoid immune destruction. Some rely on their weak antigenic nature, or the fact that maybe they were strongly antigenic but have mutated to become weakly antigenic. Some release specific factors, capable of killing immune cells that come too close to them. And others may also attract new kinds of foreign cells that can inactivate the immune cells coming to fight them [3].

To summarize, the immune response to tumors still remains imperfectly understood, and most theories on the way the immune system could be regulated to fight cancer have not been clinically experimented yet. But recent researches have unveiled a lot of evidence suggesting that the immune system plays a crucial role in preventing tumors development, at least in some forms of cancer, and the increase of certain cancers in immunocompromised individuals tend to validate that [3]. It seems that the avoidance of immune destruction should be considered a new major hallmark [11].

2.3 Targeted cancer therapy

There are currently multiple types of cancer treatments: surgery, chemotherapy, radiotherapy, immunotherapy, stem cells transplant, hormone therapy... [16] but the one that drives this work is *targeted therapy*.

This type of treatment is the result of over 30 years of research on the mechanisms of cancer. It relies on the ability to analyze a patient's cancer genetic information (tumor cells signaling pathways and gene regulatory networks), and prescribe drugs accordingly to take down one or more of the hallmark capabilities of cancer. The principle behind this approach is simple, *"if a capability is truly important for the biology of tumors, then its inhibition should impair tumors growth and progression"* [11]. As illustrated in Figure 2.11, drugs are currently developed in a way that their efficiency relies on the fact that they address one of the hallmarks[5] [11].

One of the bright sides of drugs targeting specific capabilities is that, while disabling only one capability will impair cancer progression, undesirable side effects on healthy tissues are much more limited [11] than in other kinds of treatments, like chemotherapy. Unfortunately, this is not so simple. Positive clinical responses to these kinds of treatments usually only last for a short period of time. Indeed, most patients seem to inevitably relapse [11]. One of the explanations for this phenomenon is that core hallmark capabilities are regulated by partially redundant pathways. So targeted drugs may not completely shut down a capability, allowing cancer cells to still survive with lessened capabilities until their offspring eventually adapts to these new constraints, for example, thanks to a new mutation [11].

Another form of drug resistance is the ability for cancer cells to cope for the lack of an hallmark by relying more on another one [11]. Recent treatments of human glioblastoma using anti-angiogenic therapies have seen the cancer cells increase their invasion and metastasis activity, thus gaining access to the preexisting vasculature of healthy tissues [17].

[5] Some drugs actually proceed to reinstate more than one barrier as some signaling pathways have an impact on multiple relevant phenotypes.

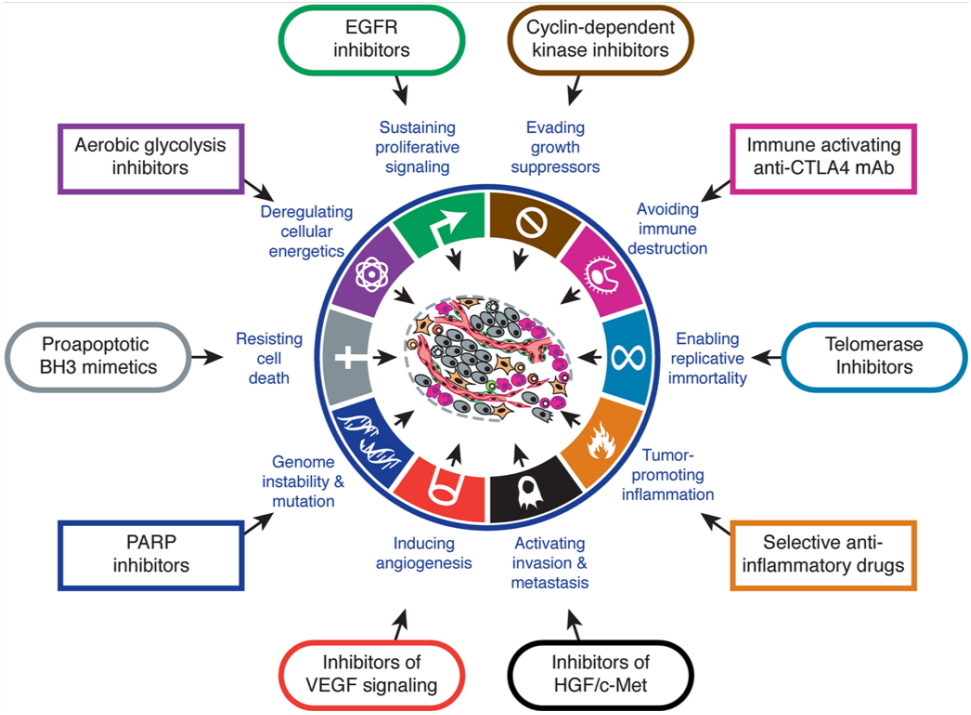

Figure 2.11: Hallmark capabilities targeted drugs. Drugs targeting each individual hallmark capability and enabling characteristic have been developed and are in clinical trial or already approved. Drugs listed here are just examples, there are actually way more candidate drugs with different molecular targets and modes of action in development [11].

Nonetheless, it does not mean targeted cancer therapy is not the way to go, it just means it must go further. There is only a limited number of signaling pathways that support a given hallmark and only so many ways cancer cells can survive without one or more of the hallmarks. Therefore, the current challenge is to integrating the available data on every single pathway and regulatory network involved in supporting hallmark capabilities in bigger models. Then, find the right combination of targeted drugs to completely dismantle tumors [11].

2.4 Further readings

While this introduction covers all the necessary basis to understand the context of this work and allow an understandable reading of this manuscript, the reader eager to get more information on the subject of cancer is invited to read Douglas Hanahan and Robert A. Weinberg's reviews *"The Hallmarks of Cancer"* [5] and *"The Hallmarks of Cancer: The Next Generation"* [11]. A lot more details regarding each concept discussed in these reviews can also be found in Robert A. Weinberg's book *"The Biology of Cancer"* [3]. These three readings are a gold mine of information on cancer research and biology and have been a great source of inspiration for the writing of this biological introduction.

Acknowledgment

Figures 2.1, 2.2, 2.3, 2.4, 2.5 and 2.8 are reprinted from [3], Copyright (2014), with permission from Garland Science, Taylor & Francis Group, LLC.
Figure 2.9 is reprinted from [5], Copyright (2000) with permission from Elsevier.
Figures 2.6, 2.7, 2.10 and 2.11 are reprinted from [11], Copyright (2011) with permission from Elsevier.

CHAPTER

3

GOAL OF THE SYSTEM

"For decades now, we have been able to predict with precision the behavior of an electronic integrated circuit in terms of its constituent parts–its interconnecting components, each responsible for acquiring, processing, and emitting signals according to a precisely defined set of rules. Two decades from now, having fully charted the wiring diagrams of every cellular signaling pathway, it will be possible to lay out the complete integrated circuit of the cell upon its current outline. We will then be able to apply the tools of mathematical modeling to explain how specific genetic lesions serve to reprogram this integrated circuit in each of the constituent cell types so as to manifest cancer. With holistic clarity of mechanism, cancer prognosis and treatment will become a rational science, unrecognizable by current practitioners. It will be possible to understand with precision how and why treatment regimens and specific antitumor drugs succeed or fail. We envision anticancer drugs targeted to each of the hallmark capabilities of cancer; some, used in appropriate combinations and in concert with sophisticated technologies to detect and identify all stages of disease progression, will be able to prevent incipient cancers from developing, while others will cure preexisting cancers, elusive goals at present. One day, we imagine that cancer biology and treatment–at present, a patchwork quilt of cell biology, genetics, histopathology, biochemistry, immunology, and pharmacology–will become a science with a conceptual structure and logical coherence that rivals that of chemistry or physics."

Douglas Hanahan and Robert A. Weinberg, 2000

Now that the biological context has been fully introduced, this chapter describes the aim of the system developed in this work. First the concept of expert system is explained, then the two most common biological model types are briefly discussed. Finally, the specific context of this work is described; it's target users, their needs and how this project envisions addressing them.

3.1 Expert Systems

In his book *"Introduction to Expert Systems"*, Peter Jackson gives the following definition: *"An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice."* [18].

In other words, an expert system is defined by the following characteristics [18]:

- It can accomplish entirely a task that requires domain-specific human expertise, or it can act as an assistant to a human decision maker. The decision maker might be an expert, the purpose of the system then being to increase its productivity.
- It simulates human reasoning.
- It performs reasoning over representations of human knowledge. The knowledge in the program being usually separated from the reasoning in two different modules. These two modules are referred to as *knowledge base* and *inference engine* respectively.
- It is able to solve problems by heuristic or approximative methods. It does not require perfect data as for algorithmic solutions, and the solution it provides may be subject to some degree of certainty.
- It is able to solve problems with a realistic complexity requiring a substantial amount of human expertise.
- It must have high performance and reliability.
- It must be able to justify its solution by providing proof to convince the user that its reasoning is correct.

3.2 Model types and visualizations

When talking about biological network computer models, two main types can be distinguished; *qualitative models*, focused on the interactions between the different entities og the network in terms of activation or inhibition; and *quantitative models*, encoding the actual chemical reactions and concentrations inside the network, as well as the way they impact the levels and concentrations of the implied reactants and products.

3.2.1 Qualitative network models

Figure 3.1 displays a qualitative model representation of the human apoptosis signaling and regulation network. As for most of these kinds of models, more than the simple connections between the different entities, it displays whether the effect of one entity over another is an activation (represented by a link ended by an arrow) or an inhibition (represented by a link ended by a bar). While these kinds of representation are already way more readable than the chemical representations, a model like the one in Figure 3.1 still contains too much information to be easily read and interpreted by a human being.

3.2.2 Quantitative network models

Figure 3.2 displays a subset (the whole thing being way bigger than the qualitative model displayed in Figure 3.1) of a quantitative model representation of the human apoptosis signaling and regulation network. The visual representation of this model displays how the product of an active protein can be a catalyst to the chemical reaction changing another protein's state. While these models are really useful to run precise simulations and find the expected concentrations and combinations of drugs that should have the best results, their representations omits these parts anyway and are of poor interest compared to a qualitative model for a visual analysis.

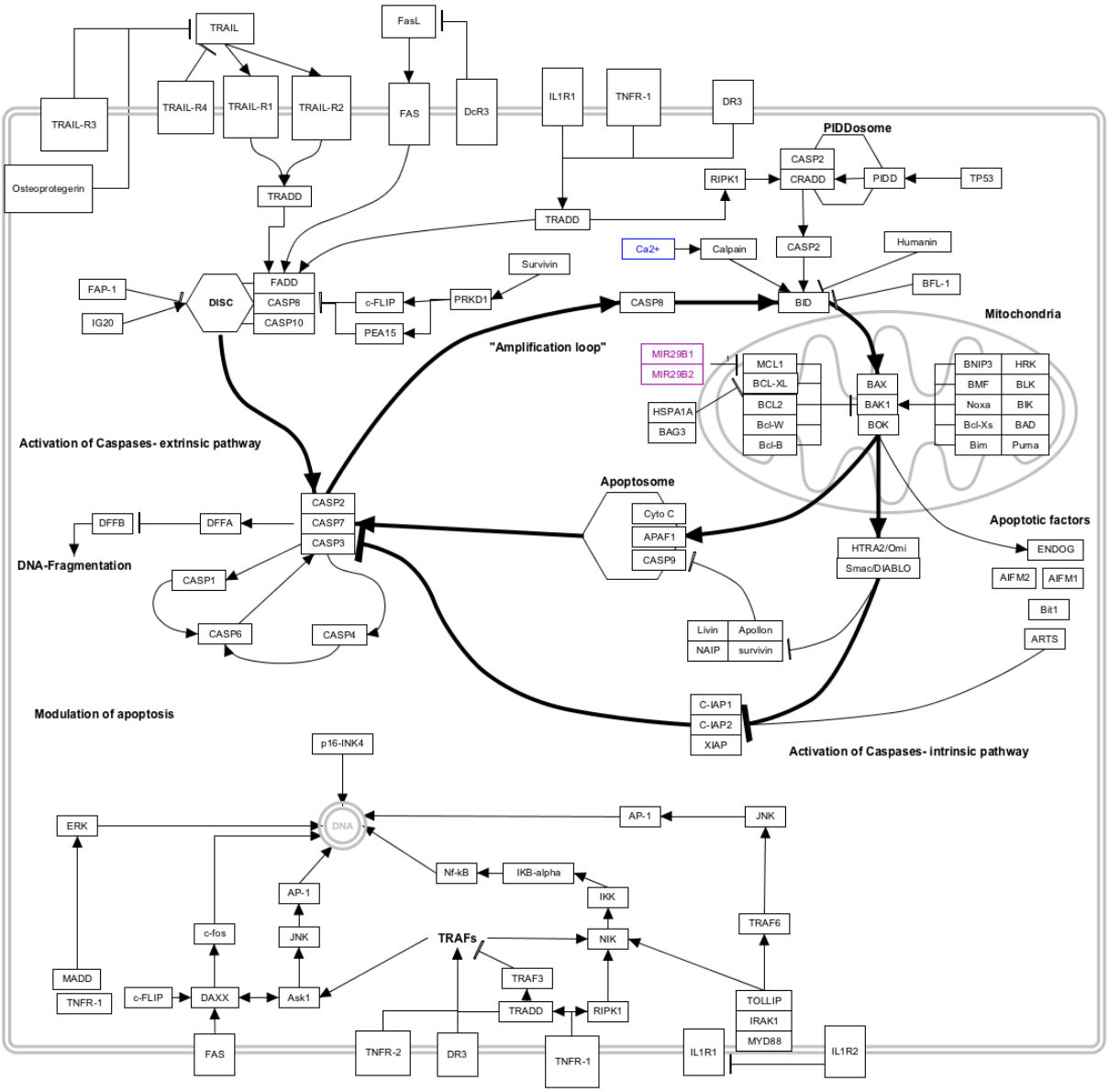

Figure 3.1: Visualization of a qualitative model of the human apoptosis signaling and regulation network. Arrows represent an activation while arcs ended by a bar represent an inhibition. Model imported from WikiPathways [19] and rendered with PathVisio (`https://www.pathvisio.org`).

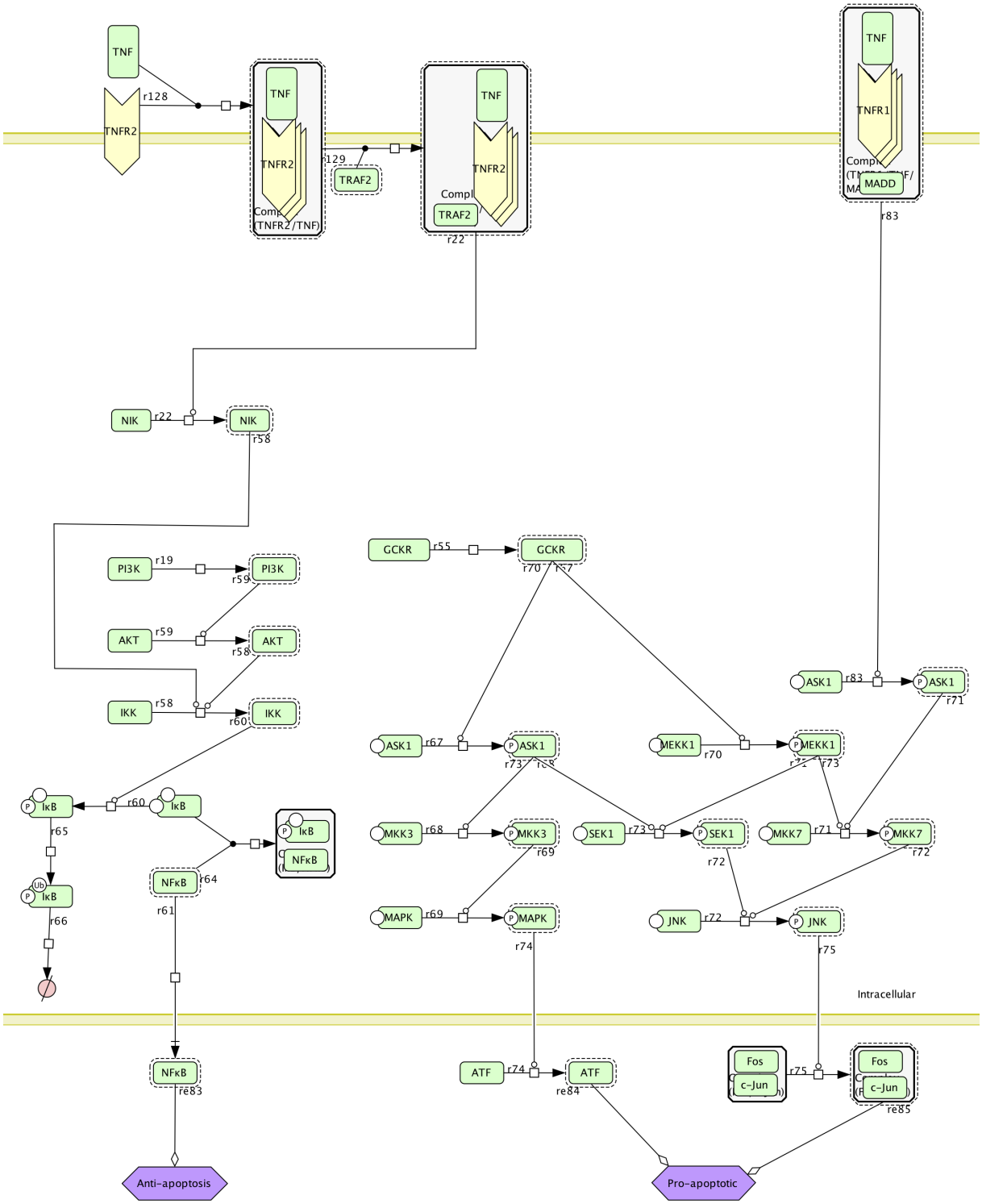

Figure 3.2: Quantitative model visualization of a subset of the human apoptosis signaling and regulation network. Model imported from the PANTHER Database [20] and rendered with CellDesigner (`http://www.celldesigner.org`).

3.3 Domain context

With the arising, in the last 20 years, of cheaper DNA sequencing methods and the arrival of cheap an powerful gene editing tools like CRISPR [2], enormous progress has been made in the study of signaling pathways and gene regulatory networks and therefore, in the evolution of targeted cancer treatment. Based on the analysis of the sequenced DNA and the inference of the associated networks, biologists are now able to provide oncologists with analyses identifying new opportunities of treatments for their patients.

The problem is, data generated by biologists is usually extremely detailed models (both quantitative and qualitative) optimized to run computer simulations and not to generate visual representations. Their actual rendering is way too big, too complex and too detailed to be easily readable and efficiently used by an oncologist (see Figures 3.2 and 3.1). In order to provide oncologists with useful, readable and understandable information, biologists need to simplify these networks representations, and enrich them with the conclusions of their analysis. The network visualizations resulting of such work should:

- focus on the main actors of the target pathway: the relevant cell surface receptors, the main proteins, known for their responsibility in the analyzed pathways and the proteins for which relevant drugs have been developed.
- display the actual phenotypes that will be expressed at the end of each pathway like apoptosis, angiogenesis, metastasis...
- hint at the kind of drugs that could be used in order to improve the patient situation and the proteins they target.
- display a qualitative relation between all elements of the network.

Figure 3.3 shows an example of such simplified qualitative pathway. It clearly displays the main proteins directly involved, the resulting phenotypes, the drugs that could potentially be useful to treat the analyzed cancer and the type of interaction between all components.

3.4 The proposal

Currently, to obtain the simplified networks discussed in Section 3.3, a lot of work is required by biologists and most of the work must be done manually. While data is available, there is no easy way to automatically simplify the input networks, remap them and add the relevant information regarding drug proposals. The goal of this work is then to address this problem by providing the bases of an expert system capable of helping biologists in this task and increase their productivity.

The solution developed in this paper focuses on two main use-cases: the simplification of qualitative model representations, and the enrichment of these models with relevant drug interactions. The resulting system should then be able to :

- provide a solution that can take in input a complete gene regulatory network in a standard format,
- simplify this network in a way that its display is as easy to read and understand as possible,
- complete this network with appropriate hints about drugs prescriptions,
- output that simplified network in an appropriate model standard or as a rendered image file.

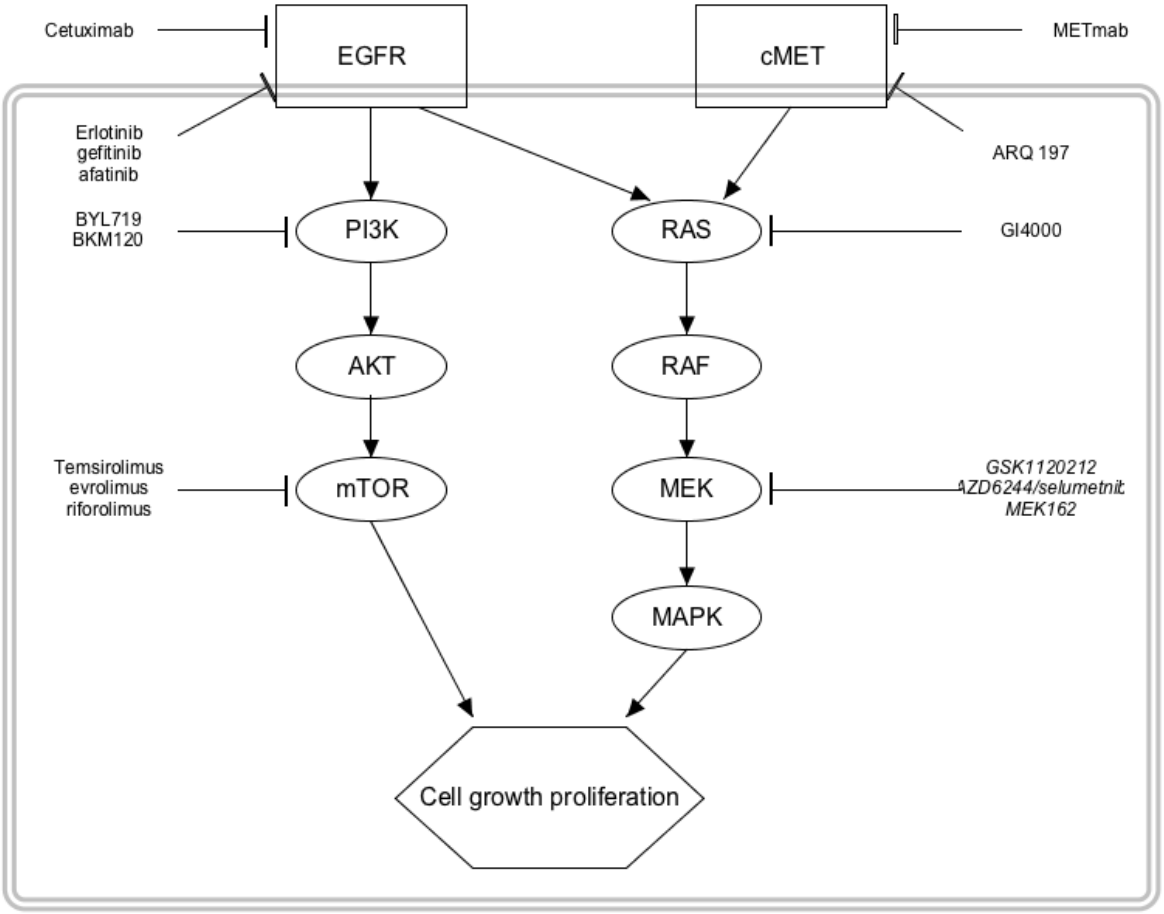

Figure 3.3: Simplified model representation of the pathways involved in lung cancer including proposed treatment options [21]. This visual representation clearly displays, for the target cancer, the signaling pathway involved in cell growth regulation, its main proteins and cell surface receptors, and a hint at some drugs that might have a positive effect on the patient.

Also, in accordance with the characteristics of an expert system described in Section 3.1, the system will:

- perform these tasks based on human reasoning logic,
- compute its reasoning based on the same input data as the biologist usually doing it,
- provide solutions as accurate as possible based on the available data,
- be able to process actual data from relevant sources,
- be able to justify its final solution by providing details on its reasoning.

CHAPTER

4

STATE OF THE ART

Up to this point, the necessary biological background and the goals of this work have been introduced. With that in mind, this chapter now gives an overview of the state of the art in terms of technologies, data availability and public software. First, a few of the main standards for data modeling are analyzed and compared. Then, the various public model databases are overseen, and finally, the functionalities of some specific softwares relevant to this project and their interactions are discussed.

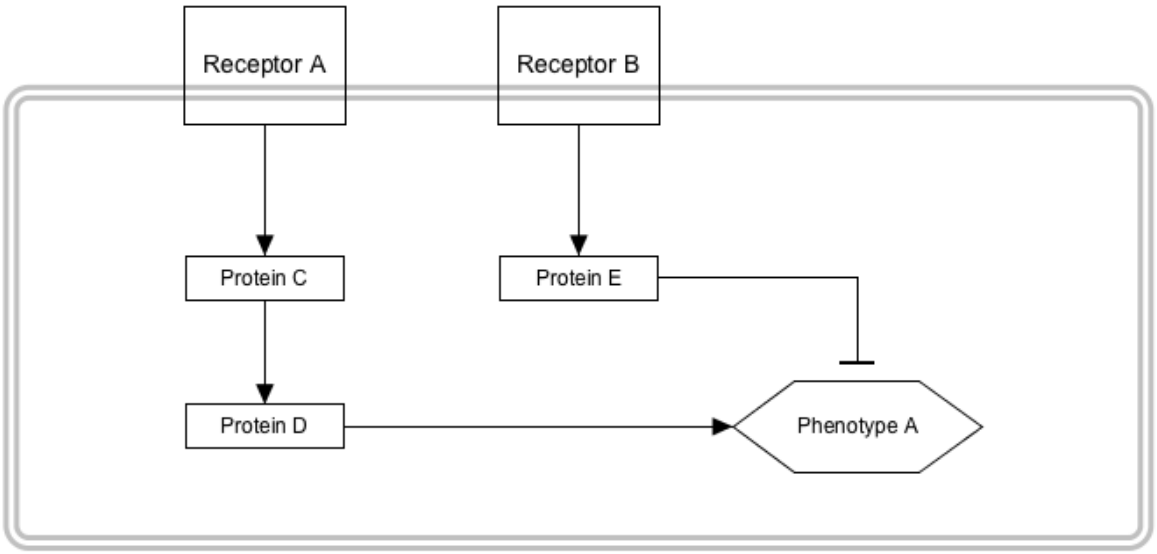

Figure 4.1: Sample signaling pathway. Receptor A stimulation activates protein C, which activates protein D, activating the phenotype A. Receptor B stimulation activates protein E which inhibits the phenotype A

4.1 The standards

There are a lot of standards for modeling biological networks. They were all created in a effort to unify the way data is encoded and shared between softwares and organizations. But with time passing by, only a handful of them steps out, all with a slightly different approach in mind. They are supported by most softwares and databases and keep evolving to extend their capabilities and allow new uses cases. Here under is a quick description and comparison of these biggest players in the field. Also, in an effort to compare the most relevant ones, their descriptions display their actual encoding of the sample signaling pathway presented in Figure 4.1.

4.1.1 The SBML standard

The **System Biology Markup Language (SBML)** [22] is a free and open XML-based format for the computer modeling of biological processes. It can be used to encode almost any biological network, including models of metabolisms, cell signaling and gene regulatory networks. The main purpose of SBML is the quantitative modeling of systems consisting in basic dynamic biochemical reaction networks with a focus on the analysis and simulation of such networks [23, 24].

The basic structure of an SBML model is pretty simple. It is composed of a list of *species*, located in one or more *comportments*, and a list of *reactions* describing all transformation, transport or binding process that can change the amount of one or more species in a compartment. But this implies that the basic qualitative model example shown in Figure 4.1 cannot be encoded as is in SBML. It has to be encoded as a succession of chemical reactions as represented in Figure 4.2. The SBML encoding of this model would then look approximately like in Figure 4.3.

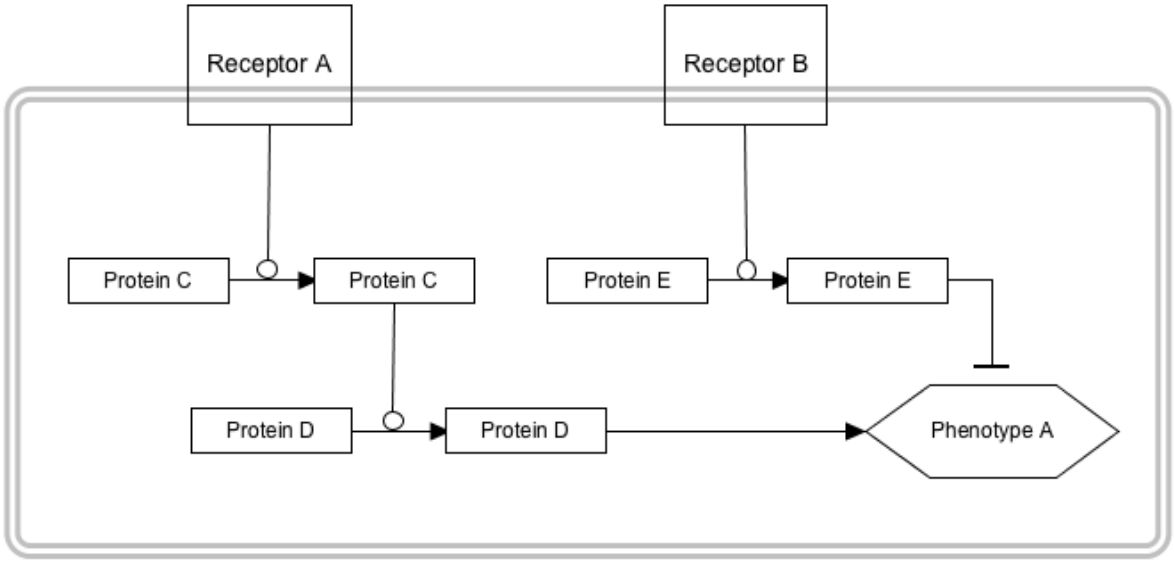

Figure 4.2: Visual representation of the SBML Core encoding of the sample network. Receptor A acts as a catalyst for the reaction changing the state of Protein C, which itself becomes a catalyst for the state change of Protein D, thus inducing Phenotype A. While receptor B is a catalyst for Protein E state change, thus inhibiting Phenotype A.

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2" version="4">
  <model metaid="SampleModel" id="SampleModel">
    <listOfCompartments>
      <compartment metaid="default" id="default" size="1" units="volume"/>
    </listOfCompartments>
    <listOfSpecies>
      <species metaid="s1" id="s1" name="Receptor A" compartment="default"
          initialAmount="0"/>
      <species metaid="s3" id="s3" name="Protein C" compartment="default" initialAmount
          ="0"/>
      <species metaid="s4" id="s4" name="Protein D" compartment="default" initialAmount
          ="0"/>
      <species metaid="s9" id="s9" name="Phenotype 1" compartment="default"
          initialAmount="0"/>
            [...]
    </listOfSpecies>
    <listOfReactions>
      <reaction metaid="re7" id="re7" reversible="false">
        <listOfReactants>
          <speciesReference metaid="CDMT00001" species="s3"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference metaid="CDMT00002" species="s3"/>
        </listOfProducts>
        <listOfModifiers>
          <modifierSpeciesReference metaid="CDMT00003" species="s1"/>
        </listOfModifiers>
      </reaction>
      [...]
    </listOfReactions>
  </model>
</sbml>
```

Figure 4.3: SBML Core encoding of the sample network. *species* objects encode the various network nodes while *reaction* objects encode the reactions between those species. In this example, reaction re7 encodes the reaction changing Protein C state, catalyzed by Receptor A.

Nonetheless, in order to extend its capabilities, the current level of SBML includes 14 extension packages. Here is a quick description of three of them relevant to this work:

Qualitative Models (Qual). An extension to support qualitative models. That is, models wherein species do not represent quantity of matter, and processes are not reactions [25]. Qual models structure is way simpler than the SBML Core structure. It contains a list of *species* that can represent anything, and a list of *transitions*, composed of inputs, outputs, and the necessary information to know when a transition must fire, if it's positive or negative (activation or inhibition), and the related species level variation. The main idea behind this extension is to provide the ability to encode logical regulatory networks (boolean or multi-valued) and standard Petri nets (see Chapter 6). With the use of this extension, the sample model presented in Figure 4.1 can now be encoded as is and its SBML encoding would look like Figure 4.4.

Layout. An extension to support the storage of spacial information regarding the network diagram [26].

Rendering. Associated to the Layout extension, it supports the storage of graphical symbols and glyphs used to render a model's diagram [27].

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level3/version1/core" xmlns:qual="http://www.
    sbml.org/sbml/level3/version1/qual/version1" level="3" version="1" qual:required="
    true">
  <model metaid="SampleModel" id="SampleModel">
    <listOfCompartments>
      <compartment metaid="default" id="default"/>
    </listOfCompartments>
    <qual:listOfQualitativeSpecies>
      <qual:qualitativeSpecies qual:id="s1" qual:compartment="default" qual:constant="
          false" qual:name="Receptor A"/>
      <qual:qualitativeSpecies qual:id="s3" qual:compartment="default" qual:constant="
          false" qual:name="Protein C"/>
      <qual:qualitativeSpecies qual:id="s4" qual:compartment="default" qual:constant="
          false" qual:name="Protein D"/>
      <qual:qualitativeSpecies qual:id="s6" qual:compartment="default" qual:constant="
          false" qual:name="Phenotype 1"/>
      [...]
    </qual:listOfQualitativeSpecies>
    <qual:listOfTransitions>
      <qual:transition qual:id="tr_1">
        <qual:listOfInputs>
          <qual:input qual:id="in_1" qual:qualitativeSpecies="s1" qual:transitionEffect
              ="none" qual:sign="positive"/>
        </qual:listOfInputs>
        <qual:listOfOutputs>
          <qual:output qual:id="out_1" qual:qualitativeSpecies="s3" qual:
              transitionEffect="production" qual:outputLevel="1"/>
        </qual:listOfOutputs>
        <qual:listOfFunctionTerms>
          <qual:defaultTerm qual:resultLevel="1"/>
        </qual:listOfFunctionTerms>
      </qual:transition>
      [...]
    </qual:listOfTransitions>
  </model>
</sbml>
```

Figure 4.4: SBML Qual encoding of the sample network. *qual:qualitativeSpecies* objects encode the nodes of the network while *qual:transition* objects encode the transitions between those nodes. Here, transition tr_1 represents the activation of Protein C by Receptor A.

Another interesting characteristic of SBML is the support of *annotations*. This mechanisms allows applications to enrich their SBML models with additional descriptive data about parts of the model. These annotations usually contain information about the nature of a protein or a reaction and links to additional information but some applications, like CellDesigner (see Section 4.3), also use them to store software specific data.

The SBML standard is supported by most of the popular biological networks databases including PANTHER, BioModels and Reactome (see Section 4.2) and is also supported by more than 290 softwares including CellDesigner, Cytoscape, PathVisio and all the softwares included in the SBW suite (see Section 4.3.

4.1.2 The BioPax standard

Biological Pathway Exchange (BioPAX) is a standard language that aims to enable integration, exchange, visualization and analysis of biological pathway data [28]. It can be used to represent signaling pathways, molecular and genetic interactions and gene regulation networks. But unlike SBML, its main focus is on modeling qualitative networks. It does not contain mathematical relations but provides more details about the relations between the different entities [23]. Its is defined in OWL DL, a Web Ontology Language sub-language, and encoded using XML and RDF.

Basically, BioPAX models are structured in a way that everything is an entity, and entities are of three main types [24]:

- **Physical entities.** Than can be genes, proteins, molecules...
- **Interactions.** That represent the various interactions between entities.
- **Pathways.** That represents a set of interactions.

Therefore, the sample network described in Figure 4.1 can be encoded as is in BioPAX and its OWL encoding would look like Figure 4.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:bp="http://www.biopax.org/release/biopax-level3.owl#">
  <owl:Ontology rdf:about="">
    <owl:imports rdf:resource="http://www.biopax.org/release/biopax-level3.owl#" />
  </owl:Ontology>
  <bp:Pathway rdf:about="id1">
    <bp:pathwayComponent rdf:resource="id8" />
    [...]
    <bp:displayName rdf:datatype="string">SamplePathway</bp:displayName>
  </bp:Pathway>
  <bp:Protein rdf:about="adbc2">
    <bp:displayName rdf:datatype="string">Protein C</bp:displayName>
    <bp:entityReference rdf:resource="id3" />
  </bp:Protein>
  <bp:Complex rdf:about="dgfh5">
    <bp:displayName rdf:datatype="string">Receptor A</bp:displayName>
    <bp:entityReference rdf:resource="id2" />
  </bp:Complex>
  [...]
  <bp:BiochemicalReaction rdf:about="id8">
    <bp:right rdf:resource="adbc2" />
    <bp:left rdf:resource="dgfh5" />
  </bp:BiochemicalReaction>
  [...]
</rdf:RDF>
```

Figure 4.5: BioPAX (OWL) encoding of the sample network. The *bp:Protein* object encodes Protein C while the *bp:Complex* object encodes Receptor A and the *bp:BiochemicalReaction* object represents the chemical reaction existing between these two entities.

Like SBML, Biopax is supported by most of the popular databases, including BioModels and Reactome (see Section 4.2), but the range of supporting softwares seems to be smaller. Nonetheless, it is supported by Cytoscape and PathVisio amongst others (see Section 4.3).

4.1.3 The SBGN standard

The **Systems Biology Graphical Notation (SBGN)** is a project whose goal is to standardize the graphical notations used in visual representations of biological process models [29]. It defines three visual languages:

The Process Description language (PD), whose goal is to specify the temporal course of biochemical interaction in a network. It would be the right language to represent SBML Core models for example.

The Entity Relationship language (ER), that allows the representation of the relationships between a model entities, regardless of the temporal aspects.

The Activity Flow language (AF), used to represent the flow of information between biochemical entities in a network. It would be more suited for the map representation of SBML Qual models.

SBGN files are encoded using the SBGN-ML XML-based format. Compared to other languages like SBML, SBGN-ML doesn't describe species, transitions or reactions anymore. It focuses on their graphical representation by defining a set of glyphs and arcs (although these arcs and glyphs still have a biologically-related class that defines their style). Figure 4.6 shows the SBGN-ML code of an SBGN AF representation of the sample model described in Figure 4.1.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<sbgn xmlns="http://sbgn.org/libsbgn/0.2">
    <map language="process description">
        <glyph class="macromolecule" id="e7019">
            <label text="Protein E"/>
            <bbox w="90.0" h="25.0" x="318.41858" y="177.11253"/>
        </glyph>
        <glyph class="phenotype" id="d2d4c">
            <label text="Phenotype A"/>
            <bbox w="142.2972" h="59.856968" x="451.27246" y="244.34749"/>
        </glyph>
        <glyph class="submap" id="f6ede">
            <label text="Receptor B"/>
            <bbox w="91.18426" h="66.45219" x="317.82648" y="35.7737"/>
        </glyph>
        [...]
        <arc class="production" id="idaf6e8dac" source="f6ede" target="e7019">
            <start x="363.41858" y="102.22588"/>
            <end x="363.41858" y="177.11253"/>
        </arc>
        <arc class="inhibition" id="id49da8430" source="e7019" target="d2d4c">
            <start x="408.41858" y="189.61253"/>
            <next x="522.4211" y="189.61253"/>
            <end x="522.4211" y="244.34749"/>
        </arc>
        [...]
    </map>
</sbgn>
```

Figure 4.6: SBGN-ML encoding of the sample network. In this format, no more species and reactions but *glyphs* and *arcs* defining the visual representation of the network.

The SBGN standard is widely supported with more than 30 applications, including CellDesigner, PathVisio and the SBW Layout viewer (see Section 4.3). SBGN exports are also provided by more than 12 databases, among which the PANTHER database, BioModels and Reactome (see Section 4.2).

4.1.4 The CSML standard

The **Cell System Markup Language (CSML)**[1] is an XML-based file format for visualizing, modeling and simulating biological pathways. Version 3 of CSML was introduced with the main focus of supporting Hybrid Functional Petri net based visualization and simulation (see Chapter 6). At the time of its creation, it covered functionality lacking from other big standards as SBML, like the absence of graphical elements. But with the evolution of SBML and BioPAX, with newer levels and extensions addressing these gaps, and the arrival of SBGN, advantages of CSML are not so clear anymore. It now looks like it has failed to make its way as a global standard and doesn't seem to be developed anymore. Though it is still used by a commercial application called Cell Illustrator that allows intuitive modeling, visualization and simulation of biological pathways[2].

4.1.5 COMBINE and the COMBINE Archive

The **COmputational Modeling in BIology NEtwork (COMBINE)** is an initiative to coordinate the development of various community standards for computational modeling. Its goal is to foster the development of inter-operable and non-overlapping standards covering all aspects of modeling biology. Standards related to COMBINE include, among others, BioPAX, SBGN and SBML [30].

COMBINE is also at the initiative of **the COMBINE archive**, a project whose goal is to create a single file, containing all documents necessary for the description of a model and its associated data and procedures. This includes all models needed to run simulations, associated data files, experiments descriptions and results, and every other relevant data for the study of a system. The archive is encoded using the Open Modeling EXchange format (OMEX) [31].

4.2 Databases

There are a few public databases providing biological network models in a variety of fields. This section presents some of the most relevant ones in the field of cancer research.

4.2.1 BioModels

The **BioModels database**[3] is a public repository hosting models of biological systems. Their main purpose is to provide reproducible, high-quality, free of use models published in scientific literature. Hosted models cover various processes. They are usually described in peer-reviewed scientific literature and some of them are automatically generated from other pathway resources like KEGG. Models are manually curated and enriched with references to relevant data [32]. Aside from standard image formats, all models can be downloaded in SBML, as well as BioPAX level 2 and 3, although those are automatically generated and can lack some information.

4.2.2 KEGG

The **Kyoto Encyclopedia of Genes and Genomes (KEGG)**[4] is a database resource integrating genomic, chemical and functional information. It contains a lot of data: signaling pathways, their genes and proteins, the actual chemical components and reactions involved and even the related drugs. Thanks to its API, it can be queried in many ways. Therefore, it is wildly used as a reference knowledge base for integration and interpretation of genome sequencing data [33]. KEGG pathway maps can be downloaded as KEGG Markup Language (KGML) files, a KEGG specific encoding format whose specification is available on their website.

[1] `http://www.csml.org`
[2] `http://www.cellillustrator.com`
[3] `http://www.ebi.ac.uk/biomodels/`
[4] `https://www.kegg.jp/`

4.2.3 Reactome

Reactome[5] is an open-source, open access, manually curated and peer-reviewed pathway database. [34] It provides tools for the visualization, interpretation and analysis of pathways to support basic and clinical research, genome analysis, modeling, systems biology and education. It's probably one of the biggest public database with more than 2200 human pathways, all properly classified, linked and annotated. All pathways can be viewed and analyzed online or exported in multiple formats including SBML, SBGN, and BioPAX level 2 and 3.

4.2.4 PANTHER

The **Protein ANalysis THrough Evolutionary Relationships (PANTHER)**[6] [20] classification system was designed to classify proteins and genes in order to facilitate high-throughput analysis. Part of this classification includes PANTHER Pathway [35], over 177 pathways, all drawn using CellDesigner and containing the mapping information for every component. All PANTHER pathways can be exported in SBML and SBGN.

4.2.5 WikiPathway

Based on the same MediaWiki software that powers Wikipedia, **WikiPathways**[7] aims at facilitating the contribution and maintenance of pathways information by the biology community. More than a simple database, it provides an easy to use interface, allowing anyone in the community, from students to field experts, to contribute in adding, maintaining and reviewing content. It also provides a custom graphical pathway editing tool and a web API for applications to easily connect to it. Pathways are encoded in GPML, PathVisio's default encoding format (see Section 4.3), but most of them can be exported in BioPAX or in standard images formats.

4.3 Public softwares and libraries

From models design to simulation, there are hundreds of softwares available to work with biological networks. This section gives an overview of some of them relevant to this work, as well as some software libraries.

4.3.1 Cell Designer

CellDesigner[8] is a free application designed to easily draw and edit gene-regulatory and biochemical networks [36]. It uses the SBML standard as data storage file format, with the addition of some private annotations, but SBGN export is also supported. CellDesigner can connect to other applications via the Systems Biology Workbench (see Section 4.3.4). It is used by the PANTHER database for the design of their pathways and can directly connect to it as well as other databases such as BioModels.

4.3.2 PathVisio

PathVisio[9] is a free, Java-based, open-source pathway analysis and drawing software. It provides a simple user interface and can be used to edit biological pathways and visualize experimental data on them [37]. It can be directly connected to the WikiPathways database to retrieve GPML models, and with the help of extension plugins, it supports most relevant standards including SBML, SBGN and BioPAX.

[5] `https://reactome.org`
[6] `http://www.pantherdb.org`
[7] `https://www.wikipathways.org/`
[8] `http://www.celldesigner.org/`
[9] `https://www.pathvisio.org`

4.3.3 Cytoscape

Cytoscape[10] is an open-source application for visualizing complex networks and integrating them with any kind of data. It supports a lot of fields, by itself or via the hundreds of available plugins, but is extensively used for its capacities in molecular and system biology. Among other things, it can be used for visualizing, modeling and analyzing molecular and genetic interaction networks. [38].

4.3.4 SBW

Researchers in systems biology make use of a large number of software applications for modeling systems, simulating models or storing and analyzing data. But more than the bother of having to go from one tool to the other, the problem is that most of these tools come with their own file format. Another problem is that some tools duplicate the capabilities of others. Indeed, there is very little code reuse in the biology community and as most projects are short-lived, they are usually not developed very far, forcing new projects to re-implement the same functionalities in order to add their own [39].

The **System Biology Workbench (SBW)** has been developed in an attempt to solve these problems. It's an open-source framework that allows applications, written in different programming languages and running on different platforms, to communicate with each other and share their capabilities via a fast, binary encoded, message system. It comes as a client-server infrastructure and the client-side includes integration libraries for multiple programming languages including C, C++, Java, Delphi, Perl, Python and Matlab. The binary installer also comes bundled with a set of integrated tool to design, layout and simulate biological models[11].

4.3.5 Converters

The way standard file formats are designed is usually dependent on the use cases behind the models. Some are meant to encode boolean regulatory networks and other may be more suited to encode hybrid Petri nets for example (see chapter 6). But fortunately, most models can be generated from one another, or at least to some extent. To help with that, a few converters are available and here is a short list of the most relevant ones:

SBML2SBGNML[12]**.** A converter between SBML and SBGN-ML files. It supports conversion both ways, from SBML Core and Qual to SBGN PD and AF and vice versa.

BioPAX2SBML. A tool that can convert BioPAX level 2 and 3 files to SBML Core + Qual [40].

KEGGTranslator. A powerful tool, able to convert KEGG pathway maps encoded in KGML to a multitude of other formats including SBML (Core and Qual), BioPAX and SBGN [41].

4.3.6 Software libraries

In order to favor their support and integration in applications, most big standards also provide software libraries that can be directly integrated and used to work with their supported file formats.

libSBML. A free, open-source programming library to help developers read, write, manipulate, translate, and validate SBML files. It supports all core versions and most extensions (some are still being developed). It comes with APIs for multiple languages including C, C++, C#, Java, Python and Matlab and is distributed under the LGPL license.

[10] `http://www.cytoscape.org`
[11] `http://jdesigner.sourceforge.net/Site/Welcome.html`
[12] `https://github.com/NRNB-GSoC2017-SBML2SBGNML-Converters/SBML2SBGNML`

libSBGN. A library dedicated to writing and reading SBGN-MLfiles. It also implements files validation and conversion to SBML and BioPAx. It has a Java and a C++ API and is distributed under the LGPL 2.1 and Apache 2.0 licenses [42].

Paxtools[13]**.** A Java library allowing software to read, write, validate, analyze and manipulate BioPAX models. It's distributed by the BioPax team under the LGPL 2.0 license.

[13] `http://biopax.github.io/Paxtools/`

CHAPTER

5

THE SOFTWARE

Now that the required background material has been provided and the main modeling techniques have been presented, this chapter describes the software developed during this work. First, the usage target and philosophy of the solution are discussed as well as the delivered products. Then the solution architecture is detailed and the technical choices are explained.

5.1 Usage target

While researching and developing this project, it clearly appeared that software development and usage in this field did not follow the same rules as usual. As stated earlier in this manuscript, there are three big problems regarding software in biological research:

1. Researchers work with dozens of small tools, all focused on a specific functionality. There is no big, multi-functional toolkit.
2. As most projects behind these tools are short-lived, their functionality is usually not really advanced.
3. Moreover, there is very little code reuse between the developers of these tools. The result is that, when a new tool is developed to test new functionalities, all the underlying functionality necessary to implement the new ones is fully redeveloped even though it is probably already available in another tool.

So, more than just producing an application able to handle the target use cases, one of the goals of this project was to address these problems. Even though, ironically, the produced solution is yet another tool, its open, easily integrable and extensible architecture allows for developers to integrate its capacities in other softwares and extend them with their own (see Section 5.4 for more details).

5.2 The Regulatory Networks Expert System library

The first output of this project is the **Regulatory Networks Expert System library (RNESlib)**. It is delivered as a .Net Core library and provides the basic structure of the expert system as well as the necessary modules to give it the ability to address the two use-cases discussed in chapter 3 (see Chapters 7 and 8 for more details on the mechanism behind these

modules). It can be integrated in any .Net Core 2.1 application and is dependent on libSBML for the management of SBML models (see Section 5.5 for more details).

5.3 Test Application

The second output of this work is the **RNES command line application (RNESCLI)**. It comes as a .Net Core command line application. It exposes the current capabilities of RNESlib, but it is also a great sample code to apprehend the integration of RNESlib in other projects. It can be run on any x64 platform with .Net Core 2.1 installed, from Windows command line or any Linux or macOS terminal.

5.4 Solution architecture

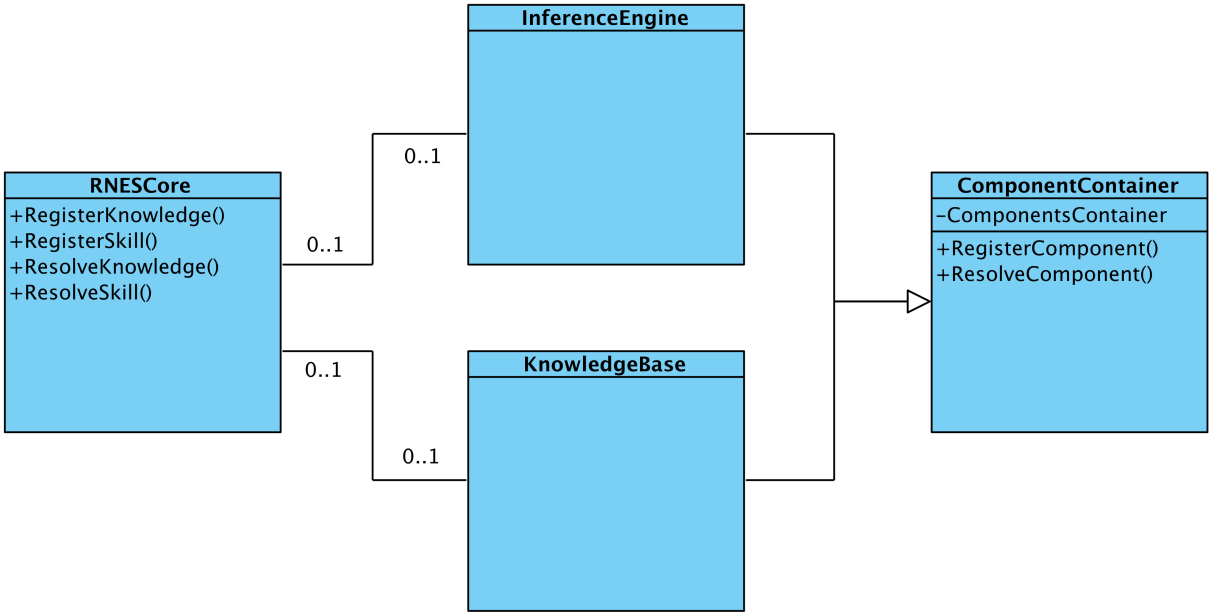

Figure 5.1: RNESlib high level architecture.

The solution architecture has been designed to be as simple as possible with the following aims in mind:

- Integrating and using the library should be straightforward. A few code line should suffice.
- The current functionality should be easily maintainable and expendable.
- As it is meant to be used by the scientific community, it should be highly reliable and thus easy to test.

As displayed in Figure 5.1, the high level architecture relies on four main blocks:

- The *KnowledgeBase*, allowing the registration and resolving of data related components.
- The *InferenceEngine*, allowing the registration and resolving of processing related components.
- The *RNESCore*, main component of the application. It hosts the KnowledgeBase and the InferenceEngine and exposes their ability to register and resolve components.
- The *Skill and knowledge Components*, that can be any class providing the system the ability to retrieve data or process this data and generate an output.

As an illustrative example, Figure 5.2 shows the class diagram of the components bundled with RNESlib and Figure 5.3 contains a snippet of the C# code used RNESlib's network simplification functionality in RNESCLI.

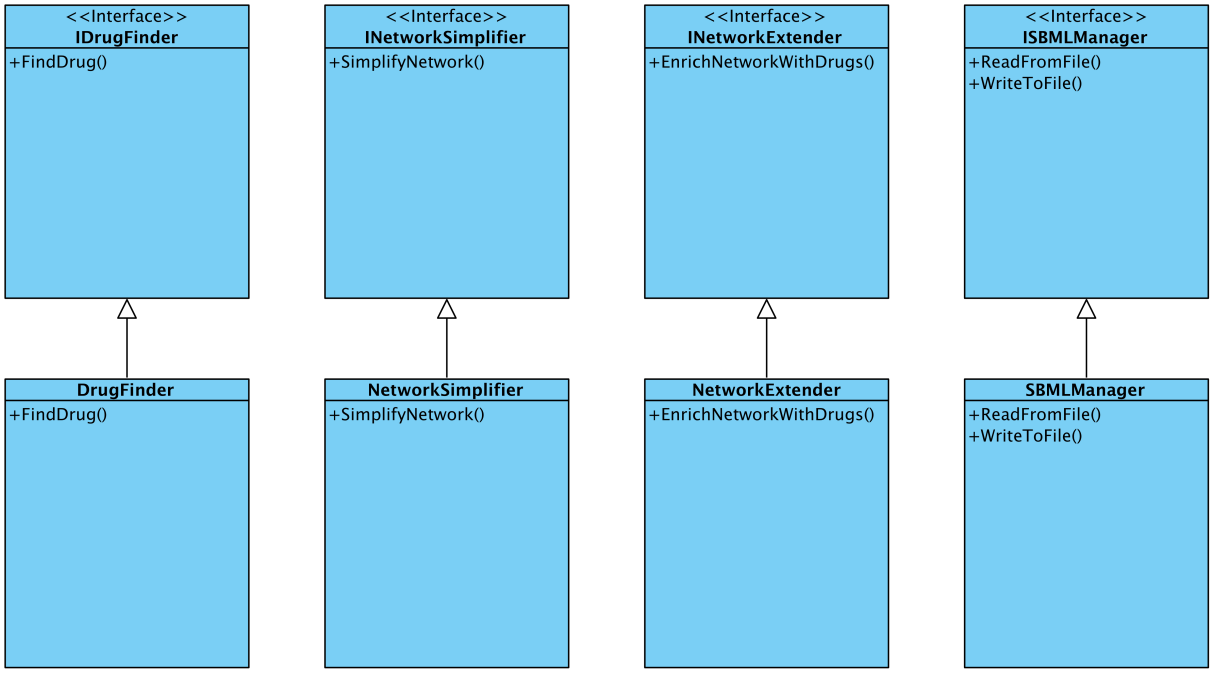

Figure 5.2: RNESlib network simplification and enrichment components class diagram.

```
public class RNESCli{
    public void main(){string[] args}
    // retrieve cli arguments and validate them
    // ...

    var rnesLib = new RNESLib();
    // Register nescessary components
    rnesLib.RegisterKnowledge<IDrugFinder, DrugFinder>()
    rnesLib.RegisterSkill<INetwrokSimplifier, NetworkSimplifier>()
    rnesLib.RegisterSkill<INetworkExtender, NetwrokExtender>()
    rnesLib.RegisterSkill<ISBMLManager, SBMLManager>()

    // Resolve the needed components
    // Note that some components like the DrugFinder and the NetworkSimplifier are not resolved,
    // they are used by the NetwirkExtender and the system will automatically instanciate them
        and provide
    // them to it when it is resolved.
    var sbmlManager = rnesLib.ResolveSkill<ISBMLManager>();
    var networkExtender = rnesLib.ResolveSkill<INetworkExtender>();

    // Call the necessary operations
    // Read the input file and convert it to a boolean model
    var booleanModel = sbmlManager.ReadFromFile(inputFilePath);
    // Simplify the model and enric it with the right drugs
    booleanModel = networkExtender.SimplifyAndEnrichNetworkWithDrugs(booleanModel);
    // Write the output in an SBML file
    SBMLManager.WriteToFile(booleanModel)
}
```

Figure 5.3: Code sample showing the usage of RNESlib by RNESCLI.

Adding functionalities to the system can be done in just a few steps:

1. Implement the necessary classes to handle the data and capacities needed by the functionality.
2. Every one of these classes must implement an interface and expose the necessary methods.
3. Register every component class in the system via its interface.
4. Resolve the necessary objects and call the desired methods.

Finally, as every supported capability is defined in classes with a single responsibility and exposed by an interface, it makes the system highly modular and easily testable. Indeed, if a user wants to write a new implementation of the simplification algorithm, to try a new approach for example, as long as its new class implements the same interface as the original, he just needs to register it in the system instead of the actual implementation and then everything will work seamlessly. Moreover, having every implementation properly interfaced makes it easy to write test systems that integrate mocks of some components to simulate results.

5.5 Input/Output

Currently, the only input file format supported is SBML Qual (see Chapter 4), and to keep the software consistent, computed results are also re-encoded in SBML Qual. Multiple reasons have motivated the choice of SBML as the first supported standard:

- SBML seems to be the most widely supported standard in the current software base. Even though the support of the Qual extension is less frequent.
- SBML comes with *SBMLlib*, a free and easy to use library to work with SBML files. Compared to the other standards, SBMLlib is bundled with interfaces to many programming languages, including C#. And as the underlying library is in C++, it stays portable.
- SBML Qual is the easiest formalism to encode qualitative models of biological regulatory networks. It supports the encoding of models for which very few information is known, thus allowing a wider range of input models.
- Of all the software libraries available to work with the main standards, SBMLlib has the best documentation.
- As SBML is supported by merely all of the biggest databases, it is easier to find compatible input data.

5.6 Programming language

While most of the softwares in the field are developed in Java, the choice has been made to use C# and .Net Core in this project. Here are some of the arguments that motivated this choice:

- .Net Core is free, open-source and multi-platform.
- It is supported by Visual Studio, Microsoft's main IDE, an intuitive and powerful environment.
- Even though it is a pretty recent technology, it benefits from the experience of the .Net Framework. The deployment of ASP.NET Core applications is already well supported and it is really impressive how easily a fully functional web application can be developed and deployed with minimal knowledge.

- Curent version (2.1) supports web-based applications and console applications but Version 3.0 will bring back WPF and WinForm APIs allowing the development of desktop applications.

5.7 Sources

Complete source code and documentation are available on GitHub at `https://github.com/TonusV/RNES` .

CHAPTER

6

NETWORKS MODELING

To infer behaviors and treatment options from biological networks, they need to be modeled and simulated. Depending on the abstraction level needed, there are multiple modeling tools available with different capabilities. Some allow qualitative network modeling, some are more suited for quantitative networks, some have a deterministic approach while others are based on stochastic theories... This chapter gives a comparative overview of the main deterministic tools currently used for biological networks modeling. A conclusion then explains network modeling technique that has been chosen for the software produced in this thesis. Stochastic models are not addressed as they do not fit the needs of this work.

6.1 Boolean networks

Boolean networks are one of the more simplistic qualitative modeling tools available to work with biological networks. In these networks, every entity is either ON or OFF and the state of an entity is determined by logical rules. Changes in the network are deterministic and synchronous [43].

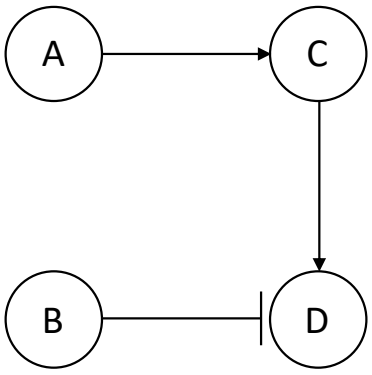

Figure 6.1: Sample boolean regulatory network. B is activated by A, D is activated by B and D is inhibited by C

For example, the regulatory network displayed in Figure 6.1, where nodes A, B en C could be proteins and D a gene, would be defined by the following equations:

- $\mathbf{C(t+1) = A(t)}$
  (protein C is activated by protein A)
- $\mathbf{D(t+1) = C(t) \wedge \neg B(t)}$
  (Gene D is activated by protein C and inhibited by protein B)

Table 6.1 then displays the truth table of this network, illustrating the impact of the multiple combinations of proteins A and B activation on protein C and gene D expression. It is clear, looking at this table, that for gene D to be expressed, protein A must be active and protein C inactive, thus solving the equation.

t				t+1				t+2			
A	B	C	D	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	0	1	0	1	0	1	1
1	1	0	0	1	1	1	0	1	1	1	0

Table 6.1: Truth table of the sample boolean regulatory network. The table clearly shows that, at t+2, the only steady state where gene D is activated is when protein A is active and protein B inactive.

The same conclusion can also be inferred by solving the equation $\mathbf{D(t+1) = C(t) \wedge \neg B(t)}$. $\mathbf{C(t)}$ can be replaced by $\mathbf{A(t-1)}$, resulting in $\mathbf{D(t) = A(t-2) \wedge \neg B(t-1)}$ only solved by A being *true* and B being $false$.

This qualitative modeling method is very efficient to analyze large biological networks, but it highly simplifies the underlying biochemical processes. A gene expression can only be ON or OFF, while in reality, some regulatory mechanisms rely on different levels of expression and thus cannot be modeled properly.

6.2 Generalized logical networks

Introduced by René Thomas in 1991 (see article [44]), generalized logical networks are an extension of boolean networks where variables can have more than two values, and transitions can occur synchronously or asynchronously [43]. They are a good approach to model non-linear interactions in biological regulatory networks while keeping a deterministic qualitative approach.

6.3 Differential equations

Differential equations are another tool that can be used to construct biological regulatory networks based on timed experimental data. There are multiple implementations of this model, not all well suited to represent complex mechanisms because they rely on linear systems. But basically, differential equation based models are able to compute state changes at a specific time, discrete or continuous, by using functions that show the effect of the activation or inhibition of other components [43].

This kind of model is quite popular as it allows a continuous, deterministic approach more precise than discrete or stochastic models. But it requires a large amount of data to compute accurate equations.

6.4 Standard Petri nets

Petri nets have been introduced by Carl Adam Petri in 1962 to model and analyze processes. Because of their strong mathematical basis, precise statements can be made regarding the behavior and state of a modeled system. But it also forces their users to define them rigorously. Petri nets are based on four items: places, transitions, arcs and tokens. Their visual representations is also strongly formalized [45]. An example of the visual representation of such networks is shown in Figure 6.2.

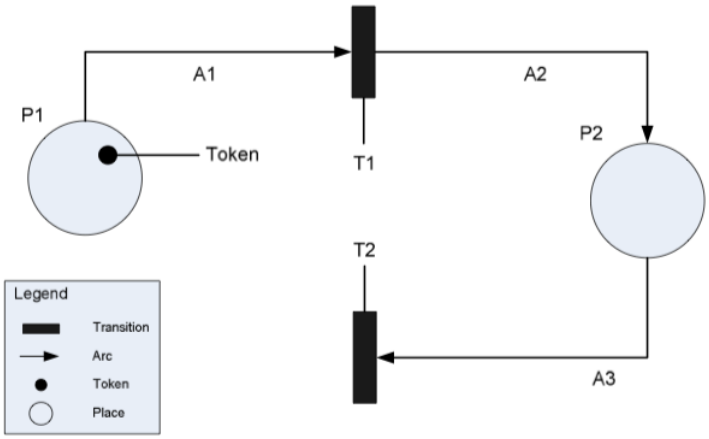

Figure 6.2: Standard Petri net visual representation. The places (P1 and P2) are represented by a circle, the transitions (T1 and T2) by black rectangles, the arcs (A1, A2, and A3) by arrows and the token by a big dot. [45]

Petri nets behave according to the following rules [45]:

- A place can be an input place (if it has outgoing transitions), an output place (if it has incoming transitions), or both.
- A place can contain zero or more tokens.
- A transition consumes and produces tokens. It can fire if there is at least one token in each of its input places.
- Arcs are just arrows that link places and transitions.
- If a transition needs to consume more than one token from a place to fire or produces more than one token, either additional arcs are added so that each arc has a weight of 1, or the weight of the arcs is adjusted accordingly.
- A transition can consume more or less tokens than it produces.
- Transitions fire instantaneously, there is no notion of time.
- The amount of tokens across the network defines the state of the system.

Petri nets are a more recent approach to biological networks modeling. Although they have been used for the first time to this purpose in 1993 [46], their usage has only become popular in the last decade. They are well suited to model qualitative data but it becomes really hard, and sometimes even impossible, to use them for precise quantitative modeling [45].

6.5 Hybrid Functional Petri nets

In order to palliate to the weaknesses of standard Petri nets when it comes to quantitative modeling of complex systems, multiple incremental extensions have been created; all proposing various improvements to the original formalism. One of the more recent extended Petri net proposition is called Hybrid Functional Petri nets. In addition to the original Petri net items, it allows the usage of multiple new features [45]:

- Continuous places and arcs, allowing the representation of continuous transitions over time that can be combined with the original discrete transitions.
- Test arcs, that allow a transition to check the content of a place and fire only if its equal or higher to a certain threshold, but without consuming content from this place.
- Inhibitory arcs that, like test arcs, check the content of a place. But if its equal or higher to the threshold, they prevent the transition from firing.

This extension of Petri nets (that also has a couple of extensions available) is way more suited for the quantitative modeling of biological systems. Continuous transitions and test arcs allow easier modeling of chemical reactions changing concentrations over time, and the catalysis or inhibition of these reactions by other reactants [45].

6.6 Comparison

There is not one and only good way to model biological networks. Depending on the data available and the type of information that must be computed, some choices may be more adequate than others. For example, boolean networks are perfectly suited for the qualitative modeling of big networks. Even though they completely ignore some underlying biological mechanisms, it is not a problem if one does not need them. Hybrid Functional Petri nets, on the other hand, would be perfect for the modeling of quantitative networks with precise experimental data over the chemical reactions happening in the network.

Even though they are not described in this chapter, if the model needs to take into account parameters like the probability of a reaction to fire over time or the impact of external noise on reactions, one may then prefer to use probabilistic models like stochastic Petri nets.

One point to take into account is also the usage evolution of the modeled system. If it is used at first for qualitative modeling but should be reused later to add experimental data and run quantitative simulations, the better choice would be to start with Standard Petri nets. The model could then be updated to an Hybrid Functional Petri net when more data is available. Table 6.2 shows a comparative overview of the modeling techniques previously described in this chapter over three parameters: their deterministic approach, their suitability for qualitative or quantitative modeling and their ability to be used to run continuous simulations over time.

Model	Deterministic	Qualitative	Quantitative	Continuous time simulation
Boolean networks	✓	✓		
Generalized logical networks	✓	✓		
Differential equations	✓		✓	✓
Standard Petri nets	✓	✓		✓
Hybrid Functional Petri nets	✓		✓	✓

Table 6.2: Comparison of the most relevant modeling tools for biological networks.

6.7 Conclusion

To suit the needs of this project, a deterministic, qualitative model was needed. Indeed, to generate the target network visualizations, it must be clear whether the impact of a unit on another is an activation or an inhibition, and what the state of the system could be with the usage of specif drugs. In concordance with the choice of using SBML Qual models as input and output of the application (see Chapter 5), the two possible choices were logical regulatory networks and standard Petri nets. While the first idea was to go for standard Petri nets, the final choice was the use of boolean networks. Keeping in mind that the goal of this implementation is to simplify networks based on a logical approach, and later on, enrich them with hints of drugs that might have a beneficial effect, the choice of boolean networks was motivated by the following arguments:

- they fulfill the minimal requirements in terms of data to be able to generate the target views,
- most SBML Qual models do not contain the necessary information to properly model Petri nets so using a simpler modeling allows for more supported inputs,
- actual simulations are not needed,
- Software implementations need to stay as simple as possible in order to be understandable and easily maintained.

CHAPTER

7

NETWORKS SIMPLIFICATION

As the general structure of the software and the technological choices have been described, this chapter now presents the implementation steps of the way human thinking was emulated in the network simplification algorithm. The first section details the boolean regulatory model implementation used in the software to work with the networks. Then, the multi-step approach of the implementation of the network simplification algorithm is presented.

7.1 Networks modeling

As specified in Chapter 6, the modeling technology that has been chosen to work on biological regulatory networks in this project is the use of boolean networks. The way these networks have actually been implemented can be described as follows:

- Every model is composed of a set of *nodes*, each node representing a species of the network; and a set of *edges*, each edge representing an interaction between two nodes.
- Every node has a *state* (active, inactive or undefined) and a *status* (protected or unprotected).
- Every edge has an *input* and an *output* node, a *type* (positive or negative) and a state (active, inactive or undefined).

Also, to understand the further reasoning, here is the basic set of rules that apply to the models encoded in the system:

- A *protected* node cannot be removed from the model during simplification.
- A *positive* edge implies that the input node as a positive effect on the output node, represented as an *activation*, while a *negative* edge represents an *inhibition* of the output by the input.
- Neither activation nor inhibition edges must be interpreted as strict state changers. A node will be considered active if at least one of its incoming positive edge is active or one if its incoming negative edge is inactive. For example, if a node C is activated by a node A and inhibited by a node B, the equation defining its state would be $\mathbf{C(t+1)} = \mathbf{A(t)} \vee \neg\mathbf{B(t)}$.

- Auto-regulation edges are automatically removed from the system, whether they are defined in the original model or are the result of a simplification. They make no sense in a static representation that does not take transitions time into account.
- The same edge (same input, output and type) is only added once to the model.

Figure 7.1 shows the class diagram of this implementation.

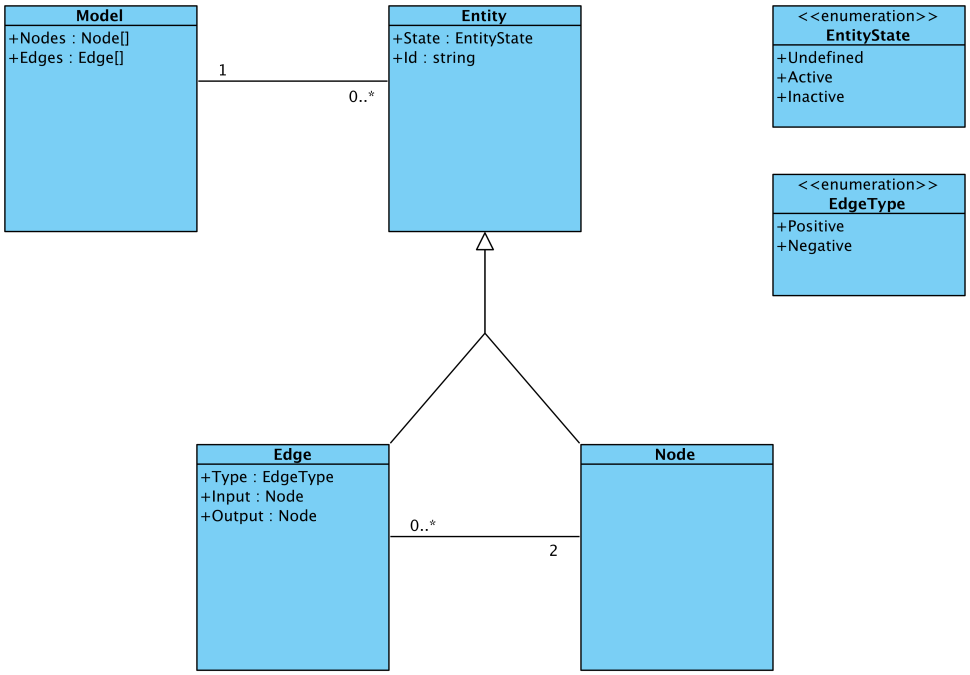

Figure 7.1: Custom boolean network implementation class diagram

Finally, the basic simplification rules that apply to the model can be defined as follows. Considering three nodes $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$:

$$\begin{aligned} \mathbf{A} \Rightarrow \mathbf{B} \Rightarrow \mathbf{C} &= \mathbf{A} \Rightarrow \mathbf{C} \\ \mathbf{A} \Rightarrow \neg\mathbf{B} \Rightarrow \mathbf{C} &= \mathbf{A} \Rightarrow \neg\mathbf{C} \\ \mathbf{A} \Rightarrow \mathbf{B} \Rightarrow \neg\mathbf{C} &= \mathbf{A} \Rightarrow \neg\mathbf{C} \\ \mathbf{A} \Rightarrow \neg\mathbf{B} \Rightarrow \neg\mathbf{C} &= \mathbf{A} \Rightarrow \mathbf{C} \end{aligned}$$

This can of course be demonstrated by solving the underlying logical equations.
Considering the first rule for example,

$$\mathbf{A} \Rightarrow \mathbf{B} \Rightarrow \mathbf{C}$$

could be represented by the following equations:

$$\begin{aligned} \mathbf{B}(\mathbf{t}+\mathbf{1}) &= \mathbf{A}(\mathbf{t}) \\ \mathbf{C}(\mathbf{t}+\mathbf{1}) &= \mathbf{B}(\mathbf{t}) \end{aligned}$$

Solving these equations would produce

$$\mathbf{C}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t})$$

Which is equivalent to

$$\mathbf{A} \Rightarrow \mathbf{C}$$

To better comprehend the steps described further in this chapter and the next, every step will be illustrated using the pathway displayed in Figure 7.2. It is a qualitative representation of the Vascular Endothelial Growth Factor (VEGF) pathway. This pathway regulates genes responsible for angiogenesis, cells survival and proliferation. The original pathway model was imported from the KEGG database[1] and converted to SBML Qual using KEGGTranslator. Visualizations are generated using Cytoscape.

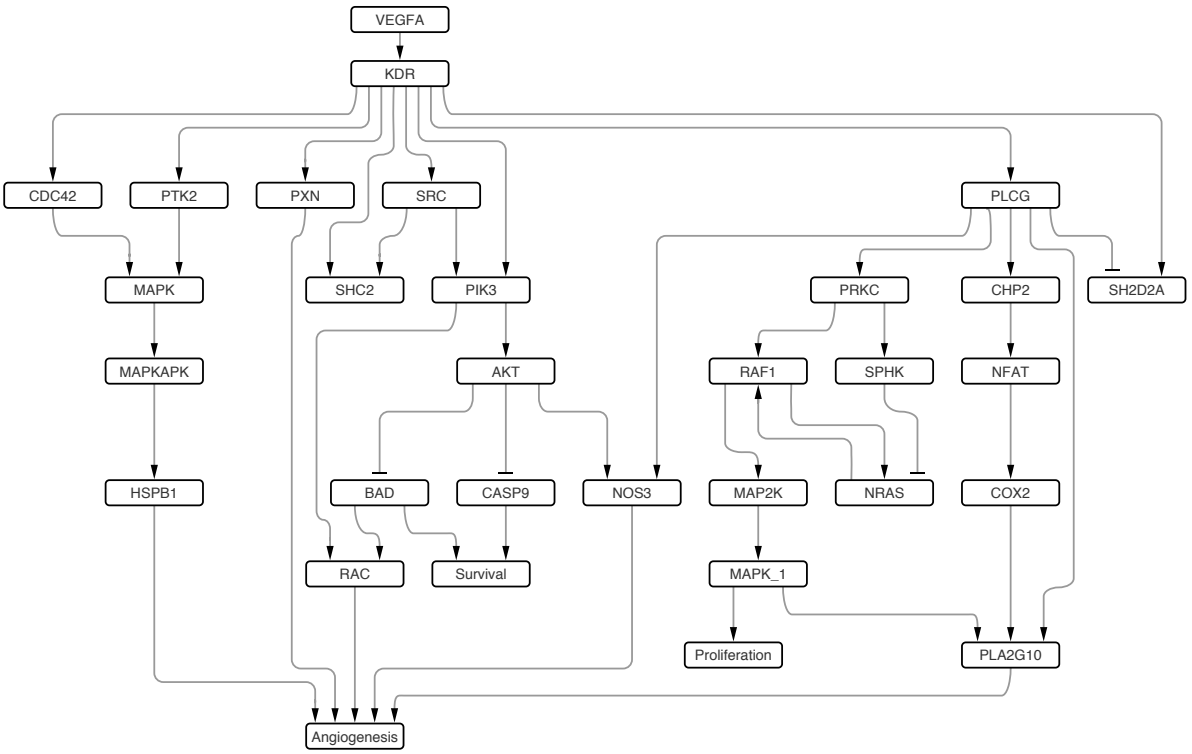

Figure 7.2: Original model visualization. Visualization of a qualitative model of the VEGF signaling pathway.

7.2 First steps

Once the models are loaded and before they can be simplified, some nodes need to be protected, as they are necessary for the understanding of the output network. By default, the decision was made to protect all model input and output nodes. That means, every node that has only incoming or outgoing edges. Usually, for a signaling pathway, input nodes would be cell surface receptors and output nodes the impacted phenotypes. For the example in Figure 7.2, the only input node is the VEGFA receptor and the output nodes are three phenotypes (angiogenesis, proliferation and survival) and two proteins that have no associated phenotype in that model.

Once this was done, the first hunch when developing the simplification algorithm was to remove the node "chains". What is meant by chains is any node that has only one incoming edge and one outgoing edge. In other words, every set of three nodes $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ defined by the following type of equations system:

$$\mathbf{B}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t})$$
$$\mathbf{C}(\mathbf{t}+\mathbf{1}) = \mathbf{B}(\mathbf{t})$$

[1] https://www.genome.jp/kegg-bin/show_pathway?hsa04370

By applying the rules defined in the previous section, these nodes can easily be removed, as well as their related edges. The three entities can then be replaced by a single edge between the input node of the original incoming edge and the output of the original outgoing edge. And of course, this is done recursively on all graph nodes until none can be removed anymore. Figure 7.3 shows the result of this operation on the original VEGF pathway. The size and complexity of the diagram are already reduced but this is not enough.

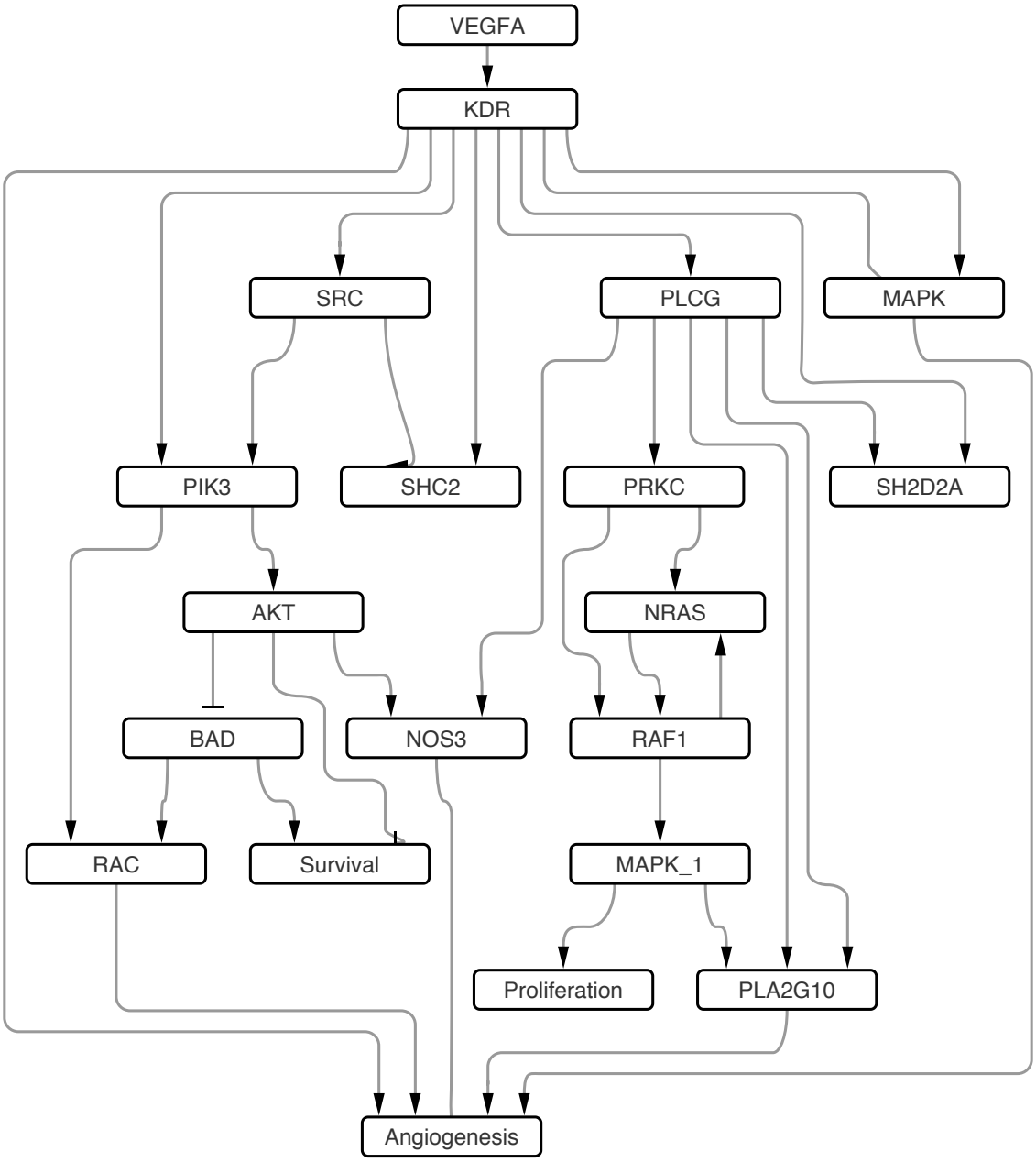

Figure 7.3: Original model after a first simplification step. All node chains have been simplified, resulting in a sensible reduction of the model.

7.3 Extrapolation to complex nodes

While removing the node chains showed some promising results, this was clearly not enough. A method had to be found to generalize this principle and address more complex nodes. By extrapolating the same rules used before, complex nodes can be considered as sets of chains, all passing through the same node. Considering, for example, the following nodes $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and $\mathbf{E}$ so that:

$$\mathbf{A} \Rightarrow \mathbf{C} \qquad \mathbf{C} \Rightarrow \mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{C} \qquad \mathbf{C} \Rightarrow \mathbf{E}$$

By cross-combining these nodes, we can in fact rewrite this as four node chains like

$$\mathbf{A} \Rightarrow \mathbf{C} \Rightarrow \mathbf{D} \qquad \mathbf{A} \Rightarrow \mathbf{C} \Rightarrow \mathbf{E} \qquad \mathbf{B} \Rightarrow \mathbf{C} \Rightarrow \mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{C} \Rightarrow \mathbf{E}$$

and simplify them using the same logic as earlier, thus obtaining the following edges and allowing the removal of the $\mathbf{C}$ node and its associated edges.

$$\mathbf{A} \Rightarrow \mathbf{D} \qquad \mathbf{A} \Rightarrow \mathbf{E} \qquad \mathbf{B} \Rightarrow \mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{E}$$

Mathematically, it would come to solving the following equation system:

$$\begin{gathered}\mathbf{C}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \mathbf{B}(\mathbf{t}) \\ \mathbf{D}(\mathbf{t}+\mathbf{1}) = \mathbf{C}(\mathbf{t}) \\ \mathbf{E}(\mathbf{t}+\mathbf{1}) = \mathbf{C}(\mathbf{t})\end{gathered}$$

resulting in:

$$\begin{gathered}\mathbf{D}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \mathbf{B}(\mathbf{t}) \\ \mathbf{E}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \mathbf{B}(\mathbf{t})\end{gathered}$$

As for the first mechanism, this is applied recursively until no more nodes can be removed. Figure 7.4 shows the result of this improved mechanism on the original network. It seems to be extremely efficient compared to the previous algorithm.

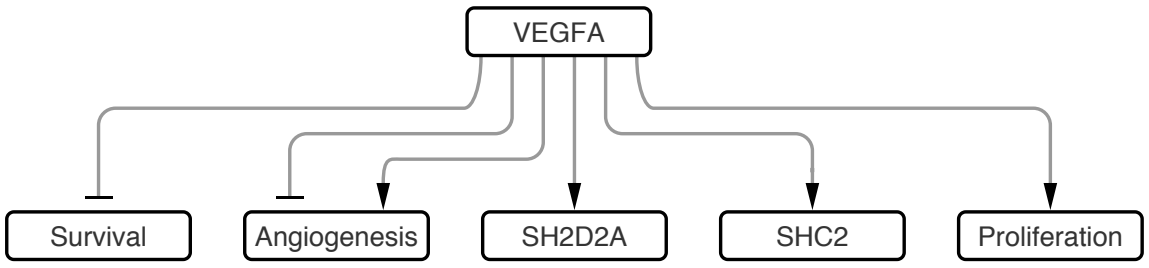

Figure 7.4: Original model after the application of the improved simplification algorithm. The network is now reduced at its simplest expression.

7.4 Avoiding inconsistency

While Figure 7.4 seems to be the most reductive representation of the original pathway, something is wrong with it. The way it is displayed, the VEGFA receptor both activates and inhibits angiogenesis. In other words, by continuously solving equations, the system ends up with a node whose state description is of the type $\mathbf{B}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \neg\mathbf{A}(\mathbf{t})$ which is always true.

This is a reality in some way, but presenting it like this renders the pathway inconsistent for the reader. So the decision was made to tweak the simplification algorithm so that a new edge

can only be added to the model if it doesn't already contain the same edge with the opposite type. In this case, the original node and edges are kept in the model. This way, the reader can understand that, while the activation of one protein can have contradictory effects on one phenotype, it can still be fixed by acting on intermediary nodes. Figure 7.5 Shows the result of this small improvement on the simplification of the VEGF pathway.

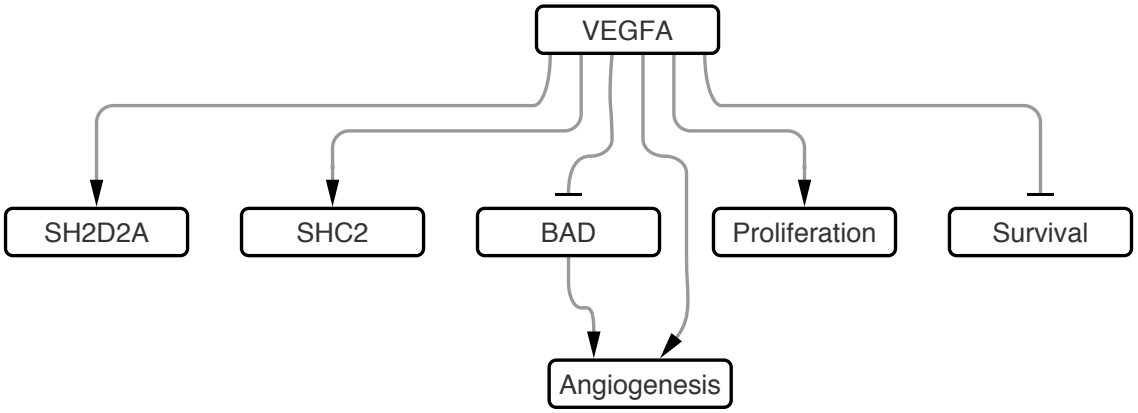

Figure 7.5: Original model after using the third version of the simplification algorithm. The resulting pathway is now as small as possible without introducing contradictory edges between the VEGFA and Angiogenesis nodes.

Note: the watchful reader might raise the concern that this renders the final output dependent of the order in which the nodes have been encoded and treated. Indeed, to avoid that problem, nodes and edges need to be sorted before being processed so that the final result is always the same.

7.5 One last optimization

After the last fix, one more problem appeared with this optimization algorithm. But it doesn't occur with the VEGF pathway model used as example here. In some cases, when a node must be kept to avoid contradictory edges, simplifying the remaining edges related to this node could lead the algorithm to add more new edges than it actually removes. Consider the following nodes $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and $\mathbf{E}$, and the following set of edges:

$$\mathbf{A} \Rightarrow \mathbf{C} \qquad \mathbf{C} \Rightarrow \mathbf{D} \qquad \mathbf{A} \Rightarrow \neg\mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{C} \qquad \mathbf{C} \Rightarrow \mathbf{E}$$

The removal of $\mathbf{C}$ and its related edges could be done by introducing the following four new edges:

$$\mathbf{A} \Rightarrow \mathbf{D} \qquad \mathbf{A} \Rightarrow \mathbf{E} \qquad \mathbf{B} \Rightarrow \mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{E}$$

But $\mathbf{A} \Rightarrow \mathbf{D}$ cannot be added to the model because $\mathbf{A} \Rightarrow \neg\mathbf{D}$ already exists. So the $\mathbf{C}$ node must be kept as well as the $\mathbf{A} \Rightarrow \mathbf{C}$ and $\mathbf{C} \Rightarrow \mathbf{D}$ edges. It means that, if the algorithm keeps these but applies the possible optimizations to the other edges, the resulting model would contain:

$$\mathbf{A} \Rightarrow \mathbf{C} \qquad \mathbf{C} \Rightarrow \mathbf{D} \qquad \mathbf{A} \Rightarrow \neg\mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{D} \qquad \mathbf{B} \Rightarrow \mathbf{E} \qquad \mathbf{A} \Rightarrow \mathbf{E}$$

The system would then have introduced one more edge than the original situation, rendering the model more complex. To counter this effect, a last improvement was made to the algorithm so that an optimization only occurs if it can remove a node, or wouldn't add more edges to the model than it removes.

CHAPTER

8

NETWORKS ENRICHMENT

Now that the mechanisms behind the networks simplification algorithm have been detailed, this chapter will go further into details to explain how this algorithm was improved to be able to enrich the resulting models with new nodes, containing hints on potentially beneficial drugs.

8.1 Targeted treatment

To be able to find out what drugs could be used to treat the analyzed cancer and improve the patient's health, the application must be able to find out, based on the available data, which drugs have an impact on which protein and what kind of impact. But it still needs to know which phenotypes must be targeted. In the scope of this work, we consider this information to be provided by the user.

Also, for the sake of keeping the example easy to understand, the available drugs data has been limited to a set of three drug types:

- An AKT inhibitor
- A VEGFA activator
- A MAP2K inhibitor

While real drugs targeting these proteins exist in the scope of cancer treatment, it is easier to ignore their names here.

8.2 Protecting drug-related nodes

The first step in the implementation of this functionality was to get back all the nodes that could be targeted by drugs in the model. To do so the system parses all the nodes and checks in its database if there is a drug available that could impact it. If there is one, the node will be marked as protected so it can't be removed by the optimization algorithm. Figure 8.1 shows the state of the original model after a run of the optimization algorithm, but with all nodes that could potentially be targeted by a drug protected, as well as the model input and output nodes.

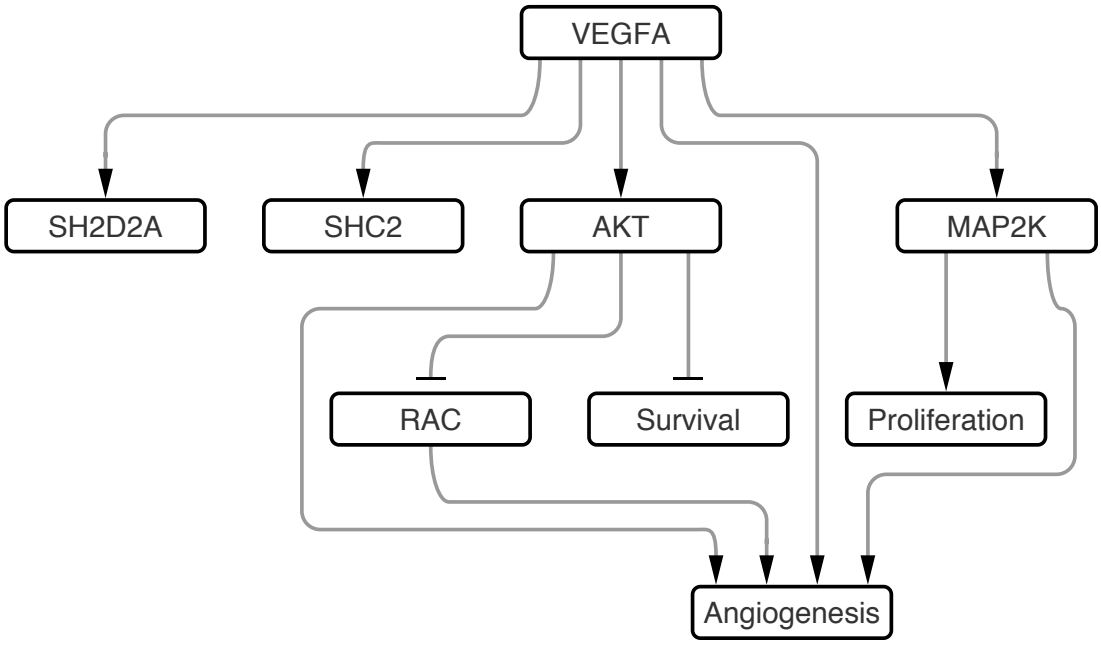

Figure 8.1: Original model optimization with drug-related nodes protected. The resulting pathway now contains more nodes than the previous version but it is necessary to integrate relevant drugs in the model.

8.3 Adding the right drugs

Now that the algorithm was able to simplify a model while keeping all nodes potentially relevant for the targeted treatment of the encoded phenotypes, it needed to extend it with the right drug nodes. But before going into the details of how this was done, let's remember that the output of this functionality is not an approved treatment proposition. The drugs added in the model are added based solely on boolean qualitative data.

The only thing that can be inferred is that each one of these drugs has a positive effect on at least one of the targeted phenotypes via one or more pathways. To know more precisely which of these drugs would create the best combination for an effective treatment and what effect can really be expected from their usage, simulations need to be run on quantitative models containing more precise data on the nodes activation levels. With that in mind, here is the explanation of how the algorithm was implemented in the application.

Mathematically, the concept is simple: considering two nodes $\mathbf{A}$ and $\mathbf{B}$, and the phenotype node $\mathbf{P}$ whose state equation is

$$\mathbf{P}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \mathbf{B}(\mathbf{t})$$

if the user wants this phenotype to be expressed, the system needs to find all the drug nodes whose activation would provide a solution to this equation. On the contrary, if the user wants this phenotype to be inhibited, the system will find all the drug nodes whose activation solve the equation

$$\neg\mathbf{P}(\mathbf{t}+\mathbf{1}) = \mathbf{A}(\mathbf{t}) \vee \mathbf{B}(\mathbf{t})$$

At the software level, here is an explanation of how the system solves these equations based on the available drugs. Basically, the process starts from the targeted phenotypes and runs what

could be qualified as a "backward simulation". From the first node, the only thing known by the application is the desired result (activation or inhibition). The algorithm then runs according to the following steps:

1. If the state of the current node is not undefined, return.
   If not:
2. Change the current node state to active or inactive based on the desired result.
3. Check if there is a drug able to apply the desired result on the current node.
4. If there is one, add the drug node and the appropriate edge to the model.
5. Get all the incoming edges of the current node.
6. For each one of these edges, computes the desired state of their input node based on the current node desired state and the edge type.
7. Call this algorithm recursively on the input node of every incoming edge with the right desired state until all possible pathways have been covered.

Once this algorithm is over, the model has been enriched with every drug that could possibly have the desired effect on the targeted phenotypes. Figure 8.2 shows the result of this update on the original VEGF pathway having targeted the inhibition of survival and proliferation.

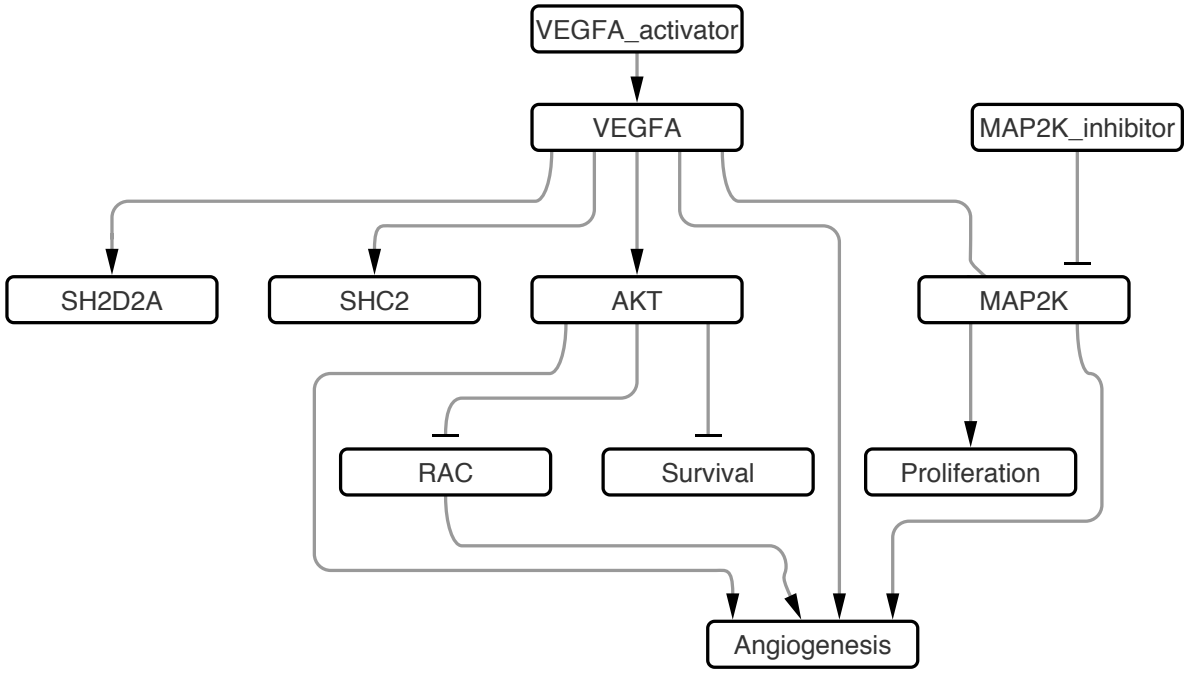

Figure 8.2: Optimization and enrichment of the VEGF signaling pathway targeting the inhibition of survival and proliferation.

8.4 One last step

Now that the application was able to find the right drugs to target one or more phenotypes, it also knew exactly what nodes should be kept as they were of interest for the user. With that in mind, a last improvement was made to the software so that, once the drugs have been added, a model simplification can be rerun on the enriched model with a couple of updates:

- The protection is removed from the drug-related nodes for which no drug has been added to model.
- The protection of model input nodes and model output nodes that are not part of the phenotypes targeted by the user are also removed.

This allows the final generation of a minimalist model containing only the strictly necessary data, based on the user needs. Figure 8.3 shows the result of this last optimization on the original model, now reduced to 6 nodes and 5 edges (coming from 31 nodes and 45 edges).

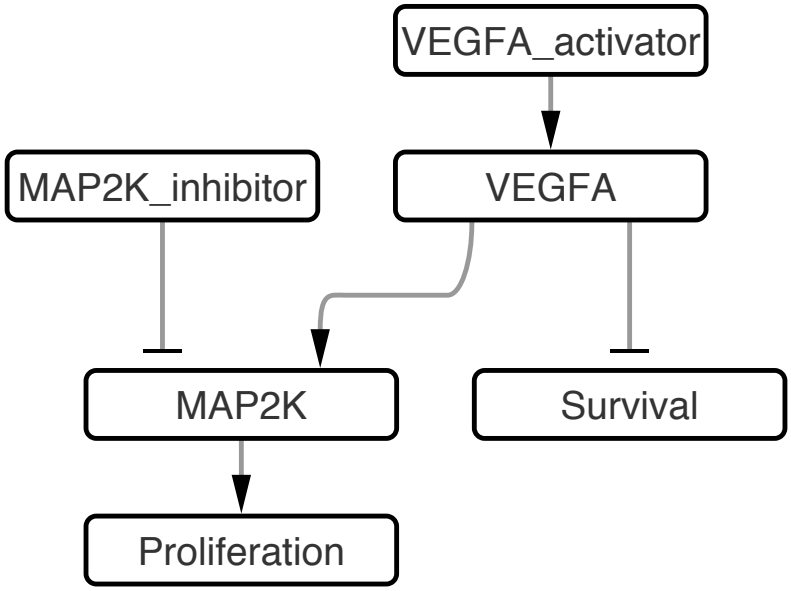

Figure 8.3: Final optimization of the VEGF signaling pathway model, targeting the inhibition of survival and proliferation.

CHAPTER

9

PERSPECTIVES

This chapter discusses future perspectives for the project based on various strengths and weaknesses of the current implementation.

9.1 User interface

A CLI application should suffice to test and use functionality to some extent. It might also be considered a good feature as it allows tools to be easily integrated in automatic workflows. Nonetheless, for this solution to have a chance to be widely adopted, a more user-friendly application with an actual UI is a must have. It makes it easier for the user to apprehend the software without having to read the documentation to find out what arguments should be passed to the command line.

As long as developing a UI, the way to go seems to be a web-based interface. More than the fact that it is really popular these days, it is actually a smart, cost-efficient move. Web-base UIs can easily be integrated in desktop applications using tools like Electron[1] while the opposite statement is not true. By doing so, if the application later needs to be extended as a full web application, the UI can just be reused at practically no cost.

9.2 Complex networks

While the simplification and enhancement algorithms have given pretty good results in the tested networks, they have their limitations. For example, looking at the regulatory network in Figure 9.1, displaying multiple interconnected signaling pathways regulating some cancer phenotypes, the original network has 49 nodes and 100 edges with numerous cross connections.

The passage of this network through our algorithms with the growth arrest phenotype as only target and one drug available (an AKT inhibitor) outputs a network containing 12 nodes and 30 edges as displayed in Figure 9.2. This is roughly a reduction of 70%, which is not that bad, but still, the resulting network remains too complex to be quickly readable by a human being.

[1] https://electronjs.org

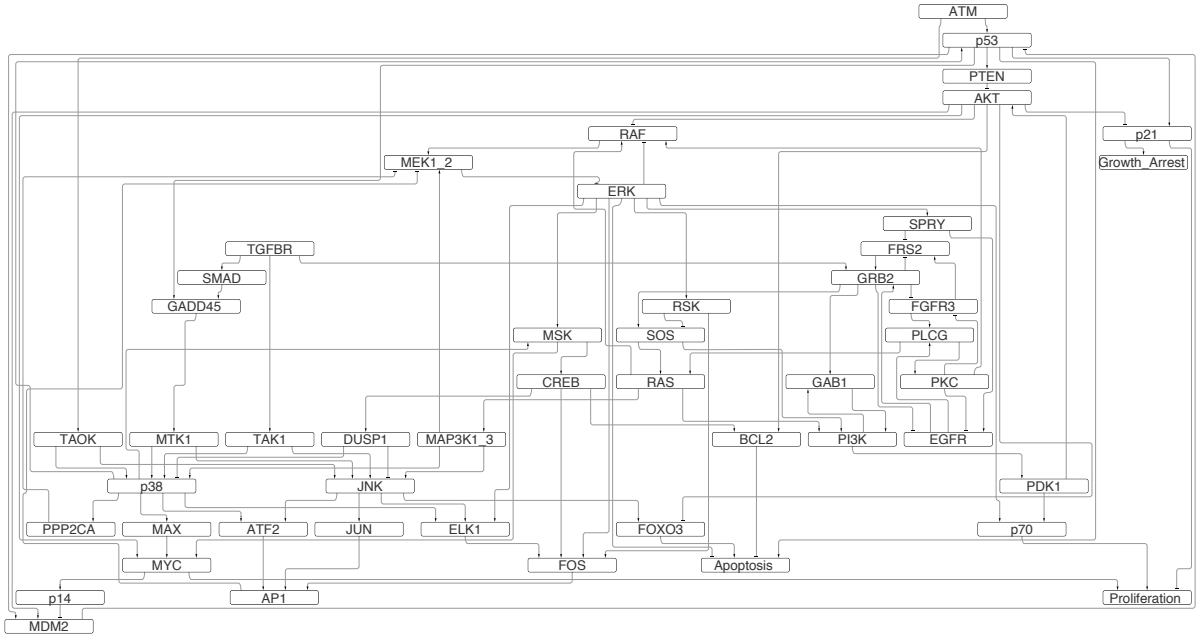

Figure 9.1: Mitogen-Activated Protein Kinase (MAPK) signaling pathways. These interconnected signalling pathways are involved in multiple cancer-related phenotypes such as apoptosis, cell growth and proliferation [47].

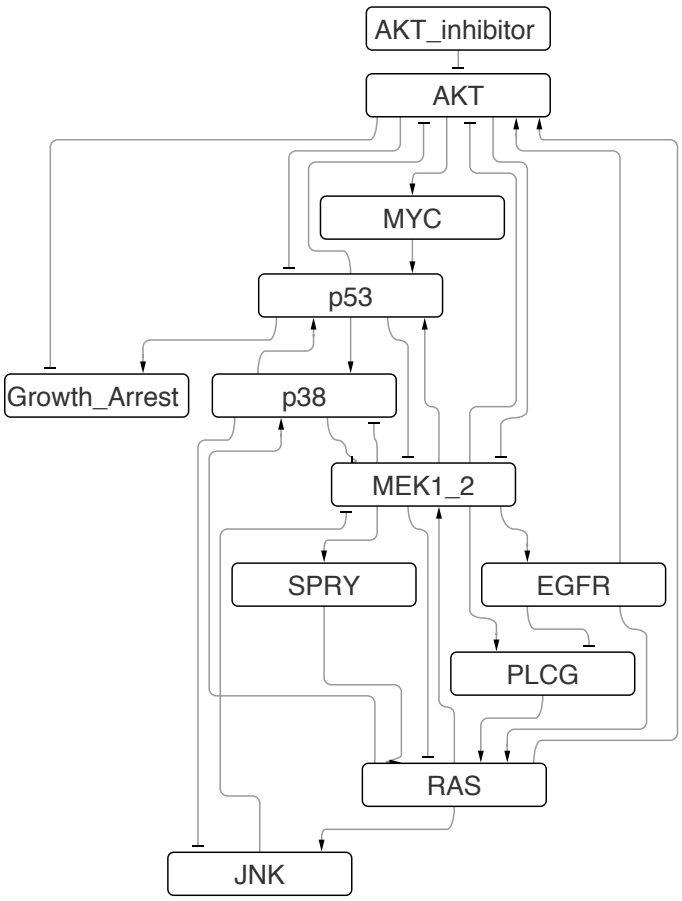

Figure 9.2: Simplified and enhanced MAPK signaling pathways.

There are some leads that could be investigated to try and shrink this kind of network a bit more. Gathering some nodes based on codependent relations to form node groups might render these networks easier to read. Running simulations on the modeled networks and try to identify non-relevant signaling loops might help too. But it would probably be difficult to go further without more data on the actual activation levels of the transitions. We might then be forced to upgrade our system to support more complex models.

9.3 Supporting Petri nets

Boolean networks are a really good starting point as they allow the processing of a wide range of models, and their encoding represents a system very close to the target visual representation. However, as stated in Section 9.2, they have their limitations. The data they express can only allow networks simplification up to some point, and the drug-mapping algorithm can only insure that the drugs added to the system could have a positive effect on the target phenotype depending on the actual levels of activations of the nodes implied in the regulation pathways between this drug and the phenotype.

Supporting the modeling of standard Petri nets would give the application the ability to be more efficient. Properly configured Petri nets would provide the ability to simplify the networks more realistically based on the transitions firing pre-conditions that represent the activation levels of the associated nodes. More nodes could be removed and contradictory edges could be resolved based on the difference between their token production and consumption. Moreover, it could be done without having to support another encoding format than SBML Qual as it supports the encoding of standard Petri nets.

9.4 Visual rendering

A feature that would be a great improvement, if a UI-based application was developed, is the visual rendering of the networks. It would bring multiple advantages and could allow the user to:

- avoid the obligation to use another tool to validate the results and exploit them;
- interact directly with the network;
- manually select the target phenotypes on the screen;
- force the protection or removal of some nodes;
- manually layout the networks;
- ...

But that would also require a lot of work and bring multiple questions and problems to the table:

- Displaying the network on screen would mean to be able to display it properly, not as a pile of nodes and edges.
- Even though automatic layout algorithms do exist, they usually don't work very well for big networks.
- The visual rendering style of nodes should be chosen.
- A biological visualization, using for example the SBGN standard, would be the best choice but that would imply identifying the nature of all nodes of the network.

- While some input networks might have rendering and layout information included or annotations with information on the nature of the species, this is not always the case.
- Even if it is the case, once the network is simplified, some of this information might not be relevant anymore.
- ...

Nonetheless, there is interesting work to do in this field.

9.5 Software integration

No matter the type of application, a good integration with existing softwares seems to be an important feature. For the product to be adopted, it needs to be usable in the current environment and then maybe start growing and integrate more functionality. A nice way to provide that integration would be to add the possibility to connect to the SBW (see Chapter 4) and interact with the other compatible softwares. The SBW is also a good way to share functionalities with software developed in other languages and that couldn't integrate RNESlib.

9.6 Standards support

Even though the integration with SBML Qual is already a good thing, with SBML probably being the most popular standard, supporting more formats as input and output would open the application to a wider user base. It could also open the door to new functionalities like conversion from one format to another, at least to the extent of the information supported by the internal boolean model.

9.7 Public databases

The actual implementation of the RNESCLI tool requires the user to pass the list of available drugs as an argument. For the application to be more easily used in real conditions, providing drug-related knowledge modules that can connect to public databases like KEGG or PANTHER to retrieve the necessary information would be an interesting feature. It would only require the implementation of new modules that can be seamlessly substituted to the original one, letting the users or developers decide what public data source they want to use.

CHAPTER

10

CONCLUSION

The aim of this thesis was to lay down the foundations of an expert system, able to help biologists in the simplification of gene regulatory networks and the research of potentially beneficial targeted treatments. The first chapter introduced the biological background of cancer and the basic principles of targeted therapy. Then, the work of biologists to provide oncologists with relevant treatment options was discussed and, after characterizing the traits of an expert system, a solution was proposed.

After providing an overview of the current state of the art in terms of software and technologies, the implemented solution was presented with all its targets, constraints and technological choices. The next chapters then provided a more detailed explanation on the way the simplification and network enrichment algorithms were conceived, as well as their mathematical bases. And finally, future perspectives of the solution where discussed.

While the implemented solution does provide the ability to successfully simplify and enhance biological regulatory networks, it might not meet the requirements of an expert system per se. One of the main missing features being the justification of its reasoning on the processed network. This could be addressed by providing a way to output intermediary networks, or a list of the computed simplifications and drugs signaling pathways. But a good solution should probably involve a user interface. It would open the application to more use cases and give it a real chance to be adopted as a viable tool for actual biologists.

Also, a user interface would make it easier to layout the resulting pathways and really create a visual representation that can be used, as is, by an oncologist. The actual implementation makes it mandatory to use another tool to visualize and layout the resulting networks and it is yet another constraint as this tool has to support the SBML Qual standard. Finally, supporting more standard formats would help intercommunication with other applications and make it easier to integrate the current solution in an existing workflow.

More than the development of the software itself, the biggest challenge throughout this work was certainly to fill the necessary knowledge gap to finally be able to start producing something. Indeed, starting the research with absolutely no knowledge of the underlying biological concepts behind cancer, it was quite difficult to find its way through the enormous amount of information out there. Also, more than the difficulty of finding the relevant documentation, most of this

documentation is targeted at biologists which doesn't help the learning curve.

But finally, after months of research, gathering information and trying to make biological and computational concepts fit together, the development could start. Though, the elaboration of the final solution has probably been more guided by the software side than the biological side. More than trying to address a field problem, the produced system is oriented to solving a user problem with its current environment and is probably of more interest to the biological software development world than to the biologists themselves.

Looking at the current state of software dedicated to biological research, at least based on the information gathered during this work, it seems like the field is in desperate need of a common goal in term of software development. An awful lot of small tools are developed by researchers to address specific models or algorithms, but it seems like nothing is ever integrated together to try and form a bigger product, capable of addressing a wider range of problems. No effort is made at gathering and combining different concepts or different process stages in the same application.

The software produced in the scope of this work might be too specific in terms of capabilities to trigger the attention yet. But it could be an interesting basis to build upon for future projects, enlarging its functionalities and improving it so that it really becomes a relevant application for the field. Anyway, the only certainty is that software and artificial intelligence have a great role to play in helping researchers in the fight against cancer.

BIBLIOGRAPHY

[1] Jason A. Reuter, Damek V. Spacek, and Michael P. Snyder. High-Throughput Sequencing Technologies. *Molecular Cell*, 58(4):586–597, 08 2015.

[2] Colette Moses, Benjamin Garcia-Bloj, Alan R. Harvey, and Pilar Blancafort. Hallmarks of cancer: The CRISPR generation. *European Journal of Cancer*, 93:10–18, 2018.

[3] R. Weinberg. *The Biology of Cancer, Second Edition.* Taylor & Francis Group, 2013.

[4] James D Watson and Francis HC Crick. The structure of DNA. In *Cold Spring Harbor symposia on quantitative biology*, volume 18, pages 123–131. Cold Spring Harbor Laboratory Press, 1953.

[5] Douglas Hanahan and Robert A. Weinberg. The Hallmarks of Cancer. *Cell*, 100:57–70, 1 2000.

[6] Yosef Yarden and Mark X. Sliwkowski. Untangling the ErbB signalling network. *Nature Reviews Molecular Cell Biology*, 2:127 EP –, 2 2001.

[7] Frank Emmert-Streib, Matthias Dehmer, and Benjamin Haibe-Kains. Gene regulatory networks and their applications: understanding biological and medical problems in terms of networks. *Frontiers in Cell and Developmental Biology*, 2:38, 2014.

[8] Nicoline Y. den Breems, Lan K. Nguyen, and Don Kulasiri. Integrated signaling pathway and gene expression regulatory model to dissect dynamics of Escherichia coli challenged mammary epithelial cells. *Biosystems*, 126:27–40, 2014.

[9] Fan Zhang, Runsheng Liu, and Jie Zheng. Sig2GRN: A software tool linking signaling pathway with gene regulatory network for dynamic simulation. *BMC Systems Biology*, 10:541–548, 12 2016.

[10] Menderes Yusuf Terzi, Muzeyyen Izmirli, and Bulent Gogebakan. The cell fate: senescence or quiescence. *Molecular Biology Reports*, 43:1213–1220, 2016.

[11] Douglas Hanahan and Robert A. Weinberg. Hallmarks of Cancer: The Next Generation. *Cell*, 144:647–674, 3 2011.

[12] Nikki C. K. Cheng, Anna M Chytil, Yu Shyr, Alison Joly, and Harold L. Moses. Transforming growth factor-beta signaling-deficient fibroblasts enhance hepatocyte growth factor signaling in mammary carcinoma cells to promote scattering and invasion. *Molecular cancer research : MCR*, 6 10:1521–33, 2008.

[13] J W Shay and W N Keith. Targeting telomerase for cancer therapeutics. *British Journal of Cancer*, 98(4):677–683, 2 2008.

[14] Daniel Gomez, Romina Armando, Hernn G Farina, Pablo Menna, Carolina Cerrudo, Pablo Ghiringhelli, and Daniel Alonso. Telomere structure and telomerase in health and disease. *International Journal of Oncology*, 41(5):1561–1569, 11 2012.

[15] Jaewon Min, Woodring E Wright, and Jerry W Shay. Alternative lengthening of telomeres can be maintained by preferential elongation of lagging strands. *Nucleic Acids Research*, 45(5):2615–2628, 3 2017.

[16] National Cancer Institute. Types of Cancer Treatments, 4 2017.

[17] Lee M. Ellis and David A. Reardon. The nuances of therapy. *Nature*, 458:290, 03 2009.

[18] Peter Jackson. *Introduction to Expert Systems, third edition.* Addison Wesley, 1998.

[19] Denise N Slenter, Martina Kutmon, Kristina Hanspers, Anders Riutta, Jacob Windsor, Nuno Nunes, Jonathan Mlius, Elisa Cirillo, Susan L Coort, Daniela Digles, Friederike Ehrhart, Pieter Giesbertz, Marianthi Kalafati, Marvin Martens, Ryan Miller, Kozo Nishida, Linda Rieswijk, Andra Waagmeester, Lars M T Eijssen, Chris T Evelo, Alexander R Pico, and Egon L Willighagen. WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research. *Nucleic Acids Research*, 46:D661–D667, 2018.

[20] Huaiyu Mi, Xiaosong Huang, Anushya Muruganujan, Haiming Tang, Caitlin Mills, Diane Kang, and Paul D. Thomas. PANTHER version 11: expanded annotation data from Gene Ontology and Reactome pathways, and data analysis tool enhancements. *Nucleic Acids Research*, 45:183–189, 2017.

[21] Joseph Vadakara and Hossein Borghaei. Personalized medicine and treatment approaches in non-small-cell lung carcinoma. *Pharmacogenomics and personalized medicine*, 5:113–123, 09 2012.

[22] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novre, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[23] Leo Lahti. A brief overview on the BioPAX and SBML standards for formal presentation of complex biological knowledge. 09 2011.

[24] Lena Strmbck and Patrick Lambrix. Representations of molecular pathways: an evaluation of SBML, PSI MI and BioPAX. *Bioinformatics*, 21(24):4401–4407, 2005.

[25] Claudine Chaouiya, Duncan Bérenguier, Sarah M. Keating, Aurélien Naldi, Martijn P. van Iersel, Nicolas Rodriguez, Andreas Dräger, Finja Büchel, Thomas Cokelaer, Bryan Kowal, Benjamin Wicks, Emanuel Gonçalves, Julien Dorier, Michel Page, Pedro T. Monteiro, Axel von Kamp, Ioannis Xenarios, Hidde de Jong, Michael Hucka, Steffen Klamt, Denis Thieffry, Nicolas Le Novère, Julio Saez-Rodriguez, and Tomáš Helikar. SBML qualitative models: a model representation format and infrastructure to foster interactions between qualitative modelling formalisms and tools. *BMC Systems Biology*, 7(1):135, Dec 2013.

[26] Ralph Gauges, Ursula Rost, Sven Sahle, and Katja Wegner. A model diagram layout extension for SBML. *Bioinformatics*, 22:1879–1885, 2006.

[27] Frank T. Bergmann, Sarah M. Keating, Ralph Gauges, Sven Sahle, and Katja Wegner. SBML Level 3 package: Render, Version 1, Release 1. *Journal of Integrative Bioinformatics*, 15, 2018.

[28] Emek Demir, Michael P Cary, Suzanne Paley, Ken Fukuda, Christian Lemer, Imre Vastrik, Guanming Wu, Peter D'Eustachio, Carl Schaefer, Joanne Luciano, Frank Schacherer, Irma Martinez-Flores, Zhenjun Hu, Veronica Jimenez-Jacinto, Geeta Joshi-Tope, Kumaran Kandasamy, Alejandra C Lopez-Fuentes, Huaiyu Mi, Elgar Pichler, Igor Rodchenkov, Andrea Splendiani, Sasha Tkachev, Jeremy Zucker, Gopal Gopinath, Harsha Rajasimha, Ranjani Ramakrishnan, Imran Shah, Mustafa Syed, Nadia Anwar, Özgün Babur, Michael Blinov, Erik Brauner, Dan Corwin, Sylva Donaldson, Frank Gibbons, Robert Goldberg, Peter Hornbeck, Augustin Luna, Peter Murray-Rust, Eric Neumann, Oliver Ruebenacker, Matthias Samwald, Martijn van Iersel, Sarala Wimalaratne, Keith Allen, Burk Braun, Michelle Whirl-Carrillo, Kei-Hoi Cheung, Kam Dahlquist, Andrew Finney, Marc Gillespie, Elizabeth Glass, Li Gong, Robin Haw, Michael Honig, Olivier Hubaut, David Kane, Shiva Krupa, Martina Kutmon, Julie Leonard, Debbie Marks, David Merberg, Victoria Petri, Alex Pico, Dean Ravenscroft, Liya Ren, Nigam Shah, Margot Sunshine, Rebecca Tang, Ryan Whaley, Stan Letovksy, Kenneth H Buetow, Andrey Rzhetsky, Vincent Schachter, Bruno S Sobral, Ugur Dogrusoz, Shannon McWeeney, Mirit Aladjem, Ewan Birney, Julio Collado-Vides, Susumu Goto, Michael Hucka, Nicolas Le Novère, Natalia Maltsev, Akhilesh Pandey, Paul Thomas, Edgar Wingender, Peter D Karp, Chris Sander, and Gary D Bader. The BioPAX community standard for pathway data sharing. *Nature Biotechnology*, 28:935–642, 09 2010.

[29] Nicolas Le Novère, Michael Hucka, Huaiyu Mi, Stuart Moodie, Falk Schreiber, Anatoly Sorokin, Emek Demir, Katja Wegner, Mirit I Aladjem, Sarala M Wimalaratne, Frank T Bergman, Ralph Gauges, Peter Ghazal, Hideya Kawaji, Lu Li, Yukiko Matsuoka, Alice Villéger, Sarah E Boyd, Laurence Calzone, Melanie Courtot, Ugur Dogrusoz, Tom C Freeman, Akira Funahashi, Samik Ghosh, Akiya Jouraku, Sohyoung Kim, Fedor Kolpakov, Augustin Luna, Sven Sahle, Esther Schmidt, Steven Watterson, Guanming Wu, Igor Goryanin, Douglas B Kell, Chris Sander, Herbert Sauro, Jacky L Snoep, Kurt Kohn, and Hiroaki Kitano. The Systems Biology Graphical Notation. *Nature Biotechnology*, 27:735–741, 08 2009.

[30] Michael Hucka, David P. Nickerson, Gary D. Bader, Frank T. Bergmann, Jonathan Cooper, Emek Demir, Alan Garny, Martin Golebiewski, Chris J. Myers, Falk Schreiber, Dagmar Waltemath, and Nicolas Le Novre. Promoting Coordinated Development of Community-Based Information Standards for Modeling in Biology: The COMBINE Initiative. *Frontiers in Bioengineering and Biotechnology*, 3:19, 2015.

[31] Frank T. Bergmann, Richard Adams, Stuart Moodie, Jonathan Cooper, Mihai Glont, Martin Golebiewski, Michael Hucka, Camille Laibe, Andrew K. Miller, David P. Nickerson, Brett G. Olivier, Nicolas Rodriguez, Herbert M. Sauro, Martin Scharm, Stian Soiland-Reyes, Dagmar Waltemath, Florent Yvon, and Nicolas Le Novère. COMBINE archive and OMEX format: one file to share all information to reproduce a modeling project. *BMC Bioinformatics*, 15(1):369, 2014.

[32] Nicolas Le Novère, Benjamin Bornstein, Alexander Broicher, Mélanie Courtot, Marco Donizelli, Harish Dharuri, Lu Li, Herbert Sauro, Maria Schilstra, Bruce Shapiro, Jacky L. Snoep, and Michael Hucka. BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Research*, 34(Database issue):689–691, 01 2006.

[33] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Research*, 44:457–462, 2016.

[34] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter DEustachio. The Reactome Pathway Knowledgebase. *Nucleic Acids Research*, 46:649–655, 2018.

[35] Huaiyu Mi and Paul Thomas. *PANTHER Pathway: An Ontology-Based Pathway Database Coupled with Data Analysis Tools*, pages 123–140. Humana Press, 2009.

[36] 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.

[37] Martina Kutmon, Martijn P. van Iersel, Anwesha Bohler, Thomas Kelder, Nuno Nunes, Alexander R. Pico, and Chris T. Evelo. PathVisio 3: An Extendable Pathway Analysis Toolbox. *PLOS Computational Biology*, 11:1–13, 02 2015.

[38] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13:2498–2504, 2003.

[39] *SBW - A Modular Framework for Systems Biology*, 12 2006.

[40] Finja Bchel, Clemens Wrzodek, Florian Mittag, Andreas Drger, Johannes Eichner, Nicolas Rodriguez, Nicolas Le Novre, and Andreas Zell. Qualitative translation of relations from BioPAX to SBML qual. *Bioinformatics*, 28(20):2648–2653, 2012.

[41] Clemens Wrzodek, Andreas Drger, and Andreas Zell. KEGGtranslator: visualizing and converting the KEGG PATHWAY database to various formats. *Bioinformatics*, 27(16):2314–2315, 2011.

[42] Martijn P. van Iersel, Alice C. Villger, Tobias Czauderna, Sarah E. Boyd, Frank T. Bergmann, Augustin Luna, Emek Demir, Anatoly Sorokin, Ugur Dogrusoz, Yukiko Matsuoka, Akira Funahashi, Mirit I. Aladjem, Huaiyu Mi, Stuart L. Moodie, Hiroaki Kitano, Nicolas Le Novre, and Falk Schreiber. Software support for SBGN maps: SBGN-ML and LibSBGN. *Bioinformatics*, 28(15):2016–2021, 2012.

[43] Hanif Yaghoobi, Siyamak Haghipour, Hossein Hamzeiy, and Masoud Asadi-Khiavi. A review of modeling techniques for genetic regulatory networks. *Journal of Medical Signals and sensors*, 2(1):61, 2012.

[44] Ren Thomas. Regulatory networks seen as asynchronous automata: A logical description. *Journal of Theoretical Biology*, 153(1):1–23, 1991.

[45] Wim Bos. Modeling biological systems using Petri nets. 2008.

[46] Venkatramana N Reddy, Michael L Mavrovouniotis, Michael N Liebman, et al. Petri net representations in metabolic pathways. *ISMB*, 93:328–336, 1993.

[47] Luca Grieco, Laurence Calzone, Isabelle Bernard-Pierrot, Franois Radvanyi, Brigitte Kahn-Perls, and Denis Thieffry. Integrative Modelling of the Influence of MAPK Network on Cancer Cell Fate Decision. *PLOS Computational Biology*, 9(10):1–15, 10 2013.