HAL
archives-ouvertes.fr

Uncovering operational interactions in genetic networks using asynchronous boolean dynamics

Laurent Tournier, Madalena Chaves

▶ **To cite this version:**

Laurent Tournier, Madalena Chaves. Uncovering operational interactions in genetic networks using asynchronous boolean dynamics. [Research Report] RR-6703, INRIA. 2008. <inria-00325914v2>

HAL Id: inria-00325914
https://hal.inria.fr/inria-00325914v2

Submitted on 23 Oct 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Uncovering operational interactions in genetic networks using asynchronous boolean dynamics

Laurent Tournier — Madalena Chaves

N° 6703

August 2008

Thème BIO

ISSN 0249-6399

Uncovering operational interactions in genetic networks using asynchronous boolean dynamics

Laurent Tournier* , Madalena Chaves†

Thème BIO — Systèmes biologiques
Projet Comore

Rapport de recherche n° 6703 — August 2008 — 29 pages

Abstract: To analyze and gain intuition on the mechanisms of complex systems of large dimensions, one strategy is to simplify the model by identifying a reduced system, in the form of a smaller set of variables and interactions that still capture specific properties of the system. For large models of biological networks, the diagram of interactions is often well represented by a Boolean model with a family of logical rules. The state space of a Boolean model is finite, and its asynchronous dynamics are fully described by a transition graph in the state space. In this context, a method will be developed for identifying the active or operational interactions responsible for a given dynamic behaviour. The first step in this procedure is the decomposition of the asynchronous transition graph into its strongly connected components, to obtain a "reduced" and hierarchically organized graph of transitions. The second step consists of the identification of a partial graph of interactions and a sub-family of logical rules that remain operational in a given region of the state space. This model reduction method and its usefulness are illustrated by an application to a model of programmed cell death. The method identifies two mechanisms used by the cell to respond to death-receptor stimulation and decide between the survival or apoptotic pathways.

Key-words: Boolean networks, Asynchronous transition graph, Model reduction, Biological networks.

* Laurent.Tournier@sophia.inria.fr
† Madalena.Chaves@sophia.inria.fr

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Identification d'interactions opérationnelles dans des modèles booléens asynchrones de réseaux génétiques

Résumé : Pour appréhender la dynamique d'un système complexe de grande dimension, il est souvent utile, lorsque cela est possible, d'identifier un système réduit, comprenant un sous-ensemble de variables et d'interactions, qui capture les principales propriétés dynamiques du système initial. Dans la modélisation de grands réseaux de régulation provenant de la biologie cellulaire, comme par exemple des réseaux génétiques, l'utilisation de systèmes dynamiques discrets, voire booléens, basés sur des graphes d'interactions entre les différentes variables est intéressante. Ces systèmes ont notamment un ensemble d'états fini, et leurs dynamiques, synchrone ou asynchrone, peut être elle aussi représentée par une structure finie, sous la forme d'un graphe de transition entre états. Dans ce contexte, nous développons ici une méthode visant à identifier, dans un système booléen asynchrone, le sous-ensemble d'interactions actives, ou opérationnelles, responsables d'un comportement dynamique donné. La première étape de cette procédure consiste en l'application de la décomposition en composantes fortement connexes du graphe de transition asynchrone, permettant ainsi d'obtenir un graphe de transition "réduit" et hiérarchisé. A partir de ce graphe, la seconde étape consiste en l'identification d'une sous-famille de règles logiques qui restent opérationnelles dans une région donnée de l'espace d'états. La praticabilité et l'utilité de cette méthode sont illustrées par un système de dimension 12 modélisant la mort programmée de la cellule (apoptose). En particulier, la méthode permet d'identifier les deux mécanismes utilisés par la cellule pour répondre à une stimulation de certains récepteurs, et décider entre la survie ou la mort programmée.

Mots-clés : Réseaux booléens, Graphe de transition asynchrone, Réduction de modèles, Réseaux biologiques.

1 Introduction

Discrete and, in particular, Boolean models have been playing an increasingly important role in the study and analysis of complex biological systems [3,6,7,12,25–27]. Based on the idea that each variable may take values only on a finite set, discrete models offer a very attractive framework for the systematic study of the dynamics of large systems, which may range from a few to hundreds of variables and their interactions.

The discrete modelling approach is highly relevant for many of the currently data acquisition techniques for signalling and genetic regulatory networks (microarrays, fluorescence markers, electrophoretic mobility shift assays, etc.) which involve more qualitative measurements. Messenger RNAs and proteins are frequently described as weakly/strongly expressed, and concentrations are reported to increase/decrease by a factor of N relative to a given reference value. Discrete ranges of values appropriately describe these type of data, and a discrete system may be expected to give a good idea of the system's dynamics from the available data (for example, on multistationary, stability, or oscillatory behavior). At the same time, since this dynamical information is essentially independent of the system's parameters (such as kinetic rates, binding rates, or degradation constants) and depends only on the interconnection structure, discrete networks provide a measure of the robustness of a system [6]. For instance, the transition graph of the network indicates how much a given trajectory may be affected by perturbations, or whether the system is capable of maintaining a given dynamical behaviour despite fluctuations in the environment. Indeed, a major advantage of discrete and boolean modelling is the possibility of fully characterizing all *qualitative* dynamical trajectories of a particular network, based simply on the structure of links and interactions between nodes. This general characterization and "easier" handling of the state space, counterbalance the loss of detailed information on time evolution and (more realistic) continuous concentration changes.

The study of complex systems with many variables frequently raises questions concerning the possibility of simplifying or reducing the system in some way. To simplify the analysis and gain intuition, it is often useful to identify a smaller, easier to analyze, family of variables and interactions that still faithfully describe the original system and exhibit the same overall qualitative dynamics. Likewise, it is often of interest to find out whether different groups of variables are associated with different dynamics [29]. Another related question is whether all interactions operate at all times, or whether different groups of interactions become active or operational at different times, in response to a precise context [19]. Similarly, finding interactions which prevent a given target function is useful from the point of view of therapeutical interventions, for instance [20, 26]. These are all challenging problems, and while some model reduction methods exist, they are generally aimed at special classes of systems (a survey of methods used for control systems can be found in [1]; a method for identifying the variables responsible for complex cell behavior was proposed in [29]). One of the objectives of this paper is to show that, to some extent, answers to these questions may be obtained through the discrete systems framework and some of its techniques.

The current work will focus on the analysis of the dynamics of asynchronous Boolean networks. In this class of networks, the variables are assumed to take only two values (0/1, expressed/not expressed), and the order of the variables' updates is assumed to be asynchronous, allowing realistic context-sensitive updating strategies. The whole state space of such networks is easily described and the asynchronous transitions' graph computationally feasible (for "medium" size systems, *e.g.*, 8-20 variables).

In this context, a model reduction technique is proposed in this paper, which is suitable to deal with Boolean networks motivated by biological (amely, signalling or genetic) regulatory networks of intermediate dimension. The model reduction technique combines and adapts two methods: the classical decomposition of a graph into its strongly connected components [8] and an identification algorithm described in [21, 36]. The first part of the model reduction technique involves the simplification of the asynchronous transition graph into its strongly connected components. These components are then organized into hierarchical levels, such that any given trajectory can only move into the next level in the hierarchy, but never into the previous level. A new "reduced" transition graph is then constructed which describes the transitions between the strongly connected components (Section 3). The second part of the model reduction involves the identification of the "operational" network (active interactions) that is responsible for the dynamics of the system from a given level in the hierarchy (Section 4).

Finally, in Section 5, we show that knowledge on reaction rates can also be incorporated into the discrete model by means of probabilities, in order to obtain a more realistic description of the system that still retains the simplicity of discrete networks. The system's reaction rates can be used to stipulate *classes* of updating strategies (or updating orders) in the asynchronous transition graph. One possible class of updating strategies consists of associating a (fixed) matrix of transition probabilities to the graph edges, a process which corresponds to generating a Markov chain. Several relevant quantities can then be computed, such as the expected times for convergence to a given attractor.

The methods proposed above are illustrated by an application to an apoptosis (or programmed cell death) network [7, 28], with $n = 12$, as described in Section 2.3. The dynamics of the network in response to death-receptor stimulation is studied, and two core groups of variables and pathways are identified: it is shown that these correspond to two mechanisms responsible for the decision between programmed cell death or cell survival. In addition, associating a transition probabilities' matrix to the apoptosis network allows us to estimate, among other quantities, the probability of cell survival or death upon stimulation of death receptors.

2 Asynchronous boolean models of gene regulatory networks

Discrete dynamical systems have been widely studied for decades, as they provide a good mathematical and algorithmic framework to model systems where variables are known or assumed to take values only in a finite set (as opposed to a continuum of values). In particular, the discrete framework has been often applied to model biological regulatory networks, such as gene networks [18, 33]. The mathematical basis of discrete models consists of a finite set of discrete variables (sometimes boolean) that interact with one another through discrete *activation functions*. Usually, these interactions are comprised in a (finite) directed graph, called *interaction graph*. This graph, together with the family of activation functions, define the structure of a discrete system. Each variable will evolve according to a given rule, constructed from the interaction graph. In order to describe the dynamics of such a system, over a discrete time, one defines an *operating mode*, by giving a *strategy* that determines the updating order of the variables over time.

There exist two main operating modes studied in the literature. The first one is the *synchronous* strategy, where all variables are simultaneously updated at each discrete instant (see [18] for an extensive study of this synchronous strategy; see also [3, 30, 37]). The dynamics implied by the synchronous strategy presents some nice mathematical properties (mainly, the transition graph is deterministic) that allow one to simulate high-dimensional networks, randomly generated, in order to find statistically relevant types of dynamical behavior [18]. However, if one wants to model a given biological system in a more realistic manner, the synchronous updating strategy may be quite a strong assumption, poorly related to the reality. This is why other approaches have been proposed, by developing *asynchronous* strategies, where discrete variables are updated in a heterogeneous way over time. Discrete networks with asynchronous updating orders are often called Thomas' networks [33–35], and are much better suited to model the dynamical behavior of biological regulatory networks. Before giving more details on the type of asynchronicity that will be used in this paper, we first recall some basic definitions about the structure of discrete networks.

2.1 Structure of a boolean network

In this paper, we will consider *boolean* networks, where the variables (which represent, for instance, the level of expression of genes, or the level of concentration of different species, such as proteins) can take only two qualitative values. "0" represents a basal level (inhibition -or weak activation- of the transcription of a gene, or *absence*[1] of a biochemical species) and "1" represents a high level (activation of the transcription of a gene, or *presence* of a species). We note here that there exist more general frameworks, where the variables can take more than two qualitative values (see for instance [5, 12, 14] or [31]). This choice is discussed below.

As most of the work on discrete or boolean gene networks is based on the same mathematical objects (with slightly different definitions), the following part is only a brief summary of the main definitions and notations that will be used in the rest of this paper (for a detailled explanation of these definitions, one can refer to the

[1] generally, a given species is considered to be *absent* if its concentration is lower than a given threshold, *present* otherwise

extensive literature on discrete networks). Let us begin with the definition of the interaction graph, which is the core of the structure of a discrete model.

Definition 1 (Interaction Graph) *The interaction graph of a n-dimensional boolean network is defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the set of nodes (each node may represent a biological species) and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of directed edges (representing the interactions between these species). The edge (v_j, v_i) exists if node v_j influences node v_i (e.g. v_j activates or inhibits v_i).*

Each node $v_i \in \mathcal{V}$ has a set of inputs (possibly empty), which are the nodes that influence its evolution:

$$\mathcal{I}(v_i) = \{v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\} \subset \mathcal{V}.$$

For each $v_i \in \mathcal{V}$, the cardinality of the set $\mathcal{I}(v_i)$ (the number of its inputs) is often called the *connectivity* of node v_i and is generally denoted by k_i. In order to give a complete definition of the structure of the network, we now define the activation functions.

Definition 2 *The structure of a n-dimensional boolean network is defined by an interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ together with a collection $\mathcal{F} = \{f_i : i = 1, \ldots, n\}$, of boolean functions:*

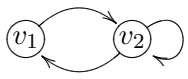
$$f_i : \{0,1\}^{k_i} \longrightarrow \{0,1\},$$

where, for each $i \in \{1, \ldots, n\}$, f_i designates the activation function of node v_i, and k_i its number of inputs.

Let x_i denote the boolean variable associated with node v_i. The updated value of x_i, denoted by x_i' is therefore given by:

$$x_i' = f_i\left(x_{i_1}, \ldots, x_{i_{k_i}}\right), \quad \text{where: } \left\{v_{i_1}, \ldots, v_{i_{k_i}}\right\} = \mathcal{I}(v_i).$$

To illustrate these definitions, let us consider a simple 2-dimensional network, given by the following interaction graph and set of rules:



$$\begin{cases} f_1(x_2) & = & \overline{x_2}, \\ f_2(x_1, x_2) & = & \text{xor}(x_1, x_2) = (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}). \end{cases}$$

The notations used in this example are classical in Boole algebra: if x and y denote two boolean variables, $\overline{x}$ denotes the negation of x, and $x \wedge y$, $x \vee y$ denote, respectively, the product (logical function and) and summation (logical function or) of x and y. The symbol xor denotes the exclusive or. In this example, the updated values of x_1 and x_2 are thus:

$$\begin{cases} x_1' & = & \overline{x_2}, \\ x_2' & = & (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}). \end{cases}$$

Remark 1 *In the study of discrete models of biochemical networks, the arrows of the interaction graph are usually signed, indicating whether the arrow represents an* activation *($\xrightarrow{+}$ or $\rightarrow$) or an* inhibition *($\xrightarrow{-}$ or $\dashv$). It is to be noted that, for a general network given by Def. 2, it is not always possible to associate a sign to each arrow in an unequivocal manner. In the previous example, for instance, the arrows (v_1, v_2) and (v_2, v_1) cannot be signed as the interactions they represent are neither activations nor inhibitions. Nevertheless, boolean networks constructed from the biological description of a particular biological system can often be signed unequivocally. We will thus adopt this signing convention in the following.*

Remark 2 *In the above approach, connectivity is defined prior to the activation functions. This can lead to inaccuracies if the real connectivity (as defined in [36]) of the function does not match its apparent connectivity. Thus, the function $f(x_1, x_2) = (x_1 \wedge x_2) \vee (x_1 \wedge \overline{x_2})$ has an apparent connectivity of 2, whereas its real connectivity is 1 (indeed $f(x_1, x_2) = x_1$). In the terminology of [30], x_2 is called a fictitious (or non essential) variable for function f. This issue becomes particularly important if one wants to identify a network from given data. It will be addressed in more detail in Section 4.*

2.2 Synchronous vs asynchronous dynamics

Consider a n-dimensional network given by $\mathcal{N} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ (see Def. 2). The state space of $\mathcal{N}$ is the set $\Omega = \{0,1\}^n$ whose cardinality is 2^n. As the state space is finite, one can represent the discrete dynamical behavior of the network with a finite directed graph, called *transition graph*. In order to define it properly, we need to assign an operating mode (or updating strategy) for the network $\mathcal{N}$. From a mathematical point of view, an operating mode is defined as a sequence $\{\Psi(t) \subset \mathcal{V} \mid t \in \mathbb{N}\}$, where, for each $t \in \mathbb{N}$, $\Psi(t)$ is a subset of $\mathcal{V}$ indicating which nodes are updated at time t. Therefore, if the state of the network at time t is given by the boolean vector:

$$X(t) = (x_1(t), \ldots, x_n(t)) \in \Omega,$$

then the next state $X(t+1)$ (also called the successor) is computed as follows:

$$X(t+1) = (x_1(t+1), \ldots, x_n(t+1)) \in \Omega,$$

$$\text{where } \begin{cases} x_i(t+1) = x_i(t) & \text{if } v_i \notin \Psi(t+1), \\ x_i(t+1) = x'_i(t) & \text{if } v_i \in \Psi(t+1). \end{cases}$$

In the synchronous approach, all nodes are simultaneously updated at each time t, which can be written as:

$$\forall t \in \mathbb{N}\,,\ \Psi(t) = \mathcal{V}.$$

In this case, the temporal evolution of the network is *autonomous*[2], in the sense that any vector $X \in \Omega$ has a (unique) successor $F(X) = (f_1(X), \ldots, f_n(X))$, and that successor is independent of time t. We can then construct a directed transition graph: its set of nodes is Ω and its set of directed edges is defined by the "successor" function. The main property of the synchronous graph is that it is *deterministic, i.e.* each state has a unique successor (see [12]). In particular, this property implies that each connected component of the graph contains a unique attractor, and this attractor is either a cycle or a fixed point (see illustrating example below). More precisely, the connected components are in fact the basins of attraction of their attractor.

Remark 3 *It can be shown that, in general, if the updating strategy is independent of the time ("context free"), then the dynamics can be brought back to an autonomous dynamical system, with a deterministic transition graph. For instance, this is the case when only one node is updated at each time t, but always in a predefined fixed order (see [36]).*

As previously said, the synchronous updating strategy is a very strong assumption, not very realistic if one wants to model the dynamical behavior of a given biological system. Actually, as discrete interactions are coarse-grained models of sometimes very complex biochemical processes (often implying several biochemical reactions), it is preferable to consider context-sensitive, asynchronous updating strategies.

In order to give a precise definition of an asynchronous transition graph, we first introduce the following notation:

- For each $X \in \Omega = \{0,1\}^n$, $F(X) \in \Omega$ designates the synchronous successor of X:

$$\forall i \in \{1, \ldots, n\}\,,\ F_i(X) = x'_i.$$

- For each $X \in \Omega$ and each $i \in \{1, \ldots, n\}$, $\widetilde{X}^i$ designates the vector:

$$\widetilde{X}^i = (x_1, \ldots, x_{i-1}, \overline{x_i}, x_{i+1}, \ldots, x_n) \in \Omega.$$

- For each $X \in \Omega$, let $U(X) = \{v_i \in \mathcal{V} \mid x_i \neq x'_i\} \subset \mathcal{V}$. $U(X)$ designates the (possibly empty) set of nodes that can actually be updated when the system is in the state X.

We also state the following assumptions, that we will suppose verified throughout this paper:

[2] this is directly linked with the concept of autonomous differential equations, in the framework of continuous dynamical systems

Assumption 1 *At each discrete time t, at most one node is updated (no update means the network is in a steady state).*

Assumption 2 *Each state $X \in \Omega$ such that $X \neq F(X)$ has exactly $|U(X)|$ successors.*

The first assumption forbids the simultaneous update of several nodes (which is reasonable from the biological point of view), whereas the second one implies that every possible update is taken into account (*i.e.* if at state X the node v_i is liable to change, then that update *must* be present in the transition graph). With these assumptions, we can now define the transition graph:

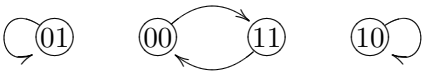
Definition 3 (Asynchronous Transition Graph) *The asynchronous transition graph of the network $\mathcal{N} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ is the directed graph $G = (V, E)$ where the set of nodes V is the state space $\Omega = \{0,1\}^n$ and the set of directed edges E is defined by:*

$$E = \left\{ \left(X \to \widetilde{X}^i \right) \mid X \in \Omega, v_i \in U(X) \right\}.$$

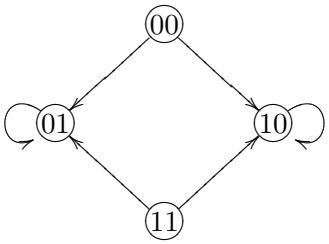
As an illustrating example, let us consider the 2-dimensional network:

$$\mathcal{G}: \quad \overline{v_1} \rightleftarrows \overline{v_2} \qquad \mathcal{F}: \begin{cases} f_1(x_2) = \overline{x_2}, \\ f_2(x_1) = \overline{x_1}. \end{cases}$$

Its synchronous transition graph is:



with three attractors: two steady states and one cycle of length 2 (in this particular case, the basins of attraction are reduced to the attractors themselves). According to Definition 3, its asynchronous transition graph is:



As we can see in this example, the asynchronous transition graph is non deterministic (the states 00 and 11 both have two successors), contrary to the synchronous case. The basins of attraction may contain several attractors (here it is easy to see that the attractors of the system are the two equilibrium points 01 and 10, but the definition of attractors itself needs to be clarified in more complex networks, we will come back to this point in the following part).

In the comparison between synchronous and asynchronous dynamics, a well-known result, illustrated in this example, is that, provided Assumptions 1 and 2 are satisfied, the asynchronous and synchronous steady states are the same. The proof is straightforward, and is not given here.

The non determinism of the asynchronous transition graph is a fundamental property. It allows to consider any possible trajectory implied by the structure of the network. If one wants to study one particular trajectory, then a particular path in the graph has to be chosen, which is equivalent to the choice of a particular updating strategy. Considering biological applications, such a choice will be based on the information available on the system. Unfortunately, the knowledge of the system is often incomplete. An advantage of this framework is the possibility to test and analyze different plausible *sets* of updating orders, as will be done later in this paper (it corresponds to the notion of *priority classes* developed in [12]). The main advantage of the asynchronous graph is that it comprises all the possible choices in a finite structure, which allows to find general dynamical properties valid whatever the updating strategy. Obviously, although finite, the size of the transition graph

grows exponentially with the dimension of the system (in the boolean case, its size is exactly 2^n). This limits the use of general graph algorithms to relatively low dimensional systems (on the order of $n = 10$-20), with respect to the synchronous case, where the dimension of the system under study can be higher [37].

Before presenting the general methodology, let us make a final remark about boolean and more general discrete models of biological systems. In the literature on discrete systems, in general, discrete variables may take more than two values (see *e.g.* [5, 12, 14, 31]). From a biological point of view, the use of discrete variables allows representing *several qualitative values* of an intrinsically continuous biological variable, such as the concentration of a species, or the transcription rate of a gene. As already evoked, the discrete modelling process consists, roughly, in assigning a threshold value to each continuous variable, and then considering only the relative position of the variable with respect to the threshold, instead of its exact, real, value. The principal advantage of using qualitative values is that the interactions among variables will also be represented with discrete functions, taking values over finite sets. However, when a variable x influences two different groups of variables in different ways, then it is more appropriate to assign two different threshold values to the variable x. If there are two different thresholds, say $\theta_1 < \theta_2$, then the qualitative value $\tilde{x}$ of x is no longer boolean, but belongs to $\{0, 1, 2\}$:

$$\tilde{x} = \begin{cases} 0 & \text{if} \quad 0 \leq x < \theta_1 \\ 1 & \text{if} \quad \theta_1 < x < \theta_2 \\ 2 & \text{if} \quad x > \theta_2. \end{cases}$$

Such a generalized discrete framework is appealing from a modelling point of view, but can actually be integrated into the boolean framework with the following observation (initially described in [35]). Define $\tilde{x}^1$ and $\tilde{x}^2$ by:

$$\tilde{x}^1 = \begin{cases} 0 & \text{if} \quad 0 \leq x < \theta_1 \\ 1 & \text{if} \quad x > \theta_1, \end{cases} \qquad \tilde{x}^2 = \begin{cases} 0 & \text{if} \quad x < \theta_2 \\ 1 & \text{if} \quad x > \theta_2. \end{cases}$$

By definition, these two variables are boolean and contain all the information in $\tilde{x}$, provided the following boolean constraints are imposed:

$$\tilde{x}^1 = 0 \;\Rightarrow\; \tilde{x}^2 = 0\,, \qquad \tilde{x}^2 = 1 \;\Rightarrow\; \tilde{x}^1 = 1.$$

By "decoupling" each discrete variable in this way, that is substituting each discrete variable with q thresholds by q boolean variables with the corresponding boolean constraints, a new completely boolean network (with boolean constraints) is obtained. As can be seen in this example, the dimension of the network increases, as several boolean variables are needed to represent a single, multi-valued, discrete variable. From a mathematical point of view, however, the boolean and the generalized discrete frameworks are therefore equivalent. To take advantage of the tractability of the boolean framework, and of existing results on boolean systems (notably the identification algorithm presented in Section 4), we choose from now on to restrict ourselves to boolean networks, without considering generalized discrete systems.

2.3 Working example: an apoptosis signalling pathway

The model reduction method will be illustrated by application to an apoptosis network (Fig. 1). Apoptosis, or programmed cell death, is a physiological process which allows an organism to remove damaged or unwanted cells in a "clean" and natural way. The signalling pathways leading to apoptosis play fundamental roles in embryonic development and in adult organisms, by maintaining normal cellular homeostasis in organs and other cellular tissues [9]. Malfunctioning apoptotic pathways may lead to various diseases, such as cancer (in this case cells do not die, there is insufficient apoptosis), or immunodeficiency and infertility (in this case too many cells die, there is too much apoptosis) [9].

The apoptosis signalling pathway to be considered in this paper (Fig. 1) is based on the model presented in [7], which is, in fact, a discrete version of a continuous model of apoptosis first developed in [28]. A brief description of the network is provided next, and the reader is referred to [7, 28] and references therein for more details.

The network is composed essentially of a pro-apoptotic and an anti-apoptotic pathway, which are activated by the same signal: stimulation of death receptors by a factor such as Tumor Necrosis Factor α (denoted TNF in

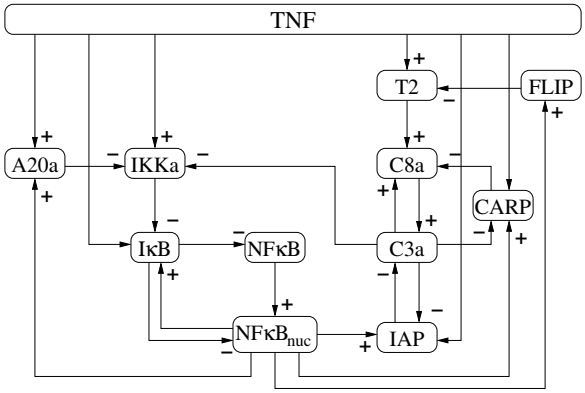


Figure 1: Interaction graph of the simplified model of regulation of apoptosis via the NFκB pathway. As noted in Remark 1, some edges do not have a fixed sign (influence of TNF on IκB, IAP and CARP).

Fig. 1). The pro-apoptotic pathway is based on the model developed in [10], and consists of a family of proteins called caspases, represented by active caspases 3 and 8 (resp., C3a and C8a) in Fig. 1. The caspases play the main role in apoptosis, as they cleave (or break into small pieces) the principal proteins in the cell, eventually leading to a "clean disposal" of the cell in response to an apoptotic signal. The anti-apoptotic pathway is based on the pioneering work of [17] and on the models developed in [17, 22], and represents the Nuclear Factor κB (NFκB) signalling pathway. It is well known that the NFκB signalling pathway is responsible for activating transcription of both pro- and anti-apoptotic genes [13, 23], and thus plays an important role in regulating apoptosis. The components of the NFκB pathway are as follows (in biological terminology): Nuclear Factor κB in the cytoplasm (NFκB) and in the nucleus (NFκB$_{\text{nuc}}$); inhibitor of NFκB (IκB); inhibitor of IκB kinases (IKKa); inhibitor of apoptosis proteins (IAP); caspase-8 and -10-associated RING proteins (CARP); a protein associated with inhibition of complex T2 (FLIP); and a protein regulating IKK activity (A20a).

Binding of TNF to a death receptor activates the anti-apoptotic pathway and, after a certain delay (upon formation of a second complex, denoted by T2), the pro-apoptotic pathway is also activated. The anti-apoptotic pathway activates synthesis of various proteins (IAP, CARP, FLIP) that will contribute to inhibit and regulate the caspases. Therefore, TNF stimulation triggers two opposite effects: activation and inhibition of caspases. The dynamics of the pro- and anti-apoptotic pathways, as well as the interconnections between them, will ultimately lead to a decision between cell death or cell survival. An abundance of active caspases (such as C3a) together with a low concentration of IAP typically leads to cell death. In contrast, a high concentration of IAP and a low level of active caspases typically characterizes a living cell (in this case, enough molecules IAP are present to down-regulate the level of active caspases). In the network represented in Fig. 1, and in particular in its corresponding boolean model (Table 1), steady states corresponding to cell death should satisfy C3a $= 1$ and IAP $= 0$, while steady states corresponding to cell survival should satisfy C3a $= 0$ and IAP $= 1$.

The boolean model in Table 1 has been slightly simplified from that in [7], namely the mRNAs have been removed and only the corresponding proteins nodes are represented. This does not affect the overall dynamics, but reduces the number of variables to facilitate the use of the asynchronous algorithms. The analysis of the transition graph of the boolean network will allow us to study the dynamics of the system, in particular the effect of the structure of the network in creating and/or maintaining a balance between the pro- and anti-apoptotic pathways, and ultimately the decision between death or survival.

3 Hierarchical organization of the asynchronous transition graph

In this section, a general methodology to analyze the asynchronous transition graph of a boolean network is presented. This methodology is based on different algorithms that are classical in the field of graph theory (mainly the strongly connected components decomposition and the topological sort). One can refer to [8] for a detailled analysis of these algorithms.

Node	Boolean rule
TNF	TNF (input of the whole system)
T2	$\mathrm{TNF} \wedge \overline{\mathrm{FLIP}}$
IKKa	$\mathrm{TNF} \wedge \overline{\mathrm{A20a}} \wedge \overline{\mathrm{C3a}}$
NFκB	$\overline{\mathrm{I}\kappa\mathrm{B}}$
NFκB$_{\mathrm{nuc}}$	$\mathrm{NF}\kappa\mathrm{B} \wedge \overline{\mathrm{I}\kappa\mathrm{B}}$
IκB	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{IKKa}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{IKKa}})]$
A20a	$\mathrm{TNF} \wedge \mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}}$
IAP	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{C3a}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{C3a}})]$
FLIP	$\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}}$
C3a	$\overline{\mathrm{IAP}} \wedge \mathrm{C8a}$
C8a	$\overline{\mathrm{CARP}} \wedge (\mathrm{C3a} \vee \mathrm{T2})$
CARP	$[\mathrm{TNF} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \wedge \overline{\mathrm{C3a}})] \vee [\overline{\mathrm{TNF}} \wedge (\mathrm{NF}\kappa\mathrm{B}_{\mathrm{nuc}} \vee \overline{\mathrm{C3a}})]$

Table 1: Boolean rules for the apoptosis network depicted in Fig. 1. See explanation of variables in the text. Note that the variable TNF can be considered an input of the system, as its activation function is $\mathrm{TNF}' = \mathrm{TNF}$.

The current method has been specifically designed to handle boolean models of biological genetic regulatory networks, where the type of knowledge is qualitative (a gene is either expressed or not) rather than quantitative. The method provides answers to many biological issues raised by this type of systems, such as the existence and characterization of attractors, reachability or controllability of a given state, or more general "robust" properties which depends strongly on the structure of the network. Some of the general results presented here are related to results of [5, 12, 14].

3.1 SCC decomposition and hierarchical organization

The notion of hierarchical organization of a directed graph (or *digraph*) relies on the well known strongly connected components (SCC) decomposition algorithm.

Let us recall some basics about digraphs (see [8] for more details). Let $G = (V, E)$ be a digraph. Two vertices $u, v \in V$ are *mutually reachable* (denoted $u \sim v$) if and only if there exist two (directed) paths ρ and ρ' such that ρ joins u to v and ρ' joins v to u. This relation is clearly an equivalence relation on the set V of vertices. The *strongly connected components* of the digraph G are then defined as the elements of $V/\sim$, that is to say the equivalence classes of the relation $\sim$. In other words, a strongly connected component of G is a maximal set of vertices $C \subseteq V$ such that for every pair $u, v \in C$, u and v are reachable from each other.

The SCC *decomposition* of a digraph G consists in computing the strongly connected components of G: $C_1, \ldots, C_p$ and then to compute the digraph $G^{scc} = (V^{scc}, E^{scc})$ defined as follows:

- $V^{scc} = \{C_1, \ldots C_p\}$,
- given $1 \leq i, j \leq p$, the directed edge (C_i, C_j) belongs to E^{scc} if and only if there are $u \in C_i$ and $v \in C_j$ such that $(u, v) \in E$.

It can be easily proved (see [8]) that the digraph G^{scc} contains no (oriented) cycles. It is called a *dag* (for directed acyclic graph). This is a key property of G^{scc}, because every dag can be *topologically sorted* (see [8], section 22.4). A topological sort of a dag can be viewed as a classification of its vertices in several hierarchical levels $H_1, H_2, \ldots$ such that the vertices of the first level H_1 are vertices with no predecessors, and the predecessors of vertices of level H_i, $i > 0$, are contained in inferior levels H_j with $j < i$ (see Fig. 2). The decomposition and hierarchical organization of a digraph G can be computed in linear time with respect to the number of vertices and edges of G [8].

The main interest of this hierarchical organization, applied to the asynchronous transition graph of a boolean network, is that, whatever path we choose in the graph (*i.e.* whatever updating order we choose

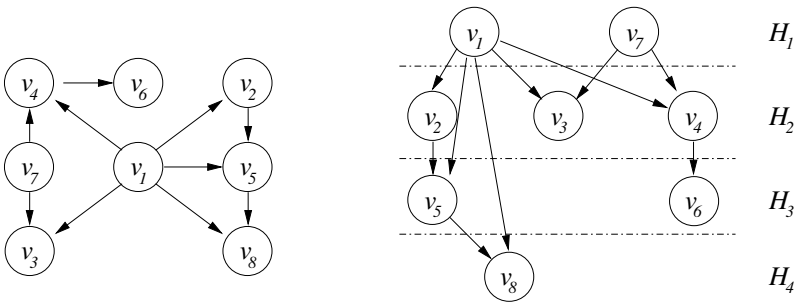


Figure 2: A dag (on the left) and its topological sort (on the right), with four hierarchical levels.

for the variables), once the path leaves a hierarchical level H_i, it cannot return to this level. So, any path will travel "down" the hierarchical levels: $H_{i_1} \rightarrow H_{i_2} \rightarrow \dots$ (with $i_1 < i_2 < \dots$). Due to this property, we can now give a precise definition of the term *attractor* for a boolean network evolving according to an asynchronous strategy.

Definition 4 (Attractor) *Let $\mathcal{N}$ be a boolean network. An SCC $c^* \in V^{scc}$ that has no successor in G^{scc} is called an (asynchronous) attractor of $\mathcal{N}$.*

In graph theory, such SCCs are often called *terminal* SCCs. In other words, the asynchronous attractors of a boolean network are the strongly connected components of the transition graph that cannot be escaped by the system, whatever the updating strategy.

However, it should be noted that it is still possible to construct specific asynchronous updating strategies such that the system gets "stuck" in a non terminal SCC. Indeed, for any SCC that contains at least two states, it is obvious that we can find a particular strategy that allows the system to remain indefinitely in this component (see Fig. 3 for an illustration). Nevertheless, such intermediate SCCs will not be considered as attractors in this paper, as we seek general dynamical properties that are valid for all the choices of updating rules.

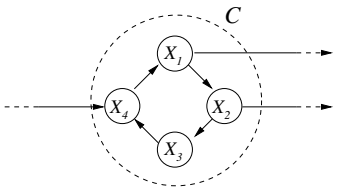


Figure 3: Example of an intermediate SCC in the transition graph. It is possible to find an updating order of the variables so that the system runs indefinitely through the cycle $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$. However, as C has successors (through X_1 and X_2), there exist updating orders that allow the system to quit C, therefore C will not be called an attractor.

The hierarchical organization of the transition graph allows the formulation of simple algorithmic definitions of attraction and reachability sets. Let $c \in V^{scc}$ be a SCC of the transition graph. Algorithm 1 computes both the attraction and the reachability sets of c (as V^{scc} has p elements, SCCs are represented by integers lying in $\{1, \dots, p\}$).
These algorithms lead to the following definition.

Definition 5 *Let c be a SCC of the asynchronous transition graph. The sets $\mathcal{A}(c)$ and $\mathcal{R}(c)$ computed by Algorithm 1 are, respectively, the attraction set of c (i.e. the set of all SCCs that can lead to c) and the reachability set of c (i.e. the set of all SCCs that can be reached from c). If c is an attractor of the network (in other words, if $\mathcal{R}(c) = \{c\}$), the set $\mathcal{A}(c)$ is its basin of attraction.*

These definitions of attraction and reachability must be understood "in a broad sense". Indeed, Definition 4 of an attractor c^* is *strong*, in the sense that no trajectories are allowed to move out of c^*. In contrast, an attractor

Algorithm 1 - Computation of attraction and reachability sets in G^{scc}.

`Input:` $c \in \{1, \ldots, p\}$ (a SCC of the asynchronous transition graph).
`Output:` $\mathcal{A}(c), \mathcal{R}(c)$: attraction and reachability sets of c.
1: $\mathcal{A}(c) :=$ ATTR(c)
2: $\mathcal{R}(c) :=$ REACH(c)

where ATTR and REACH are two simple recursive functions:

1: **ATTR**(c):
2: $A := \{c\}$
3: $P :=$ predecessors(c)
4: **if** $P \neq \varnothing$ **then**
5: **for all** $\gamma \in P$ **do**
6: $A := A \cup$ ATTR(γ)
7: **end for**
8: **end if**
9: **return** A

1: **REACH**(c):
2: $R := \{c\}$
3: $S :=$ successors(c)
4: **if** $S \neq \varnothing$ **then**
5: **for all** $\gamma \in S$ **do**
6: $R := R \cup$ REACH(γ)
7: **end for**
8: **end if**
9: **return** R

The functions $predecessors$ and $successors$ return, respectively, the -possibly empty- sets of immediate predecessors and successors of a node in the dag G^{scc}. As explained in the text, if the node $\gamma \in V^{scc}$ belongs to a hierarchical level H_i, then its predecessors (resp. its successors) can only lie in hierarchical levels H_j with $j < i$ (resp. with $j > i$).

such as $X_1 \to X_2 \to X_3 \to X_4$ in Fig. 3 would be a *weak* attractor, since some trajectories may move out of the cycle. These notions of *strong* and *weak* attractors are remniscent of those found in [4], in the context of equilibria of differential inclusions. In [4] a set of equilibria is said to be weakly asymptotically stable if there is *at least one solution* of the differential inclusion for which the set is asymptotically stable in the classical sense. The strong notion holds if the set is asymptotically stable in the classical sense *for every solution* of the differential inclusion. In a discrete graph, if $a_1, \ldots, a_r$ designate the attractors of the network, an element $c \in \mathcal{A}(a_1)$ may lead to attractor a_1 (*i.e.*, there exists an updating strategy such that c leads to a_1). If one wants the basin of attraction of a_1 in a strict sense, that is, the set of SCCs that *always* lead to a_1 (whatever the updating order), then one has to compute the set:

$$\mathcal{A}^s(a_1) = \mathcal{A}(a_1) \setminus \left(\bigcup_{i=2}^{r} \mathcal{A}(a_i) \right),$$

which may in some cases be reduced to the singleton $\{a_1\}$.

3.2 Application to the apoptosis network

The SCC decomposition and hierarchical organization were applied to the NFκB signalling pathway described in Section 2.3. The results presented here were obtained with codes implemented in `Matlab`. Following the `matlab_bgl`[3] library specifications, the graphs are represented with sparse matrices, which allow a quite efficient implementation.

We recall that the system under study is of dimension $n = 12$, and that one particular variable, TNF, is an input (*i.e.* its activation function is TNF$' =$ TNF). The state space is $\Omega = \{0, 1\}^n$, and the size of the asynchronous transition graph G is $2^n = 4096$. The number of strongly connected components is $p = 1472$, therefore the size of the graph G^{scc} is only 40% of the size of G. After the hierarchical organization of this graph, we found only 38 hierarchical levels, and 3 attractors. Fig. 4 represents a scheme of this graph with its main elements.

[3] see `http://www.stanford.edu/~dgleich/programs/matlab_bgl/`

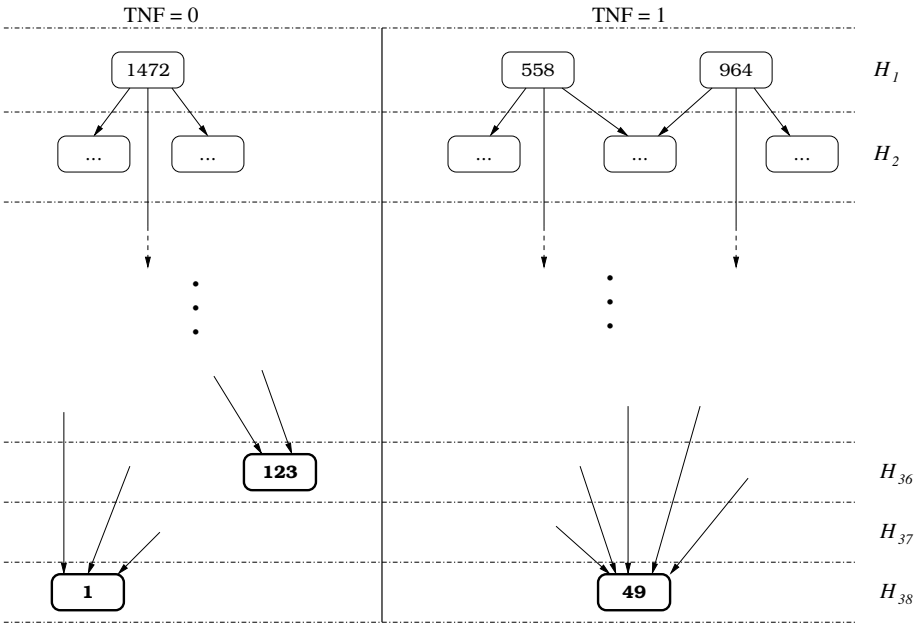


Figure 4: Scheme of the main elements of the hierarchical graph G^{scc} for the apoptosis network. The vertical line separates the two connected components (generated by the input TNF), and the horizontal lines separates the different hierarchical levels. The SCCs are designated by their integer index (between 1 and $p = 1472$). The only SCCs that are represented here are the roots (SCCs that belong to the first hierarchical level) and the attractors (in bold characters).

The fact that TNF is an input implies that its value remains constant, whatever the path in the graph. Mathematically, this means that G^{scc} has two connected components[4]. They will be denoted $\mathcal{T}^0$ (where TNF $=$ 0) and $\mathcal{T}^1$ (where TNF $= 1$). In other words, the components $\mathcal{T}^0$ and $\mathcal{T}^1$ are two subsets of nodes of G^{scc} that are completely separated (there exist no directed edge going from a SCC in $\mathcal{T}^0$ to a SCC in $\mathcal{T}^1$, and vice versa).

Remark 4 *The number of states contained in all the SCCs in $\mathcal{T}^0$ is exactly $2^{n-1} = 2048$ (the same is true for $\mathcal{T}^1$). Actually, it is easy to generalize this: if the system has r inputs, then the asynchronous transition graph (and its SCC graph) will have exactly 2^r connected components. Each of these components, in the transition graph, will contain 2^{n-r} states.*

We found three terminal SCCs in our graph, which means that the system has three different attractors. Using the SCC labels returned by the hierarchization, the SCCs 1 and 123 contain only one state (they are therefore equilibrium points) whereas the third one, 49 is a more complex SCC with 56 states. Table 2 indicates the boolean values taken by the variables within each attractor. The two equilibria belong to $\mathcal{T}^0$ (TNF is absent), the first one corresponds to survival of the cell (the caspases C3a and C8a are absent) and the second one corresponds to the triggering of apoptosis (with activation of the caspases). The complex attractor (SCC 49) belongs to $\mathcal{T}^1$ (TNF is present). As we can see in Table 2, within this attractor the caspases are activated while NFκB, IκB and other factors oscillate. At first, this might seem to indicate that apoptosis will be the final outcome, but, as will be seen later, upon TNF removal, the cell may still choose either the survival or apoptotic equilibria.

In the previous example, the computations of the asynchronous transition graph and of its hierarchical organization take only a few seconds. The implementation of these graphs is based on sparse matrices (see Fig. 5), which makes the storage of data and the run of the different algorithms rather efficient (including the computation of attraction and reachability sets). However, as the size of the transition graph is 2^n, the

[4]The notion of connected component is not to be confused with the notion of *strongly* connected component that we have used so far. A connected component of a digraph G is defined as a SCC (see the previous section) but replacing the directed graph G with a non directed version of G (*i.e.* where the direction of the edges is omitted).

	TNF	T2	IKKa	NFκB	NFκB$_{nuc}$	IκB
attractor 1	0	0	0	0	0	1
attractor 123	0	0	0	0	0	1
attractor 49	1	*	0	*	*	*
	A20a	IAP	FLIP	C3a	C8a	CARP
attractor 1	0	1	0	0	0	1
attractor 123	0	0	0	1	1	0
attractor 49	*	0	*	1	1	0

Table 2: Boolean patterns of the three attractors. The first line is the SCC 1 (equilibrium point that corresponds to cell survival), in the following it will be denoted a_2. The second line is the SCC 123 (equilibrium point that corresponds to the triggering of apoptosis), it will be denoted a_3. The third line is the SCC 49, which contains 56 states: it will designated as "apoptotic oscillations", and will be denoted a_1 in the rest of the paper. The symbol * means that the corresponding variable has no fixed value in the attractor and oscillates between 0 and 1.

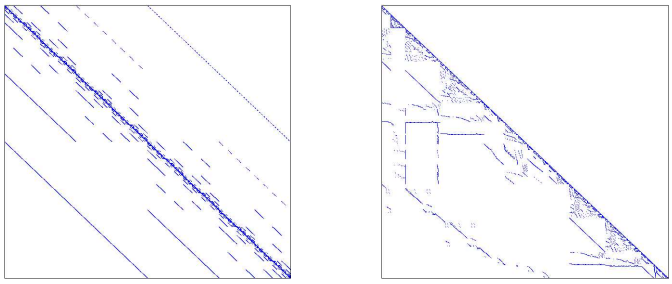


Figure 5: Patterns of the (sparse) adjacency matrices of the asynchronous transition graph (size 4096×4096, on the left) and of the SCC graph (size 1472×1472, on the right). The filling rates of these matrices are respectively: 0.13% and 0.39%. We can visualize the fact that G^{scc} has no cycle, as its adjacency matrix is lower triangular.

time complexity of its construction grows exponentially. Moreover, the non determinism of the graph makes its analysis more difficult than in the synchronous case (in particular in the search of attractors, or attraction basins). At present, with not fully optimized codes, the computation time remains reasonable for dimensions n around 15. This is a major difference with the synchronous framework, where the determinism of the transition graph makes it possible to analyze much higher dimensional systems [37].

The hierarchical organization has the advantage of characterizing the dynamics *for all* possible updating orders. It allows for instance to detect "spurious" behaviors, that may appear when the network get apparently stuck in a non terminal SCC. It also comprises all the possible dynamical behaviors in a finite structure, avoiding generation of large numbers of simulations.

4 Identification of operational interactions

The family of SCC and their transition graph describe a new state space (reduced from 2^n to p states), which characterizes the dynamics of a new, reduced, system. A second stage in our model reduction procedure involves the determination of a set of rules governing this new system. Starting from the simplified asynchronous transition graph G^{scc}, the goal of this part is to reconstruct, as far as possible, the *active*, or *operational* interactions along time (the definition of an operational interaction can be found in the following). Recall that the SCCs are hierarchically organized, in such way that trajectories can go in one sense from one level to another. Thus, the G^{scc} graph is particularly well suited for an identification process which consists, roughly, of finding all the operational interactions from a level l down to the terminal level (*i.e.* down to the possible attractors).

In general, this identification method can be applied to any, hierarchically organized, state space. It can be used to uncover groups of variables (and interactions) that are mainly responsible for the dynamical behavior of the system in a given region of the state space. More precisely, the method identifies groups of interactions responsible for the system's asymptotic behaviour, from a given level in the state space, or within a "self-contained" subgraph.

4.1 Synchronous identification algorithm

The identification technique developed here has been adapted from [21], where an algorithm was proposed for the identification of boolean networks in a synchronous framework. The term *identification* must be understood in a precise sense, related to the boolean structure of the networks under study. Basically, given a family of transition pairs $\{s_i \rightarrow s_j\,,\ \ s_i, s_j \in \Omega\}$, one wants to *reconstruct* the structure of a network, that is: its interaction graph $\mathcal{G}$ and (possibly) its set of boolean rules $\mathcal{F}$. In order to do that, a set of qualitative data is used, that consists in a set of sequences of successive boolean patterns. The algorithm, named REVEAL (for *REVerse Engineering ALgorithm*) is based on the concept of Shannon's entropy (see Definition 6), which is a classical notion in information theory. A detailed proof of its correctness, together with an analysis of its complexity can be found in [36].

The main limitation of this algorithm is that data (typically, time series issued from DNA microarrays), are supposed to be *synchronous*. Indeed, in order to reconstruct the interaction graph, it is assumed that the temporal successions of boolean vectors are sequences $\left(X^t\right)_{t=0,1,\ldots}$ that satisfy:

$$\forall t\,,\ X^{t+1} = F(X^t),$$

where F designates the "synchronous successor" function. From a biological point of view, this is of course a very strong assumption, and it is highly unlikely that all nodes of the network have indeed been updated once and only once between two successive experiments. Therefore, it is unclear whether this algorithm identifies truly the structure of a biological network from experimental data. Nevertheless, as we will show in the next section, it is relevant to adapt this algorithm to the asynchronous case, as useful results that provide biological intuition can still be obtained. Indeed, we will not seek to infer a network from experimental measures, but we will use this algorithm to identify, in an already known network, the sub-network that is operational in a given region of the state space.

In the rest of this section, a brief description of this algorithm is given (the reader is referred to [21, 36] for more details). In the following, we recall that the *successor* function F must be understood in the synchronous sense. In order to lighten the notations and ease the description, we will consider that data are arranged in boolean truth tables. The inputs of the algorithm are thus two boolean matrices In and Out, with q rows (the size of the sample) and n columns (the size of the network). If, for $j \in \{1, \ldots, q\}$, $\mathcal{L}_{In}(j)$ (resp. $\mathcal{L}_{Out}(j)$) designates the j-th rows of In (resp. Out), then vector $\mathcal{L}_{Out}(j) \in \{0,1\}^n$ is the (synchronous) successor of vector $\mathcal{L}_{In}(j)$. In other words, $\mathcal{L}_{Out}(j) = F(\mathcal{L}_{In}(j))$. Moreover, we assume that all rows of In are distinct. A simplified statement of REVEAL is given in Alg. 2.

The notion of Shannon entropy used below is quite classical in the field of information theory, we just recall here a simple definition, adapted to our needs:

Definition 6 (Entropy) *Let M be a $q \times r$ boolean matrix, and let $M_1, \ldots, M_r$ designate its column vectors. The entropy of M is the quantity $H(M)$ defined by:*

$$H(M) = H(M_1, \ldots, M_r) = -\sum_{x \in \{0,1\}^r} P_M^x \log(P_M^x),$$

where $\log$ designates the logarithm to the base 2, and $P_M^x \in [0,1]$ is the proportion of rows of M that are equal to the boolean vector x.

The principle of REVEAL consists in analysing the columns of In and Out (respectively denoted $\mathcal{C}_{In}(i)$ and $\mathcal{C}_{Out}(i)$, for each $i \in \{1, \ldots, n\}$) to find the set of inputs $\mathcal{I}(v_i)$ of each node v_i. More precisely, the set

Algorithm 2 - REVEAL: identification of synchronous boolean networks.

`Input:` In, Out: boolean matrices of size $q \times n$.
`Output:` set of boolean networks ($\mathcal{G}$ and $\mathcal{F}$) consistent with inputs.

mark all nodes $\{v_1, \ldots, v_n\}$ as *untreated*
for $k = 0, 1, \ldots, n$ **do**
  /* testing connectivity k */
  for all *untreated* node v_i **do**
    for all $(i_1, \ldots, i_k)$ **ordered** k-tuple of $\{1, \ldots, n\}$ **do**
      if $H\left(\mathcal{C}_{Out}(i), \mathcal{C}_{In}(i_1), \ldots, \mathcal{C}_{In}(i_k)\right) = H\left(\mathcal{C}_{In}(i_1), \ldots, \mathcal{C}_{In}(i_k)\right)$ **then**
        $\rightarrow (v_{i_1}, \ldots, v_{i_k})$ are inputs of node v_i
        $\rightarrow$ find corresponding rule f_i
        $\rightarrow$ mark node v_i as *treated*
      end if
    end for
  end for
end for

of (ordered) k-tuples $(i_1, \ldots, i_k)$ of $\{1, \ldots, n\}$ is computed, for k growing to 0 to n (k is the connectivity). When the following condition is satisfied:

$$H\left(\mathcal{C}_{Out}(i), \mathcal{C}_{In}(i_1), \ldots, \mathcal{C}_{In}(i_k)\right) = H\left(\mathcal{C}_{In}(i_1), \ldots, \mathcal{C}_{In}(i_k)\right),$$

then the nodes v_{i_1}, ..., v_{i_k} are a possible set of inputs for node v_i (see [36] for a proof of this statement). The corresponding boolean rule f_i can be (partially) reconstructed by analysing the submatrix of In and Out formed by the columns $\mathcal{C}_{Out}(i)$ and $\mathcal{C}_{In}(i_1), \ldots, \mathcal{C}_{In}(i_k)$.

Remark 5 *Note that, if a node v_i admits the couple (v_1, v_2) to be a set of inputs, it is clear that any tuple $(v_{i_1}, \ldots, v_{i_k})$ that contains (v_1, v_2) will be a set of inputs of v_i as well. Therefore, the loop on k (increasing from 0 to n) ensures that the sets of inputs computed are of minimal size. Moreover, the connectivity of function f_i is a* real *connectivity (see Remark 2), which means that it has no fictitious variable.*

To conclude this brief description, let us make several comments about the identification Algorithm 2 (these comments -and their proofs- can be found with more details in [36]). First, if the size q of the sample is equal to 2^n (it is the maximal value of q), then the underlying structure of the network is identified unequivocally and is unique. If, on the other hand, we have $q < 2^n$, then obviously the identified structure is not garanteed to be unique. Moreover, the identified boolean rule may be only partially reconstructed. This algorithm finds all *minimal* structures (wiring and rules) that are consistent with the data (see [36] for precise definitions of minimality and consistency).

Finally, the complexity of this algorithm is in $O\left(qn^2 2^n\right)$, as $n \rightarrow \infty$. It is therefore exponential in n. Nevertheless, it can be brought back to a polynomial algorithm if one imposes the maximal connectivity to be a constant K (which seems reasonable form a biological point of view). This leads the time complexity of the algorithm to be (at worst) in $O\left(qK^2 n^{K+1}\right)$ [36].

4.2 Algorithmic search for operational interactions

As already said, the identification power of Algorithm 2, using biological experimental data, is questionable, as one is not sure that, between two successive patterns, each node has been updated once and only once. The same problem exists in the asynchronous framework, as we have no *a priori* information about the updating strategy. However, we will not use this algorithm to identify a network from experimental data, but to identify temporal operational interactions, in a well defined network.

In order to properly define and describe the algorithmic search for operational interactions, let us first state the following definitions, that are classical in graph theory.

Definition 7 *Let $G = (V, E)$ be a directed graph, with set of vertices V and set of directed edges E.*

- *If V' is a subset of V, the* subgraph *of G (induced by V') is the directed graph $G(V') = (V', E(V'))$, where $E(V')$ is the subset of E containing only the directed edges whose head and tail both belong to V'.*
- *If E' is a subset of E, the graph $G' = (V, E')$ is called a* partial graph *of G. In that case, graph G' is said to be* included *in graph G.*

Let $\mathcal{N} = (\mathcal{G}, \mathcal{F})$ be a (given) n-dimensional boolean network, and let $G = (V, E)$, $G^{scc} = (V^{scc}, E^{scc})$ denote, respectively, its asynchronous transition graph and its (hierarchically organized) SCC decomposition. If $c \in V^{scc}$ denotes a SCC of G, Algorithm 1 computes the set $\mathcal{R}(c)$, which is the reachability set of c. This set contains all SCCs (and thus all states) that are reachable by the system starting from c, whatever the updating order of the variables. Therefore, it is possible to reconstruct, from $\mathcal{R}(c)$, the subgraph of G^{scc} (respectively, the subgraph of G) that contains all possible SCC trajectories starting from c (resp., all possible state trajectories starting from any state in c). Let $G^{scc}(c)$ (resp., $G(c)$) denote this subgraph. It is straightforward that, by Definition 3 of the asynchronous transition graph, we are able, from the graph $G(c)$, to reconstruct the corresponding part of the synchronous successor function F. This construction is described in Algorithm 3 - step 1, and strongly relies on Assumptions 1 and 2. Using Algorithm 2 on those synchronous data will allow us to identify all interactions which are effectively active in $\mathcal{R}(c)$. This process is described in the second step of Algorithm 3.

We now introduce our concept of *operational interactions*, those that are essential to describe the logical rules associated with the subgraph $G(c)$. Roughly, an operational interaction is defined as an edge in the diagram of interactions which, if removed, induces changes in $G(c)$.

Definition 8 *Consider an n-dimensional boolean system $X' = F(X)$, with a diagram of interactions $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let $G = (\Omega, E)$ be the corresponding asynchronous state transition graph. Let $c \subset \Omega$ and let $\mathcal{R}(c)$ denote the set of states reachable from any state in c. Consider the subgraph generated by $\mathcal{R}(c)$: $G(c) = (\mathcal{R}(c), E(\mathcal{R}(c)))$.*

- *An edge $e \in \mathcal{E}$, is a* non-operational interaction associated with c *if the asynchronous subgraph $\hat{G}(c)$, generated by $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \setminus \{e\})$ (starting from c), satisfies: $\hat{G}(c) = G(c)$.*
- *An edge $e \in \mathcal{E}$, is an* operational interaction associated with c *if e is not non-operational.*
- *A family of interactions $\mathcal{E}_c \subset \mathcal{E}$ is said to be a* minimal family of operational interactions associated with the set c *if, for all $e \in \mathcal{E}_c$, e is an operational interaction, and for all $e' \in \mathcal{E} \setminus \mathcal{E}_c$, e' is a non-operational interaction associated with c.*

The diagram representing the graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$ will be called operational graph*.*

In other words, a minimal family of operational interactions (for the transition graph generated by a set of states c) contains as small a set as possible of the original interactions that still generates the original graph $G(c)$. A minimal family of operational interactions is obtained after "removing" all the interactions that do not affect the original graph $G(c)$. A related notion is that of *minimal cut sets* (MCS) for logical interaction graphs, suggested and used in [20]. A MCS has been defined with respect to a certain target function or response, and by analogy with the analysis of (continuous, stoichiometric) metabolic networks. A MCS is a minimal set of reactions whose removal will prevent the target response. The notion of operational interaction defined here differs from that of MCS. Mainly, the operational graph identifies a subnetwork responsible for a certain dynamical behaviour, that is comprised in a certain region of the transition graph. The reduction of the whole state space to this particular region allows to ensure all non-operational interactions to be removed. In contrast, MCS are not defined in terms of transition graphs.

The result of Algorithm 3 will be returned as an interaction graph (see Def. 1), comprising interactions that remain operational after the system has reached the SCC c. As the set $\mathcal{R}(c)$ is only a subset of V^{scc}, matrices In and Out are only a partial truth table of the synchronous successor function. As a consequence, the interaction

Algorithm 3 - Identification of (asynchronous) operational interactions.

Step 1: Construction of synchronous data from a subgraph Γ of the asynchronous transition graph G.

`Input:` $\Gamma = (V(\Gamma), E(\Gamma))$: subgraph of G.
`Output:` In, Out: boolean matrices of size $q \times n$ (partial truth tables).

$q := 1$
for all state $X \in V(\Gamma)$ **do**
  $In(q,:) := X$ /* fill in the q-th row of In */
  $S := X$ /* will contain the synchronous successor of X */
  for all asynchronous successors Y of X (in Γ) **do**
    find i such that $Y = \widetilde{X}^i$
    $S(i) := not(X(i))$
  end for
  $Out(q,:) := S$
  $q := q + 1$
end for
return In and Out

Step 2: Identification of operational interactions from a SCC.

`Input:` G, G^{scc}: (asynchronous) transition graph and SCC decomposition.
  $c \in V^{scc}$: a strongly connected component of G.
`Output:` interaction graphs comprising operational interactions.

compute reachability set $\mathcal{R}(c)$ /* use Algorithm 1 */
compute subgraph $G^{scc}(c)$ of G^{scc} induced by $\mathcal{R}(c)$
compute corresponding subgraph $G(c)$ of G
compute synchronous data (matrices In and Out) of subgraph $G(c)$ /* use step 1 */
call REVEAL on matrices In and Out to compute operational graphs

graph returned is not guaranteed to be unique. Nevertheless, as explained in previous part, REVEAL captures all *minimal* interaction graphs, that is, interaction graphs with minimal connectivities (the term connectivity must be understood in the sense of *real* connectivity, as noted in Remark 2) which are consistent with In and Out. If the initial interaction graph $\mathcal{G}$ is already in a minimal form, then it is easy to see that among all graphs returned by Alg. 3, there exists one that is included (in the sense of Def. 7) in $\mathcal{G}$. This particular graph will be denoted $\mathcal{G}_c$. More precisely, $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$, where $\mathcal{E}_c \subset \mathcal{E}$, which means that any operational interaction identified is actually an interaction comprised in the initial network. In the major part of our preliminary tests, the operational graph returned by Alg. 3 is actually unique, and is equal to $\mathcal{G}_c$. It is notably the case for the apoptosis network (see next section).

Moreover, as the reachability set $\mathcal{R}(c)$ captures all possible asynchronous successors of c, it is easy to see that, if we successively compute the operational graphs along a particular SCC trajectory: $(c_1, c_2, \ldots, c_l)$ (where c_l is an attractor of the system), then we have the following inclusions:

$$\mathcal{G} \supset \mathcal{G}_{c_1} \supset \mathcal{G}_{c_2} \supset \cdots \supset \mathcal{G}_{c_l}.$$

(Symbol $\subset$ designates the graph inclusion defined in Def. 7). In other words, this means that it is possible to visualize, along a trajectory, at which step an interaction (represented by a directed edge of $\mathcal{G}$) may become non-operational. Ultimately, the final graph $\mathcal{G}_{c_l}$ comprises interactions that remain always operational through the whole trajectory (up to the attractor).

4.3 Application to the apoptosis network

In this section, Algorithm 3 is applied to the apoptosis NFκB signalling pathway. This system is interesting because it exhibits two global dynamical properties that are often studied in systems biology. The first one is

the *multistationarity*, *i.e.* the coexistence of several attractors. According to Fig. 4, this happens when TNF is absent (the two attractors are steady states). The second one is the presence of *oscillations*, which appear when TNF is activated. These two general properties are often related to the interaction graph, and in particular to the presence of positive and negative feedback loops. Two famous conjectures, stated in the eighties by R. Thomas [33], have been proved in different mathematical frameworks (see for instance [15, 24, 32]). The first one states that the presence of positive feedback loops is a necessary condition for multistationarity. The second one states that the presence of negative feedback loops is a necessary condition for the presence of oscillations (in continuous frameworks, oscillations may be damped, and can then be related to the biological concept of *homeostasis*). In the following, Alg. 3 is used to identify which feedback loops of Fig. 1 are effectively responsible for the presence of oscillations and for multistationarity.

According to the analysis presented in Section 3.2, the state space of the apoptosis NFκB system can be separated in two regions, $\mathcal{T}^0$ and $\mathcal{T}^1$, according to the value of the input TNF. Within region $\mathcal{T}^1$ (TNF $= 1$), we found a unique attractor, which is the SCC 49 (see Fig. 4). Let us denote this attractor a_1. It contains 56 states, and the variables that can oscillate within a_1 are indicated in Table 2. As it is the only attractor present in $\mathcal{T}^1$, we know that all trajectories starting from any state of $\mathcal{T}^1$ will eventually reach it and remain in it for all subsequent times. When applying Algorithm 3 to a_1, we obtain the operational graph $\mathcal{G}_{a_1}$ depicted in Figure 6.

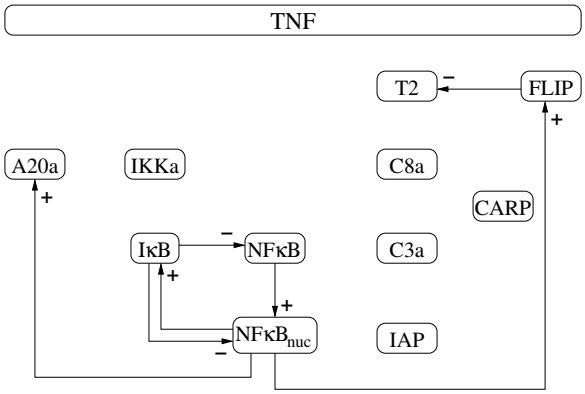
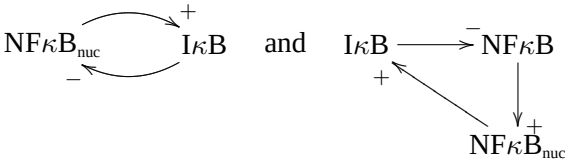


Figure 6: Operational graph of the NFκB pathway in apoptotic oscillations (attractor a_1, TNF $= 1$). The isolated variables have a fixed value, that can be found in Table 2. Interactions have been signed with respect to boolean rules (Table 1).

This interaction graph is included in the initial one (Fig. 1), and comprises all interactions that remain active in a_1. As we can see in Fig. 6, six of the twelve variables are isolated, that is, they keep a constant value over time. Their values can be found in Table 2 (in particular, the caspases C3a and C8a are activated). Among the six remaining variables, only three are involved in feedback loops: IκB, NFκB and NFκB$_{nuc}$ (the three other variables are affected by those three, but do not affect them). Using Table 1, one can associate with each interaction a sign ($+$ or $-$ depending whether it is an activation or an inhibition). One can see that both feedback loops:



are negative, as they contain an odd number of inhibitions. Thomas' conjectures [15, 24, 32] lead to the following conclusions:

(i) there cannot be more than one steady state because the graph has no positive loop,

(ii) oscillations are possible because the graph has negative loops.

This is indeed what is observed in the system. As these loops are the only ones that are active in attractor a_1, it can be infered that they are the ones responsible for the presence of oscillations in the system.

Let us now consider the region $\mathcal{T}^0$, where TNF $= 0$. As we can see in Figure 4, this region contains two attractors, corresponding to two equilibrium points. The first equilibrium (SCC 1, denoted a_2) corresponds to the "survival" of the cell, with the inhibition of the caspases, whereas the second one (SCC 123, denoted a_3) corresponds to the triggering of apoptosis, with activation of the caspases. In order to find the feedback loop that is responsible for the coexistence of these two equilibria, we have to find a SCC c^* that can lead to both a_2 and a_3, *i.e.*:

$$c^* \in \mathcal{A}(a_2) \cap \mathcal{A}(a_3)$$

(where $\mathcal{A}(a_2)$ and $\mathcal{A}(a_3)$ are the attraction sets of a_2 and a_3). There are several possible choices for c^*. As we only want to identify the loop responsible for the coexistence of the two equilibria, we choose c^* to be the one with the highest hierarchical level. The operational graph $\mathcal{G}_{c^*}$ obtained is depicted in Figure 7.

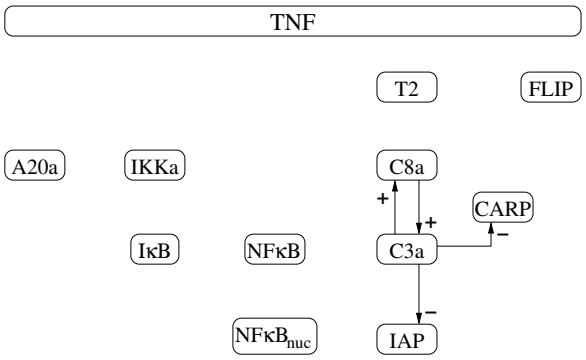
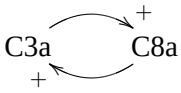


Figure 7: Operational graph of the NFκB pathway just before the choice between attractors a_2 (survival) and a_3 (apoptosis) (*i.e.* in the region where TNF$= 0$). Interactions have been signed with respect to boolean rules (Table 1).

Note that this graph contains only one feedback loop:



which is a classical double activation system, involving the caspases. From Thomas' conjectures, there can be no oscillations, only multistationarity. This is indeed the case, as there are exactly two steady states: C3a $=$ C8a $= 1$ (apoptosis) and C3a $=$ C8a $= 0$ (survival).

As already evoked, various proofs of Thomas' conjectures have already been given in different frameworks, including the boolean case [24]. These two examples allow to go a little further in the study of feedback loops. Indeed, the existing theorems are very general, and give necessary conditions for the existence of oscillations or multistationarity. What is shown here is that it is algorithmically possible, in the boolean framework, to identify, in a complex interaction graph such as the one in Fig. 1 (comprising multiple positive and negative loops), subsets of loops which are effectively responsible for those two dynamical behaviors.

Each of the two operational graphs represents the main (and ultimately active) interactions in two different biological scenarios. The first operational graph (Fig. 6), represents the interactions that remain active after a *sufficiently long time interval of TNF stimulation has elapsed*. That is, immediately upon TNF stimulation, the system responds and evolves towards a certain configuration; once this has been reached, most interactions have been "stabilized" or achieved a natural balance. At this stage only remain active the cycle corresponding to NFκB-activated transcription of IκB, and the subsequent inhibition of NFκB. In this case, oscillations in these two variables may be observed [17], but all the other variables are constant. In particular, the model predicts that inhibitor of apoptosis proteins (IAP) is present at a low concentration, but the caspases at high

concentration so, from the biological point of view, it may be expected that a very long stimulation with TNF will eventually lead to apoptosis. However, at this stage, the system may still "reverse" its apoptotic decision if TNF is shut down: the system will then leave the configuration shown in Fig. 6, and postpone the survival/death decision. In the absence of TNF, the system will evolve towards the configuration shown in Fig. 7, where only the positive feedback cycle representing the caspase cascase [10] remains functional. From the biological point of view this means that, depending on the state of the system when TNF was shut down (or its initial state), the cell may decide between survival (represented by attractor a_2) or initiating apoptosis (represented by attractor a_3). Once the trajectory of the system enters one of these two attractors, the survival/death decision is final, and no reversal is possible.

5 Probabilistic analysis of asynchronous dynamics

In this section, we expand the analysis of the asynchronous transition graph to a quantitative level, by introducing transition probabilities. The asynchronous transition graph of a given network is a very general object, in the sense that it contains information on all the possible trajectories of the system: that is, for each state the graph indicates all the next possible states, or successors. Roughly, each successor results from the update of one different variable, but the graph does not contain any indication on which of the successors is more probable at a given time. By associating a probability of transition with each graph edge, more quantitative biological knowledge can be straigthforwardly incorporated into discrete dynamics' models. Our methodology uses the SCC decomposition and the hierarchical organization of the transition graph, and is related to the work in [11, 12, 30, 31], where slightly different types of discrete models are discussed.

Application of our method to the apoptosis/NFκB network leads to estimation, among other quantities, of the probability of cell survival or apoptosis upon stimulation of death receptors. Such quantities can be compared with experimental data (see also numerical results in [7] and references therein).

5.1 Construction of a probabilistic transition graph

In order to confront the general discrete asynchronous dynamics, given by the transition graph G (see Definition 3) with biological experiments, one has to be a little bit more specific on the updating strategies. As said before, the principle of asynchronous analysis is to consider *all* possible strategies, and to find dynamical properties that are valid whatever the strategy, and thus robust with respect to the structure of the network under study. Previous algorithms and results are essentially *qualitative*, as they are mainly attached to the structure, without considering updating strategies at all. If one wants to represent the different choices of updating strategies, in a more *quantitative* manner, one way is to introduce probabilities in the transition graph G, and to consider a choice of the updating order of the variables (that is, the choice of a particular trajectory in G) as the choice of a trajectory in a stochastic process.

We recall that G is implemented by means of its adjacency matrix, that will be denoted $A(G)$. Recall that each state of the system is a boolean vector $X \in \Omega = \{0,1\}^n$. Such a vector X is unequivocally associated with a unique integer $s(X) \in \{1, \ldots, 2^n\}$ by the following relation:

$$X = (x_1, \ldots, x_n) \in \{0,1\}^n \quad \longmapsto \quad s(X) = \left(\sum_{j=1}^{n} x_i 2^{i-1} \right) + 1 \in \{1, \ldots, 2^n\},$$

so that in the following, "states" (*i.e.* elements of Ω) will be represented by integers lying in $\{1, \ldots, 2^n\}$. The adjacency matrix $A(G)$ is defined as follows:

$$A(G) = (a_{i,j})_{1 \leq i,j \leq 2^n} \text{ , with } \begin{cases} a_{ij} = 1 & \text{if "state } j\text{" is a successor of "state } i\text{",} \\ a_{ij} = 0 & \text{otherwise.} \end{cases}$$

The size of $A(G)$ is $2^n \times 2^n$ and, in general, $A(G)$ is implemented as a sparse matrix (for instance, we saw earlier that for the NFκB pathway, its filling rate is around 0.13%, see Fig. 5 on the left for an illustration). As a first example, we propose a *naive* construction of the transition probabilities p_{ij}, where all asynchronous

successors of a state have the same probability. The uniform distribution of probabilities of asynchronous successors means that we make no *a priori* assumption about the updating rules (beyond of course Assumptions 1 and 2). We will later consider "biologically educated" distributions. Define the matrix $\mathcal{P}(G)$ as the $2^n \times 2^n$ real matrix:

$$\mathcal{P}(G) = (p_{i,j})_{1 \leq i,j \leq 2^n} \text{ , with: } \quad \forall 1 \leq i,j \leq 2^n \text{ , } p_{ij} = \frac{a_{ij}}{\sum_{k=1}^{2^n} a_{ik}} = \frac{a_{ij}}{N^+(i)},$$

where $N^+(i) = \sum_{k=1}^{2^n} a_{ik}$ designates the number of directed edges of G that leave i (in other words, $N^+(i)$ is the number of asynchronous successors of state i), and $p_{ij} \in [0,1]$ is the probability for the system to go from state i to state j. By construction, matrices $\mathcal{P}(G)$ and $A(G)$ share the same sparsity pattern. Furthermore, it is easy to see that $\mathcal{P}(G)$ is a *stochastic* matrix, ***i.e.*** it satisfies the two following conditions:

- all its elements p_{ij} are nonnegative,
- the sum of the elements of each row, $\sum_{j=1}^{2^n} p_{ij}$ is equal to 1.

From the construction of $\mathcal{P}(G)$, we can now consider the (asynchronous) dynamics on G as a discrete time Markov chain, over the state space $\{1, \ldots, 2^n\}$. In particular, if $P(X_0 = i_0)$ designates the probability for the system to be initially in the state $i_0 \in \{1, \ldots, 2^n\}$, then the probability that a trajectory follows the path $\mathbf{p} = (i_0, i_1, \ldots, i_q)$ (where the i_j are elements of $\{1, \ldots, 2^n\}$) is equal to:

$$P(\mathbf{p}) = P(X_0 = i_0) \prod_{j=0}^{q-1} p_{i_j, i_{j+1}}.$$

Using this probabilistic approach, the analysis of the asynchronous dynamics of a boolean network is hence brought back into the classical framework of discrete Markov chains. The use of Markov chains to describe the dynamics of gene regulatory networks is not new (see, for instance, [6,11,30] in slightly different frameworks). Indeed, the rich theory of Markov chains (strongly linked with the theory of nonnegative matrices) provides powerful mathematical tools to help the analysis of such networks. Following the methodology described in previous sections, this approach is used here on the simplified SCC graph, G^{scc}, instead of the original transition graph G. In order to define the Markov chain over G^{scc}, we first recall some classical results about nonnegative matrices, mainly taken from [2].

As $A(G)$ is a nonnegative matrix, it can be permuted into a triangular block form, that is, there exist a permutation matrix P_1, of size $2^n \times 2^n$, such that:

$$P_1 A(G) P_1^t = \begin{pmatrix} A_{11} & 0 & \ldots & 0 \\ A_{21} & A_{22} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \ldots & A_{pp} \end{pmatrix}, \tag{1}$$

where the diagonal blocks A_{ii} are square, and either irreducible, or 1×1 and null. The notion of irreducibility, a definition of which can be found in [2], is the equivalent of the notion of strong connectivity in graph theory. As a matter of fact, as $A(G)$ is the adjacency matrix of graph G, each diagonal block A_{ii} corresponds to a strongly connected component of G. When performing the SCC decomposition, each SCC is "summarized" as a node of the SCC graph G^{scc}. In Equation (1), this corresponds to the construction of a $p \times p$ triangular matrix:

$$A^{scc} = \begin{pmatrix} a_{11}^{scc} & 0 & \ldots & 0 \\ a_{21}^{scc} & a_{22}^{scc} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1}^{scc} & a_{p2}^{scc} & \ldots & a_{pp}^{scc} \end{pmatrix},$$

where a diagonal element $a_{ii}^{scc} = 1$ if it corresponds to an irreducible block A_{ii}, and $a_{ii}^{scc} = 0$ if it corresponds to a 1×1 null block A_{ii}, and non diagonal elements $a_{ij}^{scc} \in \{0,1\}$ are computed according to the set E^{scc} of

directed edges of G^{scc}. The $p \times p$ matrix A^{scc} is actually the adjacency matrix of the graph G^{scc}. The fact that A^{scc} is triangular implies, as we already knew, that the graph G^{scc} is acyclic. Moreover, the non zero diagonal elements of A^{scc} can be easily identified, as they correspond either to final SCCs (*i.e.* attractors, according to Def. 4), or to transient SCCs that contain at least two states. Following the construction of A^{scc} from matrix $A(G)$, we can now construct the matrix $\mathcal{P}^{scc}$, containing the probabilities of the transitions between SCCs, from matrix $\mathcal{P}(G)$.

As $p \in \{1, \ldots, 2^n\}$ denotes the number of SCCs, each SCC will be unequivocally represented by an integer lying in $\{1, \ldots, p\}$. As explained in Section 3.1, a SCC $c \in \{1, \ldots, p\}$ is, by definition, a subset of the state space Ω. In the following, $S(c)$ denotes the subset of $\{1, \ldots, 2^n\}$ of the states belonging to c. The number of these states (*i.e.* the *size* of the SCC) will then be denoted $|S(c)|$. The matrix $\mathcal{P}^{scc}$ is a square real matrix of size $p \times p$, and its elements are computed as follows: given two SCCs $c, c' \in \{1, \ldots, p\}$,

$$p^{scc}_{c,c'} = \frac{1}{|S(c)|} \sum_{i \in S(c)} \sum_{j \in S(c')} p_{ij}. \tag{2}$$

In words, this definition means that to compute the probability of the transition from a SCC c to a SCC c', we sum all transitions from any state i in $S(c)$ to any state j in $S(c')$, and we divide this sum by the number of states in c, in order to obtain an average transition probability from c to c'. It is easy to see that, as for $A(G)$ and $\mathcal{P}(G)$, matrices A^{scc} and $\mathcal{P}^{scc}$ share the same sparsity pattern (see Fig. 5, on the right, for an illustration in the case of the apoptosis network). Matrix $\mathcal{P}^{scc}$ is therefore a lower triangular matrix. Moreover, we prove the following:

Proposition 1 *The matrix $\mathcal{P}^{scc}$ is stochastic.*

Proof.
The nonnegativity of $\mathcal{P}^{scc}$ elements is obvious. We prove here that the sum of the elements of its rows is equal to 1. Let $c \in \{1, \ldots, p\}$. We have:

$$\begin{aligned}
\sum_{c'=1}^{p} p^{scc}_{c,c'} &= \sum_{c'=1}^{p} \frac{1}{|S(c)|} \sum_{i \in S(c)} \sum_{j \in S(c')} p_{ij} \\
&= \frac{1}{|S(c)|} \sum_{c'=1}^{p} \sum_{i \in S(c)} \sum_{j \in S(c')} \frac{a_{ij}}{N^+(i)} \\
&= \frac{1}{|S(c)|} \sum_{i \in S(c)} \frac{1}{N^+(i)} \sum_{c'=1}^{p} \sum_{j \in S(c')} a_{ij}.
\end{aligned}$$

As the SCCs of a directed graph form a partition of its set of vertices, the quantity $\sum_{c'=1}^{p} \sum_{j \in S(c')} a_{ij}$ is equal to the number of edges that leave i (in graph G). By definition, it is equal to $N^+(i)$. This leads to:

$$\sum_{c'=1}^{p} p^{scc}_{c,c'} = \frac{1}{|S(c)|} \sum_{i \in S(c)} \frac{N^+(i)}{N^+(i)} = \frac{|S(c)|}{|S(c)|} = 1.$$

□

Therefore, the dynamics on the graph G^{scc} is reduced to the dynamics of the discrete time Markov chain of the triangular matrix $\mathcal{P}^{scc}$. The main advantage of considering matrix $\mathcal{P}^{scc}$ instead of matrix $\mathcal{P}(G)$ is that it satisfies some useful properties. For instance, each *ergodic* class of $\mathcal{P}^{scc}$ (roughly, an ergodic class is a set of non transient SCCs, see [2] for a precise definition) contains only one element. The corresponding Markov chain is then called *absorbing* and its absorbing elements are in fact the attractors of the system. For absorbing chains, we can use the following well known result (see, *e.g.* [16]): there exists a $p \times p$ permutation matrix P_2 such that

$$P_2 \mathcal{P}^{scc} P_2^t = \begin{pmatrix} I_r & 0 \\ R & Q \end{pmatrix}, \tag{3}$$

where $r \in \{1, \ldots, p\}$ is the number of attractors of the system, I_r denotes the $r \times r$ identity matrix, and Q is a $(p-r) \times (p-r)$ lower triangular matrix that satisfies:

$$\lim_{n \to \infty} Q^n = 0. \tag{4}$$

The form (3) is often called the *canonical form* of $\mathcal{P}^{scc}$. From (4), we can define the $(p-r) \times (p-r)$ matrix $N = (I - Q)^{-1}$ (often called *fundamental* matrix). Its entry $n_{cc'}$ gives the expected number of times that the process is in the transient SCC c' if it started somewhere in the transient SCC c. In particular, it can be used to compute the vector $\mathbf{t} = N\mathbf{1}$ (where $\mathbf{1}$ designates the column vector of whose entries are 1). Given a transient SCC c, the entry t_c of $\mathbf{t}$ gives the expected number of steps before the chain reaches an attractor, given that it started somewhere in transient SCC c. Finally, we define the $(p-r) \times r$ matrix $B = NR$, whose entry $b_{c,c'}$ is the probability that the chain reach attractor c' if it starts somewhere in the transient SCC c.

As a consequence, Markov chains provide an efficient mathematical framework, in which useful global parameters can be computed. These parameters provide important biological knowledge about the system, as they contribute to further characterize qualitative dynamical properties, that are robust with respect to the topology of the network. An example of such properties, in the case of the apoptosis network, is proposed in Section 5.3. Let us recall that, for the moment, we made no *a priori* assumption for the computation of transition probabilities. We show in the next section that, within this probabilistic framework, even incomplete biological knowledge about the system can be easily added, in order to provide more realistic probability distributions.

5.2 Towards a biological probabilistic graph

For models of biological genetic networks, partial knowledge of the parameters is often available. For example, the relative rates of two reactions are known (*e.g.*, the rate of formation of protein A is larger than that of protein B). This biological knowledge can be straigthforwardly incorporated into the transition graph, by stipulating an updating strategy such as an updating order among all variables. The matrix of transition probabilities, $\mathcal{P}(G)$, associated with the asynchronous graph in Section 5.1 was based on a uniform probability distribution. The probabilities of transition can instead be computed according to biological data, by using the notion of *priority classes* [12, 14]. Roughly, the idea is to group the variables into several groups, called priority classes, and assign a weight to each of these groups: higher weights denote a more probable transition. A similar idea was used in [6], where two classes were considered, one for proteins and another for mRNAs. The updating order stipulated that proteins were always updated first and mRNAs next. More generally, to implement the notion of priority classes, consider ρ classes $\mathcal{C}_1$, ..., $\mathcal{C}_\rho$ and their respective weights,

$$W_1 > W_2 > \cdots > W_\rho,$$

and associate with each edge $i \to j$ the value:

$$w_{ij} = W_r, \quad \text{if } X_s^i \neq Y_s^j \quad \text{and} \quad s \in \mathcal{C}_r$$

that is, if the variable s updated in the transition $i \to j$ belongs to class $\mathcal{C}_r$. If no transition from i to j is possible, then set $w_{ij} = 0$. Then define a new transition matrix, where each p_{ij} represents a weighted average:

$$\mathcal{P}_{bio}(G) = (p_{i,j})_{1 \leq i,j \leq 2^n} \ , \text{ with: } \quad \forall 1 \leq i,j \leq 2^n \ , \ p_{ij} = \frac{w_{ij} a_{ij}}{\sum_{k=1}^{2^n} w_{ik}}. \tag{5}$$

As before, a corresponding matrix, $\mathcal{P}_{bio}^{scc}$, can be constructed for the graph G^{scc}. The probability of transition between two SCCs c and c' is given by Eq. (2), where the p_{ij} are replaced by the transition probabilities computed in (5). Again, it is easy to check that $\mathcal{P}_{bio}^{scc}$ represents an absorbing Markov chain process.

5.3 Application to the apoptosis network

In order to illustrate the probabilistic approach, and the type of results it provides, an application to the apoptosis network is next described. In particular, the results show how Tumor Necrosis Factor (TNF) influences

the choice of the system between the two possible steady states: the "survival" of the cell (attractor a_2, with inhibition of the caspases) and the triggering of apoptosis (attractor a_3, with activation of the caspases). Experimentally, it is observed that a cell irreversibly enters the apoptotic pathway once a certain threshold in caspase activation has been reached (see [10] and references therein). In turn, caspase activation is observed to depend on the duration of TNF stimulation, as well as on TNF concentration (typically, the caspase activation threshold is reached faster for higher TNF concentrations). In [7], this property was statistically observed by computing many different trajectories (of a continuous, piecewise linear system), with different updating strategies. Within the framework presented in this paper, the probability that the cell follows the "survival" or "apoptosis" pathway can be computed directly from the matrices $\mathcal{P}^{scc}$ (or $\mathcal{P}^{scc}_{bio}$), without performing large numbers of simulations. Recall that a trajectory of the system will converge towards a_2 or a_3 in the absence of TNF. Technically, the shutdown of death receptor stimulation is represented in our system by switching the input variable TNF from 1 to 0. For each state X in $\mathcal{T}^1$ (where TNF is equal to 1), a successor X_s is computed in $\mathcal{T}^0$ (where the variable TNF is 0, and all other variables stay unchanged). Then, using the matrix $\mathcal{P}^{scc}$ and its canonical form (3), the probabilities to reach attractors a_2 and a_3 from initial state X_s are computed. Figure 8 presents the result of this numerical experiment.

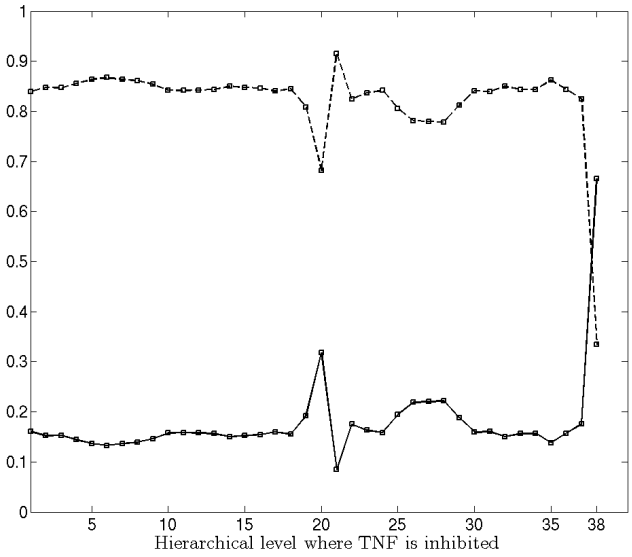


Figure 8: These two curves represent the system's response to TNF switch off, after the system has reached a certain hierarchical level H_i (x-axis). The dashed line represents the average probability for the system to reach attractor a_2 (*i.e.*, for the cell to reach the "survival" equilibrium), starting from a state in hierarchical level H_i. The straight line represents the probability to reach attractor a_3 (*i.e.*, the "apoptosis" equilibrium), starting from H_i. These *in silico* experiments were carried on with the "naive" stochastic matrix $\mathcal{P}^{scc}$.

Contrary to [7], where time is represented by a continuous variable, in the asynchronous graphs there is no direct measure of time. However, the hierarchical levels of the graph G^{scc} do give an indication of time progression. Indeed, consider two states along any given trajectory, $X_1, X_2 \in \mathcal{T}^1$ where X_1 belongs to a hierarchical level H_i and X_2 belongs to H_j, with $j > i$. Then we can say that TNF stimulation has been longer for X_2 than for X_1. The x-axis of Figure 8, which represents the hierarchical levels, is thus a relative measure of the duration of TNF.

Following the procedure developed in Section 5.2, a more realistic matrix $\mathcal{P}^{scc}_{bio}$ can be constructed. Based on the parameters reported in [10, 28] (and references therein), four priority classes were established for the apoptosis network depicted in Figure 1. These classes are based essentially on the relative magnitudes of the degradation rates. The "faster" class of proteins (*i.e.*, those with higher degradation rates) consists of NFκB, NFκB$_{nuc}$, IκB, and the inhibitor CARP; the second class contains the complexes T_2 and IKKa; the third class

contains the caspases C3a and C8a; finally, the fourth class contains the remainder of the variables. The classes and their assigned weights are summarized in Table 3.

Class	Weights	Variables
$\mathcal{C}_1$	$w_1 = 7$	NFκB, NFκB$_{nuc}$, IκB, CARP
$\mathcal{C}_2$	$w_2 = 5$	T_2, IKKa
$\mathcal{C}_3$	$w_3 = 3$	C3a, C8a
$\mathcal{C}_4$	$w_4 = 1$	A20a, IAP, FLIP

Table 3: Priority classes and respective weights.

The same numerical experiment described above was carried out, now using matrix $\mathcal{P}_{bio}^{scc}$: computation of the probability of convergence to attractor a_2 or a_3 in response to TNF stimulation shutdown at level H_i. The results are presented in Fig. 9.

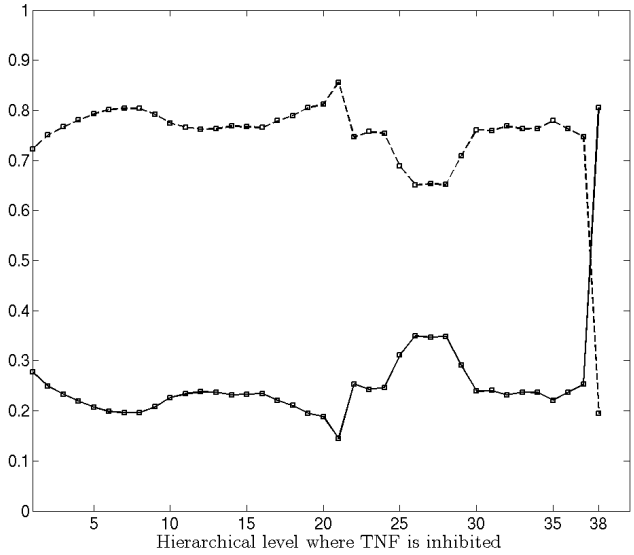


Figure 9: The two curves represent the same experiment carried out in Figure 8, with "more realistic" transition matrix $\mathcal{P}_{bio}^{scc}$.

Examination and comparison of Figures 8 and 9 shows that, as might be expected, the apoptotic pathway becomes more likely as TNF stimulation time increases. More specifically, we observe that, for both the uniform and the biologically informed probability distributions, the apoptotic pathway becomes more probable than the survival pathway only when the last hierarchical level has been reached. It is striking that this property appears in both curves, suggesting the existence of a threshold state configuration which is independent of the updating strategy, and constitutes a robust feature of the system (further numerical experiments not shown here, with randomly chosen priority classes, also exhibited similar curves). In other words, if one wants to promote apoptosis, TNF should be sustained long enough for the system to reach attractor a_1, *i.e.* apoptotic oscillations.

On the other hand, there are quantitative differences between the two figures which give an indication of the (kinetic) variability of the system. Thus, the survival pathway is always more probable for levels H_1 to H_{37}, but the average survival probability is higher in Fig. 8 (around 85%), than in Fig. 9 (around 75%). Similarly, the apoptosis pathway is always more probable for level H_{38}, but with a much lower probability in Fig. 8 (66%), than in Fig. 9 (around 80%). Also, in Fig. 8, an apoptosis intermediate peak is observed at

hierarchical level H_{20}. This peak is smoothed out in Fig. 9, which indicates that it might be an artefact due to the uniform distribution of probabilities in $\mathcal{P}^{scc}$.

The contribution of this Markov chain approach to the analysis of biological networks is thus two-fold: first, the common traits of the curves obtained with different matrices $\mathcal{P}^{scc}$ suggest global qualitative dynamical properties, which are robust with respect to the structure of the system, that is to say, independent of the choice of the updating order of the variables. Second, the quantitative aspects capture the variability and possible operating range of the network. Further applications of this technique include hypothesis testing and validation. For instance, one can easily analyze the effect of new interactions in the system's structure; similarly, the relative impact of two proteins can be studied by comparing the response of the system with different priority classes and updating strategies.

6 Conclusion

A method for model reduction of boolean networks has been developed, based on the hierarchical decomposition of asynchronous graphs. The first aspect to be analyzed is the decomposition of the state space of the n-dimensional boolean network into strongly connected components (SCCs), and the construction of the graph of transitions among them. The SCCs can be viewed as the "new states" of a "new" reduced system, since very often the number of SCC is less than or equal to the number of states of the original system. The second aspect in the model reduction method is the reconstruction of the boolean rules that represent the graph of transitions among the SCCs. An identification algorithm (known as REVEAL) was adapted and used to determine a family of boolean rules that describe the dynamics represented by a (sub-)graph of transitions.

More generally, the model reduction method uses the structure of interactions to isolate and identify smaller subsystems (or groups of variables and interactions) responsible for a given qualitative dynamical behaviour. This is a particularly relevant characterization for biological systems where experimental data consists (mostly) of qualitative measurements. The techniques described here are based on the fact that, for a boolean system, the whole state space can be easily enumerated; this introduces one other limitation to the method, on the size n of the network that can be computationally managed. Networks of intermediate size (up to $n = 15$) are easily computed. For larger networks, one may still use this method, by first isolating more basic modules. Each module would then be separately reduced and treated as one "node" with its boolean rule.

As illustrated with the apoptosis example, model reduction using the asynchronous boolean graph decomposition is a powerful potential source of valuable knowledge on a system. All the possible qualitative trajectories of the system are characterized, as well as their robustness to environmental perturbations. It is possible to identify the mechanism (in the form of smaller groups of variables and interactions) which is responsible for a given asymptotic behaviour of the system, for instance, the existence of oscillatory dynamics or (multi-)stability. Finally, the asynchronous transition graph can be naturally associated with a matrix of transition probabilities. Biological knowledge on the system's kinetics can thus be incorporated to obtain a more quantitative description of the system. These are also useful tools to test hypothesis and generate predictions concerning the structure of interconnections and the importance of each variable to the overall dynamics.

References

[1] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics, AMS Publications*, 280:193–219, 2001.

[2] A. Berman and R.J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Classics in Applied Mathematics. SIAM Press, 1994.

[3] S. Bornholdt. Less is more in modeling large genetic networks. *Science*, 310:449–451, 2005.

[4] R. Casey, H. de Jong, and J.L. Gouzé. Piecewise linear models of genetic regulatory networks: equilibria and their stability. *J. Math. Biol.*, 52:27–56, 2006.

[5] C. Chaouiya, E. Remy, B. Mossé, and D. Thieffry. Qualitative analysis of regulatory graphs: a computational tool based on a discrete formal framework. In *First Multidisciplinary International Symposium on Positive Systems: Theory and Applications (POSTA'03)*, volume 294 of *LNCIS*, pages 119–126. Springer, 2003.

[6] M. Chaves, R. Albert, and E.D. Sontag. Robustness and fragility of boolean models for genetic regulatory networks. *J. Theor. Biol.*, 235:431–449, 2005.

[7] M. Chaves, T. Eissing, and F. Allgöwer. Regulation of apoptosis via the NFκB pathway: modeling and analysis. In A. Deutsch, N. Ganguly, and A. Mukherjee, editors, *Dynamics on and of complex networks*. Birkhauser, 2008. to appear.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2001. (2nd edition).

[9] N.N. Danial and S.J. Korsmeyer. Cell death: critical control points. *Cell*, 116:205–216, 2004.

[10] T. Eissing, H. Conzelmann, E.D. Gilles, F. Allgöwer, E. Bullinger, and P. Scheurich. Bistability analysis of a caspase activation model for receptor-induced apoptosis. *J. Biol. Chem.*, 279:36892–36897, 2004.

[11] B. Faryabi, J.-F. Chamberland, G. Vahedi, A. Datta, and E.R. Dougherty. Optimal intervention in asynchronous genetic regulatory networks. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):412–423, 2008.

[12] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.

[13] C. Frelin, V. Imbert, V. Bottero, N. Gonthier, A.K. Samraj, K. Schulze-Osthoff, P. Auberger, G. Courtois, and J. F. Peyron. Inhibition of the NF-κB survival pathway via caspase-dependent cleavage of the IKK complex scaffold protein and NF-κB essential modulator NEMO. *Cell Death Differ.*, 15:152–160, 2008.

[14] A.G. Gonzalez, A. Naldi, L. Sànchez, D.Thieffry, and C. Chaouiya. GINsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. *BioSystems*, 84(2):91–100, 2006.

[15] J.-L. Gouzé. Positive and negative circuits in dynamical systems. *J. Biol. Sys.*, 6:11–15, 1998.

[16] C.M. Grinstead and J.L. Snell. *Introduction to probability*. AMS Bookstore, 1997.

[17] A. Hoffmann, A. Levchenko, M.L. Scott, and D. Baltimore. The IkB-NFkB signaling module: temporal control and selective gene activation. *Science*, 298:1241–1245, 2002.

[18] S.A. Kauffman. *The origins of order*. Oxford University Press (New York), 1993.

[19] B.N. Kholodenko, A. Kiyatkin, F.J. Bruggeman, E. Sontag, H.V. Westerhoff, and J.B. Hoek. Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *Proc. Natl. Acad. Sci USA*, 99:12841–12846, 2002.

[20] S. Klamt, J. Saez-Rodriguez, J. Lindquist, L. Simeoni, and E. Gilles. A methodology for the structural and functional analysis of signaling and regulatory networks. *BMC Bioinformatics*, 7(1):56, 2006.

[21] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architecture. In *Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[22] T. Lipniacki, P. Paszek, A.R. Brasier, B. Luxon, and M. Kimmel. Mathematical model of NFkB regulatory module. *J. Theor. Biol.*, 228:195–215, 2004.

[23] N.D. Perkins. Integrating cell-signalling pathways with NF-κB and IKK function. *Nature Rev. Mol. Cell Biol.*, 8:49–62, 2007.

[24] E. Remy, P. Ruet, and D. Thieffry. Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework. *Advances in Applied Mathematics*, 41(3), 2008.

[25] D. Ropers, H. de Jong, M. Page, D. Schneider, and J. Geiselmann. Qualitative simulation of the carbon starvation response in *Escherichia coli*. *Biosystems*, 84(2):124–152, 2006.

[26] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, and Burkhart Schraven. A logical model provides insights into t cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, Aug 2007.

[27] L. Sánchez and D. Thieffry. A logical analysis of the *Drosophila* gap-gene system. *J. Theor. Biol.*, 211:115–141, 2001.

[28] M. Schliemann. Modelling and experimental validation of TNFα induced pro- and antiapoptotic signalling. Master's thesis, University of Stuttgart, Germany, 2006.

[29] H. Schmidt and E.W. Jacobsen. Identifying feedback mechanisms behind complex cell behavior. *IEEE Control Syst. Mag*, 4:91–102, 2004.

[30] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang. Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. *BioInformatics*, 18(2):261–274, 2002.

[31] H. Siebert and A. Bockmayr. Temporal constraints in the logical analysis of regulatory networks. *Theoretical Computer Science*, 391:258–275, 2008.

[32] C. Soulé. Graphic requirements for multistationarity. *ComPlexUs*, 1:123–133, 2003.

[33] R. Thomas and R. D'Ari. *Biological feedback*. CRC Press, 1990.

[34] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. I. Structural conditions of multistationarity and other nontrivial behavior. *Chaos*, 11(1):170–179, 2001.

[35] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. II. Logical analysis of regulatory networks in terms of feedback circuits. *Chaos*, 11(1):180–195, 2001.

[36] L. Tournier. *Etude et modélisation mathématique de réseaux de régulation génétique et métabolique*. PhD thesis, Laboratoire Jean Kuntzmann (LJK-IMAG), 2005.

[37] A. Wuensche. Basins of attraction in network dynamics: a conceptual framework for biomolecular networks. In G. Schlosser and G.P. Wagner, editors, *Modularity in development and evolution*, chapter 13, pages 288–311. Chicago University Press, 2002.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399