



**HAL**  
open science

# Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery

Célia Biane, Franck Delaplace

► **To cite this version:**

Célia Biane, Franck Delaplace. Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019, 16 (5), pp.1574–1585. 10.1109/TCBB.2018.2889102 . hal-02264665

**HAL Id: hal-02264665**

**<https://hal.science/hal-02264665>**

Submitted on 7 Aug 2019

**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.

# Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery

Célia Biane and Franck Delaplace

**Abstract**—Complex diseases such as Cancer or Alzheimer’s are caused by multiple molecular perturbations leading to pathological cellular behavior. However, the identification of disease-induced molecular perturbations and subsequent development of efficient therapies are challenged by the complexity of the genotype-phenotype relationship. Accordingly, a key issue is to develop frameworks relating molecular perturbations and drug effects to their consequences on cellular phenotypes. Such framework would aim at identifying the sets of causal molecular factors leading to phenotypic reprogramming. In this article, we propose a theoretical framework, called *Boolean Control Networks*, where disease-induced molecular perturbations and drug actions are seen as topological perturbations/actions on molecular networks leading to cell phenotype reprogramming. We present a new method using abductive reasoning principles inferring the minimal causal topological actions leading to an expected behavior at stable state. Then, we compare different implementations of the algorithm and finally, show a proof-of-concept of the approach on a model of network regulating the proliferation/apoptosis switch in breast cancer by automatically discovering driver genes and their synthetic lethal drug target partner.

**Keywords**—Boolean control network, therapeutic target inference, cell reprogramming, abductive reasoning.

## I. INTRODUCTION

THE technological advances in molecular biology of the last two decades have led to the emergence of a new biomedical paradigm: precision medicine. This new field aims at redefining diseases and their treatments at molecular level in order to guide clinical decision making toward the most appropriate drug for the patient. However, the translation of technological advances into medical applications faces numerous challenges [1], [2]. First of all, the transformation of molecular data into biological knowledge requires integrative computational frameworks enabling a better understanding of the complex relationship linking genotypes and phenotypes. Subsequently, the formalisation and the automatization of reasoning over this biological knowledge appear necessary to guide the decision-making for new therapeutic strategies.

Complex diseases such as Cancer or Alzheimer’s are caused by diverse and multiple molecular perturbations leading to pathological cellular behavior. Such diversity and multiplicity challenge the identification of the perturbations causing the transformation of a healthy cell into a diseased one through data analysis. Molecular network modelling underlies the integrative, systems-level understanding of cells by linking molecules (genes, proteins, RNAs, metabolites) and their interactions to behaviors, interpreted as the dynamical states

emerging from these interactions. As a matter of fact, cells may be envisioned as webs of interacting macromolecules constituting the “interactome” and behavioral reprogramming observed in the course of diseases is explained by the perturbations of molecular interactions [3]. Therefore, the study of cell fate reprogramming through the lens of molecular networks constitutes the root for disease molecular etiology investigation and subsequent development of efficient therapies.

Main issues for integrative framework development then concerns the definition of the nature of the perturbations responsible for cell fate reprogramming and their effects. In biology and medicine, phenotypic changes are assessed by the measure of the state of some molecules, called *biomarkers*, that are defined as observable and objective characteristics of biological processes and used to characterize the shift between normal and pathological conditions [4], [5]. In [6], the authors relate mutations to their network effect and introduce the notion of edgetic perturbations of molecular networks: nonsense mutation, out-of-frame insertion or deletion and defective splicing are interpreted as node or arc deletions whereas missense mutation and in-frame insertion or deletion can be modelled as node or arc addition. Moreover, in [7], the authors classify mutations according to the way they affect signalling networks and distinguish mutations that constitutively activate or inhibit enzymes and mutations that rewire the network interactions. The effect of mutations on molecular networks can thus be described as elementary topological actions of deletion or insertion of nodes and arcs. Symmetrically, targeted therapies switch cancer cells phenotype toward growth arrest and apoptosis. Their actions can also be interpreted as network rewiring [5].

A phenotypic switch following disease-induced molecular perturbations or targeted therapies is therefore considered as the observable trait of a *dynamical system reprogramming* caused by *topological network actions* (TN-action). Therefore the identification of the TN-actions that are responsible for dynamical reprogramming would provide major insights for etiological investigation of disease, molecular pathogenesis and drug targets prediction by equating them to the causal genetic perturbations (*a.k.a.*, drivers) or actions of drugs. It is worth noticing that generate-and-test method checking the TN-actions exhaustively is often pointless: assuming that an expected phenotypic switch results from the application of a specific gene action up to  $m$  among  $n$  genes, then the number

of trials<sup>1</sup> equals  $\sum_{k=1}^m \binom{n}{k}$ . For example, the number of trials for targeting up to 10% on 100 genes exceeds 19 billions<sup>2</sup>. Hence, automatic inference of TN-actions is essential to meet this challenge.

In this paper, we define a theoretical framework called *Boolean control network* that relates the TN-actions to their dynamical effects. Then, we propose to use algorithmic approaches borrowed from artificial intelligence field and more specifically diagnosticability analysis to infer the TN-actions responsible for specified observable effects.

The article is organised as follows: first, we define the Boolean control network framework (Section II), then we specify the reprogramming modalities and constraints on solutions and show their resolution based on abductive reasoning (Section III), in Section IV, we present different algorithms computing the solutions and compare their performances and finally, we present a proof-of-concept of this approach for the identification of driver genes and synthetic lethal drugs on a Boolean model of breast cancer (Section V).

## II. BOOLEAN CONTROL NETWORK

In this section we first review the main theoretical elements used in this article, namely: propositional logic, Boolean network and introduce Boolean control network.

### A. Propositional logic

A propositional formula is inductively constructed from atoms composed of constants (False/0, True/1) and variables  $V$ , unary negation operator  $\neg$ , and binary logical operators (e.g.,  $\wedge$ /conjunction/AND,  $\vee$ /disjunction/OR). A *literal* is either an atom or its negation. Given a formula  $f$ ,  $V(f)$  denotes the set of variables occurring in  $f$ . For example, let  $f_\alpha$  be the propositional formula representing the exclusive OR between atom  $x_1$  and the negation of atom  $x_2$ ,  $f_\alpha = (x_1 \vee \neg x_2)$ , the variables are  $V(f_\alpha) = \{x_1, x_2\}$  and the literals are  $x_1$  and  $\neg x_2$ . Let  $X' \subseteq X$  be a subset of variables  $f_{\downarrow X'}$  is the restriction of a formula  $f$  to the literals involving the variables of  $X'$ .

A *cube* syntactically denotes a conjunction of literals and a *clause* a disjunction. In this article, cubes and clauses will be considered as literal sets when needed. A *disjunctive normal form* (DNF) of a formula is a disjunction of cubes (ie.,  $\bigvee_i \bigwedge_{j_i} l_{j_i}$ ) whereas a *conjunctive normal form* (CNF) is a conjunction of clauses (ie.,  $\bigwedge_{j_i} \bigvee_{l_{j_i}}$ ). Any formula can be transformed in DNF or in CNF. For example, a DNF of  $f_\alpha$  is  $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2)$  and a CNF is  $(\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)$ .

Let an *interpretation*  $I : V \rightarrow \{0,1\}$  be a mapping assigning a truth value to each variable<sup>3</sup>, a *model* of a formula  $f$ ,  $I \models f$ , is an interpretation such that the formula is evaluated to True and a *satisfiable* formula admits a model at least. For example,  $f_\alpha$  is satisfiable because the interpretations

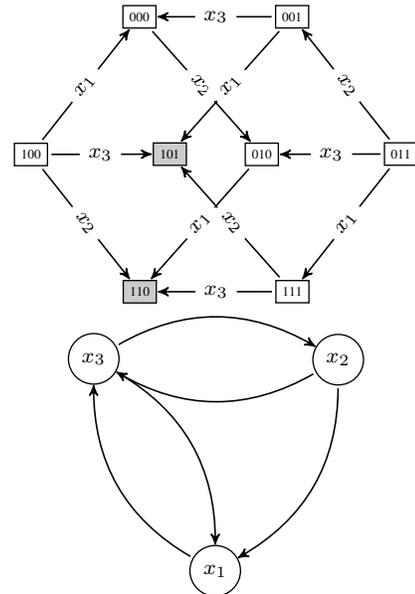
$I_1 = \{x_1 = 1, x_2 = 1\}$  and  $I_2 = \{x_1 = 0, x_2 = 0\}$  are both models of  $f_\alpha$ .

Formula  $f_1$  *entails* formula  $f_2$ , denoted by  $f_1 \models f_2$ , if and only if any model of  $f_1$  is also a model of  $f_2$  (ie.,  $f_1 \models f_2 \stackrel{\text{def}}{=} \forall I : I \models f_1 \implies I \models f_2$ ). Hence, the entailment defines a partial order on formulas.

A *minterm*  $C_I$  of an interpretation  $I$  is the unique cube such that  $V(I) = V(C_I)$  fulfilling  $I \models C_I$ . For the example  $C_1 = x_1 \wedge x_2$  and  $C_2 = \neg x_1 \wedge \neg x_2$  are the minterms of  $I_1$  and  $I_2$  respectively. A cube  $C$  entailing a formula  $f$  is said an *implicant* of  $f$  and it is *prime* if it ceases to be one when deprived of any literal. Considering the example,  $C_1, C_2$  are both prime implicants of  $f_\alpha$  with  $I_1$  and  $I_2$  as model respectively, thus entailing  $f_\alpha$ :  $C_1 \models f_\alpha, C_2 \models f_\alpha$ . Notice that by contrast to a minterm, an implicant does not necessary involve all the variables of the formula (e.g.,  $x_1$  is an implicant of  $(x_1 \vee x_2) \wedge (x_1 \vee x_3)$ ).

### B. Boolean network

The formalism of Boolean networks have been introduced for modelling the dynamics of gene regulatory networks by Stuart Kauffman [8] and René Thomas [9]. A *Boolean network* is a discrete dynamical system operating on Boolean variables  $X$  that determines the *state* evolution of variables  $x_i \in X$ . It is defined as a system of Boolean equations of the form:  $x_i = f_i(x_1, \dots, x_n), 1 \leq i \leq n$  where each  $f_i$  is a propositional formula. A Boolean state of  $s$  is an interpretation of the variables (ie.,  $s : X \rightarrow \mathbb{B}$ ) and  $S_X$  will denote the set of all states for variables of  $X$ .



$$F = \{x_1 = x_2 \vee x_3, x_2 = \neg x_3, x_3 = \neg x_2 \wedge x_1\}$$

Fig. 1. Model of asynchronous dynamics and interaction graph of  $F$ .

A Boolean network *model of dynamics* describes exhaustively all the trajectories of the system where a trajectory corresponds to a path in a labeled transition system. For each

<sup>1</sup>corresponding to the number of parts of size 1 to  $m$  in a set with  $n$  elements.

<sup>2</sup>Exactly 19 415 908 147 835 trials.

<sup>3</sup>A mapping will be described  $x = v$  instead of  $x \mapsto v$  for the sake of simplicity.

transition the states of agents are updated with respect to a pre-defined updating policy. For the *asynchronous* updating used in the article, one agent only is updated per transition. Hence, the labelled transition system for the asynchronous updating is  $\langle \longrightarrow, X, \mathbb{B}^n \rangle$  where the transition relation  $\longrightarrow \subseteq S_X \times X \times S_X$  is labelled by the updated agent,  $\xrightarrow{x_i}$  such that:

$$\begin{aligned} s_1 \xrightarrow{x_i} s_2 &\stackrel{\text{def}}{=} s_1 \neq s_2 \\ &\wedge s_2(x_i) = f_i(s_1) \\ &\wedge \forall x_j \in X \setminus \{x_i\} : s_2(x_j) = s_1(x_j) \end{aligned}$$

We denote  $\longrightarrow = \bigcup_{x_i \in X} \xrightarrow{x_i}$ . A state  $s_2$  is said *reachable* from state  $s_1$  if and only if there exists a trajectory defined by the reflexive and transitive closure of the transition relation connecting  $s_1$  to  $s_2$ ,  $s_1 \longrightarrow^* s_2$ .

A state  $s$  is an *equilibrium* for  $\longrightarrow$  if it is infinitely often reached once met, formally:

$$\forall s' \in S_X : s \longrightarrow^* s' \implies s' \longrightarrow^* s.$$

An *attractor* is a set of equilibria that are mutually reachable and from which no other states can be reached but these equilibria. Formally the states in an attractor  $S' \subseteq S$  complies to:

$$\forall s, s' \in S' : s \longrightarrow^* s'.$$

Hence, an attractor corresponds to a terminal strongly connected component in the transition graph depicting the model of dynamics.

A *stable state* or *one-state attractor* is an attractor of cardinality 1. In Figure 1, the states 101 and 110 in grey are stable. Stable states remain identical whatever the updating policy as they comply to Definition 1:

$$\text{STBL}_F(s) \stackrel{\text{def}}{=} \forall 1 \leq i \leq n : f_i(s) = s(x_i) \quad (1)$$

An *interaction graph*  $\langle X, \rightarrow \rangle$  portrays the directed interactions between variables of a Boolean network (cf., Figure 1) representing the causal dependencies between variables. An interaction  $x_i \rightarrow x_j$  exists if and only if  $x_i$  occurs as literal in a minimal DNF form of  $f_j$ , ie.,

$$x_i \rightarrow x_j \stackrel{\text{def}}{=} x_i \in V(\text{DNF}(f_j)).$$

### C. Boolean Control Network

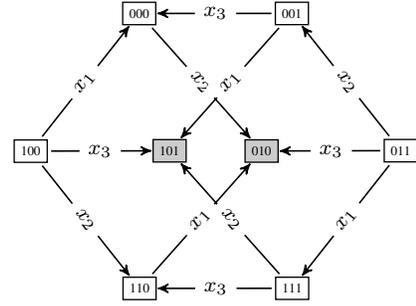
Boolean Control Network (BCN) extends Boolean network by adding *control parameters* that are Boolean variables,  $u_i \in U$  without equation definition. Hence, a BCN is defined as a function generating Boolean network parametrized by an interpretation of control parameters  $\mu \in S_U$ , called a *control input*:  $F_\mu : S_U \rightarrow (S_X \rightarrow S_X)$ . For example, an extension of the Boolean network in Figure 1 to a BCN by adding four control parameters  $u_1, u_2, u_3, u_4$  is:

$$F_{u_1, u_2, u_3, u_4} = \begin{cases} x_1 = (x_2 \wedge u_1) \vee x_3, \\ x_2 = \neg(x_3 \vee \neg u_2), \\ x_3 = ((\neg x_2 \wedge x_1) \vee \neg u_3) \wedge u_4 \end{cases} \quad (2)$$

The application of a control input  $\mu$  to a Boolean control network  $F_\mu$  therefore reprograms the dynamics. Figure 2

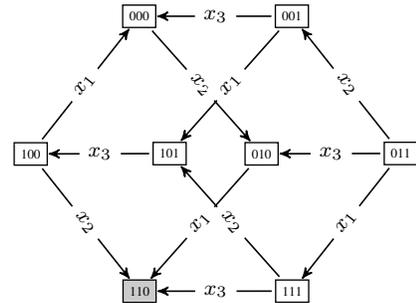
describes the dynamics resulting from the application<sup>4</sup> of two control inputs  $\mu_1 = \{u_1 = 0, u_2 = 1, u_3 = 1, u_4 = 1\}$  and  $\mu_2 = \{u_1 = 1, u_2 = 1, u_3 = 1, u_4 = 0\}$ .

$$F_{\mu_1} = \begin{cases} x_1 = x_3, \\ x_2 = \neg x_3, \\ x_3 = \neg x_2 \wedge x_1 \end{cases}$$



$$\mu_1 = \{u_1 = 0, u_2 = 1, u_3 = 1, u_4 = 1\}$$

$$F_{\mu_2} = \begin{cases} x_1 = x_2 \vee x_3, \\ x_2 = \neg x_3, \\ x_3 = 1 \end{cases}$$



$$\mu_2 = \{u_1 = 1, u_2 = 1, u_3 = 1, u_4 = 0\}$$

Fig. 2. Modification of the dynamics by control inputs  $\mu_1$  and  $\mu_2$  for the example of Figure 1.

Boolean control network provides a general framework for dynamical system reprogramming. Indeed, let  $F$  be an initial Boolean network reprogrammed into an other Boolean network  $G$  where the equations are modified, then the Boolean control network  $F_u = (u \wedge F) \vee (\neg u \wedge G)$  behaves as  $F$  if  $u = 1$  and as  $G$  if  $u = 0$ . The switch between  $F$  and its reprogramming  $G$  now depends on the value of  $u$  only.

<sup>4</sup>The formulas resulting from the instantiation of the BCN by a control input are simplified.

This encoding can be trivially extended to address a family of dynamical systems viewed as the different outcomes of reprogramming by triggering each particular system from a particular valuation of several control parameters, *e.g.*,

$$F_{u_1, u_2} = (u_1 \wedge u_2 \wedge F) \vee (\neg u_1 \wedge u_2 \wedge G_1) \\ \vee (u_1 \wedge \neg u_2 \wedge G_2) \vee (\neg u_1 \wedge \neg u_2 \wedge G_3)$$

with  $G_1, G_2, G_3$  as reprogramming outcomes. However, the control will be practically specified in another way in order to represent the effective control operated in the real system (Section II-D).

Finally, a Boolean control network can be associated to a *control constraint*  $\Phi : U^m \rightarrow \mathbb{B}$  fixing the allowed control inputs.

#### D. Control-freezing category

Among different possibilities to control a Boolean network, we focus on a particular category called *control-freezing* where the control action fixes (freezes) variable states to a specific value. We define two categories of control actions: *Definition-freezing* (D-freezing) that controls the definition of a variable and *Use-freezing* (U-freezing) controlling the use of a variable in an equation defining another variable. The D-freezing parameter governing the freeze of variable  $x_i$  will be denoted  $d_i$  and the U-freezing parameter is denoted  $u_{i,j}$  standing for the control by freeze of the variable  $x_i$  in its use in  $f_j$ . Moreover, each control parameter has two distinct regimes: either it freezes the variable to a specific value or leaves it unchanged. The convention, inspired by the freezing temperature of water  $0^\circ\text{C}$ , is as follows: the freezing action is triggered when the control parameter is set to 0 whereas a control parameter set to 1 is inactive.

*Control-freezing implementation to Boolean network.*:  $d_i^0$  is the control parameter freezing the variable state  $x_i$  to 0 and  $u_{i,j}^0$  for the variable  $x_i$  occurring in  $f_j$ . Similarly,  $d_i^1, u_{i,j}^1$  stands for 1-freezing with the same convention. For example, by considering the following controlled equation  $x_1 = (\neg x_2) \wedge d_1^0$ ,  $d_1^0$  will freeze  $x_1$  to 0 if  $d_1^0 = 0$  otherwise  $x_1$  behaves as the negation of  $x_2$  (See also (7)). The implementation of the freezing control on a Boolean network extends the formulas to obtain the expected control behavior depending on the type of control parameters:  $D^0, D^1$  or  $U^0, U^1$ .

*a) D-freezing control implementation.*: The D-freezing control of variable  $x_i$  consists in adding a D-freezing parameter to formula  $f_i$  such that setting  $\mu(d_i^k) = 0, k \in \{0, 1\}$  leads to freeze variable  $x_i$  to  $k$  or leaves it unchanged otherwise ( $\mu(d_i^k) = 1$ ). Formula  $f_i$  is completed according to this control behavior:

$$x_i = f_i(x_1, \dots, x_n) \wedge d_i^0 \quad \text{for freezing to 0} \quad (3)$$

$$x_i = f_i(x_1, \dots, x_n) \vee \neg d_i^1 \quad \text{for freezing to 1} \quad (4)$$

$D^0$  and  $D^1$  freezing parameters can be combined to trigger the freeze to different values. To avoid a contradictory freeze to 0 and 1 simultaneously, the constraint  $\Phi = d_i^0 \vee d_i^1$  is added ensuring the mutual exclusion of the parameter activities.

*b) U-freezing control implementation.*: The U-freezing control application follows the same principles as the D-freezing control but applied on the occurrence of variables in the equations of other variables.

$$x_j = f_j(x_1, \dots, x_i \wedge u_{i,j}^0, \dots, x_n) \quad \text{for freezing to 0} \quad (5)$$

$$x_j = f_j(x_1, \dots, x_i \vee \neg u_{i,j}^1, \dots, x_n) \quad \text{for freezing to 1} \quad (6)$$

Both controls can be also combined with a constraint avoiding to trigger contradictory freezing controls simultaneously (*ie.*,  $\Phi = u_{i,j}^0 \vee u_{i,j}^1$ ).

In Example (2),  $u_1$  is the U-freezing parameter of  $x_2$  to 0 ( $u_1 = u_{2,1}^0$ ) used in  $x_1$  definition,  $u_2$  can be interpreted as the U-freezing parameter of  $x_3$  ( $u_2 = u_{3,2}^1$ ), and  $u_3, u_4$  are the D-freezing parameters of  $x_3$  freezing the variable to 1 and 0 respectively ( $u_3 = d_3^1, u_4 = d_3^0$ ). Consequently, the BCN (2) can be rewritten using the appropriate naming convention as:

$$F_{u_{2,1}^0, u_{3,2}^1, d_3^1, d_3^0} = \begin{cases} x_1 = (x_2 \wedge u_{2,1}^0) \vee x_3, \\ x_2 = \neg(x_3 \vee \neg u_{3,2}^1), \\ x_3 = ((\neg x_2 \wedge x_1) \vee \neg d_3^1) \wedge d_3^0 \end{cases} \quad (7)$$

The control activity is thus fully determined by the parameters assigned to 0 in a control input  $\mu$ . The *set of active control parameters* collect these parameters to trace the control activity (*ie.*,  $\{u_i \in U \mid \mu(u_i) = 0\}$ ). In the sequel  $U$  will represent the set of control parameters indifferently of their action types ( $D$  or  $U$ ) and  $u_i \in U$  a generic freezing control parameter.

*Effects of control-freezing on the interaction graph.*: The control-freezing category models the dynamical aftermaths on Boolean network of the TN-actions on the interaction graph. D-freezing directly assigns an invariant value to variables whereas U-freezing sets locally an invariant value for their use in an equation. The immediate consequence on the interaction graph of a freezing is to disconnect a node from its inputs for D-freezing and to remove an arc for U-freezing. Therefore, D-freezing control models node action whereas U-freezing control models arc action (*cf.*, Section V for their interpretation in biological network).

### III. CONTROL PARAMETERS INFERENCE

In this section we first formally characterize the basic patterns specifying the changes of the observable molecular traits resulting from biological system reprogramming and then, we show that the abductive formulation of finding the causes of these change enable their resolution.

#### Specification of reprogramming modalities and solutions

Molecular state variations will be questioned at equilibrium conditions in a twofold way: either finding a particular property in some stable states, or finding a particular property in all of them. We thus define two modalities: the *possibility of meeting a property* in at least one stable state (PoP) and the *necessity of meeting a property* in all stable states (NoP). Let  $p$  be a Boolean function on states ( $p : S_X \rightarrow \mathbb{B}$ ) standing for

a property, the PoP and NoP inference problems are defined as follows:

Find a control input  $\mu$  fulfilling the constraints of  $\Phi$  such that:

$$\exists s \in S_X : \text{STBL}_{F_\mu}(s) \wedge p(s). \quad (\text{PoP}) \quad (8)$$

$$\forall s \in S_X : \text{STBL}_{F_\mu}(s) \implies p(s). \quad (\text{NoP}) \quad (9)$$

Different control inputs may be suitable as solutions. For instance, gaining stable state 010 for Boolean network of Figure 1 with parameters defined in (7) can be obtained with the following control inputs:

$$\{u_{2,1}^0 = 0, u_{3,2}^1 = 1, d_3^1 = 1, d_3^0 = 1\}$$

$$\{u_{2,1}^0 = 0, u_{3,2}^1 = 1, d_3^1 = 1, d_3^0 = 0\}$$

$$\{u_{2,1}^0 = 0, u_{3,2}^1 = 1, d_3^1 = 0, d_3^0 = 0\}$$

The plurality of solutions raises the question of their interpretation for identifying the root factors causing the expected effects. The causal factors are defined as the essential actions shifting the dynamics to the objective whereas the casual factors behave neutrally and do not interfere with the objective whatever their valuation. Focusing on the active parameters, only  $u_{2,1}^0 = 0$  matters for shifting the dynamics to gain 010 (first solution) since it is shared by all solutions, and without this assignment the system reprogramming fail to reach the expected objective. The other parameters becoming active are casual because they can be set to 0 or 1 without deviating the dynamics to the result.

The set of causal control parameters forms a *core*  $K^*$  defined as a minimal active parameter set under the inclusion which is equivalent to the entailment order for cubes. Considering the example, the core  $K^* = \{u_{2,1}^0\}$  is included in all other active parameter sets.

Several cores may be found for a given problem. For example, three different cores  $\{d_3^1\}, \{u_{2,1}^0\}, \{u_{3,2}^1\}$  enable the loss of equilibrium 110. Hence, the inference algorithm aims at finding all the cores in regards to a reprogramming query formulated by the possibility or the necessity of meeting a property at steady-state.

#### A. Abduction based core inference

Inferring a core corresponds to the determination of control parameters producing an expected effect. In logic finding causes from effects is an abduction problem. Abduction is a method of reasoning proposing hypotheses that provide the best explanation for observable facts in regards to knowledge of the problem constituting the theory [10], [11], [12]. In propositional logic, a cube  $C$  is an abductive explanation of a formula  $f$  formalizing the facts with respect to another formula  $\Phi$  representing the theory if and only if:  $C \wedge \Phi \models f$  and  $C$  is consistent with  $\Phi$  (i.e.,  $\Phi \wedge C$  is satisfied). Finding a parsimonious hypothesis introduces the notion of minimal solution that usually corresponds to a prime implicant. Within this framework, the possibility and the necessity of property (8, 9) must be formulated as abduction problems in propositional logic by considering that  $p$  is a propositional formula. However (8) and (9) properties are not formulated as propositions. In

Lemma 1, Possibility (10) and Necessity (11) restate this formulation in propositional logic.

*Lemma 1: Find a cube  $C_\mu$  such that:*

$$(C_s \wedge C_\mu) \wedge \Phi \models (\text{STBL}_{F_u} \wedge p); \quad (\text{PoP}) \quad (10)$$

$$C_\mu \wedge \Phi \models (\text{STBL}_{F_u} \implies p); \quad (\text{NoP}) \quad (11)$$

where  $C_s$  and  $C_\mu$  are consistent with  $\Phi$ ,  $V(C_\mu) = U$ ,  $V(C_s) = X$  and the stability condition is defined as:

$$\text{STBL}_{F_u} \stackrel{\text{def}}{=} \bigwedge_{i=1}^n (x_i \iff f_i(x_1, \dots, x_n, u_1, \dots, u_m))$$

respectively define the PoP (8) and NoP (9) problems as abductive problems in propositional logic.

*Proof 1 (Lemma 1):* To prove the equivalence of PoP (10) and NoP (11) problems with Definitions (8,9) based on propositional logic, we need to demonstrate that 1) the stability condition can be formulated as a propositional formula 2) we can find an equivalent formulation for these Definitions using abduction on propositional formulas.

1) *Stability condition defined by a propositional formula.:* By definition (1), the stability condition for a BCN involving control parameters  $u = (u_1, \dots, u_m)$  is defined as:  $\forall 1 \leq i \leq n : x_i = f_i(x, u)$ . As the set of equations is finite, the condition can be rephrased as:

$$\bigwedge_{i=1}^n (x_i = f_i(x, u)).$$

As the equivalence  $p_1 \iff p_2$  is satisfied if and only if  $I(p_1) = I(p_2)$ , the equality can be formulated by an equivalence, finally defining the stability condition as:

$$\bigwedge_{i=1}^n (x_i \iff f_i(x, u)).$$

Now we examine the equivalent formulation of PoP and NoP definitions given in (8,9) using abduction on propositional formulas and leading to Definitions (10, 11).

2) *PoP defined as an abductive problem in propositional logic.:* Definition (8) can be reformulated by introducing the entailment as:

$$\exists s \in S_X : (C_s \wedge C_\mu) \wedge \Phi \models \text{STBL}_{F_u} \wedge p,$$

where  $C_s$  and  $C_\mu$  are the minterms of a state  $s$  and a control input  $\mu$  respectively. As  $\text{STBL}_{F_u} \wedge p$  contains state variables and control parameters an implicant  $C$  fulfilling  $C \wedge \Phi \models \text{STBL}_{F_u} \wedge p$  can be divided into  $C = C_s \wedge C_\mu$ .  $C$  can be possibly completed by missing variables if needed.  $C_s$  is the minterm of an interpretation  $s \in S_X$ . Therefore determining the existence of an implicant  $C$  involving the variables of  $X$  and  $U$  is equivalent of proving the existence of its interpretation (i.e.,  $(C \models f) \iff (\exists s, s \cup \mu \models f)$ ). Thus, the existential quantifier of  $s$  can be removed.

3) *NoP defined as abductive problem in propositional logic.*

( $\implies$ ) Definition (9) of NoP can be expressed using formula entailment as:

$$\forall s \in S_X : (C_s \wedge C_\mu) \wedge \Phi \models (\text{STBL}_{F_\mu} \implies p),$$

where  $C_s$  and  $C_\mu$  are the minterms of  $s$  and  $\mu$  respectively. We define  $P = (\text{STBL}_{F_\mu} \implies p)$  and  $s_{-i}$  a state  $s$  deprived of its  $i^{\text{th}}$  component.

Let  $C_\mu$  be a solution, for all states  $s$  there exists a state  $(s_{-i}, \neg s_i)$  such that both fulfil the following  $C_s \wedge C_\mu \wedge \Phi \models P$  and  $(C_{(s_{-i}, \neg s_i)} \wedge C_\mu \wedge \Phi \models P$  respectively as the property holds for all states. By applying the consensus theorem [11], we deduce that  $C_{s_{-i}} \wedge C_\mu$  is also an implicant by removing  $s_i$  since it appears positively and negatively in two implicants. As this simplification can be applied for all states  $s \in S_x$ , we finally have:

$$\forall s_{-i} : (C_{s_{-i}} \wedge C_\mu) \wedge \Phi \models P.$$

By following the same reasoning for all  $s_{-i}$  states with the deprivation of  $s_j, j \neq i$ , we deduce the same conclusion for  $s_{-\{i,j\}}, \forall j \neq i$ . This can be repeated until no variables exist ( $C_\emptyset$ ). As  $C_\emptyset = 1$  by definition, we thus conclude that:

$$C_\mu \wedge \Phi \models P.$$

( $\Leftarrow$ ) Conversely, assume that  $C_\mu \wedge \Phi \models P$  with  $V(C_\mu) = U$  we deduce that for all  $C$  such that  $C$  is consistent with  $\Phi$  we have  $C \wedge C_\mu \wedge \Phi \models P$ . As for all  $s \in S_X, C_s$  only involves variables of  $X$  and  $\Phi$  is a constraint on parameters with variables of  $U$  then  $V(C_s) \cap V(\Phi) = \emptyset$ . Two satisfiable cubes with distinct variables are necessary consistent. Thus, we conclude that:  $\forall s \in S_X : (C_s \wedge C_\mu) \wedge \Phi \models P$ .

In conclusion, for NoP, the solution can always be determined by focusing on control parameters only. ■

In Example (7), the components of the problem for gaining state 010 (Figure 2,  $\mu_1$ ) are:

$$\begin{aligned} \text{STBL}_{F_u} &= x_1 \iff (x_2 \wedge u_{2,1}^0) \vee x_3 \\ &\wedge x_2 \iff \neg(x_3 \vee \neg u_{3,2}^1) \\ &\wedge x_3 \iff ((\neg x_2 \wedge x_1) \vee \neg d_3^1) \wedge d_3^0 \end{aligned}$$

for the stability condition,  $\Phi = d_3^0 \vee d_3^1$  for the exclusive activity of  $d_3$ , and  $p = \neg x_1 \wedge x_2 \wedge \neg x_3$  for the minterm of  $s = 010$ . For the loss of stable state 101 (Figure 2,  $\mu_2$ ), only the property differs, now defined as:  $p = \neg(x_1 \wedge \neg x_2 \wedge x_3)$  corresponding to the negation of the minterm of 101.

### B. Related works

BCN was recently introduced in systems biology to provide the theoretical foundations and computational methods for investigating cell fate reprogramming and therapeutic target discovery. In [13] the authors apply a stuck-at fault model to simulate drug intervention in an acyclic growth factors pathway by a generate-and-test method. stuck-at fault model mimics the defects on combinatorial logic circuit that correspond here to malignant mutations. Based on this model,

authors identify drug actions for single mutations by correcting all possible single faults. This framework was improved by [14] using a Max-SAT based method dedicated to acyclic networks in order to directly compute the control parameter values and final states. Inferring the drug targets on a network is also developed by [15] using algebraic techniques (Gröbner basis) in order to modify the system dynamics for creating or avoiding particular stable states. In [16], the authors propose a heuristic method with the same goal but focused on the control of key-nodes stabilizing “motifs” identifying sub-networks. Finally, we have introduced the principle of the abductive inference of cores for drug target discovery in [17] which is significantly extended here, in particular with the formalization and the generalization of the TN-actions as control freezing, and with a more efficient method for the core inference.

Our approach follows a similar orientation of these works by using BCN for modelling disease and drug actions. By comparison, the target discovery is modelled in an original way as an abductive problem. The resulting framework supports any kind of networks including cycles with actions applied on both nodes and arcs and find multiple targets qualifying the parsimonious TN-actions (cores) reprogramming the system. The proposed algorithm infers the causes of expected properties met at stable states and we formalize their query in a general setting using propositional formulas with the Necessity and Possibility modalities.

## IV. CORES COMPUTATION

For a formula  $f$  specifying the stability condition and the property, core inference consists in finding a satisfiable implicant  $C^*$  fulfilling  $C^* \models f$  that minimizes the number of active control parameters (by convention negative ones,  $\neg u_i$ ) with respect to the inclusion. The resulting core  $K^*$  is trivially deduced by collecting the negative control parameters of  $C^*$ . Computing the set of all cores is an NP-Hard problem. In this section, we present two algorithms enabling the computation of cores and study their scalability on randomly generated Boolean networks for PoP and NoP reprogramming. Both algorithms are adapted from methods developed for prime implicants computation, the first is based on 0-1 Integer Linear Programming (0-1 ILP) and presented in [18] and the second is based on a recursive approach of sum-of-products minimization applied on a binary decision diagram (BDD) representing the formula and adapted from [19].

### A. ILP-Core inference algorithm

The method, called ILP-CORE, operates on a formula  $f$  in CNF and computes the set of all the cores  $\mathcal{K}^*$  using 0 – 1 Integer Linear Programming (0-1 ILP). A 0-1 ILP problem is formulated as:

Minimize  $\sum_{j=1}^h m_j \cdot y_j$  subject to  $\sum_{j=1}^h W_{i,j} \cdot y_j \leq v_i$  for  $1 \leq i \leq r, y \in \{0, 1\}^h$ , where  $y$  is the unknown vector of size  $r$ ,  $m, v$  vectors,  $W$  matrix are the parameters of the problem and  $h$  the number of variables.

The method is based on the translation of the constraints related to core definition into 0-1 ILP constraints such that a solution  $y$  is a binary representation of an implicant  $C^*$ . The

algorithm is outlined in Algorithm 1 and the main steps are fully described in the proof of Theorem 1.

**Function** ILP-CORE( $f$ : CNF formula)

```

(min  $m.y^T, Wy \leq v$ ) = Describe constraints on
core as 0-1 ILP problem ;
//  $C^* \models f$  minimizing the number of
negative control parameters.
 $\mathcal{K}^* = \emptyset$ ;
repeat
   $y =$  Solve (min  $m.y^T, Wy \leq v$ ) with a
  0-1 ILP solver ;
  if a solution  $y$  is found then
     $K^* =$  Collect the negative control
    parameters from  $y$ ;
     $\mathcal{K}^* = \mathcal{K}^* \cup \{K^*\}$  ;
    Exclude all solutions  $K, K^* \subseteq K$  by
    adding constraints to  $Wy \leq v$  ;
  end
until No solution  $y$  is found;
return  $\mathcal{K}^*$  // the set of all cores
end

```

**Algorithm 1:** Outline of the ILP-CORE algorithm.

*Theorem 1:* The ILP-CORE algorithm finds all and only the cores.

*Proof 2 (Theorem 1):* The proof of the Theorem lies in the translation of the different constraints related to the determination of the cores in integer linear constraints.

1) *Translation of constraints on  $C^*$  to linear constraints:* The main issue of the method is to translate the required constraints applied on  $C^*$  as a set of linear constraints.

Let  $f = C_1 \wedge C_2 \wedge \dots \wedge C_r$  be the input formula in CNF where each  $C_i$  is a clause, we define  $L_f = \bigcup_{i=1}^r C_i$  the set of literals appearing in  $f$ . We associate to each literal  $l_j \in L_f$  a  $\{0, 1\}$ -variable denoted  $y_{l_j}$  meaning that the vector  $y$  is indexed by the literals of  $L_f$ .

a) *Objective function:* Let  $\bar{N}$  be the set of control parameters occurring negatively in  $L_f$ , i.e.,  $\bar{N} = \{u_j \in U \mid \neg u_j \in L_f\}$ , as each core should minimize their occurrences, the objective function is the sum of these negative control parameters:

$$\sum_{u_j \in \bar{N}} y_{\neg u_j}$$

b) *Clauses defined by inequalities:* By definition of abduction, if there exists an implicant  $C^*$  of a formula  $f$  which is consistent with a theory, then  $C^*$  is satisfiable and thus also  $f$ . A formula in CNF is satisfiable if and only if all its clauses are satisfiable and a clause is satisfiable if and only if at least one of its literal is satisfiable. Therefore, an implicant of this formula is a cube formed by taking at least one literal from each clause. This condition is formulated by a constraint for each clause, such that:

$$\forall C_i : \sum_{l_j \in C_i} y_{l_j} \geq 1$$

c) *Satisfiability of  $C^*$  defined by inequalities:* As the implicant  $C^*$  is satisfiable, it cannot contain both a literal and its negation leading to an antilogy. Let  $P = \{v_j \in X \cup U \mid v_j \in L_f \wedge \neg v_j \in L_f\}$  be the set of variables occurring both positively and negatively in  $L_f$ , then we have the following constraints excluding at least the positive or the negative literal for the variables of  $P$ :

$$\forall v_j \in P : y_{v_j} + y_{\neg v_j} \leq 1$$

2) *Conversion of  $y$  into a core  $K^*$ :*  $y$  is a binary representation of a cube  $C^*$  where  $y_{l_j} = 0$  means that  $l_j$  does not belong to  $C^*$  and  $y_{l_j} = 1$  means that  $l_j$  is contained in  $C^*$ , namely  $C^* = \{l_i \mid y_{l_i} = 1\}$ . A core  $K^*$ , being the set of negative control parameters in  $C^*$ , is deduced from  $y$  as follows:

$$K^* = \{u_j \in U \mid y_{\neg u_j} = 1\}.$$

3) *Exclusion of all sets including the core as further solutions:* The application of the algorithm computes one solution  $y$  from which a core  $K^*$  is deduced. Thus, we need to exclude it and any set including it to find other solutions. For this, we add the constraints such that a solution cannot contain exactly the same negative control parameters as the found core. Therefore the sum of the  $y$  values of the negative control parameters belonging to the core must be less than its cardinality. This linear constraint is expressed from  $K^*$  as follows:

$$\sum_{u_j \in K^*} y_{\neg u_j} \leq |K^*| - 1.$$

The method is then iterated until no more core is found to finally provide the set of all cores. ■

To properly specify the PoP and NoP resolutions, the method is called with different formulas specifying the query. Applied to PoP (10), the complete formula is passed as parameter since literals of  $C^*$  contain control parameters as well as variables identifying the state. For NoP (11), as  $C^*$  must contain control parameters only, each clause is then restricted to control parameters by removing the literals involving state variables (i.e.,  $x_i \in X$ ). The constraints on control parameters  $\Phi$  are already in CNF form by definition (Section II-D).

$$\text{ILP-CORE}(\text{CNF}(\text{STBL}_{F_u} \wedge p) \wedge \Phi) \quad (\text{PoP})$$

$$\text{ILP-CORE}(\text{CNF}(\text{STBL}_{F_u} \implies p)_{\downarrow U} \wedge \Phi) \quad (\text{NoP})$$

We have tested the scalability of the ILP-CORE method for core inference by considering an arbitrary property ( $p = x_1 \wedge x_2$ ) on random monotonous Boolean networks of different node-size having all their agents controlled (except markers  $x_1$  and  $x_2$ ). To fit to biological networks requirements, the interaction graph is scale-free with an average degree about 3. We computed the mean standard deviation of the computation time for 20 tests for PoP experiments and NoP experiments until networks of size 8, then 10 tests for NoP experiments on networks with larger sizes. We also measured the number of clauses of the formula in CNF representing the constraints and the observations. The computation<sup>5</sup> stopped when the

<sup>5</sup>performed on a HP EliteBook 820 G2 Intel Core (2.30 GHz i5-5300U CPU with 16 Go of memory).

# agents	Computation Time (sec.)		CNF size	
	Mean	Std. dev.	Mean	Std.dev
10	0.0296402	0.0150994	26.1	13.1385
20	0.0530403	0.0415523	30.8	42.8186
30	0.767525	2.24556	167.1	8.32814
40	0.0795605	0.0220037	227.25	7.34757
50	1.38763	4.48424	287.9	10.9732
60	1.35175	5.7259	347.3	13.6925

TABLE I

MEAN AND STANDARD DEVIATION OF COMPUTATION TIME FOR CORE INFERENCE WITH ILP AND NUMBER OF CLAUSES IN THE CNF FORMULA FOR POP

# agents	Computation Time		CNF size	
	Mean	Std. dev.	Mean	Std.dev
6	0.0826805	0.0472362	34.9	14.9628
7	0.146641	0.13613	71.55	34.2936
8	0.634924	0.629443	197.6	106.937
9	3.69098	3.64535	534.2	185.472
10	18.1445	18.1445	1272.9	810.262
11	188.473 (1)	294.761	3539.7	1996.91
12	-	-	-	-

TABLE II

MEAN AND STANDARD DEVIATION OF COMPUTATION TIME FOR CORE INFERENCE WITH ILP AND NUMBER OF CLAUSES IN THE CNF FORMULA FOR NOP

computation of cores from a formula exceeded 15 minutes. The experimental results shown in Table I show that the algorithm is highly scalable for the PoP resolution. However, the NoP resolution is much more costly and it is limited in the size of networks it can handle (see Table II for experimental results, the numbers in parenthesis represent the number of networks for which no solutions were computed after a 15' period). This can (at least partly) be explained by the explosion in the number of clauses of the CNF formula. Note that wide standard deviation are correlated to an absence of solution.

### B. BDD-Necessity procedure

Cores computation for NoP with ILP is limited by the explosion in the number of clauses of the CNF formula. To circumvent this limitation, we have developed an alternative method based on prime implicants computation using reduced ordered Binary Decision Diagrams (ROBDDs). ROBDDs enable a compact and canonical representation of Boolean functions (for background knowledge on ROBDD, see [20] or [21]). This method, called BDD-NECESSITY, consists in representing the formula as a Positive Reduced Ordered Binary Decision Diagram (PROBDD) and computing the prime implicants on this representation.

A PROBDD is a decision tree reduced by three reduction rules: merging of isomorphic subgraphs, elimination of nodes whose both edges points to the same node and elimination of nodes whose high edge point to the terminal node 1 and low edge point to the terminal node 0. The first two rules are common to the ROBDD representation while the last reduction rule is specific to our computation. At worst, the PROBDD and ROBDD representation are equivalent but, in

numerous cases, this supplementary reduction rule reduces the size of the representation of the formula without impairing the computation of cores. Indeed, the rule eliminates positive control literals, that are not part of cores (by definition of cores, that contain only negative control parameters). For instance, the ROBDD and PROBDD of the formula  $f$  specifying the observations and constraints for the necessity of  $p = \neg x_1$  with Boolean network  $F$  presented in Figure 1 are shown in Figure 3.

$$\begin{aligned}
 f = & (\neg d_1^0 \vee d_2^1 \vee d_3^1) \wedge (\neg d_1^0 \vee \neg d_2^0 \vee d_3^1) \wedge (\neg d_1^0 \vee \neg d_2^0 \vee \neg d_3^1) \\
 & \wedge (\neg d_1^0 \vee d_2^1 \vee \neg d_3^1) \wedge (\neg d_2^0 \vee d_1^1 \vee \neg d_3^1) \wedge (d_1^1 \vee d_2^1 \vee \neg d_3^1) \\
 & \wedge (\neg d_1^0 \vee \neg d_3^0 \vee \neg d_2^1) \wedge (\neg d_1^0 \vee \neg d_2^1 \vee d_3^1) \wedge (\neg d_3^0 \vee d_1^1 \vee \neg d_2^1) \\
 & \wedge (d_1^1 \vee \neg d_2^1 \vee d_3^1) \wedge (d_2^0 \vee d_3^0 \vee d_1^1 \vee \neg d_2^1 \vee \neg d_3^1) \\
 & \wedge (d_1^0 \vee d_1^1) \wedge (d_2^0 \vee d_2^1) \wedge (d_3^0 \vee d_3^1)
 \end{aligned}$$

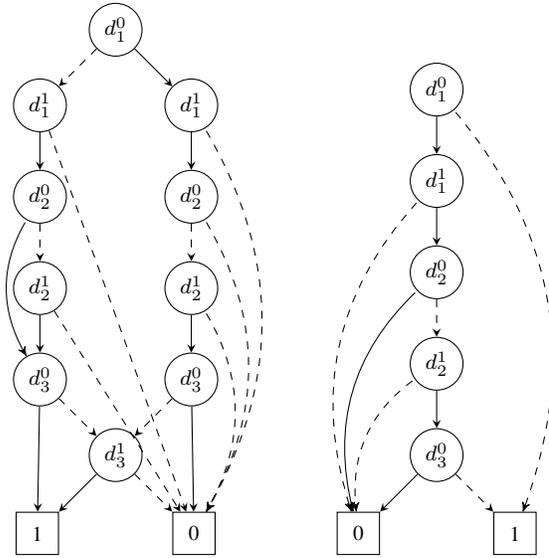


Fig. 3. Formula  $f$  and its ROBDD (left) and PROBDD (right) representation. Dashed arrows represent low edges and full arrows represent high edges.

Then, the prime implicants are computed on this representation of the formula (see [19] for a description of prime implicants computation on ROBDD). The principle underlying the algorithm is that all paths of a ROBDD from the root to the terminal node 1 can be seen as cubes constituting a disjoint cover of the function. A set of implicants of the function can then be found by computing all paths such that a variable is present as its negative literal in the implicant if the path go through its outgoing edge labeled low, as its positive literal if the path go through its outgoing edge labeled high and is absent of the cube if the path does not go through a node representing the variable. Therefore, computing the prime implicants from this representation consist in recursively dividing the PROBDD into three sets: the *low* set, corresponding to the prime implicants containing the negative literal, the *high* set, containing the positive literal and the *don't care* set that do not contain the variable. The don't care set is obtained by the computation of the conjunction of the low and high sub-PROBDD. Then, the low set and the high set are deprived of the don't care set and the same operations are recursively

# agents	Computation Time	
	Mean	Std.dev
8	0.212161	0.0936584
10	1.45237	0.555331
12	15.1898	5.64182
14	71.3143	45.4836
16	435.971	178.016

TABLE III

MEAN AND STANDARD DEVIATION OF COMPUTATION TIME FOR CORE INFERENCE WITH BDD-NECESSITY

applied on the three obtained sets. The recursion ends when the input ROBDD is a constant and the base cases are as follows: the prime set is empty if the input PROBDD is constant 0 and the prime set includes only the empty set when the input ROBDD is constant 1. From the graphical representation of the set of primes, the cores correspond to the sets of low edges in 1-paths. The result of such computation for the ROBDD and PROBDD of Figure 3 are shown in Figure 4.

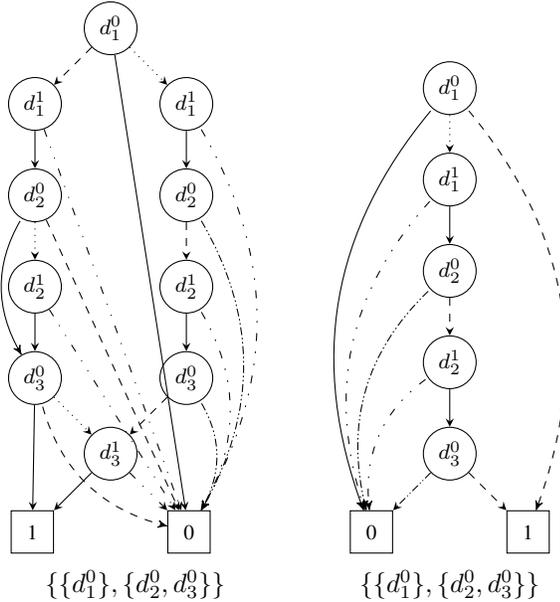


Fig. 4. Graphical representation of the prime implicants of the BDDs in Figure 3 (top) and their corresponding cores (down). Dashed arrows represent low edges, full arrows represent high edges and dotted arrows represent don't care edges.

In Table III, we show the results of the experimental measures of the mean and standard deviation of the cores' computation time of the BDD-Necessity. These results show that the BDD-Necessity procedure is better suited than the procedure based on ILP for the computation of cores for NoP. BDD Necessity can handle Boolean networks until size 16 with a control on all variables definitions, which represent an increase of 45% compared to the ILP method.

In summary, the resolution of PoP queries on formulas in CNF containing numerous variables but few clauses can be efficiently computed with ILP method, while the resolution of NoP queries, based on formulas which contain fewer variables but larger number of clauses, is more efficient on compact representations of formulas, such as BDDs.

## V. APPLICATION TO BREAST CANCER

This section shows the application of TN-actions inference for the study of breast cancer. Mainly, cancer cells differ from normal cells by their uncontrolled proliferation and apoptotic evasion. Accordingly, targeted drugs aim at inducing apoptosis or stop the proliferation of cancer cells [22]. We therefore developed a model (Section V-A) focusing on the regulation of division and apoptosis. We infer the causal TN-actions leading to a loss or gain of apoptosis (Section V-B) and then analyse the results (Section V-C).

### A. Apoptosis/Cell Division Boolean network

The model focuses on the regulation of cell division and apoptosis by the EGFR signalling pathway and a BRCA1/TP53 DNA damage response module. These genes have been identified as central in the process of tumor formation in breast cancer [23], [24]. The model incorporates the positive and negative interactions between nuclear TP53 and MDM2 described by [25], the main messengers of the PI3K/AKT and MAPK signalling following EGFR activation described by [26] and adds BRCA1 and PARP1 regulation of DNA damage. These pathways are gathered into a unique Boolean network through the lens of their role in the regulation of the G1/S transition and the triggering of apoptosis in case of DNA damage. The corresponding Boolean network<sup>6</sup>, constructed from published literature and signalling pathways databases (KEGG [27] and Signor [28]), is shown in Figure 5 and the molecular mechanism for each interaction is detailed and referenced in the conference version of this article [29]. The Boolean dynamics is bistable characterizing two cellular functions in normal cells: either (1) the cell enters division by activation of the G1/S transition and inhibition of apoptosis, or (2) it enters in apoptosis and arrest the cell cycle.

### B. Inference query

As network reprogramming effects biomarker profile changes, it is required to 1) identify the biomarkers discriminating phenotypes and 2) define the reprogramming queries based on these biomarkers for causal genes and drug actions inference.

Since the proliferative activity of cells depends on the balance between division and apoptosis, we selected CYCLIN D1 and BAX as biomarkers as they are the key effector of the G1/S transition of cell division and initiation of apoptosis [30], [31]. The pair (CYCLIN D1, BAX) distinguishes four phenotypes: apoptosis, division, quiescence (apoptosis balanced by division) and dormancy (neither apoptosis nor division) [32] through to the following signatures: (0, 1) for apoptosis, (1, 0) for division, (1, 1) for quiescence and (0, 0) for dormancy. Dormancy and quiescence are cellular states described in Cancer associated to a misbalance between apoptosis and proliferation and whose signaling mechanisms remain poorly known [33].

<sup>6</sup>For the sake of simplicity, the names of genes (by convention written in upper case letters) can also denominate the proteins they encode.

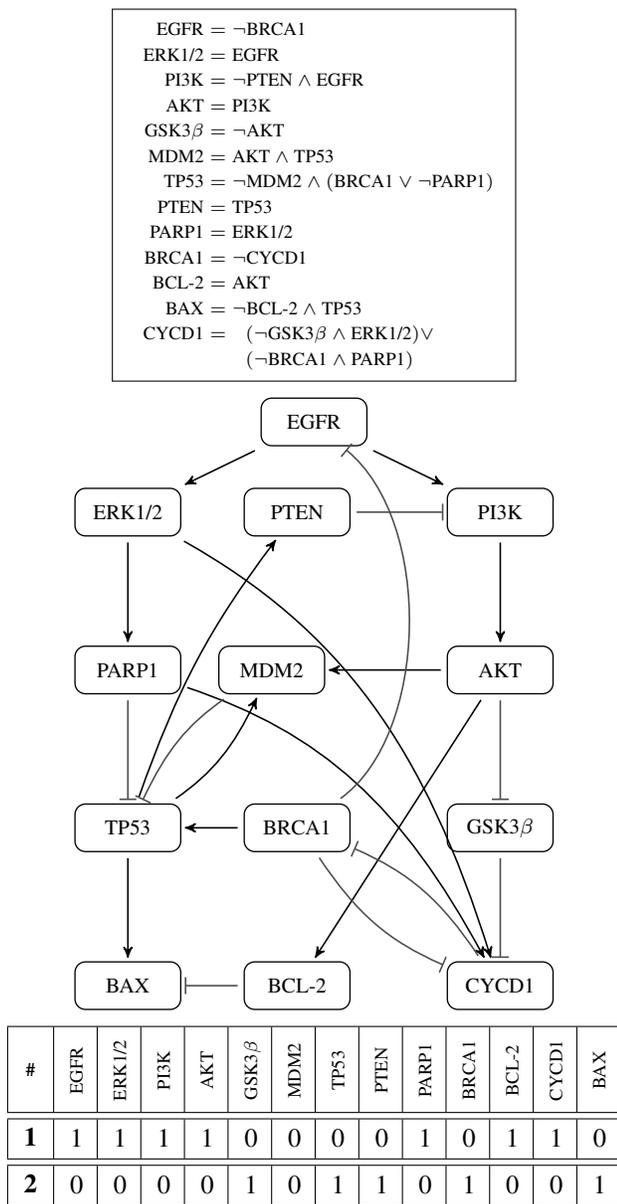


Fig. 5. Boolean network with its regulatory graph representing the activatory (headed arrows) and inhibitory (blunted arrows) interactions, and stable states (at bottom).

Since cancer cells are characterized by their inability to trigger apoptosis, the reprogramming query for the inference of causal genes corresponds to the loss of apoptosis. Conversely, as drugs induce apoptosis in cancer cells, the reprogramming query for the inference of drug actions corresponds to the gain of apoptosis. Apoptosis is formalized as a property by the minterm of (0, 1) signature:  $p = \neg$ CYCD1  $\wedge$  BAX. The loss of apoptosis thus corresponds to the necessity of  $\neg p$  since the apoptosis must not occur in any stable state. To recover this marking, the query can be either the necessity or the possibility of  $p$ . We have tested both and the solutions providing stable states are the same.

Finally, the genetic events are modelled by control parameters as follows: the loss of expression of a gene following loss-of-function mutations or other genetic events such as gene

deletion corresponds to  $D^0$ -freezing; gene over-expression following gain-of-function mutations or other genetic events such as gene amplification are represented by  $D^1$ -freezing; and the loss of interactions between two molecules is interpreted as  $U^0$ -freezing. The Boolean network (Figure 5) is automatically completed with control parameters by following the rules set out in Section II-D. Notice that  $U^1$ -freezing does not seem interpretable in terms of biological events and not used here.

- Health  $\rightarrow$  Cancer: necessary loss of apoptosis -

NODE ACTION	Single $D$ -freezing
	BRCA1 = 0
	TP53 = 0
	PI3K = 1
	AKT = 1
	BCL-2 = 1
	MDM2 = 1
ARC ACTION	Double $D$ -freezing
	GSK3 $\beta$ = 0, ERK1/2 = 1
	PTEN = 0, EGFR = 1
	GSK3 $\beta$ = 0, EGFR = 1
NODE ACTION	Single $U^0$ -freezing
	TP53 $\rightarrow$ BAX
ARC ACTION	Double $U^0$ -freezing
	BRCA1 $\rightarrow$ EGFR, TP53 $\rightarrow$ PTEN
	BRCA1 $\rightarrow$ EGFR, BRCA1 $\rightarrow$ CYCD1
	BRCA1 $\rightarrow$ EGFR, BRCA1 $\rightarrow$ TP53
	BRCA1 $\rightarrow$ EGFR, PTEN $\rightarrow$ PI3K
	BRCA1 $\rightarrow$ EGFR, GSK3 $\beta$ $\rightarrow$ CYCD1

- BRCA1 mutation (Cancer)  $\rightarrow$  Cell death: possible gain of apoptosis -

NODE ACTION	Single $D$ -freezing
	BRCA1 = 1
	PARP1 = 0
	ERK1/2 = 0
	EGFR = 0
ARC ACTION	Single $U^0$ -freezing
	ERK1/2 $\rightarrow$ PARP1
	EGFR $\rightarrow$ ERK1/2
NODE ACTION	Double $U^0$ -freezing
	PARP1 $\rightarrow$ CYCD1, PARP1 $\rightarrow$ TP53

TABLE IV

FREEZING ACTIONS CAUSING THE GAIN OR LOSS OF APOPTOSIS.

### C. Analysis of the results.

We inferred the actions from combination of  $D^0/D^1$ -freezing on all variables (molecules) except markers and the  $U^0$ -freezing on all interactions separately to compare them. The computed TN-actions are shown in Table IV. The TN-actions for the gain of apoptosis have been inferred from the model with BRCA1-deficiency (BRCA1 = 0).

Applied to the loss of apoptosis with  $D$ -freezing, the method retrieves the main driver genes identified in breast cancer namely BRCA1, TP53, PI3K and EGFR [34], [35]. Moreover, it segregates tumor suppressor genes (*ie.*, frequently affected by gain-of-function mutations in cancers) from oncogenes (*ie.*, frequently affected by loss-of-function mutations

in cancers) [36], [37]:  $D^0$ -frozen genes all correspond to tumour suppressors and  $D^1$ -frozen genes to oncogenes. For the gain of apoptosis after application of BRCA1 deficiency, the single D-freezing inferred actions recover the necessity of blocking PARP1, the synthetic lethal partner of BRCA1. The pair BRCA1/PARP1 are called synthetic lethal partners because the use of PARP inhibitors in patients with BRCA1-deficiency prevents any possibility of DNA-repair resulting in permanent DNA damage inducing apoptosis of the cancer cell [38], [39]. Finding such partnerships is critical for anti-cancer treatment [40] but since the cancer target differs from the drug target, they are hard to recover experimentally and computationally.

The algorithm also predicts double D-freezing actions for the necessary loss of apoptosis which suggest that overexpression of EGFR alone would not be sufficient to provoke a cancerous phenotype and must be combined with either loss of PTEN or GSK3 $\beta$ . The validation of such result is less obvious than the former and is based on the concomitant overexpression of EGFR and loss of PTEN/GSK3 $\beta$ . Work in [41] confirms the existence of a co-occurrence of EGFR over-expression and loss of PTEN in 20% of the tumors of the studied population. Moreover, authors also show that PTEN loss is associated to resistance to EGFR inhibitors. This work suggest the existence of the predicted cooperation between these genes. The authors of [42] show that in cell lines resistant to erlotinib (an EGFR inhibitor) the inactive form of GSK3 $\beta$  (p-GSK3 $\beta$ ) is upregulated and the downstream pathways that should be inhibited by its active form are activated. The authors suggest that combinatorial treatment targeting both EGFR signalling and the pathways including GSK3 $\beta$  (Wnt/mTOR) may circumvent the resistance to drugs. Testing such suggestion would require further development of the model by including Wnt/mTOR signaling crosstalk.

It is also predicted that EGFR inhibition would be synthetic lethal with BRCA1 mutations. This is supported by the observation that the proliferation properties of BRCA-deficient cells are sensitive to EGFR inhibition by erlotinib [35]. We found no published work suggesting that ERK1/2 inhibition in such cells would be synthetic lethal.

In summary, in the studied model the method accurately predicts cancerous genes and drug targets and segregate oncogenes from tumor suppressors. The inference also recovers cooperative gene mutations and synthetic lethal partnerships. The double freezing results provide some insights on the necessary cooperative combination of perturbations that are difficult to assess experimentally [43], [44]. Moreover by inferring cores, the method separate causal genes to casual ones (passengers) and determine frequent drivers as well as rare ones which is more difficult to obtain by statistical analysis that prioritize genes from the frequency of their occurrence [45]. Usually, drivers are classified in subtypes where a specific drug target is associated for each subtype. In the proposed approach the drug target may be directly inferred from the application of the TN-actions corresponding to drivers on the initial Boolean network. Finally, arc inference ( $U^0$ -freezing) refines the results on nodes ( $D^0$ -freezing) and, to the best of our knowledge, the resulting predictions are not

experimentally confirmed.

## VI. CONCLUSION

In this article, we have proposed a Boolean control network framework to model perturbations of biological networks. Based on this framework, we have designed a new abduction based inference method that identifies the minimal causes for reprogramming the behavior of the network. A library called ACTONETLIB was developed in Mathematica to support the application on concrete cases. It has been validated on a breast cancer model and has shown that the method can retrieve driver genes and drug targets.

A perspective of this work is to include the notion of resistance in the inference. Two sorts of resistances were established: the primary arising prior to a classical treatment and the secondary which is an adaptive negative response to a treatment. As the method infers all the causes responsible for a biomarker profile shift, the primary resistance is interpreted in our framework as the variation of the input Boolean network of a patient in comparison to a generic one in which the drug targets were deduced. In this context, we need to specialize the network to a patient. The issue for the secondary resistance is more complex and necessitates to predict the further alterations of the network once a TN-action is applied. The prediction of secondary resistance requires to extend the BCN model by including the notion of temporal sequence of control inputs instead of a single control input.

## REFERENCES

- [1] G. S. Ginsburg and H. F. Willard, "Genomic and personalized medicine: foundations and applications," *Translational Research*, vol. 154, no. 6, pp. 277–287, 2009.
- [2] R. Mirnezami, J. Nicholson, and A. Darzi, "Preparing for precision medicine," *New England Journal of Medicine*, vol. 366, pp. 489–491, 2012.
- [3] M. Vidal, M. E. Cusick, and A.-L. Barabási, "Interactome networks and human disease.," *Cell*, vol. 144, no. 6, pp. 986–998, 2011.
- [4] K. Strimbu and J. a. Tavel, "What are Biomarkers?," *Current Opinion in HIV and AIDS*, vol. 5, no. 6, pp. 463–466, 2011.
- [5] P. Csermely, T. Korcsmáros, H. J. M. Kiss, G. London, and R. Nussinov, "Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review," *Pharmacology and Therapeutics*, vol. 138, no. 3, pp. 333–408, 2013.
- [6] Q. Zhong, N. Simonis, Q.-R. Li, B. Charlotiaux, F. Heuze, N. Klitgord, S. Tam, H. Yu, K. Venkatesan, D. Mou, V. Swearingen, M. a. Yildirim, H. Yan, A. Dricot, D. Szeto, C. Lin, T. Hao, C. Fan, S. Milstein, D. Dupuy, R. Brasseur, D. E. Hill, M. E. Cusick, and M. Vidal, "Edgetic perturbation models of human inherited disorders.," *Molecular Systems Biology*, vol. 5, no. 321, p. 321, 2009.
- [7] P. Creixell, E. M. Schoof, C. D. Simpson, J. Longden, C. J. Miller, H. J. Lou, L. Perryman, T. R. Cox, N. Zivanovic, A. Palmeri, A. Wesolowska-Andersen, M. Helmer-Citterich, J. Ferkinghoff-Borg, H. Itamochi, B. Bodenmiller, J. T. Erler, B. E. Turk, and R. Lindig, "Kinome-wide decoding of network-attacking mutations rewiring cancer signaling," *Cell*, vol. 163, no. 1, pp. 202–217, 2015.
- [8] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, pp. 177–178, 1969.
- [9] R. Thomas, "Boolean formalization of genetic control circuits," *Journal of Theoretical Biology*, vol. 42, no. 3, pp. 563–585, 1973.
- [10] C. S. Peirce, "On the natural classification of arguments," in *Proceedings of the American Academy of Arts and Sciences*, vol. 7, pp. 261–287, 1867.
- [11] W. Quine, "On cores and prime implicants of truth functions," *The American Mathematical Monthly*, vol. 66, no. 9, pp. 755–760, 1959.

- [12] P. Marquis, "Extending abduction from propositional logic to first order logic," *Lecture Notes in Artificial Intelligence*, vol. 535, pp. 141–155, 1991.
- [13] R. Layek, A. Datta, M. Bittner, and E. R. Dougherty, "Cancer therapy design based on pathway logic," *Bioinformatics*, vol. 27, no. 4, pp. 548–555, 2011.
- [14] P.-C. K. Lin and S. P. Khatri, "Application of Max-SAT-based ATPG to optimal cancer therapy design.," *BMC Genomics*, vol. 13 Suppl 6, no. Suppl 6, p. S5, 2012.
- [15] D. Murrugarra, A. Veliz-Cuba, B. Aguilar, and R. Laubenbacher, "Identification of control targets in boolean molecular network models via computational algebra," *BMC Systems Biology*, vol. 10, no. 1, p. 94, 2016.
- [16] J. G. Zanudo and R. Albert, "Cell fate reprogramming by control of intracellular network dynamics," *PLoS Computational Biology*, vol. 11, no. 4, p. e1004193, 2015.
- [17] C. Biane, F. Delaplace, and T. Melliti, "Abductive network action inference for targeted therapy," in *Electronic Notes in Theoretical Computer Science*, vol. 335, pp. 3–25, 2016.
- [18] C. Pizzuti, "Computing prime implicants by integer programming," in *Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence*, pp. 332–336, 1996.
- [19] A. Mishchenko, "An introduction to zero-suppressed binary decision diagrams," in *Proceedings of the 12th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning*, vol. 8, pp. 1–15, 2001.
- [20] H. Andersen, "An introduction to binary decision diagrams," *Lecture notes for 49285 Advanced Algorithms E97, available online, IT University of Copenhagen*, 1997.
- [21] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 677–691, 1986.
- [22] D. Hanahan and R. A. Weinberg, "Hallmarks of cancer: the next generation," *Cell*, vol. 144, no. 5, pp. 646–674, 2011.
- [23] W. Kolch, M. Halasz, M. Granovskaya, and B. N. Kholodenko, "The dynamic control of signal transduction networks in cancer cells," *Nature Reviews Cancer*, no. August, 2015.
- [24] S. A. Narod and W. D. Foulkes, "BRCA1 and BRCA2: 1994 and beyond," *Nature Reviews Cancer*, vol. 4, no. 9, pp. 665–676, 2004.
- [25] A. Ciliberto, B. Novák, and J. J. Tyson, "Steady states and oscillations in the p53/Mdm2 network," *Cell Cycle*, vol. 4, no. 3, pp. 488–493, 2005.
- [26] S. Von der Heyde, C. Bender, F. Henjes, J. Sonntag, U. Korf, and T. Beissbarth, "Boolean ErbB network reconstructions and perturbation simulations reveal individual drug response in different breast cancer cell lines," *BMC Systems Biology*, vol. 8, no. 1, p. 75, 2014.
- [27] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima, "KEGG: new perspectives on genomes, pathways, diseases and drugs," *Nucleic Acids Research*, vol. 45, no. D1, pp. D353–D361, 2017.
- [28] L. Perfetto, L. Briganti, A. Calderone, A. C. Perpetuini, M. Iannucelli, F. Langone, L. Licata, M. Marinkovic, A. Mattioni, T. Pavlidou, D. Peluso, L. L. Pettrilli, S. Pirro, D. Posca, E. Santonico, A. Silvestri, F. Spada, L. Castagnoli, and G. Cesareni, "SIGNOR: A database of causal relationships between biological entities," *Nucleic Acids Research*, vol. 44, no. D1, pp. D548–D554, 2016.
- [29] C. Biane and F. Delaplace, "Abduction based drug target discovery using boolean control network," in *Proceedings Computational Methods in Systems Biology - 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27-29, pp. 57–73, 2017.*
- [30] V. Baldin, J. Lukas, M. Marcote, M. Pagano, and G. Draetta, "Cyclin D1 is a nuclear protein required for cell cycle progression in G1.," *Genes & Development*, vol. 7, no. 5, pp. 812–821, 1993.
- [31] S. Gupta, "Molecular signaling in death receptor and mitochondrial pathways of apoptosis," *International Journal of Oncology*, vol. 22, no. 1, pp. 15–20, 2003.
- [32] M. Spiliotaki, D. Mavroudis, K. Kapranou, H. Markomanolaki, G. Kallergi, F. Koinis, K. Kalbakis, V. Georgoulas, and S. Agelaki, "Evaluation of proliferation and apoptosis markers in circulating tumor cells of women with early breast cancer who are candidates for tumor dormancy," *Breast Cancer Research*, vol. 16, no. 6, p. 485, 2014.
- [33] J. Aguirre-Ghiso, "Models, mechanisms and clinical evidence for cancer dormancy," *Nature Reviews Cancer*, vol. 7, no. 11, p. 834, 2007.
- [34] C. Kandoth, M. D. McLellan, F. Vandin, K. Ye, B. Niu, C. Lu, M. Xie, Q. Zhang, J. F. McMichael, M. A. Wyczalkowski, *et al.*, "Mutational landscape and significance across 12 major cancer types," *Nature*, vol. 502, no. 7471, pp. 333–339, 2013.
- [35] L. N. Burga, H. Hu, A. Juvekar, N. M. Tung, S. L. Troyan, E. W. Hofstatter, and G. M. Wulf, "Loss of BRCA1 leads to an increase in epidermal growth factor receptor expression in mammary epithelial cells, and epidermal growth factor receptor inhibition prevents estrogen receptor-negative cancers in BRCA1-mutant mice," *Breast Cancer Research*, vol. 13, no. 2, p. R30, 2011.
- [36] C. M. Croce, "Oncogenes and cancer," *New England Journal of Medicine*, vol. 358, no. 5, pp. 502–511, 2008. PMID: 18234754.
- [37] H. Lodish and S. L. Zipursky, "Molecular cell biology," *Biochemistry and Molecular Biology Education*, vol. 29, pp. 126–133, 2001.
- [38] H. Farmer, N. McCabe, C. J. Lord, A. N. Tutt, D. A. Johnson, T. B. Richardson, M. Santarosa, K. J. Dillon, I. Hickson, C. Knights, *et al.*, "Targeting the DNA repair defect in BRCA mutant cells as a therapeutic strategy," *Nature*, vol. 434, no. 7035, pp. 917–921, 2005.
- [39] L. Livraghi and J. E. Garber, "PARP inhibitors in the management of breast cancer: current data and future prospects," *BMC Medicine*, vol. 13, no. 1, p. 1, 2015.
- [40] W. G. Kaelin, "The concept of synthetic lethality in the context of anticancer therapy.," *Nature Reviews Cancer*, vol. 5, no. 9, pp. 689–98, 2005.
- [41] J. Y. Lee, M. Hong, S. T. Kim, S. H. Park, W. K. Kang, K.-M. Kim, and J. Lee, "The impact of concomitant genomic alterations on treatment outcome for trastuzumab therapy in HER2-positive gastric cancer," *Scientific Reports*, vol. 5, p. 9289, 2015.
- [42] G. M. Botting, I. Rastogi, G. Chhabra, M. Nlend, and N. Puri, "Mechanism of resistance and novel targets mediating resistance to EGFR and c-Met tyrosine kinase inhibitors in non-small cell lung cancer," *PLoS One*, vol. 10, no. 8, p. e0136155, 2015.
- [43] X. Wang, A. Q. Fu, M. E. Mc Nerney, and K. P. White, "Widespread genetic epistasis among cancer genes," *Nature Communications*, vol. 5, 2014.
- [44] P. C. Phillips, "Epistasis - the essential role of gene interactions in the structure and evolution of genetic systems," *Nature Reviews Genetics*, vol. 9, no. 11, pp. 855–867, 2008.
- [45] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, "Cancer genome landscapes," *Science*, vol. 339, no. 6127, pp. 1546–1558, 2013.



**Franck Delaplace** is a full professor since 2005 at Paris-Saclay University, Univ. Evry. After a PhD related to high performance parallel computation, he joins the Evry University in 1994. In 2000, he realized a scientific retraining in computational biology and bioinformatics. His current research topics concern the analysis of dynamical properties of biological networks, and computational network medicine.

**Célia Biane** is currently a post-doctoral fellow in Inria, Irisa, Univ. Rennes. She holds a PhD in computer science of Paris-Saclay University, Univ. Evry and master's degrees in neurosciences (Sorbonne University) and systemic and synthetic biology (Paris-Saclay University, Univ. Evry). Her research interests focus on the development of computational methods for systems biology and medicine and knowledge-based modelling of biological systems.