



**HAL**  
open science

# An in silico target identification using boolean network attractors: avoiding pathological phenotypes

Arnaud Poret, Jean-Pierre Boissel

## ► To cite this version:

Arnaud Poret, Jean-Pierre Boissel. An in silico target identification using boolean network attractors: avoiding pathological phenotypes. 2013. hal-01024788v1

**HAL Id: hal-01024788**

**<https://hal.science/hal-01024788v1>**

Preprint submitted on 16 Jul 2014 (v1), last revised 7 May 2015 (v5)

**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.

An *in silico* target identification using boolean  
network attractors: avoiding pathological  
phenotypes

Arnaud Poret, Jean-Pierre Boissel

November 29, 2013

© 2013 Arnaud Poret, Jean-Pierre Boissel. This document is a preprint and is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

Arnaud Poret (corresponding author)

[arnaud.poret@gmail.com](mailto:arnaud.poret@gmail.com)

Novadiscovery, 60 Avenue Rockefeller, 69008 Lyon, France

UMR CNRS 5558, 43 boulevard du 11 novembre 1918, 69622 Villeurbanne cedex, France

Jean-Pierre Boissel

[jean-pierre.boissel@novadiscovery.com](mailto:jean-pierre.boissel@novadiscovery.com)

Novadiscovery, 60 Avenue Rockefeller, 69008 Lyon, France

**Abstract**

Target identification aims at identifying biomolecules whose the function should be therapeutically modified to cure the considered pathology. An algorithm for *in silico* target identification using boolean network attractors is proposed. It assumes that the attractors of a boolean network correspond to the phenotypes produced by the modeled biological network. It identifies target combinations which allow disturbed biological networks to avoid attractors responsible for pathological phenotypes. The algorithm is tested on a boolean model of the mammalian cell cycle where the retinoblastoma protein is inactivated, as seen in diseases such as cancer. It returns target combinations able to remove attractors responsible for pathological phenotypes. The results show that the algorithm succeeds in performing the proposed *in silico* target identification. However, as with any *in silico* evidence, there is a bridge to cross between theory and practice. Nevertheless, it is expected that the algorithm is of interest for target identification.

**Contents**

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Method</b>	<b>4</b>
2.1	Basic principles . . . . .	4
2.1.1	Biological networks . . . . .	4
2.1.2	Boolean networks . . . . .	5
2.2	Definitions . . . . .	6
2.3	Steps of the algorithm . . . . .	7
2.3.1	Step 1: computing $A_{physio}$ . . . . .	8
2.3.2	Step 2: generating bullets . . . . .	9
2.3.3	Step 3: computing $A_{patho}$ . . . . .	10
2.3.4	Step 4: identifying therapeutic bullets . . . . .	11
2.3.5	Step 5: assessing therapeutic bullets . . . . .	12
2.4	Working example . . . . .	12
2.5	Implementation . . . . .	13
<b>3</b>	<b>Results</b>	<b>13</b>
3.1	Results of step 1 . . . . .	14
3.2	Results of steps 2 to 5 . . . . .	14
<b>4</b>	<b>Conclusion</b>	<b>16</b>

## 1 Introduction

Drug discovery, as its name indicates, aims at discovering new effective drugs against diseases. This process can be segmented into three steps: i) disease model provision, where experimental models are developed, ii) target identification, where new therapeutic targets are proposed and iii) target validation, where the proposed therapeutic targets are assessed. The present work focuses on the second step of drug discovery: target identification [1, 2].

Given an organism suffering from a disease, it aims at finding where to act among its multitude of biomolecules in order to alleviate, or ultimately to cure, the physiological consequences of the disease. These biomolecules on which perturbations should be applied are called targets and are targeted by drugs [3]. This raises two questions: which target should be therapeutically perturbed and what type of perturbation should be applied. Broadly, the functional perturbation of a target by a drug can be either activating or inactivating, regardless the way the drug achieves it.

One solution is to test all, or at least a large number of, biomolecules for activation or inactivation. Knowing that targeting several biomolecules is potentially more effective [4], the number of possibilities is consequently huge. This rather brute force screening can be refined with knowledge about the disease pathophysiology by identifying potential targets based on the role they play in it [5]. Even with this knowledge, experimentally assessing *in vitro* or *in vivo* the selected potential targets is far from being straightforward. Indeed, such experiments are costly in time and resources and exhibit a high risk of failure [6]. Fortunately, *in silico* experiments appear as valuable tools in improving the efficiency of therapeutic researches [7] since they are less costly in time and resources than the traditional *in vitro* and *in vivo* ones. However, the stumbling block of *in silico* experiments is that they are built from the available knowledge about the pathophysiology of interest: not all is known about everything.

Nevertheless, an impressive and ever increasing amount of biological knowledge is already available in the scientific literature, databases and knowledge bases such as, to name a few, DrugBank [8], KEGG [9], PharmGKB [10], Reactome [11] and TTD [12]. In addition to the complexity of integrating an increasing body of knowledge comes the inherent complexity of biological systems themselves [13]: this is where computational tools can help [14]. The interplay between traditional and computational biology is synergistic rather than competing [15]. Since *in vitro* and *in vivo* experiments produce rather factual results, they are a relatively trustworthy source of knowledge. Once these factual pieces of knowledge are obtained, computational tools can help to integrate them and infer new ones. This computationally obtained knowledge can be subsequently used to direct further *in vitro* or *in vivo* experiments, hence mutually potentiating the whole.

The goal of the present work is to propose a computational methodology implemented in an algorithm for target identification using boolean network attractors. It assumes that the attractors of a boolean network correspond to the phenotypes produced by the modeled biological network, an assumption

successfully applied in several works [16, 17, 18, 19, 20, 21] to cite a few. Assuming that a phenotype is an observable and hence a relatively stable state of a biological system and assuming that the state of a biological system is due to the dynamic of the underlying biological network, a phenotype is likely to correspond to an attractor. This assumption can be stated for any dynamical model but, in the present work, only boolean networks are considered. The reasons are that, in their most basic form, boolean networks do not require parameter values [22] and that parameter values are not straightforward to obtain in biology, particularly at the subcellular scale, the scale where drugs interact with their targets. Since synchronous boolean networks are easier to compute than asynchronous ones [23], only synchronous boolean networks are considered. This does not exclude the possibility, at a later stage, to extend the algorithm for both synchronous and asynchronous updating schemes.

For a biological network involved in a disease pathophysiology, two possible variants are considered: the physiological variant, bore by healthy organisms, which produces physiological phenotypes and the pathological variant, bore by ill organisms, which produces pathological phenotypes or which fails to produce physiological ones. A physiological phenotype does not impair the life quantity/quality of the organism which exhibits it while a pathological phenotype does. It should be noted that the loss of a physiological phenotype is also a pathological condition. The physiological and pathological variants differ in that the second one results from the occurrence of some modifications known to be responsible for disorders. With a pathological variant, there are two non exclusives pathological scenarios: pathological phenotypes are gained or physiological phenotypes are lost.

The primary goal of the algorithm is to identify, in the pathological variant, target combinations together with the perturbations to apply on them, here called bullets, which make it unable at producing pathological phenotypes. The secondary goal is to classify the obtained bullets according to their ability at making the pathological variant producing the lost physiological phenotypes, if any.

## 2 Method

This section briefly introduces some basic principles, namely biological networks [24, 25] and boolean networks [26], states some particular definitions and then describes the proposed algorithm. A working example to illustrate it together with its implementation are also described.

### 2.1 Basic principles

#### 2.1.1 Biological networks

A network can be seen as a graph  $G = (V, E)$  where  $V = \{v_1, \dots, v_n\}$  is the set of size  $n$  containing exactly all the nodes  $v_i$  of the network and where  $E = \{(v_{i,1}, v_{j,1}), \dots, (v_{i,m}, v_{j,m})\} \subseteq V^2$  is the set of size  $m$  containing exactly all the

edges  $(v_i, v_j)$  of the network [27, 28]. In practice, nodes represent things and edges represent binary relations  $R \subseteq V^2$  involving these things:  $v_i R v_j$ . For example, in gene regulatory networks, nodes represent gene products and edges represent gene expression modulations [29].

### 2.1.2 Boolean networks

A boolean network is a network where nodes are boolean variables  $x_i$  and where edges  $(x_i, x_j)$  represent the binary *is input of* relation:  $x_i$  is input of  $x_j$ . Each  $x_i$  has  $b_i \in \llbracket 0, n \rrbracket$  inputs  $x_{i,1}, \dots, x_{i,b_i}$ . The variables which are not input of  $x_i$  have no direct influence on it. In the case where  $b_i = 0$ ,  $x_i$  is a parameter, or an input, and does not depend on other variables. At each iteration  $k \in \llbracket k_0, k_{end} \rrbracket$  of the simulation, the value  $x_i(k) \in \{0, 1\}$  of each  $x_i$  is updated to the value  $x_i(k+1)$  thanks to a boolean function  $f_i$  and to the values  $x_{i,1}(k), \dots, x_{i,b_i}(k)$  of its inputs:

```

1 for  $k \in \llbracket k_0, k_{end} - 1 \rrbracket$  do
2    $x_1(k+1) = f_1(x_{1,1}(k), \dots, x_{1,b_1}(k))$ 
3   ...
4    $x_n(k+1) = f_n(x_{n,1}(k), \dots, x_{n,b_n}(k))$ 
5 end for

```

which can be written in a more concise form:

```

1 for  $k \in \llbracket k_0, k_{end} - 1 \rrbracket$  do
2    $\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k))$ 
3 end for

```

where  $\mathbf{f} = (f_1, \dots, f_n)$  is the boolean transition function and  $\mathbf{x} = (x_1, \dots, x_n)$  is the state vector. The value  $\mathbf{x}(k) = (x_1(k), \dots, x_n(k)) \in \{0, 1\}^n$  of  $\mathbf{x}$  at  $k$  belongs to the state space  $S = \{0, 1\}^n$  which is the set of size  $2^n$  containing exactly all the possible states. If the values of all the  $x_i$  are updated simultaneously at each  $k$  then the network is synchronous, otherwise it is asynchronous. With synchronous boolean networks, at each  $k$ ,  $\mathbf{x}(k)$  has a unique possible successor  $\mathbf{x}(k+1)$ : synchronous boolean networks are deterministic.

In the particular case where  $k = k_0$ ,  $\mathbf{x}(k_0) = \mathbf{x}_0$  is the initial state and, in deterministic dynamical systems, determines entirely the trajectory  $w = (\mathbf{x}(k_0), \dots, \mathbf{x}(k_{end}))$ .  $w$  is a sequence of length  $k_{end} - k_0 + 1$  and results from the iterative computation of  $\mathbf{x}(k)$  from  $k_0$  to  $k_{end}$ . This iterative computation can be seen as the discretization of a time interval: boolean networks are discrete dynamical systems as they simulate, discretely, the time course of the state vector.

The set  $A = \{a_1, \dots, a_p\}$  of size  $p$  containing exactly all the attractors  $a_i$  is the attractor set. Due to the determinism of synchronous boolean networks, all the attractors are cycles. A cycle is a sequence  $(\mathbf{x}_1, \dots, \mathbf{x}_q)$  of length  $q$  such that  $\forall j \in \llbracket 1, q \rrbracket$ ,  $\mathbf{x}_{j+1} = \mathbf{f}(\mathbf{x}_j)$  and  $\mathbf{x}_{q+1} = \mathbf{x}_1$ : once the system reaches a state  $\mathbf{x}_j$  belonging to a cycle, it successively visits its states  $\mathbf{x}_{j+1}, \dots, \mathbf{x}_q, \mathbf{x}_1, \dots, \mathbf{x}_j$  for infinity. In the particular case where  $q = 1$ , the cycle is a point attractor.

---

<sup>1</sup>It is assumed that  $k_0 = 1$ .

The set  $B_i \subseteq S$  containing exactly all the  $\mathbf{x} \in S$  from which  $a_i$  can be reached is its basin of attraction. With deterministic dynamical systems, the family of sets  $(B_1, \dots, B_p)$  constitutes a partition of  $S$ .

## 2.2 Definitions

Some particular concepts used in the present work should be formally defined.

- *physiological phenotype*: a phenotype which does not impair the quantity/quality of life of the organism which exhibits it.
- *pathological phenotype*: a phenotype which impairs the quantity/quality of life of the organism which exhibits it.
- *variant (of a biological network)*: a biological network derived from the original one where some modifications occurred in its structure.
- *physiological variant*: a variant which produces only physiological phenotypes.
- *pathological variant*: a variant which produces at least one pathological phenotype.
- *physiological attractor set*: the attractor set  $A_{physio}$  of the physiological variant.
- *pathological attractor set*: the attractor set  $A_{patho}$  of the pathological variant.
- *physiological boolean transition function*: the boolean transition function  $\mathbf{f}_{physio}$  of the physiological variant.
- *pathological boolean transition function*: the boolean transition function  $\mathbf{f}_{patho}$  of the pathological variant.
- *run*: an iterative computation of  $\mathbf{x}(k)$  starting from  $\mathbf{x}_0$  until an  $a_i$  is reached. It returns  $w = (\mathbf{x}(k_0), \dots, \mathbf{x}(k_{end}))$  where  $k_{end}$  depends on when  $a_i$  is reached and hence on  $\mathbf{x}_0$ .
- *physiological attractor*: an  $a_i$  such that  $a_i \in A_{physio}$
- *pathological attractor*: an  $a_i$  such that  $a_i \notin A_{physio}$
- *modality*: the functional perturbation  $moda_i$  applied on an entity  $v_j \in V$  of the network, either activating ( $moda_i = 1$ ) or inactivating ( $moda_i = 0$ ): at each  $k$ ,  $moda_i$  overwrites  $f_j(\mathbf{x}(k))$  and hence  $x_j(k+1) = moda_i$ .
- *target*: an entity  $targ_i \in V$  of the network on which a  $moda_i$  is applied.

- *bullet*: a couple  $(c_{targ}, c_{moda})$  where  $c_{targ} = (targ_1, \dots, targ_r)$  is a combination without repetition of  $targ_i$  and where  $c_{moda} = (moda_1, \dots, moda_r)$  is an arrangement with repetition of  $moda_i$ ,  $r \in \llbracket 1, n \rrbracket$  being the number of targets in the bullet.  $moda_i$  is intended to be applied on  $targ_i$ .
- *therapeutic bullet*: a bullet which makes  $A_{patho} \subseteq A_{physio}$ .
- *silver bullet*: a therapeutic bullet which makes  $A_{patho} \subsetneq A_{physio}$ .
- *golden bullet*: a therapeutic bullet which makes  $A_{patho} = A_{physio}$ .

The assumed link between phenotypes and attractors is the reason why attractors are qualified as either physiological or pathological according to the phenotype they produce. This is also the reason why, in the present work, target identification aims at manipulating the attractor set of the pathological variant.

### 2.3 Steps of the algorithm

The algorithm has two goals: i) finding therapeutic bullets and ii) classifying them as either golden or silver. A therapeutic bullet makes the pathological variant unable at reaching pathological attractors, that is  $A_{patho} \subseteq A_{physio}$ . If such a bullet is applied on the pathological variant, the organism bearing it no longer exhibits the associated pathological phenotypes. However, a therapeutic bullet does not necessarily preserve/restore the physiological attractors. If a therapeutic bullet preserves/restores the physiological attractors, namely if  $A_{patho} = A_{physio}$ , then it is a golden one but if  $A_{patho} \subsetneq A_{physio}$  then it is a silver one.

Given a physiological and a pathological variant, that is  $f_{physio}$  and  $f_{patho}$ , the algorithm follows five steps:

1. with  $f_{physio}$  it computes the control attractor set  $A_{physio}$
2. it generates bullets and, for each of them, it performs the three following steps
3. with  $f_{patho}$  plus the bullet, it computes the variant attractor set  $A_{patho}$
4. it assesses the therapeutic potential of the bullet by comparing  $A_{physio}$  and  $A_{patho}$  to detect pathological attractors
5. if the bullet is a therapeutic one then it classifies it as either golden or silver by comparing  $A_{physio}$  and  $A_{patho}$  for equality

which can be written:

- 1 with  $f_{physio}$  compute  $A_{physio}$
- 2 generate *bullet\_set*
- 3 **for** *bullet*  $\in$  *bullet\_set* **do**
- 4     with  $f_{patho}$  plus *bullet* compute  $A_{patho}$
- 5     **if**  $A_{patho} \subseteq A_{physio}$  **then**

```

6         bullet is therapeutic
7         if  $A_{patho} = A_{physio}$  then
8             bullet is golden
9         else
10            bullet is silver
11        end if
12    end if
13 end for

```

The algorithm is described step by step but can be found in one block in the supplementary information page 20.

### 2.3.1 Step 1: computing $A_{physio}$

First of all,  $A_{physio}$  has to be computed since it is the control and, as such, determines what is pathological and what is not. To do so, runs are performed with  $f_{physio}$  and the reached  $a_i$  are stored in  $A_{physio}$ . However,  $\mathbf{x}_0 \in S$  and  $size(S)$  increases exponentially with  $n$ . Even for reasonable  $n$ ,  $size(S)$  explodes: more than one million possible  $\mathbf{x}_0$  for  $n = 20$ . One solution ensuring that all the possible  $a_i$  are reached is to start a run from each of the possible  $\mathbf{x}_0$ , that is from each of the  $\mathbf{x} \in S$ . Practically, this is unfeasible for an arbitrary value of  $n$  since the required computational resources can be too demanding. For example, assuming that a run requires one millisecond and that  $n = 50$ , performing a run from each of the  $2^{50}$   $\mathbf{x} \in S$  requires nearly 36 thousand years.

Given that with deterministic dynamical systems  $(B_1, \dots, B_p)$  is a partition of  $S$ , a solution is to select a subset  $D \subseteq S$  of a reasonable size containing the  $\mathbf{x}_0$  to start from. In the present work,  $D$  is selected randomly along a uniform distribution. The stumbling block of this solution is that it does not ensure that at least one  $\mathbf{x}_0$  per  $B_i$  is selected and then does not ensure that all the  $a_i$  are reached. This stumbling block holds only if  $size(D) < size(S)$ .

Again given that synchronous boolean networks are deterministic, if a run visits a state already visited in a previous run then its destination, that is the reached attractor, is already found. If so, the run can be stopped and the algorithm can jump to the next one. To implement this, the previous trajectories are stored in a set  $H$ , the history, and at each  $k$  the algorithm checks if  $\exists w \in H : \mathbf{x}(k) \in w$ . If this check is positive then the algorithm jumps to the next run.

Since with deterministic dynamical systems attractors are cycles, the algorithm checks at each  $k$  if  $\mathbf{x}(k+1)$  is an already visited state of the current run, namely if  $\exists k' \in \llbracket 1, k \rrbracket : \mathbf{x}(k+1) = \mathbf{x}(k')$ . If this check is positive then  $a_i = (\mathbf{x}(k'), \dots, \mathbf{x}(k))$ .

This step can be written:

```

1 ask  $size(D)$ 
2  $size(D) = \min(size(D), 2^n)$ 
3 generate  $D \subseteq S$ 
4  $H = \{\}$ 
5  $A_{physio} = \{\}$ 

```

```

6 for  $x_0 \in D$  do
7    $k = 1$ 
8    $\mathbf{x}(k) = x_0$ 
9   while 1 do
10    if  $\exists w \in H : \mathbf{x}(k) \in w$  then
11      break
12    end if
13     $\mathbf{x}(k+1) = \mathbf{f}_{physio}(\mathbf{x}(k))$ 
14    if  $\exists k' \in \llbracket 1, k \rrbracket : \mathbf{x}(k+1) = \mathbf{x}(k')$  then
15       $A_{physio} = A_{physio} \cup \{(\mathbf{x}(k'), \dots, \mathbf{x}(k))\}$ 
16      break
17    end if
18     $k = k + 1$ 
19  end while
20   $H = H \cup \{(\mathbf{x}(1), \dots, \mathbf{x}(k))\}$ 
21 end for
22 print  $A_{physio}$ 
23 do step 2

```

Line 2 catches the mistake  $size(D) > size(S)$ .

The purpose of the present work is not to propose an algorithm for finding boolean network attractors since more sophisticated algorithms for such tasks are already published [30, 31]. The purpose is to exploit boolean network attractors for target identification and, as a consequence, requires these attractors to be found.

### 2.3.2 Step 2: generating bullets

Bullets are candidate perturbations to apply on the pathological variant to make it unable at reaching pathological attractors and hence unable at producing pathological phenotypes. Generating a bullet is to choose some  $targ_i \in V$  and to choose for each of them a  $moda_i \in \{0, 1\}$ . In the present work, there is no time sequencing in target engagement nor in modality application. This means that, given a bullet and during a given run, all the  $targ_i$  are engaged simultaneously and constantly and that the  $moda_i$  do not change. As a consequence, for a given bullet, choosing more than once the same  $targ_i$  is senseless while it is possible to choose the same  $moda_i$  for more than one  $targ_i$ . Therefore, a bullet is a combination  $c_{targ}$  without repetition of  $targ_i$  together with an arrangement  $c_{moda}$  with repetition of  $moda_i$ .

If bullets containing  $r$  targets have to be generated then there are  $n!/(r! \cdot (n-r)!)$  possible  $c_{targ}$  and, for each of them, there are  $2^r$  possible  $c_{moda}$ . This raises the same difficulty than with the state space size explosion since there are  $(n! \cdot 2^r)/(r! \cdot (n-r)!)$  possible bullets. For example, assuming that  $n = 50$  and that  $r = 3$ , there are more than 150 thousand possible bullets. Knowing that the algorithm, as explained below, computes one attractor set per bullet, the computation time becomes practically unfeasible.

To overcome this barrier, the algorithm asks for  $r$  as an interval  $\llbracket r_{min}, r_{max} \rrbracket$ ,

asks for a maximum number  $max_{targ}$  of  $c_{targ}$  to generate and asks for a maximum number  $max_{moda}$  of  $c_{moda}$  to test for each  $c_{targ}$ . Then, the algorithm generates a set  $C_{targ}$  of  $c_{targ}$  with  $size(C_{targ}) \leq max_{targ}$  by randomly selecting, along a uniform distribution and without repetition, entities in the network. In the same way, the algorithm generates a set  $C_{moda}$  of  $c_{moda}$  with  $size(C_{moda}) \leq max_{moda}$  by randomly choosing, along a uniform distribution and with repetition, modalities as either activating ( $= 1$ ) or inactivating ( $= 0$ ).

The result is the bullets: per  $r \in \llbracket r_{min}, r_{max} \rrbracket$ , a  $C_{targ}$  together with a  $C_{moda}$ . As with the state space size explosion, the stumbling block of this method is that it does not ensure that all the possible  $c_{targ}$  together with all the possible  $c_{moda}$  are tested. This stumbling block holds only if  $max_{targ} < n!/(r! \cdot (n-r)!)$  or  $max_{moda} < 2^r$ .

This step can be written:

```

1 ask  $r_{min}, r_{max}, max_{targ}, max_{moda}$ 
2  $r_{max} = \min(r_{max}, n)$ 
3  $golden\_set = \{\}$ 
4  $silver\_set = \{\}$ 
5 for  $r \in \llbracket r_{min}, r_{max} \rrbracket$  do
6    $max_{targ}^r = \min(max_{targ}, n!/(r! \cdot (n-r)!))$ 
7    $max_{moda}^r = \min(max_{moda}, 2^r)$ 
8    $C_{targ} = \{\}$ 
9    $C_{moda} = \{\}$ 
10  while  $size(C_{targ}) < max_{targ}^r$  do
11    generate  $c_{targ} \notin C_{targ}$ 
12     $C_{targ} = C_{targ} \cup \{c_{targ}\}$ 
13  end while
14  while  $size(C_{moda}) < max_{moda}^r$  do
15    generate  $c_{moda} \notin C_{moda}$ 
16     $C_{moda} = C_{moda} \cup \{c_{moda}\}$ 
17  end while
18  do steps 3 to 5
19 end for
20 print  $golden\_set$ 
21 print  $silver\_set$ 

```

Line 2 catches the mistake  $r > n$ . Lines 3 and 4 create the sets in which the therapeutic bullets found in step 4 are classified as either golden or silver in step 5. Lines 6 and 7 catch the mistake where  $max_{targ}$  or  $max_{moda}$  is greater than its maximum, which depends on  $r$ , hence the creation of  $max_{targ}^r$  and  $max_{moda}^r$  to preserve the initially supplied value. Lines 11 and 15 ensure that only new  $c_{targ}$  and  $c_{moda}$  are generated.

### 2.3.3 Step 3: computing $A_{patho}$

Having the control attractor set  $A_{physio}$  and a bullet  $(c_{targ}, c_{moda}) \in C_{targ} \times C_{moda}$ , the algorithm computes the variant attractor set  $A_{patho}$  under the effect of the bullet by almost the same way  $A_{physio}$  is computed in step 1. However, in

this step,  $\mathbf{f}_{patho}$  is used instead of  $\mathbf{f}_{physio}$  and the bullet is applied: at each  $k$ ,  $f_j(\mathbf{x}(k))$  is overwritten by  $moda_i \in c_{moda}$ , that is  $x_j(k+1) = moda_i$ , provided that  $v_j = targ_i \in c_{targ}$ .

In order to apply all the generated bullets, the algorithm uses two nested *for* loops. For each  $c_{targ} \in C_{targ}$  it uses successively all the  $c_{moda} \in C_{moda}$ . For each  $(c_{targ}, c_{moda})$ , the algorithm computes the corresponding  $A_{patho}$  and does steps 4 and 5.

This step can be written:

```

1  for  $c_{targ} \in C_{targ}$  do
2    for  $c_{moda} \in C_{moda}$  do
3       $H = \{\}$ 
4       $A_{patho} = \{\}$ 
5      for  $x_0 \in D$  do
6         $k = 1$ 
7         $\mathbf{x}(k) = x_0$ 
8        while 1 do
9          if  $\exists w \in H : \mathbf{x}(k) \in w$  then
10             break
11          end if
12           $\mathbf{x}(k+1) = \mathbf{f}_{patho}(\mathbf{x}(k))$ 
13          for  $targ_i \in c_{targ}$  do
14            for  $v_j \in V$  do
15              if  $v_j = targ_i$  then
16                 $x_j(k+1) = moda_i$ 
17              end if
18            end for
19          end for
20          if  $\exists k' \in \llbracket 1, k \rrbracket : \mathbf{x}(k+1) = \mathbf{x}(k')$  then
21             $A_{patho} = A_{patho} \cup \{\mathbf{x}(k'), \dots, \mathbf{x}(k)\}$ 
22            break
23          end if
24           $k = k + 1$ 
25        end while
26         $H = H \cup \{\mathbf{x}(1), \dots, \mathbf{x}(k)\}$ 
27      end for
28      do step 4 and 5
29    end for
30  end for

```

Lines 13–19 are where bullets are applied.

### 2.3.4 Step 4: identifying therapeutic bullets

To identify therapeutic bullets among the generated ones, for each  $(c_{targ}, c_{moda})$  tested in step 3 and once the corresponding  $A_{patho}$  is obtained, the algorithm compares it with  $A_{physio}$  to check if  $A_{patho} \subseteq A_{physio}$ . This check ensures that, under the effect of the bullet, all the pathological attractors are removed and

that if new attractors appear then they are physiological ones. If this check is positive then the bullet is therapeutic and the algorithm pursues with step 5.

This step can be written:

```

1 if  $A_{patho} \subseteq A_{physio}$  then
2     do step 5
3 end if

```

### 2.3.5 Step 5: assessing therapeutic bullets

Therapeutic bullets are qualified as either golden or silver according to their ability at making the pathological variant reaching the physiological attractors. All therapeutic bullets, being golden or silver, remove the pathological attractors without creating new ones, that is  $A_{patho} \subseteq A_{physio}$ . However, this does not imply that therapeutic bullets preserve/restore physiological attractors. A golden bullet preserves/restores the physiological attractors:  $A_{patho} = A_{physio}$  while a silver bullet does not:  $A_{patho} \subsetneq A_{physio}$ .

In this setting, golden bullets are perfect therapies while silver bullets are not. However, since precious things are rare and since gold is rarer than silver, finding a golden bullet is less likely than finding a silver one. Indeed, given that more constraints are required for a therapeutic bullet to be a golden one, it is more likely that the found therapeutic bullets are silver ones.

Practically, again in the present setting, an organism bearing a pathological variant treated with a therapeutic bullet no longer exhibits the associated pathological phenotypes. Moreover, if the therapeutic bullet is a golden one, the organism exhibits the same phenotypes than its healthy counterpart. However, if the therapeutic bullet is a silver one, the organism fails to exhibit at least one physiological phenotype. With a silver bullet this is a matter of choice: what is less detrimental between a silver bullet and no therapeutic bullet at all.

This step can be written:

```

1 if  $A_{patho} = A_{physio}$  then
2      $golden\_set = golden\_set \cup \{(c_{targ}, c_{moda})\}$ 
3 else
4      $silver\_set = silver\_set \cup \{(c_{targ}, c_{moda})\}$ 
5 end if

```

## 2.4 Working example

The algorithm is used on a published boolean model of the mammalian cell cycle [18] to test and illustrate it. This model is chosen for several reasons: i) a synchronous updating scheme is performed: to date, the algorithm focuses on synchronous boolean networks, ii) a mammalian biological system is modeled: the closer to human physiology the model is the better it illustrates the intended applications, iii) the cell cycle is at the heart of cancer: this gives relevancy to the working example, iv) the network comprises ten nodes: easily computable in face of its state space and v) the attractors are already computed: useful to validate the algorithm in finding them.

Below are the boolean functions of the working example where, for the sake of readability,  $x_i$  stands for  $x_i(k)$  and  $x_{i+}$  stands for  $x_i(k+1)$ .

$$\begin{aligned}
CycD_+ &= CycD \\
Rb_+ &= (\neg CycD \wedge \neg CycE \wedge \neg CycA \wedge \neg CycB) \vee (p27 \wedge \neg CycD \wedge \neg CycB) \\
E2F_+ &= (\neg Rb \wedge \neg CycA \wedge \neg CycB) \vee (p27 \wedge \neg Rb \wedge \neg CycB) \\
CycE_+ &= E2F \wedge \neg Rb \\
CycA_+ &= (E2F \wedge \neg Rb \wedge \neg Cdc20 \wedge \neg (Cdh1 \wedge UbcH10)) \\
&\quad \vee (CycA \wedge \neg Rb \wedge \neg Cdc20 \wedge \neg (Cdh1 \wedge UbcH10)) \\
p27_+ &= (\neg CycD \wedge \neg CycE \wedge \neg CycA \wedge \neg CycB) \\
&\quad \vee (p27 \wedge \neg (CycE \wedge CycA) \wedge \neg CycB \wedge \neg CycD) \\
Cdc20_+ &= CycB \\
Cdh1_+ &= (\neg CycA \wedge \neg CycB) \vee Cdc20 \vee (p27 \wedge \neg CycB) \\
UbcH10_+ &= \neg Cdh1 \vee (Cdh1 \wedge UbcH10 \wedge (Cdc20 \vee CycA \vee CycB)) \\
CycB_+ &= \neg Cdc20 \wedge \neg Cdh1
\end{aligned}$$

Having the working example, two variants of it are needed: the physiological one and the pathological one. The physiological variant is the network as is while the pathological variant is the network plus a constitutive activation/inactivation of at least one of its entities. For simplicity, and given the relatively small number of entities, only one is chosen: the retinoblastoma protein  $Rb$  for which a constitutive inactivation is applied. To implement this, the corresponding  $f_i$  becomes  $Rb_+ = 0$  in  $\mathbf{f}_{patho}$ .  $Rb$  is chosen because its inactivation occurs in many cancers [32] and, as a consequence, a network bearing a constitutive inactivation of it should be a relevant example of a pathological variant.

## 2.5 Implementation

The algorithm is implemented in Fortran 95 compiled with GFortran<sup>2</sup> for performance and also in GNU Octave<sup>3</sup> for convenience. The code is available on GitHub<sup>4</sup> at <https://github.com/arnaudporet/kali-targ> under the BSD 3-Clause License.

## 3 Results

This section exposes results produced with the algorithm on the working example.

<sup>2</sup><https://www.gnu.org/software/gcc/fortran/>

<sup>3</sup><https://www.gnu.org/software/octave/>

<sup>4</sup><https://github.com/>

### 3.1 Results of step 1

Thanks to the relatively small size of the working example,  $size(D)$  is set to  $size(S) = 1024$ . Since  $size(D) = size(S)$ , all the attractors are found. The algorithm returns the following attractors:

$$\begin{array}{r}
 a_1 = \\
 CycD : \quad 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \\
 Rb : \quad 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 E2F : \quad 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \\
 CycE : \quad 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \\
 CycA : \quad 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \\
 p27 : \quad 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \\
 Cdc20 : \quad 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \\
 Cdh1 : \quad 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \\
 UbcH10 : \quad 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \\
 CycB : \quad 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1
 \end{array}$$

$$\begin{array}{r}
 a_2 = \\
 CycD : \quad 0 \\
 Rb : \quad 1 \\
 E2F : \quad 0 \\
 CycE : \quad 0 \\
 CycA : \quad 0 \\
 p27 : \quad 1 \\
 Cdc20 : \quad 0 \\
 Cdh1 : \quad 1 \\
 UbcH10 : \quad 0 \\
 CycB : \quad 0
 \end{array}$$

Attractors are presented as matrices where, for an attractor of length  $q$ , lines correspond to the  $x_i(k), k \in \llbracket 1, q \rrbracket$  and columns to  $\mathbf{x}(k)$ .  $A_{physio} = \{a_1, a_2\}$  which corresponds to the results obtained by the authors of the working example. By the way,  $a_1$  and  $a_2$  are the two physiological attractors. In term of phenotypes,  $a_1$  corresponds to the accomplishment of a normal cell cycle while  $a_2$  corresponds to quiescence.

### 3.2 Results of steps 2 to 5

Results of steps 2 to 5 are grouped since only therapeutic bullets found in step 4 and classified in step 5 are returned. The algorithm is launched with  $r_{min} = 1$  and  $r_{max} = 2$ . Again due to the relatively small size of the working example,  $max_{targ}$  and  $max_{moda}$  are set to their maximum, namely  $max_{targ} = 45$  and  $max_{moda} = 4$ . As a consequence, all the possible bullets with  $r \in \llbracket 1; 2 \rrbracket$  are tested. The algorithm returns the following therapeutic bullets:

$+CycD$		<i>silver</i>
$+CycD$	$-p27$	<i>silver</i>
$-CycD$	$+Rb$	<i>silver</i>
$+CycD$	$-Rb$	<i>silver</i>

where  $+$  means a therapeutic activation and  $-$  a therapeutic inactivation. It should be noted that no golden bullets are found, a not surprising result since they are rarer than silver ones. Given these results, therapeutic activation of  $Rb$  alone, which is pathologically inactivated, is not enough to remove the pathological attractors. As seen in the third bullet, a therapeutic activation of  $Rb$  must be accompanied by a therapeutic inactivation of  $CycD$ .

To better illustrate what is performed to obtain these therapeutic bullets, below is  $A_{patho}$  without any bullet:

$$a_3 = \begin{array}{l} CycD : \\ Rb : \\ E2F : \\ CycE : \\ CycA : \\ p27 : \\ Cdc20 : \\ Cdh1 : \\ UbcH10 : \\ CycB : \end{array} \begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{array}$$

$$a_4 = \begin{array}{l} CycD : \\ Rb : \\ E2F : \\ CycE : \\ CycA : \\ p27 : \\ Cdc20 : \\ Cdh1 : \\ UbcH10 : \\ CycB : \end{array} \begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{array}$$

$a_4 = a_1 \in A_{physio}$ :  $a_4$  is a physiological attractor. It is possible that the pathological variant produces physiological attractors:  $A_{patho}$  is not the set containing exactly all the pathological attractors, it is the attractor set of the pathological variant and it is possible that  $A_{physio} \cap A_{patho} \neq \emptyset$ . However,  $a_3 \notin A_{physio}$ : it is a pathological attractor. In term of phenotypes,  $a_3$  may correspond to a somewhat degenerated cell cycle and is what a therapeutic bullet, being golden or silver, is intended to avoid.

Again to better illustrate what is performed to obtain the therapeutic bullets, below is  $A_{patho}$  with the third bullet:

$CycD$ :	0
$Rb$ :	1
$E2F$ :	0
$CycE$ :	0
$CycA$ :	0
$p27$ :	1
$Cdc20$ :	0
$Cdh1$ :	1
$UbcH10$ :	0
$CycB$ :	0

which is  $a_2$ . As expected for a therapeutic bullet, the pathological attractor  $a_3$  is removed but the physiological attractor  $a_1$  is not restored: the third bullet is a silver one. Consequently, with this bullet no cell cycle occurs, being normal or degenerated, and the only reachable phenotype is quiescence. While disabling the cell cycle of cancer cells is beneficial, what about disabling the cell cycle of normal cells. As mentioned above, with silver bullets this is a matter of choice.

## 4 Conclusion

Under the assumption stating that dynamical system attractors and biological network phenotypes are linked when the first one models the second one, the results show that the algorithm succeeds in performing the proposed *in silico* target identification. It returns four therapeutic bullets for a pathological variant of the mammalian cell cycle relevant in diseases such as cancer. Consequently, the algorithm can be used on other synchronous boolean models of biological networks involved in disease pathophysiologies for *in silico* target identification. However, both the physiological and pathological variants have to be known: this can constitute a limit in the application of the proposed methodology.

Target identification, being *in silico* or not, is a step belonging to a wider process: drug discovery. Having an *in silico*, or even an *in vitro*, demonstrated potential target is far from having a new drug. Further work and many years are necessary before obtaining a drug which is effective *in vivo*. For example and among other characteristics, such a drug has to be absorbed by the organism, has to reach its target and has to be non toxic at therapeutic dosages. Furthermore, as with any *in silico* evidence, it should be validated *in vitro* and ultimately *in vivo*: there is a bridge to cross between theory and practice. Nevertheless, it is expected that the algorithm is of interest for target identification.

Regarding the possible improvements which could be made on the present work, extending the algorithm for asynchronous boolean networks is important since such models are likely to better approach the dynamic of biological systems. Indeed, in biological systems, events can be subjected to stochasticity, do

not necessarily occur simultaneously or do not belong to the same time scale, three points that a synchronous updating does not take into account. Another possible improvement could be to use finer logics than the boolean one, such as multivalued logics. One of the main limitations of boolean models is that their variables can take only two values. In reality, things are not necessarily binary and variables should be able to take more possible values. Multivalued logics allow it in a discrete manner where variables can take a finite number of values between 0 (false) and 1 (true). For example, one can state that *Rb* is partly inactivated rather than totally. Such a statement is not implementable with boolean models but is with multivalued ones such as, for example, a three valued logic where *true* = 1, *moderate* = 0.5 and *false* = 0.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgments

The authors thank Claudio Monteiro Sousa and François-Henri Boissel for their critical reading of the article manuscript. This work is supported by Novadiscovery<sup>5</sup>.

## References

- [1] Mark A Lindsay. Target discovery. *Nature Reviews Drug Discovery*, 2(10):831–838, 2003.
- [2] Jonathan Knowles and Gianni Gromo. Target selection in drug discovery. *Nature Reviews Drug Discovery*, 2(1):63–69, 2003.
- [3] Peter Imming, Christian Sinning, and Achim Meyer. Drugs, their targets and the nature and number of drug targets. *Nature reviews Drug discovery*, 5(10):821–834, 2006.
- [4] Grant R Zimmermann, Joseph Lehar, and Curtis T Keith. Multi-target therapeutics: when the whole is greater than the sum of the parts. *Drug discovery today*, 12(1):34–42, 2007.
- [5] Jackson B Gibbs. Mechanism-based target identification and drug discovery in cancer research. *Science*, 287(5460):1969–1973, 2000.
- [6] KI Kaitin. Deconstructing the drug development process: the new face of innovation. *Clinical Pharmacology & Therapeutics*, 87(3):356–361, 2010.

---

<sup>5</sup><http://www.novadiscovery.com>

- [7] Denis Noble, Jeremy Levin, and William Scott. Biological simulations in drug discovery. *Drug Discovery Today*, 4(1):10–16, 1999.
- [8] David S Wishart, Craig Knox, An Chi Guo, Dean Cheng, Savita Shrivastava, Dan Tzur, Bijaya Gautam, and Murtaza Hassanali. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research*, 36(suppl 1):D901–D906, 2008.
- [9] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.
- [10] M Whirl-Carrillo, EM McDonagh, JM Hebert, L Gong, K Sangkuhl, CF Thorn, RB Altman, and Teri E Klein. Pharmacogenomics knowledge for personalized medicine. *Clinical Pharmacology & Therapeutics*, 92(4):414–417, 2012.
- [11] David Croft, Gavin O’Kelly, Guanming Wu, Robin Haw, Marc Gillespie, Lisa Matthews, Michael Caudy, Phani Garapati, Gopal Gopinath, Bijay Jassal, et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic acids research*, 39(suppl 1):D691–D697, 2011.
- [12] Xin Chen, Zhi Liang Ji, and Yu Zong Chen. Ttd: therapeutic target database. *Nucleic acids research*, 30(1):412–415, 2002.
- [13] Hiroaki Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002.
- [14] Hiroaki Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.
- [15] Barbara Di Ventura, Caroline Lemerle, Konstantinos Michalodimitrakis, and Luis Serrano. From in vivo to in silico biology and back. *Nature*, 443(7111):527–533, 2006.
- [16] Sui Huang and Donald E Ingber. Shape-dependent control of cell growth, differentiation, and apoptosis: switching between attractors in cell regulatory networks. *Experimental cell research*, 261(1):91–103, 2000.
- [17] Maria I Davidich and Stefan Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PLoS One*, 3(2):e1672, 2008.
- [18] Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.
- [19] Herman F Fumiã and Marcelo L Martins. Boolean network model for cancer pathways: Predicting carcinogenesis and targeted therapy outcomes. *PLoS one*, 8(7):e69008, 2013.

- [20] Pau Creixell, Erwin M Schoof, Janine T Erler, and Rune Linding. Navigating cancer network attractors for tumor-specific therapy. *Nature biotechnology*, 30(9):842–848, 2012.
- [21] Keith Baverstock. A comparison of two cell regulatory models entailing high dimensional attractors representing phenotype. *Progress in Biophysics and Molecular Biology*, 106(2):443–449, 2011.
- [22] Michelle L Wynn, Nikita Consul, Sofia D Merajver, and Santiago Schnell. Logic-based models in systems biology: a predictive and parameter-free network analysis method. *Integrative Biology*, 4(11):1323–1337, 2012.
- [23] Abhishek Garg, Alessandro Di Cara, Ioannis Xenarios, Luis Mendoza, and Giovanni De Micheli. Synchronous versus asynchronous modeling of gene regulatory networks. *Bioinformatics*, 24(17):1917–1925, 2008.
- [24] Xiaowei Zhu, Mark Gerstein, and Michael Snyder. Getting connected: analysis and principles of biological networks. *Genes & development*, 21(9):1010–1024, 2007.
- [25] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell’s functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.
- [26] Stefan Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of the Royal Society Interface*, 5(Suppl 1):S85–S94, 2008.
- [27] Wolfgang Huber, Vincent J Carey, Li Long, Seth Falcon, and Robert Gentleman. Graphs in molecular biology. *BMC bioinformatics*, 8(Suppl 6):S8, 2007.
- [28] Ilía Nikolaevich Bronshtein, Konstantin A Semendyayev, Gerhard Musiol, and Heiner Muehlig. Algorithms of graph theory. In *Handbook of mathematics*, chapter 5, pages 348–359. Springer, 2007.
- [29] Yufei Xiao. A tutorial on analysis and simulation of boolean gene regulatory network models. *Current genomics*, 10(7):511–525, 2009.
- [30] D Zheng, G Yang, X Li, Z Wang, and WNN Hung. An efficient algorithm for finding attractors in synchronous boolean networks with biochemical applications. *Genetics and molecular research: GMR*, 12(4):4656, 2013.
- [31] Elena Dubrova and Maxim Teslenko. A sat-based algorithm for finding attractors in synchronous boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(5):1393–1399, 2011.
- [32] Charles J Sherr and Frank McCormick. The rb and p53 pathways in cancer. *Cancer cell*, 2(2):103–112, 2002.

## Supplementary information

### The algorithm in one block

```

1 ask  $size(D)$ 
2  $size(D) = \min(size(D), 2^n)$ 
3 generate  $D \subseteq S$ 
4  $H = \{\}$ 
5  $A_{physio} = \{\}$ 
6 for  $x_0 \in D$  do
7    $k = 1$ 
8    $\mathbf{x}(k) = x_0$ 
9   while 1 do
10    if  $\exists w \in H : \mathbf{x}(k) \in w$  then
11      break
12    end if
13     $\mathbf{x}(k+1) = \mathbf{f}_{physio}(\mathbf{x}(k))$ 
14    if  $\exists k' \in \llbracket 1, k \rrbracket : \mathbf{x}(k+1) = \mathbf{x}(k')$  then
15       $A_{physio} = A_{physio} \cup \{(\mathbf{x}(k'), \dots, \mathbf{x}(k))\}$ 
16      break
17    end if
18     $k = k + 1$ 
19  end while
20   $H = H \cup \{(\mathbf{x}(1), \dots, \mathbf{x}(k))\}$ 
21 end for
22 print  $A_{physio}$ 
23 ask  $r_{min}, r_{max}, max_{targ}, max_{moda}$ 
24  $r_{max} = \min(r_{max}, n)$ 
25  $golden\_set = \{\}$ 
26  $silver\_set = \{\}$ 
27 for  $r \in \llbracket r_{min}, r_{max} \rrbracket$  do
28    $max_{targ}^r = \min(max_{targ}, n! / (r! \cdot (n-r)!))$ 
29    $max_{moda}^r = \min(max_{moda}, 2^r)$ 
30    $C_{targ} = \{\}$ 
31    $C_{moda} = \{\}$ 
32   while  $size(C_{targ}) < max_{targ}^r$  do
33     generate  $c_{targ} \notin C_{targ}$ 
34      $C_{targ} = C_{targ} \cup \{c_{targ}\}$ 
35   end while
36   while  $size(C_{moda}) < max_{moda}^r$  do
37     generate  $c_{moda} \notin C_{moda}$ 
38      $C_{moda} = C_{moda} \cup \{c_{moda}\}$ 
39   end while
40   for  $c_{targ} \in C_{targ}$  do
41     for  $c_{moda} \in C_{moda}$  do
42        $H = \{\}$ 

```

```

43      $A_{patho} = \{\}$ 
44     for  $x_0 \in D$  do
45          $k = 1$ 
46          $\mathbf{x}(k) = x_0$ 
47         while 1 do
48             if  $\exists w \in H : \mathbf{x}(k) \in w$  then
49                 break
50             end if
51              $\mathbf{x}(k+1) = \mathbf{f}_{patho}(\mathbf{x}(k))$ 
52             for  $targ_i \in c_{targ}$  do
53                 for  $v_j \in V$  do
54                     if  $v_j = targ_i$  then
55                          $x_j(k+1) = moda_i$ 
56                     end if
57                 end for
58             end for
59             if  $\exists k' \in \llbracket 1, k \rrbracket : \mathbf{x}(k+1) = \mathbf{x}(k')$  then
60                  $A_{patho} = A_{patho} \cup \{(\mathbf{x}(k'), \dots, \mathbf{x}(k))\}$ 
61                 break
62             end if
63              $k = k + 1$ 
64         end while
65          $H = H \cup \{(\mathbf{x}(1), \dots, \mathbf{x}(k))\}$ 
66     end for
67     if  $A_{patho} \subseteq A_{physio}$  then
68         if  $A_{patho} = A_{physio}$  then
69              $golden\_set = golden\_set \cup \{(c_{targ}, c_{moda})\}$ 
70         else
71              $silver\_set = silver\_set \cup \{(c_{targ}, c_{moda})\}$ 
72         end if
73     end if
74 end for
75 end for
76 end for
77 print  $golden\_set$ 
78 print  $silver\_set$ 

```