

COMPUTATIONAL METHODS FOR ASYNCHRONOUS BASINS

IAN H DINWOODIE

ABSTRACT. For a Boolean network we consider asynchronous updates and define the exclusive asynchronous basin of attraction for any steady state or cyclic attractor. An algorithm based on commutative algebra is presented to compute the exclusive basin. Finally its use for targeting desirable attractors by selective intervention on network nodes is illustrated with two examples, one cell signalling network and one sensor network measuring human mobility.

1. INTRODUCTION

There are three parts to network modeling in the life sciences: 1) build a dynamical model of interacting molecules or other agents using data, statistical learning, expertise, and literature, 2) classify the realistic attractors or steady state possibilities as uninformative, useful, or undesirable, according to the meaning in the model of the node values of the attractor, and 3) seek settings or interventions on accessible nodes to guarantee termination in a useful attractor or prevent termination in an undesirable configuration. This at least is one scenario for applied network modeling.

This paper gives an example of a 15 node dynamic network of human mobility variables including cognitive impairment to illustrate step 1). The network is based on our particular study of household remote sensor data of motion variables related to human health, a type of data that is sometimes called “life kinetics” data. This model is given as a realistic computational example for the techniques of this paper and not as a conclusive model for the medical or biological sciences. We also give computational methods for problems 2) and 3) that are applied to both a cell signalling example and the life kinetics model.

Networks in biology and life sciences are modeled as dynamical systems in order to understand interactions, causal relationships, and evolution. Boolean or on/off models are the most tractable mathematically and have the most intuitive interpretation. Boolean models continue to be used widely ([15], [20], [23], [24], [27], [36]). Cancer signalling is a common application, and a growing application area is aging and Alzheimer’s disease ([9], [19], [26]), which is related to the life kinetics example of Section 4.2.

To include variability in the timing of the interactions in the network, random perturbations of the dynamics are of interest ([6], [28], [34], [40], [43]). The randomized dynamics are a special case of “probabilistic Boolean networks” [37].

Date: December 2015, revised March 2016.

1991 Mathematics Subject Classification. Primary 37N25; Secondary 13P25, 62M05.

Key words and phrases. asynchronous update; basin of attraction; Boolean network; signalling network; sensor network.

The randomization techniques are sometimes called “asynchronous” dynamics to distinguish them from the plain deterministic dynamics where the coordinate maps are simultaneously or synchronously applied to the present state, mapping all coordinates or nodes together to the next state. The asynchronous systems are Markovian, and bring in probabilistic notions of basin of attraction that extend the traditional definition. Here we will focus on one particular scheme, known sometimes as “general asynchronous,” in which a node j is selected uniformly at random for update using its transition or update function f_j .

This paper defines precisely the exclusive asynchronous basin in Section 2 as the set of points that can never leave a deterministic basin under asynchronous updates. Theorem 2.1 gives useful properties following from the definition, especially one that no point in the exclusive basin can reach any other attractor. In Section 3 we describe algorithms based on commutative algebra that compute and analyze the exclusive asynchronous basin. In Section 4 we apply the algorithms to examples from the life sciences of 13 and 15 nodes which demonstrate the effectiveness of the computational tools. The algebraic computations were done in Singular [8], but other algebra software is also suitable, such as Cocoa [1] and Macaulay2 [14].

2. EXCLUSIVE ASYNCHRONOUS BASIN

Consider a state space $\{0, 1\}^d$, a d -fold product of logical or on/off or high/low symbols 1 and 0. These will be states or configurations on a network with d nodes. Let $F = (f_1, \dots, f_d)$ be a transition map or transition function or update function on $\{0, 1\}^d$, where $f_j : \{0, 1\}^d \rightarrow \{0, 1\}$ is the rule for node j . This map is deterministic, it is the simplified algebraic or logical model of interactions from one time step to the next and F is often called the synchronous update.

Two common choices for randomizing the timing of the updates in F are “random node” (recommended by [34] and [43] where it is called “general asynchronous” or GA), and random order (implemented in [6]). In this work we will only deal with general asynchronous, which we will call simply asynchronous.

Asynchronous updates proceed by choosing a node or coordinate $j \in \{1, 2, \dots, d\}$ with the uniform probability distribution, then changing current state $\mathbf{x} = (x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_d)$ to $\mathbf{x}' = (x_1, \dots, x_{j-1}, f_j(\mathbf{x}), x_{j+1}, \dots, x_d)$. The resulting process $\mathbf{x}_0, X_1, X_2, X_3, \dots$ is a Markov chain in state space $\{0, 1\}^d$, starting at nonrandom initial configuration \mathbf{x}_0 and with random variables X_n for successive random states (here we follow the convention that capital letters denote random variables). The probability that X_n is in a set S will be written:

$$P_{\mathbf{x}_0}(X_n \in S).$$

In theory the process has a probability transition matrix of size $2^d \times 2^d$ but usually it is not practical to work with such matrices directly. Our examples have 13 and 15 nodes respectively, giving cardinalities of 8192 and 32768 for the state space size 2^d . We hope and expect that the algebraic methods in this paper can be applied on examples of up to 50 nodes in the future, as symbolic commutative algebra software gets more powerful.

Let $F = (f_1, \dots, f_d) : \{0, 1\}^d \rightarrow \{0, 1\}^d$ be the transition function on Boolean states $\{0, 1\}^d$. For a state \mathbf{x} , define the limiting set

$$A_{\mathbf{x}} = \bigcap_{k=1}^{\infty} \bigcup_{n=k}^{\infty} \{F^n(\mathbf{x})\}$$

where F^n is the n -fold composition of F . The resulting sets, as \mathbf{x} varies in the state space $\{0, 1\}^d$, are the limiting sets or attractor sets of the system. These are fixed points or cycles, and every state enters one of them and stays there under deterministic dynamics F .

Let B_A be the usual deterministic basin of attraction for an attractor A , the points that eventually hit A :

$$(2.1) \quad B_A := \bigcup_{k=1}^{\infty} \{\mathbf{x} : F^k(\mathbf{x}) \in A\}.$$

Define the single coordinate updates on a state $\mathbf{x} = (x_1, \dots, x_d)$ by

$$F_j(\mathbf{x}) = (x_1, \dots, x_{j-1}, f_j(\mathbf{x}), x_{j+1}, \dots, x_d), \quad j = 1, 2, \dots, d.$$

Now with $B_0 = B_A$, F_j^{-1} the inverse of F_j , and B^c the complement of a set B , define the exclusive asynchronous basin of the attractor A by

$$(2.2) \quad B_i = B_{i-1} \setminus \bigcup_{j=1}^d F_j^{-1}(B_{i-1}^c), \quad i = 1, 2, 3, \dots$$

$$(2.3) \quad B_{ex,A} = \lim_{i \rightarrow \infty} B_i.$$

Probabilistically, this can be described as

$$B_{ex,A} = \{\mathbf{x} : P_{\mathbf{x}}(X_n \in B_A, n = 0, 1, 2, 3, \dots) = 1\}.$$

In words, the points in $B_{ex,A}$ are the states that can never leave the basin of attraction B_A over time under perturbations in the form of asynchronous updates. The sets B_i are the initial states \mathbf{x}_0 that keep the initial segment $\mathbf{x}_0, X_1, X_2, \dots, X_i$ in B_A , and (2.2) gets them recursively by removing the points from B_{i-1} that can move out of B_{i-1} in one initial step. At the end of this section we discuss the definition further.

Note that $B_0 \supset B_1 \supset B_2 \dots$ and the domain $\{0, 1\}^d$ is finite so the limit $B_{ex,A}$ exists, while possibly empty, and will be a fixed point for the set map T :

$$(2.4) \quad T(B) := B \setminus \bigcup_{j=1}^d F_j^{-1}(B^c), \quad i = 1, 2, 3, \dots$$

$$(2.5) \quad B_{ex,A} = T(B_{ex,A}).$$

Our definition of $B_{ex,A}$ starts with the deterministic basin B_A , and this gives useful properties on keeping the asynchronous dynamics out of other attractors.

Theorem 2.1. *If $B_{ex,A}$ is the exclusive asynchronous basin for attractor A then 1) asynchronous or synchronous dynamics starting at any point in $B_{ex,A}$ cannot reach any point of any attractor other than A , 2) $B_{ex,A}$ contains A if the attractor A is a fixed point, 3) if $B_i = B_{i-1}$ in (2.2), then $B_{ex,A} = B_i$.*

Proof. For 1), let $\mathbf{x} \in B_{ex,A} \subset B_A$. Then $F^n(\mathbf{x}) \in B_A, n \geq 0$, and \mathbf{x} can never reach any other attractor under deterministic dynamics F . For the asynchronous dynamics, (2.5) implies

$$B_{ex,A} = B_{ex,A} \bigcap \bigcap_{j=1}^d F_j^{-1}(B_{ex,A})$$

and so $B_{ex,A} \subset \bigcap_{j=1}^d F_j^{-1}(B_{ex,A})$ and $\mathbf{x} \in B_{ex,A}$ means $F_j(\mathbf{x}) \in B_{ex,A}, j = 1, \dots, d$, in other words $P_{\mathbf{x}}(X_1 \in B_{ex,A}) = 1$. Thus \mathbf{x} stays in $B_{ex,A}$ under single-node updates, and $B_{ex,A}$ is disjoint from any other attractor since $B_{ex,A} \subset B_A$. For 2), a fixed point for F is also fixed for the asynchronous updates (that is, if $A = \{\mathbf{x}\}$ and $F(\mathbf{x}) = \mathbf{x}$ then $F_j(\mathbf{x}) = \mathbf{x}$ as well, hence $P_{\mathbf{x}}(X_1 = \mathbf{x}) = 1$) and will never leave B_A . For 3), which means we can stop the iteration as soon as it repeats, observe that if $B_i = B_{i-1}$ then $B_i = B_{i+1} = B_{i+2} = \dots$ and the limit is B_i . \square

Note that $B_{ex,A}$ may be empty if A is a cycle. A simple example is useful to understand the definition and its consequences. Consider the state space diagram in Figure 1 for dynamics given in logical form by $f_1(x_1, x_2) = x_1 \text{ xor } x_2$, $f_2(x_1, x_2) = x_1$. Here $B_{00} = \{00\}$, and this is also the exclusive basin for the fixed point 00. Now

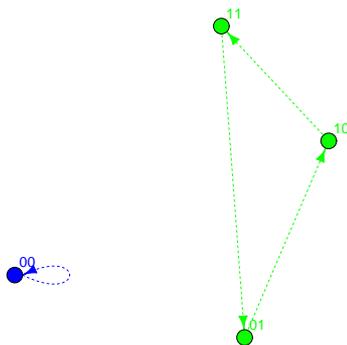


FIGURE 1. Cyclic attractor with empty exclusive basin

the cycle attractor $01 \rightarrow 10 \rightarrow 11 \rightarrow 01$ has empty exclusive basin, because any of its states can transition to 00 with asynchronous updates. For example, $11 \rightarrow 01 \rightarrow 00$ is possible by updating first coordinate 1 with map f_1 then coordinate 2 with map f_2 . So 11 can reach 00, and 00 is not in the basin of the cycle attractor as it is a fixed point, and 01, 10 are similar. Thus there are no states that cannot leave the basin $\{01, 10, 11\}$, and $B_{ex,\{01,10,11\}} = \emptyset$.

This example illustrates also why the iteration (2.2) should start at $B_0 = B_A$ in the presence of cyclic attractors, rather than starting with the inclusive basin $B_{in,A}$. Recall that the inclusive basin $B_{in,A}$ is the set of points that have positive probability of reaching the attractor A with asynchronous updates. In the above example, the inclusive basin for the fixed point 00 contains all states, and then the iteration (2.2) would give the entire state space if $B_0 = B_{in,A}$, including the cyclic attractor, as the exclusive basin $B_{ex,A}$. Thus, if we want to exclude all elements of other attractors from $B_{ex,A}$ we should start with $B_0 = B_A$ and not with $B_{in,A}$ at (2.2).

Starting with an initial state \mathbf{x}_0 in $B_{ex,A}$ as defined in (2.3) may not guarantee reaching the attractor A under asynchronous dynamics (but it does using the original deterministic map F). A simple example is a network in $d = 2$ where states 00, 11 map to 10 under F , 10 maps to 01, and 01 is fixed. Then the basin B_{01} for 01 is all four states, as is the exclusive basin $B_{ex,01}$. But the state 10 cannot reach 01 under asynchronous updates, it can only stay within $\{00, 11, 10\}$ with single node updates.

There are a variety of ways to define basins of interest with randomized versions of deterministic dynamics F , as explained and illustrated in [6] and [35]. Randomized versions can be either totally random or partially random, depending on

how much of the network is randomized. Then updates can be randomized by choosing an order randomly or a particular node or subset to update. The totally randomized method with individual random node updates is recommended in [34] (and called general asynchronous or GA dynamics) for its realism combined with computational efficiency, and this is what we use.

The definition (2.3) for the exclusive asynchronous basin is slightly different than the notion in [34] for general asynchronous dynamics. The definition used in [34] is the collection of states in the inclusive basin of A that are not in any other inclusive basin. The two definitions work out the same on the main 6-node example of their Table 1. They are also the same in the Figure 1 example, where $B_{ex,00} = \{00\}$ and $B_{ex,\{01,10,11\}} = \emptyset$. But in the previous example (in $d = 2$ where states 00, 11 map to 10 under F , 10 maps to 01, and 01 is fixed), the two notions are different: definition (2.3) for $B_{ex,01}$ gives the entire state space $\{00, 01, 10, 11\}$ since there is no other attractor basin that the Markov chain could enter, whereas the notion in [34] would make the exclusive basin equal to $\{01\}$, since it is the only point in the inclusive basin. This example supports the value of (2.3) if synchronous dynamics are considered primary and the goal is to prevent the dynamics from entering unwanted attractors under perturbations. The definition in [34] is studied in [11], the algebraic methods work there as well.

The notion of exclusive basin made precise in (2.3) is computationally efficient, it has a clear probabilistic interpretation, and it has practical value in applications like Example 4.2. It gives a robust basin for the purpose of targeting useful attractors.

3. ALGEBRAIC COMPUTATION

There are three computational problems when trying to target a given attractor under asynchronous dynamics with node interventions: 1) computing the basin of attraction B_A , 2) computing the exclusive asynchronous basin of attraction $B_{ex,A} \subset B_A$, and 3) finding cylinder sets inside $B_{ex,A}$. A cylinder set C is a set of the form $C = \{\mathbf{x} : x_{j_1} = a_1, \dots, x_{j_c} = a_c\}$ for particular 0-1 values of a_1, \dots, a_c . In other words, certain coordinates are assigned particular values and the others are free. The fixed coordinates j_1, \dots, j_c are the targets for intervention – if $C \subset B_{ex,A}$, then initializing the values of coordinates j_1, \dots, j_c at a_1, \dots, a_c for any starting state \mathbf{x} will guarantee convergence to the attractor A under deterministic dynamics and will prevent entering any other attractor under asynchronous dynamics, by Theorem 2.1.

A convenient and effective way to do the three computations uses computational commutative algebra ([7], [21], [32]) and its implementation in packages like [8]. Although the notion of exclusive asynchronous basin involves probability, the algebraic calculations are exact, unlike inferences about the basin from data generated by computer simulation of the asynchronous process. This technology works with the polynomials that vanish on a set of points (the *ideal* of the set) rather than directly with the points themselves (the *variety*). The *ideal-variety correspondence* [7] allows one to move between the two and find geometric properties of the set of points from algebraic properties of the ideal. To use commutative algebra for dynamics, one must write the dynamics as polynomial functions (see [22] and [38] for background). For example, the dynamics in Figure 1 would be written $f_1(s_1, s_2) = s_1 + s_2 - 2s_1 \cdot s_2$, $f_2(s_1, s_2) = s_1$ where the s_1, s_2 are indeterminates for

a polynomial ring $\mathbb{C}[s_1, s_2]$, technically not the same as state variables x_1, x_2 which are merely symbols for unspecified values of 0 or 1.

The remaining parts of this section are about doing the computations described in Section 2 using commutative rings in algebra software and will be of interest primarily to people implementing the methods in software.

3.1. Computing B_A . Let A be an attractor of interest, with basin of attraction B_A . We will use twice as many indeterminates as the number of coordinates d . Define the ring of polynomials $R := \mathbb{C}[s_1, \dots, s_d, t_1, \dots, t_d] = \mathbb{C}[\mathbf{s}, \mathbf{t}]$, and define ideals

$$\begin{aligned} I_{01} &= \langle s_1^2 - s_1, \dots, s_d^2 - s_d, t_1^2 - t_1, \dots, t_d^2 - t_d \rangle \\ F_{st} &= \langle f_1(\mathbf{s}) - t_1, f_2(\mathbf{s}) - t_2, \dots, f_d(\mathbf{s}) - t_d \rangle \\ F_{ts} &= \langle f_1(\mathbf{t}) - s_1, f_2(\mathbf{t}) - s_2, \dots, f_d(\mathbf{t}) - s_d \rangle \\ I_A &= \bigcap_{\mathbf{x} \in A} \langle t_1 - x_1, \dots, t_d - x_d \rangle. \end{aligned}$$

Define the ideal I_1 by

$$I_1 = (F_{st} + I_A + I_{01}) \cap \mathbb{C}[\mathbf{s}],$$

and define recursively a sequence of ideals I_2, I_3, I_4, \dots by

$$(3.1) \quad J = (F_{ts} + I_i + I_{01}) \cap \mathbb{C}[\mathbf{t}]$$

$$(3.2) \quad I_{i+1} = (F_{st} + J + I_{01}) \cap \mathbb{C}[\mathbf{s}], \quad i = 1, 2, 3, \dots$$

Stop the iteration when $\dim R/(I_i + I_{01})$ repeats in order to get the polynomials I_{B_A} that vanish on the basin of attraction B_A . This solves the first computational problem of getting B_A . This algorithm is in [13].

3.2. Computing $B_{ex,A}$. To get the exclusive asynchronous subset $B_{ex,A}$, we remove points algebraically from B_A following (2.2). Define ideals in $\mathbb{C}[\mathbf{s}, \mathbf{t}]$ for single coordinate updates of asynchronous dynamics by

$$I_{F_j} := \langle s_1 - t_1, \dots, s_{j-1} - t_{j-1}, f_j(\mathbf{s}) - t_j, s_{j+1} - t_{j+1}, \dots, s_d - t_d \rangle, \quad j = 1, \dots, d.$$

Also we will use the notation $I^{s \rightarrow t}$ for substitution of indeterminates t for s in an ideal $I \subset \mathbb{C}[\mathbf{s}]$.

The way to compute the ideal of $B_{ex,A}$ is

$$(3.3) \quad I_0 := I_{B_A}$$

$$(3.4) \quad J_i := \bigcap_{j=1}^d (R : I_{i-1}^{s \rightarrow t} + I_{F_j} + I_{01}) \cap \mathbb{C}[\mathbf{s}], \quad i = 1, 2, 3, \dots$$

$$(3.5) \quad I_i := I_{i-1} : J_i.$$

In words, we get J_1 by looking first at the complement of B_A with the colon ideal $R : I_0^{s \rightarrow t}$ in next-step variables t , and by adding I_{F_j} we add equations that constrain the starting variables s to hit the complement of B_A with the coordinate update f_j . Adding the ideal I_{01} is for technical reasons related to variable elimination and extending solutions (see the Extension Theorem in [7]). Then I_1 is the complement of J_1 , the points in \mathbf{s} variables that do not exit the basin. Stop the iteration when the ideals I_i repeat, using property 3) of Theorem 2.1, which can be detected by measuring their size with

$$\dim R/(I_i + I_{01})$$

the vector-space dimension counting the number of points in the 0-dimensional ideal. If the size of I_i stabilizes at i^* then

$$I_{B_{ex,A}} = I_{i^*}.$$

3.3. Finding Control Variables. One purpose of a signalling network model is to understand causal relationships among the nodes that can be used to target certain desirable limiting configurations. Examples are in cancer therapy ([24], [34], [33]) and Alzheimer's disease ([9], [19]). The effect of node settings is a purely mathematical one in this section. As discussed further in Example 4.2, the causal effect of network node interventions in the larger scheme of gathering data and fitting statistical models to biological data is more complicated.

Ways to find control variables, or nodes for intervention, are developed in [3]. One way is to find the prime decomposition (see [7]) of the ideal $I_{B_{ex,A}}$ which is the algebraic way to find simple cylindrical subsets of the variety $B_{ex,A}$. To summarize Theorem 2.2 of that work, we do three things to $I_{B_{ex,A}}$: 1) reduce or divide a standard basis for $I_{B_{ex,A}}$ by I_{01} to get \bar{I} , 2) compute the radical of \bar{I} , say $\sqrt{\bar{I}}$, 3) compute the primary decomposition of $\sqrt{\bar{I}}$ and look for components using variables that can be controlled.

The idea behind why this method works is as follows. Suppose the cylinder $C = \{\mathbf{x} \in \mathbb{C}^d : x_{j_1} = a_1, \dots, x_{j_c} = a_c\} \cap \{0, 1\}^d$ is contained in the basin $B_{ex,A}$. The ideal for the irreducible set $\{\mathbf{x} \in \mathbb{C}^d : x_{j_1} = a_1, \dots, x_{j_c} = a_c\}$ is prime, say P , and the ideal for $\{0, 1\}^d$ is I_{01} defined previously in Section 3.1. So we have inclusions

$$\begin{aligned} B_{ex,A} &\supset \{\mathbf{x} \in \mathbb{C}^d : x_{j_1} = a_1, \dots, x_{j_c} = a_c\} \cap \{0, 1\}^d, \\ I_{B_{ex,A}} &\subset I_{\{\mathbf{x} \in \mathbb{C}^d : x_{j_1} = a_1, \dots, x_{j_c} = a_c\} \cap \{0, 1\}^d} = P + I_{01} \end{aligned}$$

using that the ideal of the intersection of these sets (varieties more precisely) is the sum of their ideals. Then reducing the generators for $I_{B_{ex,A}}$ to get \bar{I} leads to $\bar{I} \subset P$ and since P is prime we get $\sqrt{\bar{I}} \subset P$. Then the prime ideal $P = \langle s_{j_1} - a_1, \dots, s_{j_c} - a_c \rangle$ in the ring $\mathbb{C}[\mathbf{s}]$ will show up in the prime or minimal decomposition $\cap_j P_j$ of the radical ideal $\sqrt{\bar{I}}$. The details are in Theorem 2.2 of [3].

For the simple example in Figure 1, the three calculations would go as follows. Starting with basin $B = \{00\}$, its ideal $I_B = \langle s_1, s_2 \rangle$, the polynomials whose roots define $x_1 = 0, x_2 = 0$. Then $I_{ex,B} = I_B$ since 00 is a fixed point for both deterministic and asynchronous dynamics, and $I_{ex,B} = \langle s_1, s_2 \rangle$. Finally, reducing $I_{ex,B}$ by I_{01} does not change anything because s_1, s_2 are not divisible by $s_j^2 - s_j$. Finally, the prime decomposition is nearly trivial, since the ideal $\langle s_1, s_2 \rangle$ is already prime and it corresponds to the cylinder $x_1 = 0, x_2 = 0$, just a single point.

A more illustrative example of the algebra is one where a hypothetical exclusive basin $B_{ex,A} = \{000, 001, 010\}$ is a union of two cylinder subsets 00^* and 0^*0 , with intersection the single point 000. This set has ideal given by $I_{B_{ex,A}} = \langle s_3^2 - s_3, s_2 s_3, s_2^2 + s_2 s_3 - s_2, s_1 \rangle$. Then reducing the generating elements of $I_{B_{ex,A}}$ by I_{01} and computing the radical gives ideal $\langle s_1, s_1 s_3 \rangle$, which has prime decomposition $P_1 = \langle s_1, s_3 \rangle, P_2 = \langle s_1, s_2 \rangle$, corresponding to 0^*0 and 00^* respectively.

4. EXAMPLES

Our two applications for illustrating the methods come from the life sciences. The first cell signalling example starts with dynamics from published sources, whereas the second involves a first step of statistical modeling using modern learning techniques described for example in [16]. We used the software BoolNet [29] to draw wiring diagrams, where green (light) edges mean activation and red (dark) edges mean inhibition, and to draw the state space diagram of Figure 1. We also used it to check calculations on attractors. BoolNet uses a concise and intuitive logical notation for coding functions: $\&$ codes for “and”, $|$ codes for “or” and $!$ codes for “not.” In Figure 2, the first line “NOS, Ca2” means that the first node is called NOS, and its update has a value given by the current value of the calcium node Ca2. Then the function for coordinate 1 is $f_1(\mathbf{x}) = x_{11}$, since Ca2 is node 11. For node 11, we have $f_{11}(\mathbf{x}) = x_9 \cdot (1 - x_{10})$.

4.1. Cell Signalling Network. The first example is a signalling network from Table 1 of [34]. The network models the agents (enzymes, proteins, and small molecules) that are involved in signalling stomatal closure in a cell. The closure is a function of the calcium node Ca2 and is not actually present in the coding of Figure 2. The wiring diagram is in Figure 3. The meanings of the names of the nodes in Figure 2 are given in Figure 1 of [35].

targets, functions
NOS, Ca2
NO, NOS
GC, NO
ADPRc, NO
cADPR, ADPRc
cGMP, GC
PLC, Ca2
InsP3, PLC
CIS, (cGMP & cADPR) InsP3
Ca2ATP, Ca2
Ca2, CIS & (!Ca2ATP)
KAP, !Ca2
KEV, Ca2

FIGURE 2. Stomatal Closure Dynamics in BoolNet Notation

The Boolean network has one fixed point 000000000010 in which node 11 Ca2 is off (implying no stomatal closure) and two cycles of length 4. The fixed point has deterministic basin of size 108, and an exclusive asynchronous basin of size 56, composed of two cylinders $x_1 = x_2 = x_3 = x_6 = x_7 = x_8 = x_9 = x_{11} = 0$ and $x_1 = x_2 = x_4 = x_5 = x_7 = x_8 = x_9 = x_{11} = 0$ each of size 32, and overlapping in 8 points. The computer time to get the synchronous basin was .060 seconds, whereas the asynchronous basin required 74.420 seconds (Intel x86-64 processor i7-3770 at 3.40GHz) and memory use was under 2 Gbytes always.

Any of the $8136 = 2^{13} - 56$ points outside of its exclusive basin has a positive probability of entering the deterministic basins of the two cycles under asynchronous dynamics. Both cycles show oscillatory behavior in the Ca2 node, and oscillation is of special interest to biologists [35].

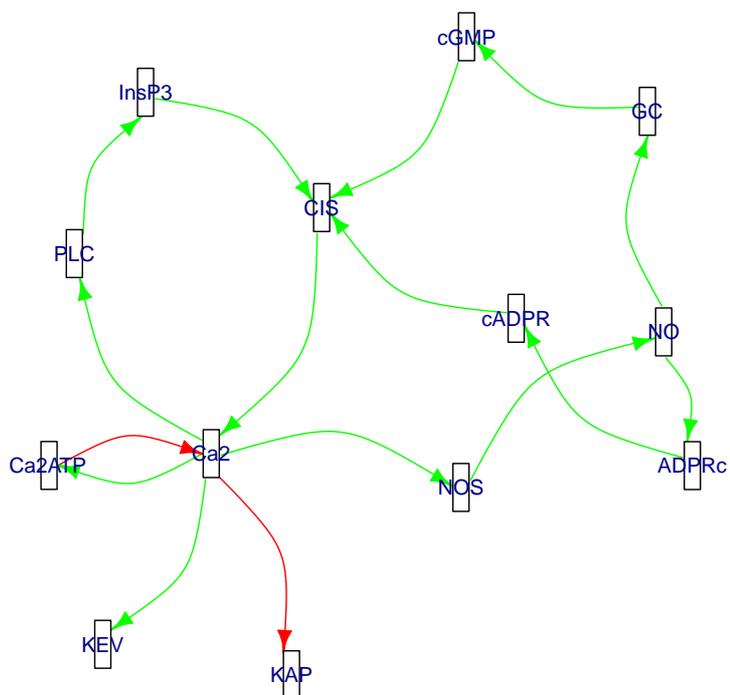


FIGURE 3. Wiring Diagram for Stomatal Closure Signalling Dynamics

One cycle shows oscillation in all 13 nodes:

```

1000001011001
0100000100010
0011000010010
0000110000110
    
```

For this cycle, analysis of the deterministic dynamics gives 24 cylinder subsets of the basin that move to the cycle, involving 5 or 6 nodes. For example, the first one is $x_1 = 0, x_4 = 0, x_7 = 0, x_{10} = 1, x_{11} = 1$. Thus, any point with these initial values on nodes 1, 4, 7, 10, and 11 (NOS off, ADPRc off, PLC off, Ca2ATP on, Ca2 on) will move to this attractor. Note that the values on these nodes will change over time, as they must since the attractor shows oscillation on all nodes. The exclusive

asynchronous basin is empty, meaning that changing the dynamics with random node updates will allow the system to move into other attractor basins with positive probability.

A second cycle is fixed in the CIS node 9:

```

1100001111001
0111000110010
0011110010110
1000111011101

```

Here the deterministic dynamics give 16 subcylinders of the basin, with some two-node sets like $x_7 = 1, x_{11} = 1$. Again, the exclusive asynchronous basin is empty.

The empty asynchronous basins show the network is fragile in the sense that randomized dynamics can change the limiting behaviour significantly with positive probability, a property also noted in [6].

4.2. Life Kinetics Network. Our second example involves longitudinal data from in-home motion sensor recordings. This data was generated by the Intelligent Systems for Assessing Aging Changes (ISAAC) study [18], an aging study conducted by the Oregon Center for Aging & Technology (ORCATECH) at OHSU. The particular data set involved 153 people, 197 homes, and spanned the time period May 2007 to February 2014. Our data file recorded summary data for each day, for each residence. The variables to be measured and computed were determined by ORCATECH based on expertise and current research including [2], [17], [30], where variables like the number of transitions from one room to another, or the time spent out of house, were shown to be relevant. It is known that variability of quantities like walking speed is important, so we also had measures of dispersion like the standard deviation of walking speed over each day, and the coefficient of variation of walking speed (standard deviation divided by mean) on each day to normalize for subject differences. The data is observational and not controlled experimental data, as variables like “walking speed” were not applied in a randomized way to subjects. So dynamical patterns in the data found with statistical learning methods are comparable to correlations, and causal effects can only be determined with further study.

A variable of interest is the binary variable MCI (*mci*) which codes for mild cognitive impairment. While MCI is not the same as Alzheimer’s disease, the two are related [31]. One goal is to understand the predictive roles of other behavioural variables. In particular, it would be useful to know if interventions may be possible on some variables to reduce the risk of future MCI.

Let us say why a Boolean network model may be useful for this data. A Bayes network can be used to model dependencies in multivariate data, but the model is static and it does not help to understand the predictive role of variables related to MCI months in advance. In the work [5], the predictive role of walking speed years in advance on MCI was established using change point methods, statistical methods which are related to on/off signalling. But [4] emphasizes that monotherapeutics (therapies focusing on one variable) have been ineffective, and presents

a new therapeutic system involving over twenty actions designed to modify behaviour variables like sleep and exercise, variables which also appear in some form here. The Boolean network method can give an enhanced understanding on how combinations of interventions have signalling effects over time. It is one of the simplest ways to understand multivariate signalling over time, our main goal. The methods of survival analysis are also applicable, but they are not able to capture cyclic behaviour of MCI in some individuals. It is possible to extend the techniques of this paper to predictive variables with more levels than two [10]. Finally, many statistical methods are useful for understanding multivariate longitudinal data and the Boolean model that we use here is just one tool.

A large scale evolving study like ISAAC will have challenging statistical issues. About 16% of the entries are missing in our working data file of 43 Megabytes, and outliers are present due to imperfect sensor readings. The reasons why we consider our fitted model in Figure 4 only a first step in understanding the dynamics of the data are: 1) our missing data procedure is crude, 2) tree classification methods are known to be sensitive to small changes in the data, and 3) the best time lag for regression is not established.

Now we describe the statistical procedure that gave the estimated dynamics of Figure 4. First, the data was revised by removing the few records with missing MCI value or missing subject identifying number. Then other missing values were imputed with the function `na.roughfix` of the `randomForest` package [25]. This is a very crude tool for missing data, but the more sophisticated function `rfImpute` would not handle so much data with only 32 Gigabytes of computer memory. A more refined analysis in the future should improve this step. Then, after imputing to get complete data, we used the variable importance feature of the `randomForest` package to rank the variables for inclusion in the final model. This method essentially looks at how much worse prediction with cross validation becomes when the variable is removed from the model (see p. 593 of [16] p. 593 for a detailed description). From approximately 30 variables, we chose 15 including the MCI value. Many of the variables are continuous, so it is necessary to discretize into high/low or on/off values for a Boolean network. Rather than using quantiles or other methods that are unrelated to signalling or prediction, we used a new predictive method that is based on random forests. For this, we get 500 classification trees from the `randomForest` command predicting MCI on one day from the predictor values on the previous day. Then we examined each tree for the value of the first split on each predictor variable. The cutoff or threshold for on/off was the mean of the 500 values of the first split (each of the 500 trees split every predictor variable in its partition of the domain for building a function $y = f(x_1, \dots, x_{14})$, where y is a binary code for MCI). The variable definitions are in Table 1.

Then finally tree classification was implemented on the resulting binary variables. For the tree classification, we used the R package `rpart` [39]. This procedure is described in [16] and is different than the interpolation method of [42]. Converting the trees to algebra was done as in [12]. The wiring diagram in Figure 5 is a graphical representation of Figure 4, where green arrows indicate an effect of switching on or up (activation) and red arrows represent switching off or down (inhibition). The wiring diagrams are slightly ambiguous because of interaction effects. The arrow shows the direction of the effect in time.

TABLE 1. ISAAC Study Variable Definitions

numtrans	number of transitions between rooms, normalized to mean 1.0 in home
numfir	number of sensor firings in the home, normalized to mean 1.0 in home
oohhours	estimate of the number of hours the participant spent out of home
timeasleep	total time asleep (minutes)
ttib	total time in bed (minutes)
waso	time awake after sleep onset (minutes)
sleeplatency	total sleep latency (minutes)
sleeplivroom	time asleep in living room (minutes)
compuse	total computer use (minutes)
numwalks	number of walks (count)
meanws	mean walking speed (cm/s)
wsq3	upper quartile of walking speed (cm/s)
wscv	coefficient of variation of walking speed
wssigma	standard deviation of walking speed (cm/s)
MCI	mild cognitive impairment diagnosis with the Jak criteria (binary indicator)

The Boolean dynamical system from statistical learning is in Figure 4. This model is one that is consistent with many observed interactions among variables related to aging, but should not be taken as a final analysis. It has 52 attractors, including 7 steady states and 45 two element cycles. Six of the steady states have MCI off, so these are interesting and useful as targets, as are the 33 cycles in which MCI is off.

	targets, functions
1	mci, !wsq3 & !compuse & (!waso & sleeplivroom waso)
2	numtrans, numfir
3	numfir, !numtrans & (!sleeplivroom & !wsq3 sleeplivroom) numtrans
4	oohhours, !ttib & (!sleeplivroom & (numfir & numwalks !numfir) sleeplivroom)
5	wsq3, !meanws & !numwalks & !mci meanws
6	meanws, wsq3 & (wscv & wssigma !wscv)
7	numwalks, wssigma & !wscv
8	wscv, wssigma & (!meanws meanws & mci)
9	ttib, timeasleep
10	wssigma, wscv & meanws
11	timeasleep, ttib
12	sleeplivroom, timeasleep & !ttib
13	waso, sleeplatency & ttib
14	compuse, !mci & wssigma
15	sleeplatency, waso & ttib

FIGURE 4. Life Kinetics Dynamics in BoolNet Notation

One of the interesting steady states is 01101 10000 00000, with coordinate labels in the order of Figure 4. Thus MCI has value 0, which is a desirable configuration. The deterministic basin has size 3456, whereas the exclusive asynchronous basin has size 960. Such a significant reduction in size is expected, as this was observed in [34] and [35]. The computational time in Singular [8] (version 3-1-6) is considerably higher for the asynchronous basin. On a desktop computer (32 Gbytes of memory, Intel x86-64 processor i7-3770 at 3.40GHz), the synchronous basin B_A took 2.320

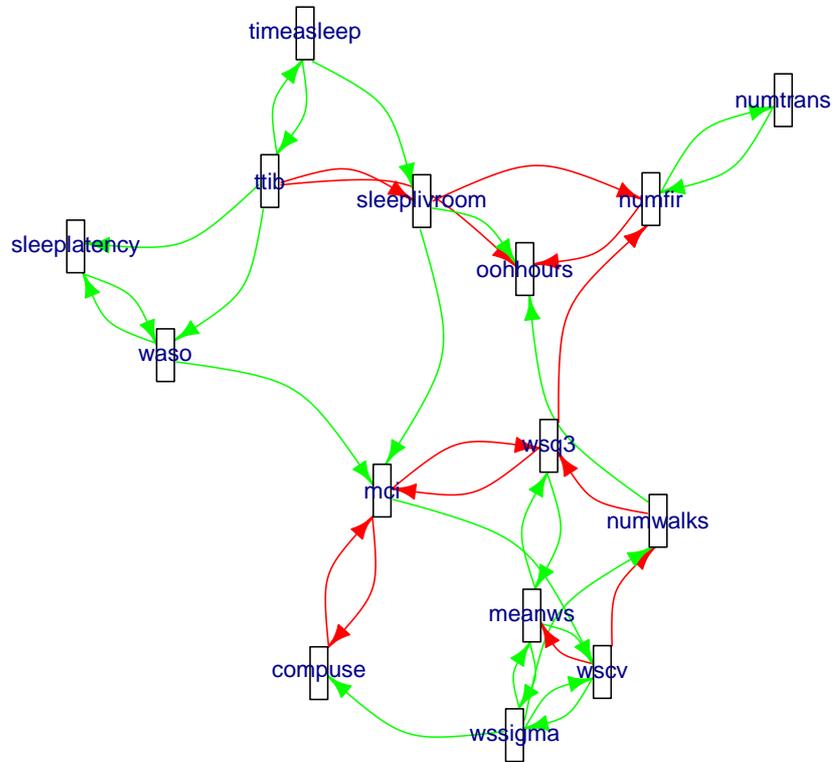


FIGURE 5. Wiring Diagram for Life Kinetics and Cognitive Impairment

seconds of computing time, whereas the exclusive asynchronous basin $B_{ex,A}$ took 1090.100 seconds. The computer memory use for the asynchronous basin calculation was less than 2 Gbytes throughout the computation. The basin $B_{ex,A}$ is the union

of five intersecting cylinder subsets:

$$C_1 = \{\mathbf{x} : x_2 = 1, x_3 = 1, x_8 = 0, x_9 = 0, x_{10} = 0, x_{11} = 0\}$$

$$C_2 = \{\mathbf{x} : x_2 = 1, x_3 = 1, x_6 = 0, x_9 = 0, x_{10} = 0, x_{11} = 0\}$$

$$C_3 = \{\mathbf{x} : x_1 = 0, x_2 = 1, x_3 = 1, x_5 = 1, x_6 = 1, x_8 = 0, x_9 = 0, x_{11} = 0\}$$

$$C_4 = \{\mathbf{x} : x_2 = 1, x_3 = 1, x_5 = 1, x_6 = 1, x_8 = 1, x_9 = 0, x_{10} = 1, x_{11} = 0\}$$

$$C_5 = \{\mathbf{x} : x_1 = 0, x_2 = 1, x_3 = 1, x_5 = 1, x_6 = 1, x_9 = 0, x_{10} = 1, x_{11} = 0\}$$

Cylinders C_3 and C_5 use node 1 (MCI) so these are of no practical use for interventions. However C_1 may be useful: it says if we initialize numtrans at high, numfir at high, wscv low, ttib low, wssigma low, and timeasleep low, then the system will lead to this steady state deterministically and will not lead to any attractor with MCI on under asynchronous dynamics. The six nodes for this intervention are known to be related to MCI individually, but this new analysis gives the precise combination of settings that will turn MCI off over time. The results are consistent with current knowledge. One can study all steady states to seek the most practical nodes for intervention. All of the 33 cycle attractors have empty asynchronous basins.

5. CONCLUSIONS

We have defined an exclusive asynchronous basin in order to eliminate initial states from a deterministic basin of attraction that could move to unwanted attractors under uncertainty in the timing of the node updates. Variations on the definition are possible by changing the initial set B_0 in the iteration. Algebraic computational methods are given. Altogether, for Boolean networks of up to around 15 nodes, the methods give a way to find node interventions that will set an initial state moving towards a desired limiting state even under random perturbations of a model.

The examples show that analyzing asynchronous dynamics is more demanding computationally than a deterministic analysis. Still, existing symbolic computational algebra software is sufficient for precise geometric calculations needed to understand network control variables on real examples. While other computational methods exist for finding deterministic and asynchronous basins, the algebraic geometry is especially good for finding combinations of node settings that target a given attractor in the presence of randomized dynamics. Calculations on a 15 node network use a space of polynomials with 30 indeterminates and this works well in practice. An interface between a package like BoolNet [29] directly to Singular [8] would be convenient and not difficult.

ACKNOWLEDGMENTS

We would like to thank Daniel Austin PhD and Jeffrey Kaye MD for their assistance with Example 4.2. The sensor data for this example was provided by ORCATECH at Oregon Health and Science University with funding provided by NIH R01 AG024059, P30 AG024978, and P30 AG008017.

REFERENCES

- [1] J. Abbott and A. M. Bigatti, *CoCoALib: a C++ library for doing Computations in Commutative Algebra*. Available at <http://cocoa.dima.unige.it/cocoalib>, 2014.

- [2] D. Austin, R. M. Cross, T. Hayes, and J. Kaye, Regularity and Predictability of Human Mobility in Personal Space. *PLoS One*, **9** (2014), e90256.
- [3] D. Austin, and I. H. Dinwoodie, Monomials and Basin Cylinders for Network Dynamics. *SIAM J. Appl. Dyn. Syst.*, **14** (2015), 25-42.
- [4] D. E. Bredesen, Reversal of cognitive decline: a novel therapeutic program. *Aging*, **6** (2014), 707-717.
- [5] T. Buracchio, H. Dodge, D. Howieson, D. Wasserman, J. Kaye, The trajectory of gait speed preceding MCI. *Arch. Neurol.*, **67** (2010), 980-986.
- [6] M. Chaves, R. Albert, and E. D. Sontag, Robustness and fragility of Boolean models for genetic regulatory networks. *Jour. Theoret. Biol.*, **235** (2005), 431-449.
- [7] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms, Second Edition*. Springer, New York, 1997.
- [8] W. Decker, G.-M. Greuel, G. Pfister, and H. Schönemann, Singular 4-0-2 — A computer algebra system for polynomial computations. <http://www.singular.uni-kl.de>, 2015.
- [9] G. V. De Ferrari, and N. C. Inestrosa, Wnt signaling function in Alzheimer's disease. *Brain Research Reviews*, **33** (2000), 1-12.
- [10] I. H. Dinwoodie, Conditional tests on basins of attraction with finite fields. *Methodology and Computing in Applied Probability*, **16** (2014), 161-168.
- [11] I. H. Dinwoodie, Vanishing configurations in network dynamics with asynchronous updates. *Proc. Amer. Math. Soc.*, **142** (2014), 2991-3002.
- [12] I. H. Dinwoodie, Polynomials for classification trees and applications, *Statistical Methods and Applications*, **19** (2009), pp. 171-192.
- [13] I. H. Dinwoodie, and K. Pandya, Exact tests for singular network data. *Ann. Inst. Statist. Math.* **67** (2014), 687-706.
- [14] D. R. Grayson, and M. E. Stillman, *Macaulay2, a software system for research in algebraic geometry*. Available at <http://www.math.uiuc.edu/Macaulay2/>, 2014.
- [15] T. Handorf and E. Klipp, Modeling mechanistic biological networks: An advanced Boolean approach. *Bioinformatics*, **28** (2012), 557-663.
- [16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2011.
- [17] T. L. Hayes, T. Riley, M. Pavel, and J. A. Kaye, Estimation of rest-activity patterns using motion sensors. *Conf Proc IEEE Eng Med Biol Soc*, vol. 2010, pp. 2147-2150, 2010.
- [18] J. A. Kaye, S. A. Maxwell, N. Mattek, T. L. Hayes, H. Dodge, M. Pavel, H. B. Jimison, K. Wild, L. Boise, and T. A. Zitzelberger, Intelligent systems for assessing aging changes: home-based, unobtrusive, and continuous assessment of ageing. *The Journals of Gerontology, Series B: Psychological Sciences and Social Sciences*, 66B(S1) (2011), i180-i190.
- [19] M. Hermes, G. Eichhoff and O. Garaschuk, Intracellular calcium signalling in Alzheimer's disease. *J. Cell. Mol. Med.*, **14** (2009), 30-41.
- [20] S. Klamt, J. Saez-Rodriguez, J. A. Lindquist, L. Simeoni, and E. D. Gilles, A methodology for the structural and functional analysis of signalling and regulatory networks. *BMC Bioinformatics*, **7** (2006), 1471-2105.
- [21] M. Kreuzer, and L. Robbiano, *Computational Commutative Algebra I*. Springer, New York, 2000.
- [22] R. Laubenbacher and B. Sturmfels, Computer Algebra in Systems Biology. *Amer. Math. Monthly*, **116** (2009), 882-891.
- [23] R. K. Layek, A. Datta and E. R. Dougherty, From biological pathways to regulatory networks. *Molecular BioSystems*, **7** (2011), 843-851.
- [24] R. K. Layek, A. Datta, M. Bittner, and E. R. Dougherty, Cancer therapy design based on pathway logic. *Bioinformatics*, **27** (2011), 548-555.
- [25] A. Liaw and M. Wiener, Classification and Regression by randomForest, *R News*, **2** (2002), 18-22, <http://CRAN.R-project.org/doc/Rnews/>.
- [26] T. Lu, L. Aron, J. Zullo, Y. Pan, H. Kim, Y. Chen, T.-H. Yang, H.-M. Kim, D. Drake, X. S. Liu, D. A. Bennett, M. P. Colaiácovo, and B. A. Yankner, REST and stress resistance in ageing and Alzheimer's disease. *Nature*, **507** (2014), 448-454.
- [27] M. K. Morris, J. Saez-Rodriguez, P. K. Sorger and D. A. Lauffenburger, Logic-Based Models for the Analysis of Cell Signaling Networks. *Biochemistry*, **49** (2010), 3216-3224.

- [28] D. Murrugarra, A. Veliz-Cuba, B. Aguilar, S. Arat, and R. Laubenbacher, Modeling stochasticity and variability in gene regulatory networks, *EURASIP Journal of Bioinformatics and Systems Biology*, **1** (2012) 2012:5.
- [29] C. Müssel, M. Hopfensitz, and H. A. Kestler, BoolNet – an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, **26** (2010), 1378-1380.
- [30] J. Petersen, D. Austin, J. Kaye, M. Pavel, and T. Hayes, Unobtrusive in-home detection of time spent out-of-home with applications to loneliness and physical activity, *IEEE Journal of Biomedical and Health Informatics*, **18** (2014), 1590-1596.
- [31] R. C. Petersen, Mild cognitive impairment as a diagnostic entity. *J. Intern. Med.*, **256** (2004), 183-194.
- [32] G. Pistone, E. Riccomagno, and H. Wynn, *Algebraic Statistics: Computational Commutative Algebra in Statistics*. Chapman and Hall, Boca Raton Florida (2001).
- [33] R. Poltz and M. Naumann, Dynamics of p53 and NF- κ B regulation in response to DNA damage and identification of target proteins suitable for therapeutic intervention. *BMC Systems Biology*, **6** (2012), 125.
- [34] A. Saadatpour, R-S. Wang, A. Liao, X. Liu, T. P. Loughran, I. Albert, and R. Albert, (2011) Dynamical and Structural Analysis of a T Cell Survival Network Identifies Novel Candidate Therapeutic Targets for Large Granula Lymphocyte Leukemia. *PLoS Comp. Biol.* **7**, e1002267.
- [35] A. Saadatpour, I. Albert, and R. Albert, Attractor analysis of asynchronous Boolean models of signal transduction networks. *J. Theor. Biol.* **266** (2010), 641-656.
- [36] R. Schlatter, K. Schmich, I. A. Vizcarra, P. Scheurich, T. Sauter, C. Borner, M. Ederer, I. Merfort and O. Sawodny, ON/OFF and Beyond - A Boolean Model of Apoptosis. *PLoS Comput. Biol.*, **5** (2009), e1000595.
- [37] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18** (2002), 261-274.
- [38] B. Stigler, Polynomial Dynamical Systems in Systems Biology. *AMS 2006 Proceedings of Symposia in Applied Mathematics*, **64** (2006), 59-84.
- [39] T. Therneau, B. Atkinson, and B. Ripley, rpart: Recursive Partitioning and Regression Trees. <http://CRAN.R-project.org>.
- [40] R. Thomas, Boolean formalization of genetic control circuits. *J. Theoret. Biol.*, **42** (1973), 563-585.
- [41] A. Wuensche, Complex and Chaotic Dynamics, Basins of Attraction, and Memory in Discrete Networks. *Acta Physica Polonica B*, **3** (2010), 463-478.
- [42] A. Veliz-Cuba, An algebraic approach to reverse engineering finite dynamical systems arising from biology. *SIAM Jour. Appl. Dyn. Systems*, **11** (2012), 31-48.
- [43] R. Zhang, M. V. Shah, J. Yang, S. B. Nyland, X. Liu, J. Yun, R. Albert and T. P. Loughran, Network model of survival signaling in large granular lymphocyte leukemia. *Proc. Natl. Acad. Sci. USA*, **105** (2008), 16308-16313.

IAN H DINWOODIE, FARIBORZ MASEEH DEPARTMENT OF MATHEMATICS AND STATISTICS, PORTLAND STATE UNIVERSITY, 724 SW HARRISON STREET PORTLAND, OR 97201, USA, ihd@pdx.edu