Multi-parameter exploration of dynamics of regulatory networks

Author: Tomas Gedeon

Multi-parameter exploration of dynamics of regulatory networks★

Tomáš Gedeon[a]

[a]*Department of Mathematical Sciences*
Montana State University, Bozeman, MT 59715

ABSTRACT

Over the last twenty years advances in systems biology have changed our views on microbial communities and promise to revolutionize treatment of human diseases. In almost all scientific breakthroughs since time of Newton, mathematical modeling has played a prominent role. Regulatory networks emerged as preferred descriptors of how abundances of molecular species depend on each other. However, the central question on how cellular phenotypes emerge from dynamics of these network remains elusive. The principal reason is that differential equation models in the field of biology (while so successful in areas of physics and physical chemistry), do not arise from first principles, and these models suffer from lack of proper parameterization. In response to these challenges, discrete time models based on Boolean networks have been developed.

In this review, we discuss an emerging modeling paradigm that combines ideas from differential equations and Boolean models, and has been developed independently within dynamical systems and computer science communities. The result is an approach that can associate a range of potential dynamical behaviors to a network, arrange the descriptors of the dynamics in a searchable database, and allows for multi-parameter exploration of the dynamics akin to bifurcation theory. Since this approach is computationally accessible for moderately sized networks, it allows, perhaps for the first time, to rationally compare different network topologies based on their dynamics.

1. Introduction

"Nowadays, many computational biologists avoid modeling-as-data-fitting, opting instead to create models in which networks are specified in terms of elements and interactions (the network "topology"), but the numerical values that quantify those interactions (the parameters) are deliberately varied over wide ranges. As a result, the study of such networks focuses not on the exact values of outputs, but rather on qualitative behavior, e.g., whether the network acts as a "switch," "filter," "oscillator," "dynamic range adjuster," "producer of stripes," etc. By investigating how such behaviors change for different parameter sets— an exercise referred to as "exploring the parameter space"—one starts to assemble a comprehensive picture of all the kinds of behaviors a network can produce." (Lander 2004 [47])

Differential equations are inseparably linked to the birth of modern science. Among Newton's fundamental contributions to science was the introduction of a phase space where only those descriptors of a physical system that have effect on its dynamics are represented. Second, the development of a concept of a differential equation that algebraically links infinitesimal changes and other quantities allowed a quantitative prediction of a future state in the phase space. Most important was his realization that, in the abstract world of the phase space, one can postulate the existence of non-measurable quantities (i.e gravitational forces), solve a differential equation for measurable quantities (i.e distances) and based on their comparison with experimental observations one is able to reject a particular form of non-measurable quantity. This laid the foundation to the fruitful cycle of theory and experiments that has driven science forward over the last 300 years.

Differential equations underwent a paradigm shift starting with the work of Poincare at the start of 20th century. The emphasis shifted from solving differential equations to qualitative theory, from a finite time horizon to the long term behavior. The resulting tools from the qualitative theory of differential equations were instrumental to understanding the wide range of phenomena from the singularities in n-body dynamics [74, 58] to the description of neural spike propagation modeled by Hodgkin-Huxley equations [41].

Naturally, when the world of molecular and cellular biology opened up to experimental measurements, the toolbox of differential equations had been readily available to try to model the dynamics of the intricate systems that underlie cellular life. The problems that were encountered were both severe and fundamental; there seemed to be no easy fix that could simultaneously address all of them.

In this paper we describe these difficulties and how a combination of Boolean networks and continuous time dynamics begin to converge to a modeling framework that is well suited for modeling regulatory networks that arise in molecular and cell biology.

There are at least three fundamental challenges when using differential equations to model the dynamics of regulatory networks: nonlinearity, parameterization and continuity of phase space, parameter space and time.

1.1. Nonlinearity

Because of the multi-scale character of the biological networks, there is a lack of "first principles" description of the nonlinear right hand side of the differential equations models. In order to analyze a differential equation, either numerically or analytically, an explicit nonlinearity is necessary. Nonlinearities arise in physics from fundamental laws like Newton's law. There are no such well established laws in molecular biology that would govern the interaction between genes, or between a protein and its receptor. What is being used instead is a parameterized nonlinearity in a form of a Hill function, or nonlinearities that arise from mass action chemical kinetics.

1.2. Parameterization

Parameters pose another big problem. In Newtonian physics, the parameters are masses of objects that interact; these are directly measurable. In contrast, in molecular biology the constants such as the "half-saturation rate", or a "reaction rate", can only be measured by fitting a solution of a differential equation with an assumed nonlinearity to the experimental data. If we assume a different nonlinearity, the value of the fitted parameter will be different. Such a parameter can only be interpreted within the framework of its model.

This makes it difficult to permanently associate a parameter value to a given system. Therefore it is challenging to build on the work of others since their parameters may have been used in a context of a slightly different model. At best, one has to re-fit the data in the context of the new model; but more often than not, one must measure the parameters again. Since measuring parameters in biology is expensive, parameter estimates are often poor. This leads to parameterization of models by fitting the models solutions to the data. This results in identifiability problems, where there are many different parameters and models that can fit the same data.

However, the exploration of the parameter space is often how the model is expected to inform biology: the interest is to know what perturbation can bring the system from one state to another, what medical interventions, modeled as a change of the parameters, have medically desirable effects, and how does underlying genetic diversity determine the medical outcome. These are all questions about how the dynamics of the system changes with parameters.

1.3. Continuous time, phase space and parameter space

Continuous time, phase space and parameter space are arguably more realistic in modeling the natural world than discrete time models. However, a continuous description is not directly representable in computers which always brings up the issue of a proper approximation of a continuous system.

On the other hand, discrete models' future behavior depends on the present state via a map, phase space is often discrete (i.e. when we represent a state of a gene by being On or Off, as in the Boolean network models), or there is only a discrete set of parameters. In a discrete description enumeration of all possibilities is, at least in principle, feasible, and thus no approximation is necessary.

2. Models of gene regulatory networks

Even though without nonlinearities and parameters, the differential equation models cannot be expected to be very useful, the demand for informative models in systems biology is growing. The potential medical applications demand an analysis of large systems with dozens of interacting genes and proteins. These systems are often characterized in terms of a *network* of interactions, which represent a direct up-or down-regulation of one gene, or protein, by another. As a result, there is a great interest in understanding the *dynamics of networks*. However, the term itself provides an insight to the conflict between expectations placed on mathematical models and its potential to deliver on them. Biologists would like to understand the range of potential dynamics that this network exhibits under different cellular

conditions and different initial data; mathematicians can address these questions using differential equations only if they are given parameters, nonlinearities and initial conditions. As discussed above, none of these are readily available.

Definition 1. *A regulatory network* **RN** $= (V, E)$ *is a graph with network nodes* $V = \{1, 2, \ldots, n\}$ *and signed, directed edges* $E \subset V \times V \times \{\rightarrow, \dashv\}$. *For* $i, j \in V$, *we will use the notation* $(i, j) \in E$ *to denote a directed edge from* i *to* j *of either sign;* $i \rightarrow j$ *will denote an activation or positive interaction, and* $i \dashv j$ *will denote a repression or negative interaction.*

We define the targets of a node i *as*

$$\mathbf{T}(i) := \{j \mid (i, j) \in E\}$$

and the sources of a node i *as*

$$\mathbf{S}(i) := \{j \mid (j, i) \in E\}$$

The goal of this paper is to describe two approaches that find a way to move beyond traditional ODE models for network dynamics. Several ideas converge together to combine the best features of the continuous and discrete dynamics into a powerful new tool.

3. Boolean networks

A Boolean model of network's dynamics starts with set $B := \{0, 1\}$. A *state* of a network is an element $x \in B^n$. A *Boolean network* is a map $f = (f_1, f_2, \ldots, f_n)$ with $f^i : B^n \rightarrow B$ and

$$f : B^n \rightarrow B^n.$$

The dynamical system generated by iterates of f is called *synchronous Boolean network*. Let H be the n-dimensional hypercube graph with vertices $v \in B^n$ and edges between $x, y \in B^n$ if, and only if, these states differ in exactly one component. Each synchronous Boolean map f induces *asynchronous* dynamics on H, given by a multi-valued map $\mathbb{F} : H \rightrightarrows H$, encoded by a *state transition graph* (STG), defined as follows: connect $x \in H$ to all its neighbors y in H with $|x_i - y_i| = 1$ for all indices i with the property that $f_i(x) \neq x_i$.

Boolean networks have been used to model gene regulatory networks (circuits) for many years [67, 63, 68, 57, 4, 18, 53]. Boolean network is determined by specification of a collection Boolean functions $(f_1, f_2, \ldots, f_n)$, which, in turn, are each given by a truth table with number of inputs equal to number of incoming edges to the corresponding node in V. Any parameterization that encodes all collections of truth tables that are compatible with the network **RN** will give parameterization of all Boolean models compatible with **RN**. However, since a change in a Boolean network involves changing a value of some function f_i by a discrete unit (from 0 to 1, or vice versa) there is no natural bifurcation theory that can delineate the region of validity of certain dynamics under changing cellular conditions.

In addition, synchronous dynamics also assume, unrealistically, that multiple genes change their states simultaneously. This problem is largely addressed by considering an asynchronous update rule for Boolean models. Lastly, Boolean networks lack the means to ask questions about the time duration of other dynamic phenomena, like length of the period of a cell cycle oscillation, since time is not a part of the modeling framework.

4. Continuous time models

In attempt to address some of these short-comings L.Glass [35, 36] introduced continuous time models with discontinuous right hand side, that are motivated by Boolean networks. They consider a system of differential equations (see [28])

$$\dot{x}_i = -\gamma x_i + F_i(\tilde{x}_1, \ldots, \tilde{x}_n) \tag{1}$$

where

$$\tilde{x}_i = 1 \quad \text{if } x_i > \theta_i; \qquad \tilde{x}_i = 0 \quad \text{if } x_i < \theta_i. \tag{2}$$

The system (1) is sometimes referred to as a *Glass system*. The Glass models have been used extensively [69, 45, 46, 6, 9, 50, 10, 15] in models of molecular network dynamics.

The connection to Boolean networks is straightforward; The function $F = (f_1, \ldots, f_n)$ reduces to a Boolean function f above, if we restrict the range of F to B^n. Note that the values of $F \in R^n$ do not depend on values of x but only on the values of $\tilde{x}$; therefore the range of F is finite - it can have at most 2^n values. More formally, R^n is decomposed into 2^n disjoint open sets, parameterized by $\alpha \in B^n$,

$$D(\alpha) := \{x \in R^n \mid x_i < \theta_i \text{ when } \alpha_1 = 0, \text{ and } x_i > \theta_i \text{ when } \alpha_1 = 1\}.$$

The solutions of (1) in $D(\alpha)$ converge toward $\frac{F(\alpha)}{\gamma}$, but only until they leave the domain $D(\alpha)$. Connecting trajectories in neighboring domains that share a threshold, whenever possible, results in a piece-wise linear flow. There are many subtleties as to when such merging of solutions in neighboring domains is possible, as well as how to extend these solutions if it is not possible. We address these issues briefly below.

In Glass networks the function f_i, viewed as a function $B^n \to B$ could be an arbitrary Boolean function. A more appropriate class of Boolean functions for modeling biological networks is the class of *monotone (unate)* Boolean functions [1, 21]

Definition 2. *A function* $f : B^n \to B$ *is increasing (decreasing) in* $\tilde{x}_i$, $i \in \{1, \ldots, n\}$, $f_{|\tilde{x}_i=0} \leq f_{|\tilde{x}_i=1}$ ($f_{|x_i=1} \leq f_{|\tilde{x}_i=0}$). *In either case,* f *is said to be monotone (or unate) in* $\tilde{x}_i$. *It is said to be monotone, if it is monotone in all* $\tilde{x}_i$. *In particular, if it is increasing in all* $\tilde{x}_i$ *then it is positive.*

For example, both logical functions AND and OR are monotone (increasing). However, the function XOR is not monotone. The classes of monotone and positive Boolean functions attracted the interests of different research communities. For review of their properties, we refer reader to Crama and Hammer [21] and Korshunov [1].

The concept of a *monotone* function captures the information embedded in the gene regulatory network where each edge is either marked as activating or repressing [27, 70]. The number of monotone Boolean functions on n variables is much smaller than that of the general Boolean functions, but it is still very large. In fact, the precise number of monotone Boolean functions is the *Dedekind number* which is not known for $n \geq 9$.

A different formalism that incorporates monotonicity implied by signed interactions in a regulatory network **RN** was used by R. Thomas and coworkers [65, 66, 68, 67, 60, 61, 64]. They proposed that if a node i affects k downstream nodes, then generically there should be k different thresholds θ_{ji} at which the effect of i on a downstream node j is activated.

The form of the differential equation (1) remains the same, but the definition of function F_i changes. In Snoussi [60], F_i consists of positive combinations of products and sums of contributions $\sigma^{\pm}(x_j)$ (see definition below) from nodes $j \in S(i)$. In a more general framework [61, 64, 56] the contributions are summed

$$F_i = \sum_{\omega \subset S(i)} \sigma^{\pm}(i, \omega), \tag{3}$$

but the sum ranges, in general, over all subsets of the input set of nodes $S(i)$ of i [63, 56]. The form of functions $\sigma^{\pm}(x_j)$ is

$$\sigma_{ij}^+(i, \omega) = \begin{cases} k_{i,\omega} & \text{if } x_j > \theta_{ij}, \\ 0 & \text{if } x_j < \theta_{ij}, \end{cases} \quad \sigma_{ij}^-(i, \omega) = \begin{cases} k_{i,\omega} & \text{if } x_j < \theta_{ij}, \\ 0 & \text{if } x_j > \theta_{ij}, \end{cases}.$$

For every node $i \in V$ the set of source nodes of i decomposes $S(i) := S^+(i) \cup S^-(i)$ into the set of positive and negative regulators of i respectively, by requiring that $j \in S^+(j)$ ($j \in S^-(j)$) if and only if $j \to i$ ($j \dashv i$). The collection of numbers $K := \{k_{i,\omega}\}$ satisfies the monotonicity assumption

$$k_{i,\omega} \leq k_{i,\omega'} \quad \text{for} \quad \omega \subset \omega'$$

for subsets $\omega \subset \omega' \subset S^+(i)$ of positive inputs. The monotonicity condition reverses

$$k_{i,\omega} \geq k_{i,\omega'} \quad \text{for} \quad \omega \subset \omega'$$

for for subsets $\omega \subset \omega' \subset S^-(i)$, see [11].

The motivating assumption is that the basic expression level of j-th input is $k_{i,j}$ for $\omega = \{j\}$; the simultaneous expression of inputs j and s, that corresponds to $\omega = \{j, s\}$, will be at the level $k_{i,js}$ that is larger than both $k_{i,j}$ and $k_{i,s}$. We call the system (1) with nonlinearities (3) a *monotone system*.

There are other slightly different formulations of this system. For instance Ironi [43, 44] replaces $\tilde{x}_j$ in general formulation (1) by a Boolean function with values in B, i.e. $Z_{ij}(x_j, \theta_{ij}) \in B$, and assumes that for any j the thresholds θ_{ij} of variable x_j are distinct and that F_i are multilinear polynomials in the Z_{ij}.

Chronologically the most recent formulation of the ODE system with discontinuous right hand side, which generalizes Glass system (1), was considered in [26]. We will call this a *switching system*. This formulation was used in a comprehensive description of the dynamics across parameter space, called DSGRN (Dynamic Signatures Generated by Regulatory Networks), which is presented in next section.

Consider

$$\dot{x}_i = -\gamma_i x_i + \Lambda_i(\sigma_{i1}^{\pm}(x_1), \dots, \sigma_{in}^{\pm}(x_n)) \tag{4}$$

where each entry is either function σ^+ or a function σ^-, defined by

$$\sigma_{ij}^+(x_j) = \begin{cases} U_{ij} & \text{if } x_j > \theta_{ij}, \\ L_{ij} & \text{if } x_j < \theta_{ij}, \end{cases} \qquad \sigma_{ij}^-(x_j) = \begin{cases} U_{ij} & \text{if } x_j < \theta_{ij}, \\ L_{ij} & \text{if } x_j > \theta_{ij}, \end{cases} .$$

Assume that the functions $\Lambda_i(y_1, \dots, y_n), i = 1, \dots, n$ are of the form

$$\prod_j \sum (y_{i_1}^j + y_{i_2}^j + \dots + y_{i_{k}(j)}^j),$$

where each y_s occurs exactly once. Such a function is automatically *multi-linear* i.e. linear with respect to each y_i, and all coefficients are 1. The values $L_{ij} < U_{ij}$ represent *lower* and *upper* activation (repression) level of gene i by gene j. Note that σ_{ij}^+ mediates an up-regulation of i by j, while σ_{ij}^- mediates a down-regulation.

We describe the difference between monotone system (3) and switching system (4). The first difference is that the nonlinearity F_i is a sum in (3) but function Λ_i in (4) is arbitrary multi-linear function. This provides more freedom to the switching system to match the choice of nonlinearity to biology represented by the network **RN.** On the other hand, the collection of constants K for (3) is only required to meet monotonicity assumption, but is otherwise arbitrary. The collection K can be computed for any choice of multilinear function Λ_i, but the type of multilinear function that has to be selected for each i in switching system (4) will restrict in subtle way what collections K are attainable as parameters $L_{ij} < U_{ij}$ are allowed to vary.

We now describe the differences between Glass system (1) and switching system (4). First, (4) allows decay constants α_i to vary between nodes i. More importantly, there is a difference between how the parameters enter function F_i vs. Λ_i. In (1) the coefficients of function F_i parameterize expression level of gene x_i, while in (4) the expression level of x_i is determined directly by the values of U_{ij}, L_{ij} along the input edges to i, together with Λ_i that determines how they are combined to the final value of Λ_i. Restricting the range of each function $\sigma^{\pm}$ to $U = 1$ and $L = 0$, the function σ^+ (or σ^-) becomes an increasing (decreasing) Boolean function of the Boolean variable $\tilde{x}_j$ (see (2)). Combining such functions in Λ_i results in a monotone Boolean function. Using general values L, U instead of Boolean values 0, 1 represents one way in which the switching system generalizes the Glass model.

The second difference comes from the number of different thresholds that (4) considers. In Glass system (1) when $x_i > \theta_i$ then the state $\tilde{x}_i$ of gene i is $\tilde{x}_i = 1$ and this value enters the function $F(\tilde{x})$. In other words, all the downstream nodes are activated. On the other hand, in a switching system (4) a node i activates downstream nodes x_j at their own thresholds θ_{ji} and it may activate some, but not others of the downstream nodes. This depends on. the relative size of the value of the activated gene Λ_i and the thresholds associated to each downstream edge under a generic condition

$$\theta_{ji} \neq \theta_{ki} \qquad \text{for all } i \text{ and all } k \neq j. \tag{5}$$

While the continuous time systems (1,3,4) were developed to address some deficiencies of Boolean networks related to the lack of continuity of time, space and parameters, these systems generate their own sets of technical problems.

The main technical difficulty in the traditional view of (1,3, 4) when viewed as systems of ordinary differential equations, is in merging solutions that exist in each domain (Figure 1a) to global solutions that are defined for all $t > 0$. One problem is posed by the solutions in domain D that enter co-dimension k-manifolds with $k \geq 2$ that are formed as intersections of more than two neighboring domains. The more serious problem that affects an open set of solutions is when the solutions from neighboring domains both enter their common co-dimension one intersection which is a
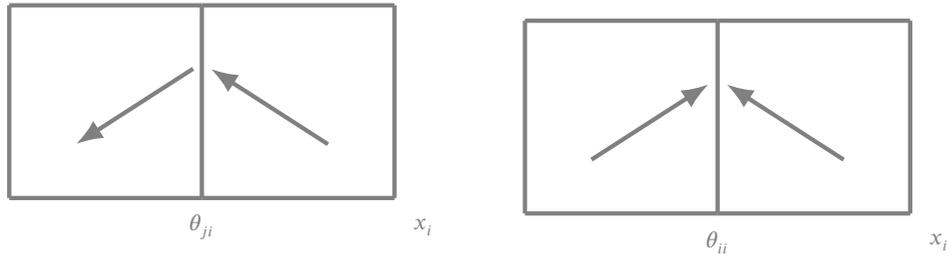
(a) Solutions can be joined across a threshold θ_{ji} at which x_i affects x_j.

(b) Negative self-regulation of node i at threshold θ_{ii}.

Figure 1:

subset of $\{x \in \mathbb{R}^n \mid x_i = \theta_{ji}\}$ (Figure 1b.) It is easy to see that this happens if, and only if, the node $i \in \mathbf{RN}$ has a negative self-loop.

There has been a substantial body of work devoted to the proper definition of solutions of (1,3, 4) for all $t > 0$. They either invoke Filippov's idea of differential inclusions and replacing (1,3, 4) by corresponding differential inclusions [31], or consider only those solutions that are limits of solutions of nearby steep sigmoidal functions ([43, 44, 71]. However, these issues distract from the fundamental problem of developing a modeling toolbox for biology and often form a un-surmountable block for biologists who may otherwise be tempted to use it.

It is our view that the system (4) should be viewed as a computational model for computing lattices of *attracting blocks,* which are sets in the phase space that are positively invariant and form neighborhoods of the attractors. In can be shown, that the lattice of attracting blocks of (4) is part of the lattice of attracting blocks for all neighboring smooth ODE systems [34]. This shifts focus from the individual trajectories to the attracting sets, which are robust under the change of nonlinearity from piecewise constant to smooth.

In order to avoid this technical difficulty, the following assumption is often made

Assumption: The network **RN** has no negative self-regulation.

In the context of the applications, a network is usually a simplification of a more complex network and inclusion of an intermediary like an mRNA, or another protein into a negative self-loop allows this assumption to be satisfied.

4.1. State transition graph

Both Boolean maps and ordinary differential equation (ODE) systems (1, 3,4) admit a discrete description of the phase space. For Boolean map f, we will always represent dynamics by iterates of its multivalued asynchronous update function $\mathbb{F} : H \rightrightarrows H$, which can be viewed as a graph on n-dimensional hypercube H.

For the Glass ODE system (1) the underlying phase space for differential equations is $\mathbb{R}^n$. However, the phase space R^n decomposes to 2^n domains $D(\alpha)$ where $\alpha \in B^n$ and there are bijections between the set of domains $D(\alpha)$, the set $\alpha \in B^n$, and the set of vertices of H.

The *state transition graph (STG)* for ODE models (1) on H is defined analogously to the Boolean model. We connect $\alpha \in H$ to all its neighbors α' in H with $|\alpha_i - \alpha'_i| = 1$ for all indices i with the property that

$$[\frac{F(\alpha)}{\gamma}]_i(\alpha) \neq \alpha_i.$$

The STG can be viewed as a multi-valued map $\mathbb{F}_G : H \rightrightarrows H$, analogous to an asynchronous update of the Boolean function $\mathbb{F} : H \rightrightarrows H$.

The connection between the dynamics of iterates of $\mathbb{F}_G : H \rightrightarrows H$ and the solutions of ODE (1) is the at the heart of the approach described in this paper. Its origin is the observation that all ODE solutions in $D(\alpha)$ converge toward $\frac{F(\alpha)}{\gamma}$, while in the domain D. The discrete description of the continuous dynamics has a natural appeal, but the details of how exactly the long term behavior of the solutions of ODE correspond to the long term behavior of iterates of $\mathbb{F}_G$ is subtle [43, 44, 50, 71, 28]. Any attempt to make this comparison on the level of an individual solution must address the technical issues outlined above.

For more general ODE systems (3, 4) the key assumption (5) implies that each variable x_i has $|T(i)|$ thresholds (recall that $T(i)$ is a set of out edges from a node i in the network **RN**). Therefore, by analogy to system (1) where each variable has one threshold, the continuous phase space R^n consists of $\Pi_{i=1}^n (|T(i)| + 1)$ domains $D(\alpha)$ where α is now a multi-index with $\alpha_i \in \{0, 1, 2, \ldots, |T(i)|\}$. Let

$$\bar{H} = \{\alpha \mid \alpha = (\alpha_1, \ldots, \alpha_n), \alpha_i \in \{0, 1, 2, \ldots, |T(i)|\}\}$$

be a generalized hypercube consisting of nodes with labels α. Each node represents a corresponding domain $D(\alpha) \subset R^n$.

We construct a state transition graph (STG) on $\bar{H}$ using the fact that solutions in each domain $D(\alpha)$ converge toward its associated *target point* $T(\alpha)) = (\Lambda_1(\alpha)/\gamma_1, \ldots, (\Lambda_n(\alpha)/\gamma_n)$. STG is a graph representation of *asynchronous update* of the discrete-valued function

$$\Lambda : D(\alpha) \to T(\alpha).$$

This means that STG is a graph on $\bar{H}$ with edges between nodes α, β with $|\alpha - \beta| \le 1$ assigned in the following way

- if $T(\alpha) \in D(\alpha)$ then there is a self-edge $\alpha \to \alpha$;

- if $|\alpha - \beta| = 1$, $\alpha_i = \beta_i - 1$ and the threshold between $D(\alpha), D(\beta)$ is θ, then
    1. $\alpha \to \beta$ if $T_i(\alpha) > \theta$;
    2. $\beta \to \alpha$ if $T_i(\beta) < \theta$.

Note that the the latter assignment is well defined unless both conditions are satisfied at the same time. Then it must be that the threshold θ is a threshold of variable x_i that affects the function Λ_i, which implies that node i regulates itself n the network **RN**. Furthermore, this self regulation must be negative, since the Λ_i value decreases when variable x_i increases from domain $D(\beta)$ to domain $D(\alpha)$. Since we assume that there are no negative self-loops in **RN**, the STG is well defined and there is at most one edge between neighboring nodes in $\bar{H}$. STG can be also viewed as *multi-valued map* $\mathcal{F} : \bar{H} \to \bar{H}$.

The state transition graph is a finite object that can be interrogated in different ways. The Boolean community has implemented a variety of algorithms that use a correspondence between the states of the STG and their description by logical expressions. This resulted in very fast algorithms that decide reachability of certain states from other states; we only mention the selected work on software implementation [30, 38] and applications [12, 5].

The DSGRN approach [26, 25, 24, 33] focuses on the compression of the information about the dynamics of $\mathcal{F}$ by computing strongly connected path components of STG and the reachability properties between them. The strongly connected path components are nodes of a *Morse graph* which is a Haase diagram of a partial order imposed by the reachability on the nodes of Morse graph, that is inherited from reachability in STG. Morse graph is an analogous concept to that of a Morse decomposition [20], restricted to a finite dynamical system $\mathcal{F} : \bar{H} \to \bar{H}$.

Each attractor A of $\mathcal{F}$ i.e. a positively invariant collection of nodes in $\bar{H}$ under $\mathcal{F}$, corresponds to a set of domains $D(\alpha) \subset R^n$ that is positively invariant under the flow of the switching system. This set of domains may consist of several disjoint non-contiguous collections of domains. The union of these domains is called an *attracting block* corresponding to A. The boundary of such an attracting block is transverse to the solutions of (4) and are therefore robust under and C^1 perturbations of the flow, and therefore C^0 perturbation to the nonlinearities. This class of perturbations includes smooth systems in the neighborhood of the piece-wise constant nonlinearities. These ideas have been used in [34] to show that in a two dimensional system under mild conditions, there is an injective map respecting the partial order that maps the Morse graph of the switching system (4) to a Morse decomposition of any differential smooth equation with nonlinearities C^0-close to switching system (4). This shows that the switching system provides a rigorous lower bound on complexity of dynamics of nearby smooth systems. Since switching systems cannot directly capture the existence of singular solutions [68, 63, 56], the fact that switching system only provides a lower bound is not surprising.

5. Dynamics in context of changing parameters.

We now come to the main part of the paper where we discuss how to study either Boolean maps or ODE systems (1-4) as the Boolean functions or nonlinearities change. In the ODE systems, this is called *bifurcation theory* where

---

one seeks to find parameter values where qualitative features of the dynamics abruptly change. Bifurcation values of the parameter bound areas of the parameter space where the dynamics is qualitatively the same.

It has been recognized at least as early as [63] that by systematic change in parameters $k_{i,\omega}$ in system (3) one obtains different qualitative dynamics in STG. This has been formalized in [56, 11] by introducing the concept of *qualitative parameters*, which parameterize potentially different dynamics of the state transition graphs. The recent work [2], inspired by [63], uses the correspondence between Boolean maps and Glass systems (1) to define bifurcation theory for Boolean maps. Recall that for a Boolean function $f : B^n \to B$ one can define the set of truth values $\mathbb{T}(f)$ and false values $\mathbb{F}(f)$ by

$$\mathbb{T}(f) := \{\alpha \in B^n \mid f(\alpha) = 1\}, \qquad \mathbb{F}(f) := \{\alpha \in B^n \mid f(\alpha) = 0\}.$$

Using this concept, a very recent work [27] organizes the structure of the set of monotone Boolean functions f_i by ordering the sets $\mathbb{T}(f_i)$ by inclusion.

A natural parameterization for systems (4) has $3E + N$ continuous parameters (L_{ij}, U_{ij}, and θ_{ij} for each edge (j, i) and γ_i for each node i). We will describe the relationship between this continuous parameter space and the finite number of admissible state transition graphs. We then compare this description to the other approaches.

5.1. Parameter graph

The parameter space of system (4) consists of a collection

$$Q := \{(L_{ij}, U_{ij}, \theta_{ij}, \gamma_i) \quad | \quad \text{for all } (i, j) \in E, i \in V, \text{ such that } L_{ij} \leq U_{ij}.\} \tag{6}$$

We will consider an open subset $P \subset Q$ where

$$\Lambda_i \neq \gamma_i \theta_{ij} \text{ for all } i, j.$$

We make a following definition that leads to a concise description of all dynamics that can be generated by a given network across all parameters.

Definition 3. *For node i let Z_i be the set of input strings $\zeta_i = (a_1, \dots, a_{p_i})$ of the length $p_i := |\mathbf{S}(i)|$, the number of sources of i, where each $a_j \in \{L_{ij}, U_{ij}\}$. A state of a node i at a parameter $p \in P$ is a collection of inequalities*

$$\Sigma_i(p) := \{\Lambda_i(\zeta_i) < \gamma_i \theta_{ji} \text{ or } \Lambda_i(\zeta_i) > \gamma_i \theta_{ji} \mid j \in \mathbf{T}(i), \zeta_i \in Z_i\}$$

where for each pair (ζ_i, j) there is exactly one inequality ($<$ or $>$) in the collection Σ_i. Importantly, since Λ_i is assumed to be a product of sums where each dependent variable occurs at most once, Λ_i is monotone in each of its arguments. The choice of inequalities has to respect this monotonicity.

A state of the system is a collection $\Sigma(p) = (\Sigma_1(p), \Sigma_2(p), \dots, \Sigma_n(p))$.

We make three key observations [34]

1. The parameter $p \in P$, and the identity of domain $D(\alpha)$ determines uniquely the values of strings $\zeta := (\zeta_1(\alpha), \dots, \zeta_n(\alpha))$.
2. Given ζ, the state of the system $\Sigma(p)$ determines position of target points of all domains $D(\alpha)$ and thus determines unique state transition graph.
3. The number of states is finite, and so there are open sets of parameters $p \in P$ that generate the same state $\Sigma(p)$.

A *parameter graph* **PG** for a network **RN** has a node for each state of the system Σ; an undirected edge between two states Σ_1 and Σ_2 exists, if Σ_1 and Σ_2 differ by exactly one inequality.

Because the states of the system depend on the algebraic form of Λ_i, the parameter graph will be different for a different collection of algebraic forms $(\Lambda_1, \dots, \Lambda_n)$. Each assumed algebraic form of the function Λ_i imposes constraints on which the monotone Boolean functions are realizable by that algebraic form [26].

It is worth noting that the set of parameter nodes is equivalent to the set of *qualitative parameters* discussed in Richard *et. al.* [56] and Bernot *et. al.* [11]. The parameter graph **PG** organizes this set into a graph that captures geometrical structure of the corresponding parameter regions in the parameter space.

Note that since each state Σ can be written as $\Sigma = (\Sigma_1, \dots, \Sigma_n)$, the parameter graph **PG** is a product of parameter graphs for each node $i \in \mathbf{RN}$

$$\mathbf{PG} = \Pi_{i=1}^n \mathbf{PG}(i).$$

The product structure significantly simplifies construction of **PG**, since **PG**(i) only depends on the type of the node i; i.e the number of input edges, the number of output edges and the algebraic form of function Λ_i that combines the inputs. These graphs can be precomputed and then combined together to construct **PG** for any network **RN**.

In Figure 2(a) we show **PG** for a network node z with two inputs from nodes x, y and one output. Here we assume that $\Lambda(\sigma^+(x), \sigma^+(y))$ where x, y are the continuous variables representing the concentration of chemical species and $\Lambda(u, v) = uv$ is a product. In order to show connection between DSGRN parameter graph and qualitative parameters of [56, 11], we associate logical variables X to x and Y to y

$$X := \left\{ \begin{array}{ll} 1 & x > \theta_{zx} \\ 0 & x < \theta_{zx} \end{array} \right. \qquad Y := \left\{ \begin{array}{ll} 1 & x > \theta_{zy} \\ 0 & x < \theta_{zy} \end{array} \right.$$

which represent Booleanization of $\sigma^+(x), \sigma^+(y)$, respectively. In Figure 2(a) we use subscripts X and Y in L_X, U_X, L_Y, U_Y instead of $L_{zx}, U_{zx}, L_{yx}, U_{yx}$ to enable comparison to Boolean description below.

In order to present a succinct representation of parameter nodes, we only include in Figure 2(a) the minimal set of inequalities from which the other inequalities characterizing the parameter node that can be deduced by using standard inequalities $L_X < U_X, L_Y < U_Y$. For instance, the top node in Figure 2(a) also includes implicit inequalities $L_X L_Y < U_X L_Y, L_X L_Y < L_X U_Y, U_X L_Y < U_X U_Y, L_X U_Y < U_X U_Y$. Observe that each node differs from neighboring nodes by exactly one inequality. We also note that the structure of this parameter graph would be the same if we chose $\Lambda(u, v) = u + v$. For functions Λ with more that two inputs, the parameter graphs may be different for different algebraic forms of Λ.

We now connect the parameter graph to the language of Boolean maps. Consider a parameter node $\Sigma = (\Sigma_1, \ldots, \Sigma_n)$ and assume that the j node of **RN** has k-inputs and one output node $|\mathbf{S}(j)| = k, |\mathbf{T}(j)| = 1$. We demonstrate that the value Σ_j corresponds to unique Boolean function $g : B^k \to B$. As a first step, we map every input sequence $\zeta_j = (a_1, \ldots, a_k), a_i \in \{L_{ji}, U_{ji}\}$ to a sequence $\eta_j \in B^k$ by mapping each U to 1 and L to 0. If an input sequence $\zeta_j \in \{L, U\}^k$ is such that the inequality $\Lambda_j(\zeta_j) < \gamma\theta$ belongs to Σ_j, then we assign the corresponding sequence $\eta_j \in B^k$ to the false set $\eta_j \in \mathbb{F}(g)$. Similarly, if an input sequence ζ_j is such that the inequality $\Lambda_j(\zeta_j) > \gamma\theta$ belongs to Σ_j, then we assign the corresponding sequence $\eta_j \in B^k$ to the true set $\eta_j \in \mathbb{T}(g)$. This uniquely associates to Σ_j a Boolean function g for which $\mathbb{F} = \mathbb{F}(g)$ and $\mathbb{T} = \mathbb{T}(g)$ with $B^k = \mathbb{T}(g) \cup \mathbb{F}(g)$. The collection $\{\mathbb{T}(g)\}_g$ over all Boolean functions $g \in B^k \to B$, ordered by inclusion, produces a partial order that recovers the lattice structure of Figure 2(a). Each set of true values $\mathbb{T}(g)$ gives rise to a description of the function g in terms of logical formulas depicted in Figure 2(b). Therefore, there is one-to-one correspondence between parameter graph in Figure 2(a) and the graph of Boolean functions Figure 2(b).

Consider now a parameter node $\Sigma = (\Sigma_1, \ldots, \Sigma_n)$ where the j node of **RN** has k-inputs and ℓ output nodes $|\mathbf{S}(j)| = k, |\mathbf{T}(j)| = \ell$ with corresponding thresholds $\theta_1 < \theta_2 < \ldots < \theta_\ell$. The value Σ_j corresponds to a collection of ℓ monotone Boolean functions $g := (g_1, g_2, \ldots, g_\ell)$ where the set of true values $\mathbb{T}(g_i)$ of a function g_i is a collection of $\eta_j \in B^k$ for which the corresponding ζ_j satisfy $\Lambda_j(\zeta_j) > \gamma_j\theta_i$. The ordering of the thresholds implies that the true values of the collection g satisfy

$$\mathbb{T}(g_\ell) \subseteq \mathbb{T}(g_{\ell-1}) \subseteq \ldots \mathbb{T}(g_1).$$

In the Figure 2(c) is the **PG**(i) for a network node i with one input and two output edges. Since each output edge corresponds to a threshold, there are two thresholds θ_1 and θ_2. To explain the shape of this graph, assume first that $\theta_1 < \theta_2$. If $L < U$ denote two possible values of the input, then we have the following collection of parameter nodes

$$(a_1)\ U < \theta_1, \qquad (b_1)\ L < \theta_1 < U < \theta_2, \quad (c_1)\ \theta_1 < L < U > \theta_2;$$
$$(d_1)\ L < \theta_1 < \theta_2 < U; \quad (e_1)\ \theta_1 < L < \theta_2 < U; \quad (f_1)\ \theta_2 < L$$

The nodes $a_2, \ldots, f_2$ in Figure 2(c) with the subscript 2 are obtained by exchanging θ_1 and θ_2 in the above formulas, which is the equivalent to changing their mutual order to $\theta_2 < \theta_1$. Nodes that differ by a single inequality are connected by an edge.

Computations of parameter graph **PG** are limited by memory storage, rather than the computational complexity. DSGRN databases with billion (10^9) nodes has been computed. These typically correspond to networks with 6-10 nodes. The precise number depends more sensitively on the complexity of regulation at each node, rather than on the
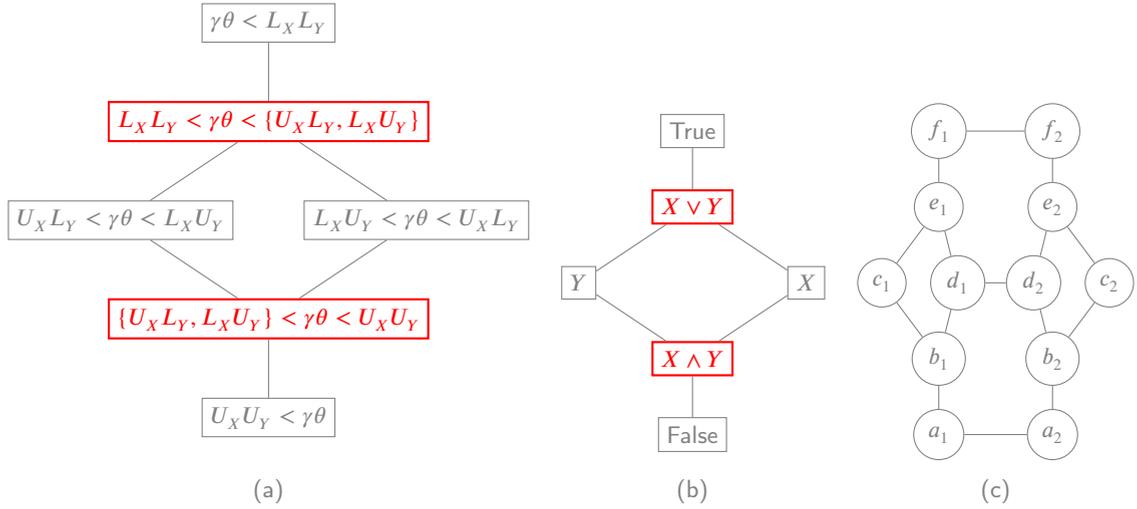
Figure 2: (a) Parameter graph for two input edges from nodes x and y with inputs $L_X < U_X$ and $L_Y < U_Y$ respectively, and one output edge. Each node is marked by the set of inequalities; we only include minimal set of inequalities sufficient to reconstruct all of them. Nodes marked red (thicker boxes) are essential parameter nodes. (b) Description of the corresponding monotone Boolean function; (c) Parameter graph for one input and two output edges. Symmetry between left and right corresponds to change of order of output thresholds.

number of nodes themselves. If a node has only up to two inputs and outputs, the parameter graphs is rather small $O(10^1)$; number of inputs increases this number extremely rapidly.

One of the way to reduce the size of the parameter graph is to employ the concept of *an essential parameter node* [26, 24, 22], related to the concept of *an essential monotone Boolean function* [27, 70]. A parameter node $\Sigma = (\Sigma_1, \ldots, \Sigma_n)$ is *essential* if for every $\Sigma_j, j = 1, \ldots, n$ the following holds:

a For each threshold θ_{ij} associated to $i \in \mathbf{T}(j)$, there is a input string ζ such that $\Lambda_j(\zeta) < \gamma_j \theta_{ij}$ and an input string $\bar{\zeta}$ such that $\Lambda_j(\bar{\zeta}) > \gamma_j \theta_{ij}$;

b For every $k \in \mathbf{S}(j)$ there exist states ζ_1, ζ_2 that differ only in $k-$th component, and a threshold θ_{ij} such that

$$\Lambda(\zeta_1) < \gamma_j \theta_{ij} < \Lambda(\zeta_2) \quad \text{or} \quad \Lambda(\zeta_2) < \gamma_j \theta_{ij} < \Lambda(\zeta_1).$$

In other words, each input edge and output edge plays a role in determining an output of the network node j.

The concept of the essential monotone function [27, 70] coincides with the concept of the essential parameter node for network nodes with a single output. In Figure 2(b) the essential Boolean functions are in red and correspond to those functions where both inputs X and Y play a role in determining the outcome.

A parameter node is *non-essential* if it is not essential. Non-essential parameter nodes can be viewed as parameter nodes of a reduced network where at least one of the input, or output edges, can be removed from the network (Figure 2(b)). DSGRN database restricted to essential parameter nodes is smaller and thus potentially computable for larger networks.

Definition 4. *A DSGRN (Database of Signatures Generated by Regulatory Networks) for a network* **RN**

$$\bigcup_{p \in \mathbf{PG}} M G(p),$$

is a collection of Morse graphs parameterized by the nodes of the parameter graph **PG**.

The idea of considering a class of nonlinearities Λ_i in (4) (or F_i in (1)), rather than analyzing a single Boolean or switching network was considered by several authors. It has been recognized by [29] that precise parameter values can

be replaced by inequalities between parameters to get an identical qualitative description of the dynamics in terms of state transitions. This has been used in [43, 7] to iteratively generate inequalities between the parameters at which the system (4) qualitatively matches the time series of the experimental data by state transitions in STG. In the language of DSGRN, as more of the time series is known, a smaller number of parameter nodes can reproduce the experimental time series. This iterative process zooms in on the parameter regions of interest. These papers use greedy methods which may not find all regions of interest. Compared to DSGRN, the advantage of these methods is, that since it does not compute the entire parameter space, it may be able to handle larger networks; the disadvantage is that for every new time series, the computation needs to be repeated from the start.

More recently [2] introduced *logical bifurcation diagrams* . This is a way to track changes in the state transition graph as a function of a particular parameter. In the language of this paper, this corresponds to a restriction of DSGRN to a particular subgraph **PG**(i) for some network node *i* and considering changes in Morse graph as a function of paths through parameter nodes. The paths of interest are either increasing or decreasing paths in the graphs **PG**(i) like those in Figure 2(a) (c). Since the entry point of [2] is the Boolean network formalism, which is discrete, they introduce the ODE system (1) to embed Boolean functions into a setting with varying parameters in the way that is described above. The emphasis is on the generation of one-dimensional paths through parameter graphs starting at a particular parameter node in **PG**(i) Since **PG**(i) is a partially ordered set there can be multiple choices for these paths.

The work [27] shows how to traverse a graph that only includes essential monotone Boolean functions with *k* inputs. This can generate potential bifurcation paths within essential monotone Boolean functions and can be used to define a neighborhood of an essential monotone Boolean function within this class.

6. Applications

The advantage of the description of the dynamics by DSGRN database is the ability to ask questions that cannot be asked by considering a single Boolean function or an ODE at a single parameter value. Global information about the dynamics of the system is represented in DSGRN in different ways. For instance, the nodes of the Morse graph that have no outgoing edges represent invariant sets that are stable in the sense that there are no other Morse nodes that can be reached from this node. We will call such nodes *stable*. If a Morse graph has two such minimal nodes, the system exhibits bistability; more nodes indicate multi-stability.

Furthermore, we use annotation of the Morse nodes of Morse graph at $p \in$ **PG** as a way to represent a *dynamical signature* of the dynamics of all parameters that belong to the parameter region represented by p. Currently we use three types of annotation. A Morse node is designated FP if it corresponds to a single node $\alpha \in \bar{H}$ and there is a self-loop $\alpha \to \alpha$ in STG. The name FP (for "fixed point") signifies that the corresponding domain $D(\alpha) \subset R^n$ will contain a fixed point of the dynamics of (1,3) or (4). Annotation XC (for Partial Cycle) designates a Morse node that represents a set of domains $D(\alpha_i)$ where there is at least one component j that is constant across the collection α_i, i.e. $[\alpha_i]_j = const$ for all i. In other words, such a component does not change its state in the set of domains $D(\alpha_i)$. Annotation FC (for Full Cycle) describes a Morse node that represents a set of domains $D(\alpha_i)$ where none of the components are constant across all i. We hasten to add that the existence of a stable Morse node FC does not guarantee that for every parameter corresponding to this parameter node, the ODE (4) has a stable periodic orbit, as has been shown by Glass [37]. However, we view such a set of periodic domains as an adequate proxy for the periodicity observed in the experimental data, which is both finite and subject to experimental noise.

One way to use annotated Morse graphs in DSGRN database is to rank networks based on their ability to robustly exhibit an oscillatory dynamics, based on prevalence of stable FC in the DSGRN database. This may be used to rank potential networks that may explain oscillatory phenomena like a cell cycle, a circadian rhythm, or the emergence of a malaria parasite from red blood cells. A network that does not show any FC in the entire parameter graph is clearly not an appropriate model for a system that exhibits periodic behavior. This trivial conclusion may lead to the rejection of many potential networks, and thus may lead to a substantial reduction of hypothesis space. This rejection can be enhanced by assuming that the network must robustly exhibit periodicity represented by FC; this allows a ranking of the networks based on the FC prevalence and a rejection of lower ranked networks.

An additional restriction comes from considering extrema (i.e minima and maxima) of an experimental time series. A path in a state transition graph of a switching system (4) also corresponds to an ordered set of minima and maxima of the coordinate functions in (4). Therefore we can evaluate not only how prevalent is the periodic behavior encoded in Morse set FC, but by comparing the sequence of maxima and minima along an arbitrary path through such an FC with the ordered set of minima and maxima of the same genes in experimental data, we can reject those networks that

do not match the experimental ordering or extrema [25],

We describe a broad set of questions that can be addressed by the approaches described in this article.

6.1. Delineation of network dynamics

E2F-Rb network [48, 3, 54, 16] plays a key role in one of the most important decisions in the human cell life cycle: whether to start DNA replication and initiate proliferation. This decision is based on several factors, but once the process has started DNA replication, it must finish. Therefore, the influence of the factors that led to the decision must be uncoupled at the moment of the decision, called the *restriction point* of the cell cycle [52, 14, 59]. Not surprisingly, mutations in E2F-Rb network are linked to many cancers [17, 19, 49]. The requirement of irreversibility and decoupling suggests that phenotypically a bistable switch may underlie the restriction point.

In [33] the authors analyzed a series of networks inspired by the work of Yao and his collaborators [76, 77]. In ([77]) the authors simplified a large E2F-Rb network by clustering nodes into three groups, with a number of potential edges between them that were consistent with the edges of the larger network. Then they numerically simulated many potential 3-node networks consistent with this information at a large, but fixed, set of parameter values, and evaluated which of these networks most robustly exhibit different types of hysteresis. In [33] this analysis was first repeated by computing DSGRN for all 3-node compatible networks. Then the authors went a step further and looked at eight 6-node networks that were expanded from the 3-node networks and asked which of them most robustly showed the hysteresis behavior. Interestingly, the network that exhibits hysteresis most robustly according to the DSGRN database, matched the topology of the network that performed the same function in yeast. Since the genes involved in the restriction point in yeast share no homology with the genes in humans [23], this result suggested that the evolution may have selected the same network topology independently multiple times to perform the same function.

In order to search for hysteresis in the E2F-Rb, [33] searched paths in the parameter graph $PG(S)$ where S was a node in the graph affected by the growth rate of the cell. Along every path they asked if the set of Morse graphs exhibited the following sequence of events: first, a single FP that represents non-proliferative state, followed by a Morse graph exhibiting bistability, and finally, at the end of the path, a Morse set with single FP representing a proliferative state. This annotated Morse graph representation of hysteresis allowed [33] to enumerate prevalence of hysteresis in parameter graphs of different networks.

6.2. Control and manipulation of dynamics

There have been several attempts to use predictions of Boolean networks to control and manipulate their dynamics. The program *kali* developed by ([55]) considered a single Boolean network and assumed that the attractors, and in particular the fixed point attractors, corresponded to network phenotypes. The goal of the program was to reduce the size of the basins of attraction of the so called pathological attractors that correspond to un-desired outcome of the network dynamics. This program allows a choice between the synchronous and asynchronous updates and allows for multi-valued logic.

Still within a single Boolean network, the series of papers [78, 80, 79, 81] assumed that there was a set of control nodes that can be externally manipulated. They used optimal control techniques to derive a control policy that minimized the likelihood of transitioning into an undesirable state in a stochastic context.

A slightly different approach was taken by Biane and Delaplace [13] who extended the Boolean formalism to *Boolean control network*, where additional *control Boolean nodes* were added to the system. These nodes are not dynamically updated, but they serve as preset control variables. Including them in a logical expression with dynamically updated variables of the Boolean network allows encoding of the node control into extended Boolean framework. One can view this approach as a discrete parameterization of a family of Boolean maps by all possible Boolean values of the control nodes. It is possible to translate the Boolean control framework into the framework of DSGRN.

A mathematically different approach was taken by ([51]). They translated the problem of finding control point candidates to a problem of solving a set of polynomial equations. These were then solved by a powerful techniques of computational algebra. The output was a set of actions that are capable to steer the network towards desirable states.

6.3. Learning parameters that fit the data

An important problem is to match a Boolean or an ODE model to the experimental data, which is often presented in the form of a time series of several (or all) network variables.

As mentioned above, the work of ([7]) and ([43]) consider ODE systems (1)-(4) with the goal of selecting a set of parameter nodes which produce solutions that match experimental data. The key observation is that the information

that needs to be matched is qualitative, since (1)-(4) produce piecewise linear solutions. Both papers ([7])-([43]) start with the assumption that the experimental trajectory has been mapped into a sequence of domains $D(\alpha)$. Then, given an initial domain, ([43]) construct a tree where each branch connects to all neighboring domains and each edge is annotated by a list of parameter inequalities that are consistent with that transition. This results in a very large graph and is not practical for networks with more than a few nodes, even for a relatively short time series.

The paper ([7]) encodes qualitative observations about data into a temporal logic formula and uses model checkers ([5, 8, 12, 32]) to check if a state transition graph generated at a particular parameter node satisfies this formula. To avoid the explicit construction of the parameter graph **PG** they use an alternative approach. They effectively use the branching graph structure of ([43]) without explicitly constructing it. It this way, for a given path through the state transition graph that is generated by the experimental data, the work ([7]) computes parameter constraints that are consistent with the path. These constraints become more restrictive as the length of the path increases.

Another approach to the matching of an experimental time series to the dynamics of network has been done in the context of DSGRN [25]. Given an experimental time series and a proposed network, the goal is to find all parameter nodes of the parameter graph where the dynamics is consistent with the time series. To do this, a partial order of the maxima and the minima of individual time series is extracted from the time series and compared to STG dynamics generated by all parameter nodes of the DSGRN database. The consideration of a partial order rather than a full order is motivated by the fact that experimental time series are often sparsely sampled and multiple genes can achieve maximum or minimum at the same time point. The result of the analysis is a collection of parameter nodes where the discrete STG dynamics exhibits the same partial order of maxima and minima as the experimental time series. Then networks can be compared based on robustness of matching i.e. prevalence of this behavior across all parameters.

There has been a lot of work devoted to learning Boolean networks from data. In the context of a parameter graph PG, this is equivalent to selecting which parameter nodes are consistent with the observed dynamics. The paper [39] introduces *caspo*, a free open-source tool to learn the (Boolean) logic models of signal transduction. It uses CellNOpt pre- and post-processing routines [62]. The output is a set of Boolean models that are capable of reproducing data. In our language, the output is a collection of parameter nodes at which the state transition graph STG has a path that is consistent with the data. The follow-up paper ([73]) goes a step further. Given a collection of potential Boolean models that are consistent with the data, it proposes a minimal set of experiments that are capable of distinguishing between them. In this case, however, the method is restricted only to Boolean networks representing **RN** with no feedback loops.

Finally, ([72]) used methods of computational algebra to learn all Boolean maps that are consistent with the data. The algorithm consists of encoding possible wiring diagrams using ideals and algebraic sets and choosing those that are minimal using primary decomposition and irreducible components.

7. Discussion

We describe how relatively independent ideas developed over the last 50 years in dynamical systems (Glass systems, switching systems) and computer science (Boolean networks, automatic model checking), started to merge into a unified modeling platform for systems biology. Facing the challenge of describing the dynamics of networks, where the nonlinearities, parameters and initial data are known only qualitatively, a combination of continuous time dynamics and its discrete time description allows enumeration of all potential dynamics compatible with the network structure. This description, by necessity, matches a qualitative input that systems biology is able to provide. However, this approach allows

- the delineation and classification of all dynamics consistent with network structure;

- the description of changes in dynamics induced by the changes in parameters;

- the evaluation of network structures for their potential fit to the experimental data;

- the rejection of proposed network structures that are incompatible with the experimental data.

These techniques, including perhaps the most clearly formulated approach called DSGRN [26, 25, 24, 34, 42, 22, 40], have been successfully used in cell biology [33, 75].

We hope that this contribution will spur more applications, as well as further the development of these techniques.

References

[1] A, K., 2003. Monotone Boolean functions. Russ. Math. Surv. 58, 929–1001.

[2] Abou-Jaoude, W., Monteiro, P., 2019. On logical bifurcation diagrams. J. Theor. Biol. 466, 39–63.

[3] Adams, M., Sears, R., Nuckolls, F., Leone, G., Nevins, J., 2000. Complex transcriptional regulatory mechanisms control expression of the e2f3 locus. Mol. Cell. Biol. 20, 3633Ð3639.

[4] Albert, R., Collins, J., Glass, L., 2013. Introduction to Focus Issue: Quantitative approaches to genetic networks. Chaos 23, 025001.

[5] Barnat, J., Briml, L., Cerna, I., Drazan, S., Fabrikova, J., Safranek, D., 2009. On algorithmic analysis of transcriptional regulation by ltl model checking. Theor. Comput. Sci. 410, 3128–3148.

[6] Batt, G., Belta, C., Weiss, R., 2007a. Model checking genetic regulatory networks with parameter uncertainty, in: Bemporad, A., Bicchi, A., Buttazzo, G. (Eds.), Hybrid Systems: Computation and Control, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 61–75.

[7] Batt, G., Page, M., Cantone, I., Goessler, G., Monteiro, P., DeJong, H., 2010. Efficient parameter search for qualitative models of regulatory networks using symbolic model checking. Bioinformatics 26, i603–i610.

[8] Batt, G., Ropers, D., de Jong, H., Geiselmann, J., Mateescu, R., Page, M., Schneider, D., 2005. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in escherichia coli. Bioinformatics 21, i19–28. doi:10.1093/bioinformatics/bti1048.

[9] Batt, G., Yordanov, B., Weiss, R., Belta, C., 2007b. Robustness analysis and tuning of synthetic gene networks. Bioinformatics 23, 2415–2422.

[10] Bernard, O., Gouze, J., 2002. Global qualitative description of a class of nonlinear dynamical systems. Artificial Intelligence 136, 29–59. doi:10.1016/S0004-3702(01)00169-2.

[11] Bernot, G., Cassez, F., Comet, J., Delaplace, F., Muller, C., Roux, O., 2004a. Application of formal methods to biological regulatory networks. J. Theor. Biol 229, 339–348.

[12] Bernot, G., Comet, J., Richard, A., Guespin, J., 2004b. Application of formal methods to biological regulatory networks. J. Theor. Biol 229, 339–348.

[13] Biane, C., Delaplace, F., 2018. Causal reasoning on boolean control networks based on abduction: theory and application to cancer drug discovery. IEEE/ACM Transactions on Computational Biology and Bioinformatics doi:10.1109/TCBB.2018.2889102.

[14] Blagosklonny, M., Pardee, A., 2002. The restriction point of the cell cycle. Cell Cycle 2, 102–109.

[15] Bornholt, S., 2008. Boolean network models of cellular regulation: prospects and limitations. J. R. Soc. Interface 2008 5, 134–150.

[16] Bouchard, C. qnd Thieke, K., Maier, A., Saffrich, R., Hanley-Hyde, J., Ansorge, W., Reed, S., Sicinski, P., Bartek, J., Eilers, M., 1999. Direct induction of cyclin d2 by myc contributes to cell cycle progression and sequestration of p27. embo j. 18: 5321-5333. EMBO J. 18, 5321Ð5333.

[17] Burkhart, D., Sage, J., 2008. Cellular mechanisms of tumour suppression by the retinoblastoma gene. Nat Rev Cancer 8, 671–82.

[18] Chaves, M., Sontag, E., Albert, R., 2006. Methods of robustness analysis for Boolean models of gene control networks. IEEE Proceedings-Systems Biology 153, 154–167. doi:10.1049/ip-syb.

[19] Chinnam, M., Goodrich, D., 2011. RB1, development, and cancer. Curr Top Dev Biol 94, 129–69.

[20] Conley, C., 1978. Isolated Invariant Sets and the Morse Index. Regional conference series in mathematics, R.I.:American Mathematical Society.

[21] Crama, Y., , Hammer, P., 2011. Boolean Functions: Theory, Algorithms, and Applications. Cambridge University Press.

[22] Crawford-Kahrl, P., Cummins, B., Gedeon, T., 2019. Comparison of two combinatorial models of global network dynamics. SIAM J. Appl. Dyn. Syst. 18, 418–457.

[23] Cross, F.R., Buchler, N.E., Skotheim, J.M., 2011. Evolution of networks and sequences in eukaryotic cell cycle control. Phil. Trans. R. Soc. B 366, 3532–3544.

[24] Cummins, B., Gedeon, T., Harker, S., Mischaikow, K., 2017. Database of dynamic signatures generated by regulatory networks (dsgrn), in: Koeppl, J.F.H. (Ed.), Computational Methods in Systems Biology - 2017. Springer. chapter 19, pp. 300–308. doi:10.1007/978-3-319-67471-1.

[25] Cummins, B., Gedeon, T., Harker, S., Mischaikow, K., 2018. Model rejection and parameter reduction via time series. SIAM Journal on Applied Dynamical Systems 17, 1589–1616.

[26] Cummins, B., Gedeon, T., Harker, S., Mischaikow, K., Mok, K., 2016. Combinatorial Representation of Parameter Space for Switching Systems. SIAM Journal on Applied Dynamical Systems 15, 2176–2212.

[27] Cury, J., Monteiro, P., Chaouiya, C., 2019. Partial order on the set of boolean regulatory functions. preprint 2019 .

[28] Edwards, R., 2001. Chaos in neural and gene networks with hard switching. Diff. Eq. Dyn. Sys. , 187–220.

[29] Edwards, R., Glass, L., 2006. A calculus for relating the dynamics and structure of complex biological networks., in: Berry,R.S. and Jortner,J. (eds) Adventures in Chemical Physics, Wiley, Hobeken, USA. pp. 151–178.

[30] Fages, F., Soliman, S., 2008. Formal cell biology in biocham, in: M. Bernardo, P. Degano, G. Zavattaro (Eds.), Formal Methods for Computational Systems Biology, Springer-Berlin-Heidelberg. pp. 54–80.

[31] Filippov, A., 1988. Differential equations with discontinuous right-hand sides. Kluwer Academic Publishers Group.

[32] Fisher, J., Henzinger, T., 2007. Executable cell biology. Nat. Biotechnol 25, 1239–1250.

[33] Gedeon, T., Cummins, B., Harker, S., Mischaikow, K., 2018. Identifying robust hysteresis in networks. PLoS Comput Bio 14, e1006121. doi:https://doi.org/10.1371/journal.pcbi.1006121.

[34] Gedeon, T., Harker, S., Kokubu, H., Mischaikow, K., Oka, H., 2017. Global dynamics for steep sigmoidal nonlinearities in two dimensions. Physica D 339, 18–38.

[35] Glass, L., Kauffman, S., 1972. Co-operative components, spatial localization and oscillatory cellular dynamics. Journal of Theoretical Biology 34, 219–37. URL: http://www.ncbi.nlm.nih.gov/pubmed/5015702.

[36] Glass, L., Kauffman, S., 1973. The logical analysis of continuous, non-linear biochemical control networks. Journal of Theoretical Biology 39, 103–29. URL: http://www.ncbi.nlm.nih.gov/pubmed/4741704.

[37] Glass, L., Pasternack, J.S., 1978. Stable oscillations in mathematical models of biological control systems. Journal of Mathematical Biology

6, 207–223. URL: http://link.springer.com/article/10.1007/BF02547797.

[38] Gonzalez, A., Naldi, A., Sanchez, L., Thieffry, D., , Chaouiya, C., 2006. Ginsim: a software suite for the qualitative modelling, simulation and analysis of regulatory networks. BioSystems 84, 91–100.

[39] Guziolowski, C., Videla, S., Eduati, F., Thiele, S., Cokelaer, T., Siegel, A., JSaez-Rodriguez, J., 2013. Exhaustively characterizing feasible logic models of a signaling network using answer set programming. Bioinformatics 29, 2320–2326.

[40] Harker, S., 2017. Dsgrn software. https://github.com/shaunharker/DSGRN. doi:10.5281/zenodo.805413.

[41] Hodgkin, A., Huxley, A., 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology 117, 500–544.

[42] Huttinga, Z., Cummins, B., Gedeon, T., Mischaikow, K., 2018. Global dynamics for switching systems and their extensions by linear differential equations. Physica D: Nonlinear Phenomena 367, 19–37. doi:10.1016/j.physd.2017.11.003.

[43] Ironi, I., Panzeri, L., 2009. A computational framework for qualitative simulation of nonlinear dynamical models of gene-regulatory networks. BMC Bioinformatics 10, S14.

[44] Ironi, L., Panzeri, L., Plahte, E., Simoncini, V., 2011. Dynamics of actively regulated gene networks. Physica D: Nonlinear Phenomena 240, 779–794. URL: http://linkinghub.elsevier.com/retrieve/pii/S016727891000360X, doi:10.1016/j.physd.2010.12.010.

[45] de Jong, H., 2002. Modeling and simulation of genetic regulatory systems: a literature review. J Comput Biol 9, 67–103. doi:10.1089/10665270252833208.

[46] de Jong, H., Gouze, J., Hernandez, C., Page, M., Sari, T., Geiselmann, J., 2004. Qualitative simulation of genetic regulatory networks using piecewise-linear models. Bull Math Biol 66, 301–40. doi:10.1016/j.bulm.2003.08.010.

[47] Lander, A., 2014. A calculus of purpose. PLoS Biology 2, 0712.

[48] Leone, G., DeGregori, J., Sears, R., Jakoi, L., Nevins, J.R., 1997. Myc and ras collaborate in inducing accumulation of active cyclin e/cdk2 and e2f. Nature 387, 422–6.

[49] Manning, A., Dyson, N., 2012. RB: mitotic implications of a tumour suppressor. Nat Rev Cancer 12, 220–6.

[50] Mestl, T., Plahte, E., Omholt, S., 1995. A mathematical framework for describing and analyzing gene regulatory networks. J. Theor. Biol. 176, 291–300.

[51] Murrugarra, D., Veliz-Cuba, A., Aguilar, B., Laubenbacher, R., 2016. Identification of control targets in boolean molecular network models via computational algebra. BMC Systems Biology 10, 94.

[52] Pardee, A., 1974. A restriction point for control of normal animal cell proliferation. PNAS 71, 1286–1290.

[53] Pauleve, L., Richard, A., 2012. Static analysis of boolean networks based on interaction graphs: A survey. Electronic Notes in Theoretical Computer Science , 93–104.

[54] Perez-Roger, I., Kim, S.H., Griffiths, B., Sewing, A., , Land, H., 1999. Cyclins d1 and d2 mediate myc-induced proliferation via sequestration of p27(kip1) and p21(cip1). EMBO J 18, 5310Ð5320.

[55] Poret, A .and Guziolowski, C., 2018. Therapeutic target discovery using boolean network attractors: improvements of kali. R. Soc. open sci 5, 121852.

[56] Richard, A., Comet, J., Bernot, G., 2005. R. thomas' modeling of biological regulatory networks: introduction of singular states in the qualitative dynamics. Fundamenta Informacae XX, 1–18.

[57] Saadtpour, A., Reka, A., 2013. Boolean modeling of biological regulatory networks: A methodology tutorial. Methods 62, 3–12.

[58] Saari, D., 1977. A global existence theorem for the four-body problem of newtonian mechanics. J. Differential Equations 26, 80–111.

[59] Sears, R., Nevins, J., 2002. Signaling networks that link cell proliferation and cell fate. J Biol Chem 277, 11617–11620.

[60] Snoussi, H., 1989. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. Dynamics and Stability of Systems 4, 565–583.

[61] Snoussi, H., Thomas, R., 1993. Qualitative dynamics of piecewise-linear differential equations: a discrete mapping approach. Bull. Math. Biol 55, 973–991.

[62] Terfve, C.e.a., 2012. Cellnoptr: a flexible toolkit to train protein signaling networks to data using multiple logic formalisms. BMC Syst. Biol 6, 133. doi:https://doi.org/10.1186/1752-0509-6-133.

[63] Thieffry, D., Romero, D., 1999. The modularity of biological regulatory networks. BioiSystems 50, 49–59.

[64] Thieffry, D., Thomas, R., 1995. Dynamical behaviour of biological regulatory networks- ii. immunity control in bacteriophage lambda. Bull. Math. Biol 57, 277–297.

[65] Thomas, R., 1973. Boolean formalization of genetic control circuits. Journal of Theoretical Biology 42, 563–585. doi:10.1016/0022-5193(73)90247-6.

[66] Thomas, R., 1991. Regulatory networks seen as asynchronous automata: A logical description. Journal of Theoretical Biology 153, 1–23. doi:10.1016/S0022-5193(05)80350-9.

[67] Thomas, R., D'Ari, R., 1990. Biological Feedback. CRC-Press.

[68] Thomas, R., Thieffry, D., Kaufman, M., 1995. Dynamical behaviour of biological regulatory networks-i. biological role of feedback loops and practical use of the concept of the loop-characteristic state. Bull Math Biol 57, 247–76.

[69] Tournier, L., Chaves, M., 2009. Uncovering operational interactions in genetic networks using asynchronous boolean dynamics. J Theor Biol 260, 196–209. doi:10.1016/j.jtbi.2009.06.006.

[70] Tournier, L., Goelzer, A., Fromion, V., 2017. Optimal resource allocation enables mathematical exploration of microbial metabolic configurations. J. Math. Biol. 75, 1349–1380.

[71] Veflingstad, S.R., Plahte, E., 2007. Analysis of gene regulatory network models with graded and binary transcriptional responses. Biosystems 90, 323–339.

[72] Veliz-Cuba, A., 2012. An algebraic approach to reverse engineering finite dynamical systems arising from biology. SIAM J. Dynam. Sys. 11, 31–48.

[73] Videla, S., Konokotina, I., Alexopoulos, L., Saez-Rodriguez, J., Schaub, T., Siegel, A., Guziolowski, C., 2015. Designing experiments to

discriminate families of logic models. Frontiers in Bioengineering and Biotechnology 3, 131.

[74] Xia, Z., 1992. The existence of noncollision singularities in newtonian systems. Annals of Mathematics 135, 411–468.

[75] Xin, Y., Cummins, B., Gedeon, T., 2019. Multi-stability in epithelial-mesenchymal transition network. preprint .

[76] Yao, G., Lee, T., Mori, S., Nevins, J., You, L., 2008. A bistable Rb-E2F switch underlies the restriction point. Nat. Cell Biol 10, 476–482.

[77] Yao, G., Tan, C., West, M., Nevins, J., You, L., 2011. Origin of bistability underlying mammalian cell cycle entry. Molecular Systems Biology 485.

[78] Yousefi, M., Datta, A., Dougherty, E., 2012. Optimal intervention strategies for therapeutic methods with fixed-length duration of drug effectiveness. Signal Process IEEE Trans 60, 4930–44.

[79] Yousefi, M., Datta, A., Dougherty, E., 2013. Optimal intervention in Markovian gene regulatory networks with random-length therapeutic response to antitumor drug. Biomed Eng IEEE Transac 60, 3542–52.

[80] Yousefi, M., Dougherty, E., 2013. Intervention in gene regulatory networks with maximal phenotype alteration. Bioinformatics 29, 1758–67.

[81] Yousefi, M., Dougherty, E., 2014. A comparison study of optimal and suboptimal intervention policies for gene regulatory networks in the presence of uncertainty. EURASIP J Bioinforma Syst Biol 2014, 6–6.