Analysis and Synthesis of Boolean Cell-cycle Gene Regulatory Networks

Bane Vasić, Vida Ravanmehr
Department of Electrical and Computer Engineering
University of Arizona
Tucson, AZ, 85721, USA
Email: {vasic,vravanmehr}@ece.arizona.edu

David W. Galbraith
BIO5 Institute and School of Plant Sciences
University of Arizona
Tucson, AZ, 85721, USA
Email: galbraith@arizona.edu

***Abstract*—We provide a summary of Boolean network models for budding yeast cell-cycle including the Irons' model. The model has 18 nodes and involves some auxiliary nodes which allow variability in time delays in the activation and degradation expression level of nodes. We also introduce a Boolean network exhibiting the cell-cycle behavior based on a family of cyclic codes, Projective Geometry codes, which encode the cell mass into the expression level of 7 genes. Both networks have a single cycle attractor and show high robustness against perturbations. A comparison between these networks reveals that using the same concepts and assumptions can capture the fundamental and desirable features in the construction of Boolean cell-cycle gene regulatory networks.**

I. Introduction

In eukaryotics, the cell-cycle, which is the process in which a single cell divides into two daughter cells, consists of four phases: G_1 in which the cell grows, S (Synthesis) in which DNA is replicated, G_2 during which the cell continues to grow and prepares for division, and M (Mitosis) in which the cell is divided into two cells. During the cell-cycle, there are a number of checkpoints that monitor certain parameters such as the cell size, shape, and DNA damage and therefore control the cell-cycle integrity.

Thanks to the great deal of available data and information on budding yeast Saccharomyces cerevisiae (S. cerevisiae), numerous mathematical models have been proposed for the gene regulatory network (GRN) controlling the cell-cycle. The most common models are ordinary differential equations (ODEs) and Boolean networks (BNs).

Chen *et al.* [1], [2] proposed several versions of the ODE model which are consistent with a wide range of wild type and mutant data. Subsequently more sophisticated ODE models were developed that focus on specific parts of the cell-cycle such as transition from G_1 to S phase [3] and exit from M phase [4], [5].

Despite the sophisticated molecular-interaction details, ODE models are difficult to analyze, particularly for networks with large number of genes. In contrast, Boolean networks are the simplest models and yet rich enough to describe the connection of underlying biological processes with the large-scale behavior of the GRN. Li *et al.* [6] proposed a Boolean network with 11 genes which plays a crucial role in ensuring stability of the cell-cycle in S. cerevisiae. However, this model has a single point attractor and does not explicitly show mitotic exit. Furthermore, it is inconsistent with a number of mutant phenotypes. The attractor basins of the BN were studied by Willadsen and Wiles [7]. The model was further extended by Trepode *et al.* [8] by introducing Markovian type of randomness in gene activations. This model demonstrated robustness to variations in gene activation probability. Lee and Huang [9] considered a similar extension in which the expression levels are given by the same deterministic function introduced in [6] but is randomly flipped with some fixed probability. Recently, Irons [10] introduced a more accurate BN by adding auxiliary nodes that can capture activation or degradation delay in gene expression levels. Irons' model has only one cycle attractor and is consistent with a wide range of mutations.

In the context of construction of gene regulatory networks, the most important theoretical questions that arise, are related to the optimality of the BN cell-cycle network topology. For instance, if we have the capability to build a synthetic cell-cycle network from a set of given standard logic gates, how would one choose the gene interactions to achieve the highest robustness? What is the smallest number of genes that guarantees stability, and what is the least amount of logic computations required to maintain it? In order to answer these questions, we introduce a Boolean gene regulatory network exhibiting the cell-cycle behavior with 7 genes that emulates the cell-cycle, while guaranteeing resilience to cellular noise which are modeled as random flips in gene expression levels. Unlike the cell-cycle models which consider the specific genes/proteins participating in the cell-cycle process, this network provides a method to capture the two fundamental characteristics of the cell-cycle (cyclicity and error correcting) with minimal number of genes and with a high robustness. The mathematical tool enabling us to achieve these features is a class of cyclic codes, Projective Geometry codes. The complete analysis is given in a journal version of this paper [11].

We note that despite differences between Irons' approach and ours, both networks have the same properties and use similar methods to capture the fundamental features of cell-cycle. First, both BNs are redundant; in Irons' BN, adding some auxiliary nodes (redundant nodes) leads to time delay in activation and degradation level of nodes. Similarly, there exists gene redundancy in our GRN which ensures self error-

correcting ability in the network. As we will show later, redundancy results in highly robust networks. Also, in both BNs, nodes (genes) are updated synchronously. Another common feature is the noise modeling in these networks. One of the methods Irons used to test the robustness of his model is artificially changing the state of a node from 0 to 1 and vice versa. In our GRN, perturbations or cellular noises are also modeled as random flips in gene expression levels which are the codewords of the Projective Geometry code. However, the difference is that in order to study mutant attractors and checkpoint attractors, Irons fixes the value of nodes (stuck-on assumption) and models permanent failures, while we use Von Neumann error model (transient failure).

The rest of the paper is organized as follows. We begin Section II by introducing the Irons' model which includes time delays in activation and degradation of genes/proteins. In Section III, the robustness of the model is studied for both wild type and mutant phenotypes, and the attractors associated to the wild type, mutants and checkpoints are presented. In Section IV, we present a Boolean network model in which the effects of external factors and cellular noise have been included. In Section V, we construct the GRN based on the cyclic Projective Geometry codes and show the attractor basins corresponding to this BN . Section VI, summarizes our results.

II. Boolean Networks and Time Delays

A. Boolean Networks

A Boolean network (BN) is a system of n nodes $V = \{v_0, v_1, \cdots, v_{n-1}\}$ each of which takes a Boolean value $v_i \in \{0, 1\}$ (0 and 1 represent an inactive and active node, respectively). The expression level of a node v_i at time $t+1$, $v_i^{(t+1)}$, is a function of expression levels of its regulators at time t, denoted by $N_i^{(t)} \subseteq V$. So, $v_i^{(t+1)} = f_i(v_{N_i}^{(t)})$ where f_i converts $v_{N_i}^{(t)}$ into a Boolean value 0 or 1. For simplicity, whenever it is clear from the context, we shall denote $N_i^{(t)}$ as simply N_i. Updating node expression levels can be done either *synchronously* or *asynchronously*. In synchronous updating, all nodes are updated at each time step t. Then, with time progression, the state of the network $\mathbf{v}^{(t)} = \{v_0^{(t)}, v_1^{(t)}, \cdots, v_{n-1}^{(t)}\}$ stays in a stable state called an attractor $\mathcal{A} = \{\mathbf{p}_0, \mathbf{p}_1, \cdots, \mathbf{p}_{q-1}\}$. This means that there is a time point t' for which for all $t \geq t'$, $\mathbf{v}^{(t)} = \mathbf{p}_i$, where $i = t - t' \pmod q$. Attractors can be cyclic ($q > 1$) or fixed point ($q = 1$). In contrast, in asynchronous updating, at most one node is updated at each time [10], [12]. Asynchronous updating is more realistic but is difficult to analyze and is currently used for theoretical analysis of very small networks.

B. Time-delayed Boolean Networks

In order to allow variability in time delays in the activation or degradation expression level of nodes, the technique of adding "auxiliary" nodes was introduced in [10] and is explained here briefly. Suppose $N_v = \{v_1, v_2, \cdots, v_k\}$ be a set of neighbors of v which can activate node v provided that $v_1, v_2, \cdots, v_k$ satisfy logical condition C_{in}. Now, in order to hold the record of C_{in} over 'τ' time steps, a node corresponding to the value of C_{in} together with 'τ' auxiliary nodes are added to the network. They create a "delay line" between N_v and v. To show how to choose 'τ' for a specific node v, Irons defined "*activation delay*" and "*degradation delay*" as described bellow.

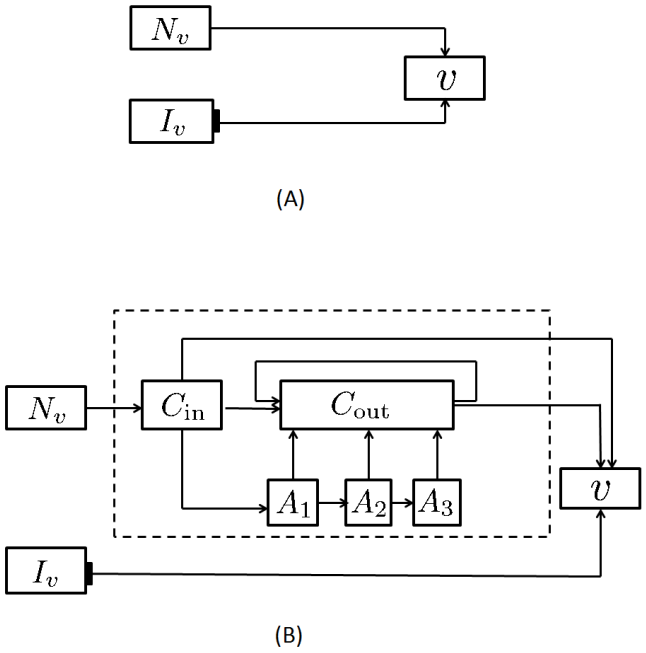

Fig. 1. An example of a Boolean network. (A) A Boolean network without time delays, (B) The Boolean network with time delays ($\tau_A = 3$ and $\tau_D = 4$). A_1, A_2 and A_3 are auxiliary nodes and C_{out} is an extra node which is added to the model with the Boolean functions described in Table II.

Let us suppose the activation delay of node v is τ_A. This means that this node will become active provided the set of nodes N_v satisfy C_{in} for τ_A continuous time steps. If the degradation delay of the active node v is τ_D, this means that v will become inactive after τ_D continuous time steps if the logical condition C_{in} is not satisfied for τ_D continuous time steps, while v maintains its activity in $\tau_D - 1$ continuous time steps. Now, by adding $\tau = \max\{\tau_A - 1, \tau_D - 1\}$ auxiliary nodes to C_{in}, both types of delays are included in the model. Fig. 1(A) shows a simple example of a Boolean network without time delays, while Fig. 1(B) shows that network with time delays $\tau_A = 3$ and $\tau_D = 4$. I_v is a set of nodes which are not involved in time-delaying interactions and in this example inhibits the node v. Boolean functions corresponding to these networks are given in Tables I and II, respectively. The dynamics corresponding to the activation and degradation delay of Fig. 1(B) is shown in Fig. 2(A) and (B).

C. Time-delayed Boolean Network Model for the Budding Yeast Cell-cycle

Now, time delays are applied to the S. cerevisiae cell-cycle GRN. The S. cerevisiae is an oval single cell that divides by

TABLE I
EXAMPLE OF BOOLEAN FUNCTION FOR THE NETWORK IN FIG. 1(A).

Node	Boolean rules which ensure the node v takes state 1
v	$I_v = 0$ AND $N_v = 1$

TABLE II
BOOLEAN FUNCTIONS CORRESPONDING TO THE TIME-DELAYED NETWORK IN FIG. 1(B).

Node	Boolean rules which ensure the node v takes state 1
A_1	C_{in}
A_2	$A_1 = 1$
A_3	$A_2 = 1$
C_{out}	initial activation conditions: C_{in} AND $A_1 = 1$ AND $A_2 = 1$ maintenance activation conditions: $C_{\text{out}} = 1$ AND (C_{in} OR $A_1 = 1$ OR $A_2 = 1$ OR $A_3 = 1$)
v	initial activation conditions: $I_v = 0$ AND C_{in} AND $A_1 = 1$ AND $A_2 = 1$ maintenance activation conditions: $I_v = 0$ AND $C_{\text{out}} = 1$ AND (C_{in} OR $A_1 = 1$ OR $A_2 = 1$ OR $A_3 = 1$)

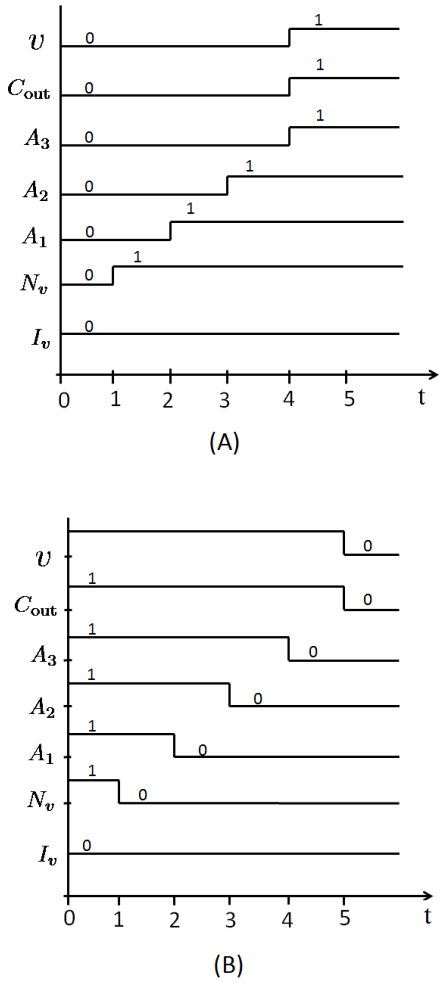

Fig. 2. Dynamics corresponding to the Boolean network in Fig. 1(B) and Boolean functions in Table II. Fig 2(A), shows the activation delay of node v with $\tau_A = 3$ and Fig. 2(B) shows degradation delay of node v with $\tau_D = 4$.

forming a bud. It has G_1 and S phases but does not have a normal G_2 phase. Irons' network has 18 nodes, 14 nodes of which represent groups of genes and proteins and the remaining 4 nodes (named B, S, M and D) represent Bud growth, DNA synthesis, entry to mitosis and cell division, respectively. Time delays and Boolean functions associated with each node are given in Table III. More details about the interactions and transitions in Irons' Boolean network model can be found in [10].

III. ROBUSTNESS OF THE MODEL

In this section, we summarize the robustness analysis of Irons' network for both wild type and mutant phenotypes, and the attractors corresponding to the wild type, mutants, and checkpoints are presented.

A. *Wild type attractors* [10]

The time delays in the wild type model is described in Table III and the only attractor corresponding to this type is shown in Fig. 3(A). This shows that the model is significantly robust to both perturbations in gene activity and variations in reaction times. Any perturbation to any node which is done by putting a node in active or inactive mode (1 or 0, respectively), does not lead the system into a different stable attractor. Also, changing the way of updating nodes and the timing of individual interactions, will not change the stable dynamics. This can be seen in Fig. 3(B) which shows the wild type attractor without additional time delays.

B. *Mutant attractors* [10]

The robustness of the model is also analyzed for a number of mutant phenotypes given in Table IV. The mutant phenotype corresponding to the protein x is shown by $\tilde{x}$. For each mutant, the main attractor has been shown in Fig. 4.

C. *Checkpoint attractors* [10]

As mentioned in the introduction, there are a number of checkpoints in the cell which ensure the cell-cycle safety. For the wild type model, the main attractor corresponding to each of the six different checkpoints is shown in Fig. 5.

The first checkpoint is *start checkpoint* which controls transition from G_1 to S by checking the cell size. The response to this checkpoint is done by fixing Cln3 to 0 and the attractor for this checkpoint is shown in Fig. 5(A).

The second checkpoint, *morphogenesis checkpoint* is also a size checkpoint which controls the transition into mitosis. The response to this checkpoint is done by fixing node B to 0 and can be seen in Fig. 5(B).

TABLE III

TIME-DELAYS AND BOOLEAN FUNCTIONS IN THE BUDDING YEAST CELL-CYCLE MODEL (FROM [10]). THE NOTATION $\{condition\}\langle\tau_A, \tau_D\rangle$ INDICATES THE ACTIVATION DELAY τ_A AND DEGRADATION DELAY τ_D.

Node	Conditions that ensure the node takes state 1
Cln3	$\{$Yhp1=0$\}\langle 2, 1\rangle$
S/MBF	Clb2=0 AND (Cln3=1 OR Cln2=1 OR S/MBF=1)
Cln2	$\{$S/MBF=1$\}\langle 2, 1\rangle$
Clb5	Cdc20=0 and $\{$S/MBF=1$\}\langle 4, 1\rangle$ Cdc20=0 and CKI=0 AND $\{$S/MBF=1$\}\langle 2, 3\rangle$
Yhp1	$\{$S/MBF=1$\}\langle 2, 6\rangle$
Clb2	CKI=0 AND Cdh1=0 AND B=1 CKI=0 AND Cdc20=0 AND B=1 CKI=0 AND Cdh1=0 AND Clb2=1 AND SFF=1 CKI=0 AND Cdc20=0 AND Clb2=1 AND SFF=1
SFF	CKI=0 AND Cdh1=0 AND B=1 CKI=0 AND Cdc20=0 AND B=1 Clb2=1 AND SFF=1
Cdc20	M=1 AND Clb2=1 AND $\{$SFF=1$\}\langle 2, 1\rangle$
FEAR	Cdc20=1
MEN	FEAR=1 AND Clb2=1
Cdc14	FEAR=1 AND MEN=1
Swi5	Clb2=0 AND $\{$SFF=1$\}\langle 2, 3\rangle$ Cdc14=1 AND $\{$SFF=1$\}\langle 2, 3\rangle$
CKI	$\{$Cdc14=1 AND Swi5=1$\}\langle 2, 1\rangle$ Cln2=0 AND Clb5=0 AND Clb2=0 AND $\{$Swi5=1$\}\langle 2, 1\rangle$ Cln2=0 AND Clb5=0 AND Clb2=0 AND CKI=1
Cdh1	Cdc14=1 Cln2=0 AND Clb5=0 AND Clb2=0
S	D=0 AND S=1 D=0 AND $\{$(Clb5=1 OR Clb2=1)$\}\langle 2, 1\rangle$
B	D=0 AND B=1 D=0 AND $\{$(Cln2=1 OR Clb5=1)$\}\langle 6, 1\rangle$
M	D=0 AND M=1 D=0 AND $\{$S=1 AND Clb2=1$\}\langle 2, 1\rangle$
D	D=0 AND $\{$M=1 AND FEAR=1 AND Cdc14=1$\}\langle 2, 1\rangle$

During mitosis, two *spindle checkpoints* prevent anaphase and mitotic exit until the chromosomes and mitotic spindle are correctly aligned. The response to the first one is done by fixing Cdc20 to 0 (Fig. 5(C)) and the response to the second one is done by fixing MEN to 0 (Fig. 5(D)).

The two remaining checkpoints are *DNA damage checkpoints* that can delay both S phase and mitotic exit. The response to these checkpoints can be seen in Figs. 5(E) and (F) by fixing S/MBF to 0 and fixing both FEAR and MEN to 0, respectively.

IV. SYNTHETIC BOOLEAN NETWORK MODEL FOR THE GENE REGULATORY NETWORKS OF CELL-CYCLE

Having studied the Irons' model for the budding yeast S. cerevisiae, we now design a synthetic network exhibiting cyclic and robust behavior. Designing such networks provides a valuable tool in understanding the essential characteristics of one of the most important networks– the cell-cycle gene regulator network. As we mentioned, in this context, the goal is to find the smallest network with these properties. In this section, we introduce a class of finite-systems models of gene regulatory networks exhibiting the cell-cycle behavior and with high robustness. First, we show how the cellular noise is modeled in our Boolean network.

TABLE IV

MUTANTS ASSOCIATED WITH EACH NODE (FROM [10]).

Node	Mutant
Cln3	$\tilde{\text{Cln}}3, \tilde{\text{Bck}}2$
S/MBF	$\tilde{\text{Swi}}4, \tilde{\text{Swi}}6$ $\tilde{\text{Swi}}4, \tilde{\text{Mbp}}1$
Cln2	$\tilde{\text{Cln}}1, \tilde{\text{Cln}}2$
Cdc20	$\tilde{\text{Cdc}}20$
FEAR	$\tilde{\text{Esp}}1$
Swi5	$\tilde{\text{Swi}}5$
CKI	$\tilde{\text{Sic}}1, \tilde{\text{Cdc}}6$

A. Boolean network model in the presence of cellular noise

Consider a Boolean network with a set of n genes $V = \{v_0, v_1, \cdots, v_{n-1}\}$ in which the expression level of gene v_i at time $t+1$ is a function of its regulators at time t, $v_i^{(t+1)} = f_i(v_{N_i}^{(t)})$ where $N_i^{(t)}$ is the regulators of v_i at time t.

In eukaryotic cells, as we mentioned before, the cell-cycle may be influenced by some checkpoints (control factors) which are included as *external factors* that are external to the cell such as the cell size but can be either inside or outside the cell. A change in an external factor level may change the expression level of genes in the GRN. Thus, the external factors are also encoded into the gene expression profile. So, if $U = \{u_0, u_1, \cdots, u_{k-1}\}$ be the set of checkpoints, then

$$v_i^{(t+1)} = f_i(v_{N_i}^{(t)}, u_{E_i}^{(t)})$$

where $E_i \subseteq U$ is the set of external factors affecting the gene i. Without loss of generality, we can assume that external factors take binary values. That is $u_j \in \{0, 1\}$ for $j \in \{0, 1, \cdots, k-1\}$.

Cellular noises may cause changes in gene expression levels which are modeled by random flipping of the output of Boolean functions, i.e., by changing 0 to 1 and vice-versa. In this paper, we assume that the bit flips are independent of each other and independent of time. Such errors are called Von Neumann type of errors [13]. So, the expression level of the gene v_i at time $t+1$ is

$$v_i^{(t+1)} = f_i(v_{N_i}^{(t)}, u_{E_i}^{(t)}) + e_i^{(t+1)},$$

where $e_i^{(t+1)} \in \{0, 1\}$ is the flip (error) of the level of gene v_i at time $t+1$, and the operator "+" which models the bit flip, denotes the modulo-2 sum or Boolean XOR. Therefore,

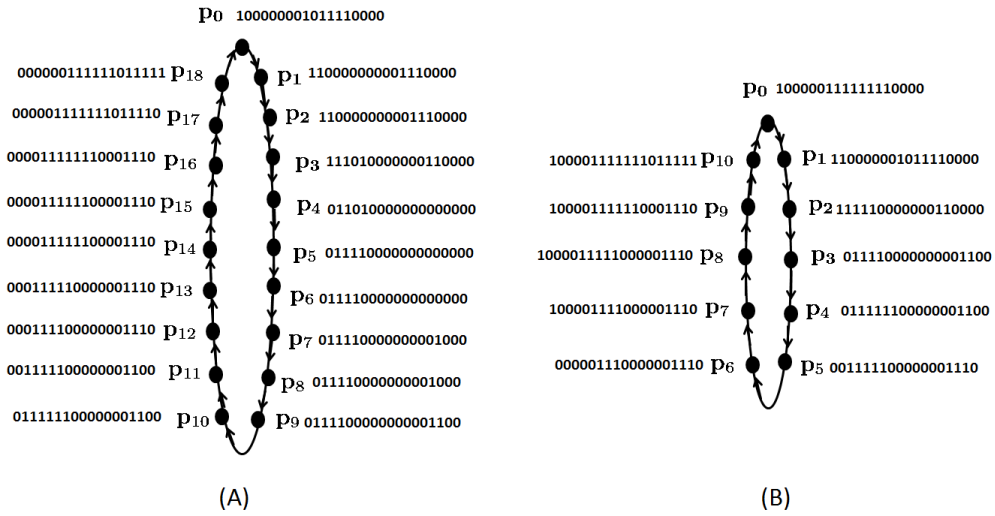

Fig. 3. Wild type attractors associated with the Boolean network model described in Table III. (A) Wild type attractor with time delays. (B) Wild type attractor without additional time delays. The sequence of 18 binary bits represents the activity corresponding to the 18 nodes: Cln3, S/MBF, Cln2, Clb5, Yhp1, Clb2, SFF, Cdc20, FEAR, MEN, Cdc14, Swi5, CKI, Cdh1, S, B, M, D, resp.

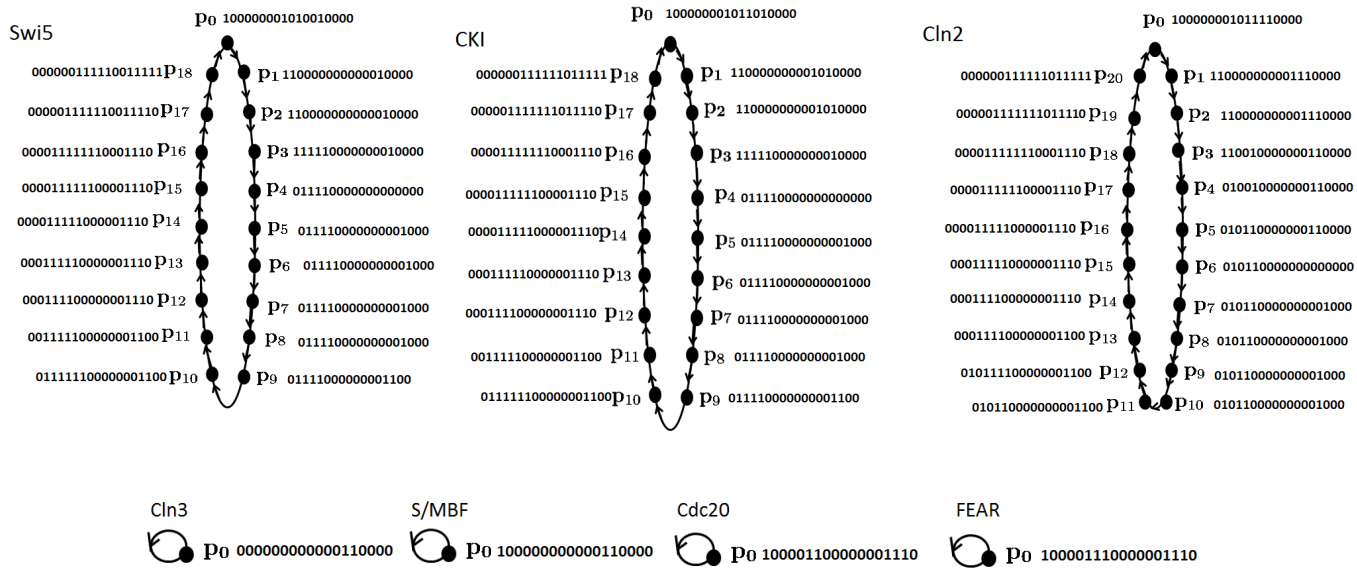

Fig. 4. Attractors corresponding to each mutant phenotypes. The sequence of 18 binary bits represents the activity corresponding to the 18 nodes: Cln3, S/MBF, Cln2, Clb5, Yhp1, Clb2, SFF, Cdc20, FEAR, MEN, Cdc14, Swi5, CKI, Cdh1, S, B, M, D, resp.

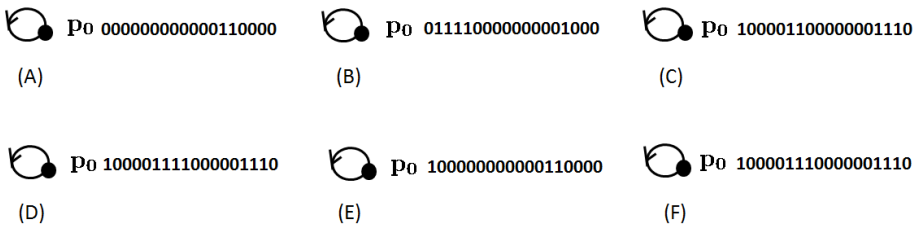

Fig. 5. Attractors corresponding to the activation of checkpoints. The single point attractors in (A)-(F) show the response to the six checkpoints discussed in text, resp. The sequence of 18 binary bits represents the activity corresponding to the 18 nodes: Cln3, S/MBF, Cln2, Clb5, Yhp1, Clb2, SFF, Cdc20, FEAR, MEN, Cdc14, Swi5, CKI, Cdh1, S, B, M, D, resp.

for the vector $\mathbf{v}^{(t+1)}$ of gene activity levels at time $t+1$ and for a given vector of external factors $\mathbf{u}^{(t)}$,

$$\mathbf{v}^{(t+1)} = \mathbf{f}(\mathbf{v}^{(t)}, \mathbf{u}^{(t)}) + \mathbf{e}^{(t+1)} \quad (1)$$

where $\mathbf{e}^{(t)} = (e_0^{(t)}, e_1^{(t)}, \ldots, e_{n-1}^{(t)})$ is the *error vector* at time t.

If the states $\mathbf{w}$ and $\mathbf{v}$ at times $t+1$ and t, respectively satisfy $\mathbf{w} = \mathbf{f}(\mathbf{v}, \mathbf{u})$, then $\mathbf{v}$ is referred to as the *predecessor* of $\mathbf{w}$ (equivalently, $\mathbf{w}$ is the *successor* of $\mathbf{v}$). If all successors of the elements of a set of states $\mathbf{V}$ are also in $\mathbf{V}$, then $\mathbf{V}$ is called a *basin of attraction*. In the absence of errors ($\mathbf{e} = \mathbf{0}$), the sequence of states $\mathbf{v}^{(t)}$ is deterministic and these states are *valid states*. In the presence of errors, some bits are altered, and the resulting state may not be valid. Such states are referred to *erroneous states*.

To ensure self error-correcting, we assume that the number of genes (n) is greater than the number of external factors (k). If $n = k$ and if external factors are independent, then it is not possible to determine whether a given state is a result of a change of the corresponding external factor or due to an internal error.

B. Encoding the cell mass

For simplicity, we assume that the cell mass takes values $0, 1, \cdots, 2^k - 1$ which can be represented by a binary vector $\mathbf{u}$ of length k. Encoding the vector $\mathbf{u}$ is done by the linear map $\mathbf{v} = \mathbf{u}G$, where G is the generator matrix of some linear code $\mathcal{C}$. The 2^k binary vectors (codewords) generated by this linear map are shown by $\mathcal{C}_m$, $m \in \{0, 1, \cdots, 2^k - 1\}$. We call the all-zero codeword $\mathcal{C}_0$ as the *dead state*, since we assume that no gene can be expressed once all n genes in the network are inhibited. With this assumption, we define a correspondence relation between the cell mass and gene expression levels, which exhibits the behavior of the cell-cycle. Without any restrictions, the gene expressions would follow the changes in the cell mass, i.e., $\mathbf{v}^{(t)} = \mathcal{C}_m$ if $\mathbf{u}^{(t)} = u_m$. But the natural behavior of the BN should be as follows. When the sensed cell mass increases from m to $m+1$, the BN should move from the state $\mathcal{C}_m$ to the state $\mathcal{C}_{m+1}$. When the cell mass reaches the value $2^k - 1$, its maximal value, the cell-cycle is completed, and the cell divides. The cell mass of the new cell becomes 1, the BN moves to the state $\mathcal{C}_1$, and the new cell-cycle begins. If the cell mass decreases or does not increase by more than the normal amount (this amount is 1), the progression of cell-cycle should be blocked, and BN should move to the dead state $\mathbf{0}$.

Thus, if BN is in the state $\mathbf{v}^{(t)} = \mathcal{C}_m$, there is only one input, namely, $\mathbf{u}_{m+1}$, which causes transition to the state $\mathcal{C}_{m+1}$. A closed mathematical representation to show this behavior can be given in the following formula.

$$\mathbf{f}(\mathcal{C}_m, \mathbf{u}_i) = \begin{cases} \mathcal{C}_m & \text{if } i = m, \\ \mathcal{C}_{m+1} & \text{if } i = m+1, \\ \mathcal{C}_0 & \text{otherwise.} \end{cases}$$

The operations on indices $1, 2, ..., 2^k - 1$ are "modulo $2^k - 1$". That is, for indices i, j and k we write $k = i + j$ to denote $k - 1 = (i-1) + (j-1) \bmod 2^k - 1$. This means that the state $\mathcal{C}_{2^k}$ is equivalent to $\mathcal{C}_1$.

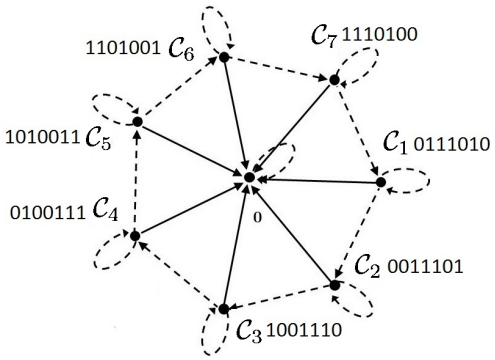

Fig. 6. Cycling through codewords as a response to the increasing cell mass. Black arrows denote transitions in the presence of wrong cell mass leading to the dead state **0** (in the center), while dashed arrows correspond to the cases when the cell mass remains the same (loops), or increases by 1 (straight lines).

V. GRN BASED ON A CYCLIC PROJECTIVE GEOMETRY CODE

To show the cyclic behavior of the network, we use a cyclic code to encode the cell mass into the gene expression levels. The cyclic code we use to encode the cell mass is the cyclic Projective Geometry code (PG(2,2)) of length 7 with the generator matrix G as follows.

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

By the formula $\mathbf{v} = \mathbf{u}G$, we have $v_0 = u_0$, $v_1 = u_1$, $v_2 = u_2$, $v_3 = u_0 + u_2$, $v_4 = u_0 + u_1 + u_2$, $v_5 = u_0 + u_1$ and $v_6 = u_1 + u_2$. Thus, the first three genes encode the cell mass and the remaining genes are redundant. This code has 8 codewords $\mathcal{C} = \{00000000, 0111010, 0011101, 1001110, 0100111, 1010011, 1101001, 1110100\}$.

The seven nonzero codewords $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_7$ encode the binary representation of the cell mass m. The codeword $\mathcal{C}_m$ corresponds to the mass m and the change from $\mathcal{C}_m$ to $\mathcal{C}_{m+1}$ requires only a cyclic shift of the codeword $\mathcal{C}_m$.

One notices that the codeword $\mathcal{C}_m$ is equal to $\mathcal{C}_{m-1}$ cyclically shifted to the right (all index operations are modulo-7, the number of genes). This is a crucial property of $\mathcal{C}$ leading to a single cycle attractor in a BN as we demonstrate in a moment (*Remark:* In reality, there are two attractors, namely, $\{\mathcal{C}_0\}$ and $\{\mathcal{C}_i : i = 1, \ldots, 2^k - 1\}$. However, we do not consider the former since it corresponds to the dead state). Also, it is easy to see that our network is the smallest nontrivial network with cyclic behavior and error-correcting ability as the cell-cycle.

Cycling through the codewords is shown in Fig. 6. Both Fig. 3(A) and Fig. 6 show the only cyclic attractors corresponding to these two BNs.

In addition to cycle progression and division, the model includes the effect of errors and checkpoints. At the cell-cycle checkpoints, the GRN verifies that all the errors are corrected

and that the combination of gene expressions is such that the next phase can start.

At the checkpoints, the satisfaction of all the parity checks is verified by calculating the syndrome. If there is at least one unsatisfied check, the error correction mechanism is activated. the error correction mode terminates when all parity checks are satisfied, which is signaled by a control variable s. Then each bit v_i takes its corrected value v_i^{corr}. It is only then that the cell-cycle can move forward. In other words, at each checkpoint, the error correction syndrome $\mathbf{s}$ of length m is calculated, and checked if $\mathbf{s} = \mathbf{0}$. The result is an internal control variable s defined as $s = \prod_{j=0}^{m-1}(s_j+1)$. In the absence of errors, s is equal to 1.

To formulate the error correction mechanism, we need a parity check matrix H of PG(2,2) code which is given as follows.

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

PG codes can be decoded by one-step majority logic [14]. For the parity check matrix of PG(2,2), updating the variable nodes is identical. For instance, $v_0^{(t+1)} = v_0^{(t)} + \mathrm{MAJ}(\langle c_0\rangle, \langle c_1\rangle, \langle c_3\rangle)$ where, $\langle c_0\rangle = v_0^{(t)} + v_4^{(t)} + v_6^{(t)}$, $\langle c_1\rangle = v_0^{(t)} + v_1^{(t)} + v_5^{(t)}$, $\langle c_3\rangle = v_0^{(t)} + v_2^{(t)} + v_3^{(t)}$ and MAJ denotes a majority function which outputs the value of the majority of its arguments. In our algorithm, we use a binary control argument c which toggles the system between "correction mode" to "shifting mode". $c = 1$ turns on the correction mode, while $c = 0$ enables the system to move to the next state or stay in the same state. Now, we describe our cell-cycle like algorithm:

Algorithm: Update Rule [11]

a) Initialization

Let $c = 0$, $t = 0$, $\mathbf{u}^{(t)} = (0,1,1)$, $\mathbf{v}^{(t)} = \mathbf{u}^{(t)}G$ = $(0,1,1,1,0,1,0)$ = $\mathcal{C}_1$.

b) Correction mode

Let $t = t + 1$ and $c = c + 1$. Thus, c equals 1 for odd t, and 0 for even t. Next, compute

$$s = \prod_{i=0}^{6}(1 + \langle c_i\rangle)$$

where $\langle c_i\rangle = v_i^{(t)} + v_{i+4}^{(t)} + v_{i+6}^{(t)}$. Then for all $i \in \{0, 1, \cdots, 6\}$, let v_i^{corr} denote the result of one-step majority logic decoding. Compute $v_i^{(t+1)}$ as:

$$v_i^{(t+1)} = \mathrm{MUX}(v_i^{(t)}, \mathrm{MUX}(v_i^{corr}, v_i^{(t)}, s), c) \quad (2)$$

where MUX function is defined as follows.

$$\mathrm{MUX}(a,b,c) = \begin{cases} a & \text{if } c = 0, \\ b & \text{if } c = 1. \end{cases}$$

c) Routing-control variable update

The time counter is incremented by 1. So, $t = t + 1$ and $c = c + 1$. Then, recompute s. $s = 1$ implies correction to a valid gene expression vector, while $s = 0$ implies decoding failure. Such a state automatically drives the BN to bring the cell to the dead state. Now, the BN checks if the elements of the external factor vector corresponding to the cell mass have the right values, the values encoded in the genes in the indices set $M = \{0, 1, 2\}$ or the cyclically shifted (to the left) gene expression values corresponding to the genes in M in the next state. Now, compute

$$c_{stay} = \prod_{i\in M}(1 + v_i^{(t)} + u_i^{(t)}),$$

$$c_{move} = \prod_{i\in M}(1 + v_{i-1}^{(t)} + u_i^{(t)}),$$

and

$$c_{dead} = (1 + c_{stay})(1 + c_{move}).$$

d) Gene expression updates

Finally, the gene expression $v_i^{(t+1)}$ is updated as follows.

$$v_i^{(t+1)} = \mathrm{MUX}(\mathrm{MUX}(v_i^{(t)}, v_{i-1}^{(t)}, c_{move}), 0, c_{dead}) \quad (3)$$

e) Go to Step b)

The BN has eight attractor basins which are all the eight codewords of PG(2,2). The attractor basin of the state $\mathcal{C}_i$ is the set of vectors $\{v : v \in E_0 \cup E_1 \cup E_6 \cup E_7\}$ where E_i for $i = 0, 1, 6, 7$, is the set of all error patterns of weight i. According to the properties of PG(2,2) code, the 16 error patterns in $E_0 \cup E_1 \cup E_6 \cup E_7$ are correctable. The attractor basins are shown in Fig. 7. Specifically, the attractor basin corresponding to the all-zero codeword has been shown in Fig. 8. It is notable that the majority logic decoding algorithm never produces an invalid word. It always converges to a codeword. Thus the decoding failure never happens. The algorithm may produce a wrong state which results in a transition to the dead state.

VI. Conclusion

We have given a summary of Irons' Boolean network model for the budding yeast S. cerevisiae which has a single cycle attractor and is robust enough to a wide range of perturbations. We have also presented an information theoretic method for constructing Boolean networks exhibiting properties of the cell-cycle. The method is based on algorithms in coding theory and results in a network robust to noisy perturbations. A comparison between these networks with different structures and approaches, shows how using common concepts leads to construction of highly robust BNs with a single cycle attractor.

Acknowledgment

This work was funded by NSF under the grant CCF-0830245. The authors would like to thank Shiva Kumar Planjery for his useful comments.

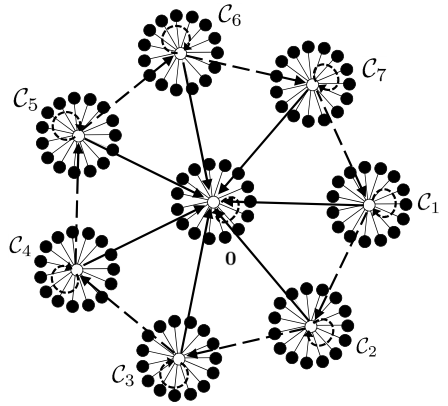

Fig. 7. Attractor basins of a Boolean network based on a PG code of length $n = 7$. ● denotes erroneous states corresponding to the all 16 correctable error patterns, and ○ denotes the codewords: $\mathbf{0}$, and $\mathcal{C}_i$, $i = 1, 2, \ldots, 7$. Black arrows denote the transitions in the presence of errors, while dashed arrows are error free transitions.

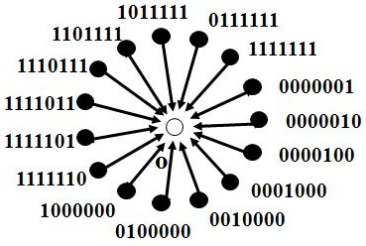

Fig. 8. Attractor basin corresponding to the all-zero codeword.

REFERENCES

[1] K. C. Chen, A. Csikasz-Nagy, B. Gyorrfy, J. Val, B. Novak, J. J. Tyson, "Kinetic analysis of a molecular model of budding yeast cell cycle," *Mol. Biol. Cell*, vol. 11, pp. 369–391, 2000.

[2] K. C. Chen, A. Csikasz-Nagy, B. Gyorrfy, F. R. Cross, B. Novak, J. J. Tyson, "Integrative analysis of cell cycle control in budding yeast," *Mol. Biol. Cell*, vol. 15, pp. 3841–3862, 2004.

[3] M. Barberis, E. Klipp, M. Vanoni, L. Alberghina, "Cell size at S phase initiation: an emergent property of the G_1/S network," *PLoS Comp. Biol.* vol. 3, p. e64, 2007.

[4] E. Queralt, C. Lehane, B. Novak, F. Uhlmann, "Downregulation of $\mathrm{PP2A}^{Cdc55}$ phosphatase by separase initiates mitotic exit in budding yeast," *Cell 125*, pp. 719–732, 2006.

[5] A. Tóth, E. Queralt, F. Uhlmann, B. Novak, "Mitotic exit in two dimensions," *J. Theor. Biol.*, vol. 248, pp. 560–573, 2007.

[6] F. Li, T. Long, Q. Quyang, C. Tang, "The yeast cell-cycle network is robustly designed," *Proceeding of the National Academy of Sciences of the United States of America*, vol. 101, pp. 4781–4786, 2004.

[7] K. Willadsen, J. Wiles, "Robustness and state-space structure of Boolean gene regulatory models," *Journal of Theoretical Biology*, vol. 249, no. 4, pp. 749–765, 2007.

[8] N. W. Trepode, H. A. Armelin, M. Bittner, J. Barrera, M. D. Gubitoso, R. F. Hashimoto, "A robust structural PGN model for control of cell-cycle progression stabilized by negative feedbacks," *EURASIP J. Bioinformatics Syst. Biol.*, vol. 2007, pp. 1–11, 2007.

[9] W. Lee, J. Huang, "Robustness and topology of the yeast cell cycle boolean network," *FEBS Letters*, vol. 583, no. 5, pp. 927–932, 2009.

[10] D. J. Irons, "Logical analysis of the budding yeast cell cycle," *Journal of Theoretical Biology*, vol. 257, pp. 543–559, 2009.

[11] B. Vasic, V. Ravanmehr, A. R. Krishnan, "An information theoretic approach to constructing robust Boolean gene regulatory networks," *Submitted to IEEE/ACM, Transactions on Computational Biology and Bioinformatics*.

[12] K. Mangla, D. L. Dill, M. A. Horowitz, "Timing robustness in the budding and fission yeast cell cycles," *PloS ONE*, vol. 5, no. 2, p. e8906, 2010. [Online]. Available: http://www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0008906.

[13] J. Von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Brain theory: reprint volume*, p. 110, 1988.

[14] S. Lin, D. J. Costello, "Error Control Coding: Fundamentals and Applications," *Englewood Cliffs, NJ: Prentice-Hall*, 1993.