A Study of 3-gene Regulation Networks Using NK- Boolean Network Model and Fuzzy Logic Networking

Trina Kok, Pratt Fellow **Box 97815** **Duke University** **Durham, NC 27708** **+1 (919) 218 7664** **tk16@duke.edu**	**Paul Wang, Professor** **Box 90291** **ECE Department** **Pratt School of Engineering** **Duke University** **Durham, NC 27708** **+1 (919) 660 5259** **ppw@ee.duke.edu**

Abstract

Boolean network theory, proposed by Stuart A. Kauffman about 3 decades ago, is more general than the cellular automata theory of von Neumann. This theory has many potential applications, and one especially significant application is in the modeling of genetic networking behavior. In order to understand the genomic regulations of a living cell, one must investigate the chaotic phenomena of some simple Boolean networks.

We studied a very basic and simple 3-genes regulation network. Different combinations of the three basic logic elements: AND, OR and COMPLEMENT resulted in different logic functions. We studied the influence of these logic functions on steady state behavior of the attractors and limit cycles patterns of cells.

In evaluating the degrees of gene expression using Boolean network theory, it is necessary to quantize the expression levels to "1" and "0". "1" indicates that the gene is expressed and a protein is formed; "0" indicates that the gene is not expressed at all. However, gene expression occurs in many stages, and it is not uncommon for the expression of a gene to cease in one of the intermediate steps. Thus, there is a need for the development of a model to represent the varying degrees of gene expression. We used Fuzzy Logic Networking to circumvent the information loss associated with quantization.

Hopefully, a complete dictionary of the classification or taxonomy, of all possible chaotic patterns can be established, as it is useful in the sense that more complex chaotic behavior resulted from gene regulation can be derived from the basic patterns in it. It is highly possible that the "reverse engineering" problem can be completely solved theoretically for the 3-gene networks.

I. Introduction: NK-Boolean Network Modeling

Boolean models, pioneered by Kauffman [1] – [2], considers each gene symbolically as either ON/1 (expressed) or OFF/0 (not expressed), so that the continuous data obtained from microarray technology has to be quantized to these two levels. States of genes in time 't' regulate the states of genes in time 't+1' via logic functions consisting of AND, OR and COMPLEMENT connectors. As regulation proceeds among genes in a parallel manner, a synchronized genetic regulatory network evolves. An NK-Boolean network is set up by the evolution of N genes with K connectivity, where N refers to the total number of genes in the network, and K refers to the maximum number of genes that regulate some single gene. The number of possible states for such a network and the amount of data necessary for its elucidation is 2^N.

Assuming a network with maximum connectivity (K=N) like those studied by Wang et al [3] – [6], there are $(2^{2^N}-2)^N$ possible logic functions. Thus it may be concluded that the Boolean network model results in an uninformative discrete representation of gene expression and activity profiles, and lead to an intractable solution.

In addressing the above concerns, it was found that real regulatory networks typically have low connectivity, which translates into low K values [7]. Only a small fraction of the 2^N possible gene expression states are fulfilled where unfulfilled states represent unstable states. Thus, a low K connectivity results in a tractable solution with a smaller number of possible logic functions. Also, the Boolean network model retains sufficient biological information to realistically model genetic regulatory networks as indicated by the study where Shmulevich and Zhang found that the model able to provide a clear distinction between different classes of sarcomas and different sub-classes of gliomas [8]. Intuitively, the Boolean network model is a suitable representation of genetic networks because genetic manipulation often involves either over-expression or deletion of a gene [9].

As a Boolean network evolves in time, a sequence of states results and converges to limit cycles or attractors eventually. Information from the initial states are no longer as important and only a small number of all the possible configurations actually occur [10], composing the limit cycle or attractor.

Definition 1: Attractor

An attractor is a set of states, invariant under the dynamically progression, towards which the neighboring states in a given basin of attraction asymptotically approach in the course of dynamic evolution. An attractor is defined as the smallest unit which cannot be itself decomposed into two or more attractors with distinct basins of attraction. [11]

Definition 2: Basin of attraction

The set of points in the state vector space of system state variables such that initial conditions chosen in this set dynamically evolve to a particular attractor. [12]

Definition 3: Limit cycle

An attracting set of state vectors to which orbits or trajectories converge and upon which trajectories are periodic. [13]

Definition 4: Length of a limit cycle

In the above sense, the length of a limit cycle represents its fundamental period and is equal to the number of states contained within the cycle.

Definition 5: Basin number

The basin number is the number of reachable states to a limit cycle or attractor.

Attractors and limit cycles of the Boolean network model can be interpreted in two ways. First, they can be seen to represent stable phenotypes of differentiated cells- muscle vs. nerve cells, or healthy vs. sick cells [14] [15]. Second, attractors and limit cycles can be regarded as cellular states- differentiation, apoptosis and cell cycle [9]. Both interpretations capture the concept of homeostasis perfectly. Homeostasis occurs when cells maintain their state despite minor disturbances in their environmental and internal stimuli. These perturbations can be interpreted as changes in state configurations of the cell, but as long as they reside within the same basin of attraction, the same attractors or limit cycles will be reached. Thus attractor or limit cycle stability increases with the size of the basin of attraction.

Regarding attractors and limit cycles as cellular states, cancer can be represented as a shift from the usually stable "differentiation" state to the "growth" state. Mutations might have reduced the size of the basin of attraction leading to the "differentiation" state, thus rendering it less stable and more susceptible to perturbations. Cancer drugs should then strive to push the cell from its "growth" state back into "differentiation" state. [9]

II. Investigation of 3-gene Boolean Network

In this study, we attempt to evaluate the evolution pattern of a NK-Boolean network whose evolution depends only on its two neighboring sites and itself. In addition, we assume N (total number of genes in the network) =3 and K (connectivity) =3, so that there are $2^N = 8$ possible states and $(2^{2^N}-2)^{N=}$ 16,387,064 possible combinations of logic function. As we believe that networks with larger N's may be broken down into networks of N = 2 or 3, we studied NK-Boolean network of the 3-gene network. (A study on the 2-gene network can be found in [16]). Of the 16,387,064 possible logic functions, about 150 examples were evaluated by hand, resulting in diagrams similar to Figure 1 and 2. A', B', C' represent genes at time 't+1' and A, B, C, represent genes at time 't'. Figure 1 illustrates two limit cycles, one with length 2 (L2) and the other with length 6 (L6). Both limit cycles have a basin number of 0. Figure 2 illustrates two attractors, one with basin number 0 (B0) and the other with basin number 6 (B6). In the syntax of this study, logic functions that involve no logic connectors are termed "PLAIN". Conversely, logic functions that involve AND, OR, COMPLEMENT connectors are termed "AND", "OR" and "NOT" respectively.

A' ← B
B' ← $\overline{C}$
C' ← A

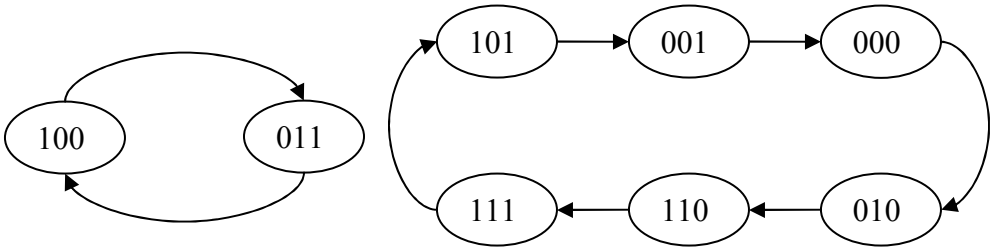

Figure 1. An example of a 3-gene network with two limit cycles

A' ← A ⩓ B
B' ← B Λ C
C' ← A Λ C

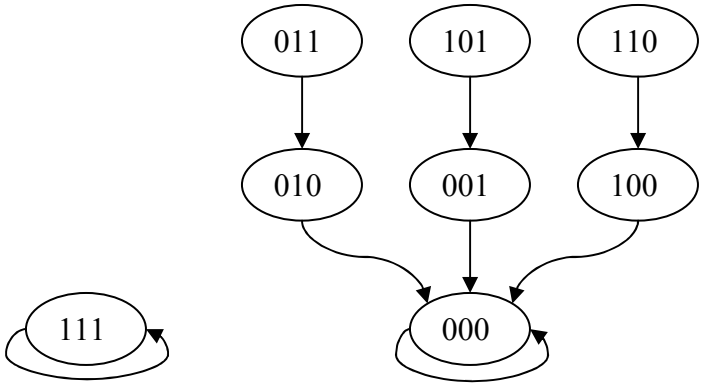

Figure 2. An example of a 3-gene network with two attractors

III. Boolean Network Evolutionary patterns

We evaluated and grouped the evolution patterns resulting from different types of logic functions. Several general rules that govern the evolution of NK-Boolean network patterns were discovered. To better summarize and compare these NK-Boolean network patterns, the logic functions and their corresponding evolution patterns are grouped into tables. Close observation of the tables leads to a number of general rules that govern the evolution of NK-Boolean network

patterns. These links between logic functions and NK-Boolean network evolution patterns will aid efforts in "reverse engineering". We list our observations as below:

Rule 1:

NOT functions produce at least 1 limit cycle. Conversely, pure AND and OR functions only produce attractors.

Rule 2:

For NOT functions,

$$A' \leftarrow f(A)$$
$$B' \leftarrow f(B)$$
$$C' \leftarrow f(C)$$

The above logic function is a special case where 4 limit cycles with length 2 always result.

Rule 3:

For NOT functions, the presence of repeated regulation will result in a NK-Boolean network pattern with length equal to its basin number.

Rule 4:

For AND and OR functions, attractors of "000" and "111" are always present.

Rule 5:

For AND functions, the dominant attractor is "000", and the least dominant attractor is "111".

Rule 6:

For OR functions, the dominant attractor is "111", and the least dominant attractor is "000".

Rule 7:

For 3 (AND 2) functions such as

$$A' \leftarrow A \wedge B$$
$$B' \leftarrow B \wedge C$$
$$C' \leftarrow A \wedge C$$

The number of branches leading to the attractor is 3 + number of repetitions. Therefore, the range of branches leading to the attractors is between 3 and 5 (since there can be 2 repetitions at most).

IV. Introduction- Fuzzy Logic Networking

The absolute binary values of Boolean network does not account for the varying degrees at which regulators affect gene expression. By placing weights on certain regulators, Fuzzy Logic Networking attempt to model the real genetic regulatory networks more realistically. Here, $\boldsymbol{X}$ represents the set of expressed genes and μ_x represents a function that maps elements from the universal set $\boldsymbol{U}$ to $\boldsymbol{X}$. $\boldsymbol{X}$ connotes the idea of "expressed" genes, and μ_x corresponds to the degree of expression of each gene, and is known as the membership function and X(u) is degree of membership of u in the fuzzy subset of X. [17]

V. Investigation of Fuzzy Logic Networking

Table 6. Five Fuzzy Logics from [50]

		x or y	x and y
Logic 1	CFMQVS	min(1,x+y)	x*y
Logic 2	max/min	max(x, y)	min(x,y)
Logic 3	probabilistic	x+y-x*y	x*y
Logic 4	MV	min(1,x+y)	max(0,x+y-1)
Logic 5	gcd/lcm	gcd(x,y)	lcm(x,y)

In the case of fuzzy subsets, we employ the same operations used in the Boolean representation i.e. AND, OR and COMPLEMENT. Due to the range of membership values, there is no one way of carrying out the AND, OR and COMPLEMENT operations on the fuzzy subsets. Table 6 presents five logics that were discussed by Reiter [18]. As observed by Reiter, each of the logic has different characteristics that are worth noting. Logic 2 is a common operation based on maximum and minimum replacing, and it used in both [17] and [19]. Based on the operation on two fuzzy subsets X and Y, we generated figures that showed the differences amongst these. The plots compare only Logic 1 to Logic 4 as Logic 5 does not map back to the interval of [0,1]. From the plots, the logic functions are approximately similar.

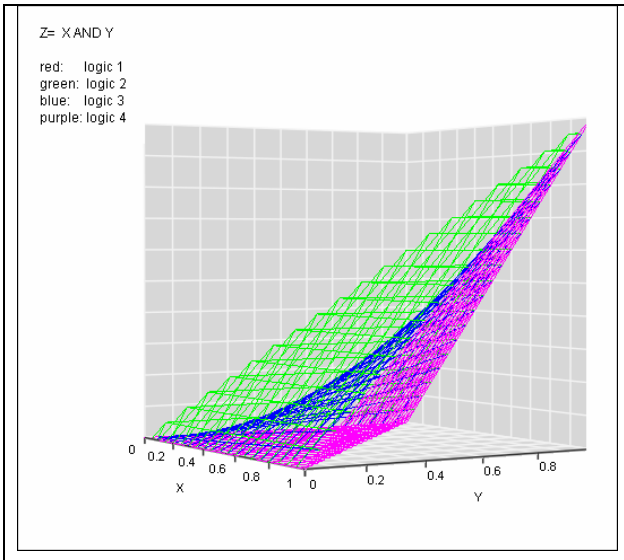

Figure 5. Comparison of "AND" values. Logic 1 is same as logic 3, so red graph is covered by blue graph.

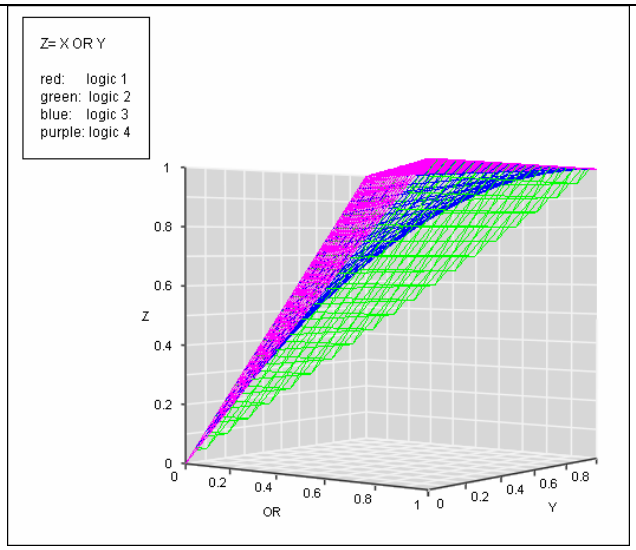

Figure 6. Comparison of "OR" values. Logic 1 is same as logic 4, so red graph is covered by purple graph.

VI. Fuzzy Logic Network Evolutionary patterns

A preliminary study of 3-gene regulation networks using fuzzy sets was carried out. About 150 examples were evaluated by hand. We observed that different (i) logic operations (Logic1, 2, 3 or 4), (ii) logic functions, and (iii) initial membership values led to different attractor and limit cycles for the 3-gene regulation network. Compared to the study of 3-gene regulation networks using NK-Boolean network, (i) and (iii) are additional parameters in the evolution pattern of the 3-gene network and add an extra dimension of complexity to the study. In order to meaningfully evaluate the attractor and limit cycle patterns, we made two assumptions. 1.) A, B, C have different initial membership values

and 2.) Logic 2 is used for the AND and OR operations. Assumption 1 is intuitively reasonable as it is unlikely that two genes will have exactly the same extent of expression.

We grouped the logic functions and their corresponding evolution patterns into tables and observed a number of general rules that govern the evolution of Fuzzy Logic network patterns. We list our observations as below:

Rule 1

NOT functions produce limit cycles.

Rule 2

A' ← A ∧ B B' ← B C' ← B ∧ C	2 (AND 2)s un-AND gene is self regulated	If un-AND gene holds highest value: B0 attractor If un-AND gene holds middle value: B1 attractor If un-AND gene holds middle value: B2 attractor

Rule 3

A' ← A ∧ B B' ← B C' ← B ∧ C	2 (OR 2)s un-OR gene is self regulated	If un-OR gene holds highest value: B2 attractor If un-OR gene holds middle value: B1 attractor If un-OR gene holds middle value: B0 attractor

Rule 4

A' ← A ∧ B B' ← B ∧ C C' ← A ∧ C	3 (AND 2)s No repeated regulation	B2 attractor	Need two steps to pick the minimum of 3 genes

Rule 5

A' ← A ∨ B B' ← B ∨ C C' ← A ∨ C	3 (OR 2)s No repeated regulation	B2 attractor	Need two steps to pick the maximum of 3 genes

VII. Conclusion

As the cell ages and undergoes irreversible changes, it either goes into an invariant state or progresses through a periodic cycle. The irreversible evolution of the NK-Boolean and Fuzzy Logic network mimic such changes very well. Future work would build on the basic connectors studied in this paper, and include computer simulations of more complicated functions involving combinations of logic connectors. ‘

The most significant result obtained from this study is a fundamental understanding of the stochastic behavior of a cell with simple assumption of a simple 3-gene network. It is affirmative that a “reverse-engineering” problem can be solved, at least theoretically via induction and reduction approach. In general, the realistic biological behavior is so complicated that it can be seem to be at the edge-of-the-chaos. Usually, a theoretical and mathematical derivation is very difficult. Nevertheless, a simplified analysis based upon a simple model would definitely provide a much needed visualization of the biological behavior and associated phenomenon. It is only through mathematical analysis would

there be a better chance in understanding the complex phenomenon. This is partially true when the simplified model can be viewed as the basic building blocks of a complicated situation.

VII. References

[1] Kauffman, S.A. (1993). *The Origins of Order, Self-Organization, and Selection in Evolution.* Oxford University Press

[2] Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly connected nets. *J. Theoret. Biol,* **22** , 437-467

[3] Wang, P.P. and Robinson, J. (2003). What is SORE? *The 7th Proceedings of the Joint Conferences on Information Sciences*

[4] Wang, P.P., Cao, Y., Robinson, J., Tokuta, Alade. (2003). A study of two-gene network- The simplest special case of SORE. *The 7 the Proceedings of the Joint Conferences on Information Science*

[5] Wang, P.P. and Tao, H., (2003). A novel method of error correcting code generation based on SORE. *The 7 the Proceedings of the Joint Conferences on Information Sciences*

[6] Wang, P.P. and Yu, J., (2003) SORE, Self Organizable & Regulating Engine- A powerful classifier. *The 7 the Proceedings of the Joint Conferences on Information Sciences*

[7] Thieffry, D., Huerta, A.M., Perez-Rueda, E. and Collado Vides J. (1998). From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in Escherichia coli. *Bioessays* , **20** , 433-440

[8] Shmulevich I. and Zhang, W. (2002). Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics* , **18** , 4, 555-565

[9] Huang, S. (1999). Gene expression profiling, genetic networks, and cellular state: an integrating concept for tumorigenesis and drug discovery . *J. Mol. Med* ., **77** , 469-480

[10] Wolfram, S. (1983). Cellular Automata, *Los Alamos Science* , **9** , 2-21

[11] Weisstein, W.E. “Limit Cycle.” *MathWorld—A Wolfram Web Resource* . http://mathworld.wolfram.com/Attractor.html

[12] Weisstein, W.E. “Limit Cycle.” *MathWorld—A Wolfram Web Resource* . http://mathworld.wolfram.com/BasinofAttraction.html

[13] Weisstein, W.E. “Limit Cycle.” *MathWorld—A Wolfram Web Resource* . http://mathworld.wolfram.com/Limitcycle.html

[14] Kauffman , S.A. (1993). *The Origins of Order, Self-Organization, and Selection in Evolution.* Oxford University Press

[15] Somogyi, R., and Sniegoski C. (1996). Modeling the complexity of genetic networks. *Complexity* , **1** , 6, 45-63

[16] Wang, P.P., Cao, Y., Robinson, J., Tokuta, Alade. (2003). A study of two-genes network- The simplest special case of SORE. *The 7 th Proceedings of the Joint Conferences on Information Science*

[17] Nguyen, H.T. and Walker, E.A (2000) A First Course in Fuzzy Logic. Chapman & Hall/CRC, 2-12

[18] Reiter, C.A. (2002) Fuzzy Automata and Life. Complexity, 7, 3, 19-29

[19] Wang, L. (1997) A Course In Fuzzy Systems And Control. Prentice Hall PTR, 29-32