# Stability Properties of Biologically Relevant Boolean Functions

Lori Layne

Advisor: Dr. Elena Dimitrova
Clemson University

September 25, 2010

# Discrete Modeling Frameworks

- (Random/Probabilistic) Boolean networks (S. Kauffman)
- Polynomial dynamical systems (T. Thomas, D. Thieffry, R. Laubenbacher, B. Stigler, B. Pareigis, A. Jarrah)
- Cellular automata (von Neumann, J. Conway, S. Wolfram)
- Graphical models (Barret, Mortveit, Reidys)
- Logical models (R. Thomas)
- Petri nets (Google 'Petri Nets World')

# Polynomial Dynamical Systems

We can represent biological networks with polynomial equations: $(f_1, \ldots, f_k) : \mathbb{F}^k \to \mathbb{F}^k$, where

- $\mathbb{F}$ is a finite set of states
- $k$ is the number of components in the network
- $f_i$ is a polynomial; gives the dynamics of component $i$

We use the *state space graph* to analyze the dynamics of the system.

# Polynomial Dynamical Systems

We can represent biological networks with polynomial equations: $(f_1, \ldots, f_k) : \mathbb{F}^k \to \mathbb{F}^k$, where

- $\mathbb{F}$ is a finite set of states
- $k$ is the number of components in the network
- $f_i$ is a polynomial; gives the dynamics of component $i$

We use the *state space graph* to analyze the dynamics of the system.

# Polynomial Dynamical Systems

We can represent biological networks with polynomial equations: $(f_1, \ldots, f_k) : \mathbb{F}^k \to \mathbb{F}^k$, where

- $\mathbb{F}$ is a finite set of states
- $k$ is the number of components in the network
- $f_i$ is a polynomial; gives the dynamics of component $i$

We use the *state space graph* to analyze the dynamics of the system.

# Example: State Space Graph of a PDS

$$f_1 = 1 + x_1 + x_2$$
$$f_2 = x_1 x_2$$

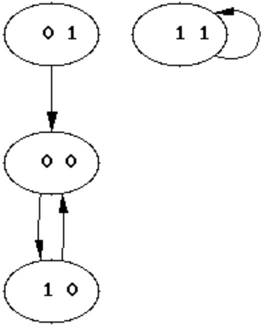



Figure: State Space Graph

# Boolean Networks

**Definition**

$F = (f_1, \ldots, f_k)$ is a *Boolean network* on $k$ variables if $f_i$ is a Boolean function in $k$ variables for $i = 1, \ldots, k$.

Note that every Boolean function can be represented as a polynomial over $\mathbb{F}_2$, and vice versa.

Boolean networks are polynomial dynamical systems over $\mathbb{F}_2$.

# Question

Do all Boolean functions reflect the behavior of biological networks??

# Canalyzing Functions

**Definition (Canalyzing function)**

A Boolean function $f(x_1, \ldots, x_k)$ is *canalyzing* if there exists an index $i$ and $a, b \in \{0, 1\}$ such that

$$f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_k) = b.$$

That is, when $x_i$ is given the input value $a$, $f$ evaluates to $b$ regardless of the input values of the other variables. Here $a$ is called the *canalyzing value* and $b$ is called the *canalyzed value*.

# Examples: Canalyzing Functions

- The AND function $x \wedge y$ is canalyzing in $x$. Since $0 \wedge y = 0$ for any input value of $y$, 0 is a canalyzing value for $x$ with canalyzed output value 0
- $XOR(x, y) := (x \vee y) \wedge \overline{(x \wedge y)}$ is not canalyzing in either variable

# Examples: Canalyzing Functions

- The AND function $x \wedge y$ is canalyzing in $x$. Since $0 \wedge y = 0$ for any input value of $y$, 0 is a canalyzing value for $x$ with canalyzed output value 0
- $XOR(x, y) := (x \vee y) \wedge \overline{(x \wedge y)}$ is not canalyzing in either variable

# Canalyzing Functions: Role and Applications

- Introduced by Kauffman [Kau93] as appropriate rules in Boolean network models of gene regulatory networks
- Reminiscent of the concept of “canalisation” introduced by the geneticist Waddington [Wad42] to represent the ability of a genotype to produce the same phenotype regardless of environmental variability
- Play an important role in the study of random Boolean networks [Kau93, Lyn95, Sta87, Ste99]

# Canalyzing Functions: Role and Applications

- Introduced by Kauffman [Kau93] as appropriate rules in Boolean network models of gene regulatory networks
- Reminiscent of the concept of “canalisation” introduced by the geneticist Waddington [Wad42] to represent the ability of a genotype to produce the same phenotype regardless of environmental variability
- Play an important role in the study of random Boolean networks [Kau93, Lyn95, Sta87, Ste99]

# Canalyzing Functions: Role and Applications

- Introduced by Kauffman [Kau93] as appropriate rules in Boolean network models of gene regulatory networks
- Reminiscent of the concept of "canalisation" introduced by the geneticist Waddington [Wad42] to represent the ability of a genotype to produce the same phenotype regardless of environmental variability
- Play an important role in the study of random Boolean networks [Kau93, Lyn95, Sta87, Ste99]

But what if the canalyzing variable does not receive the canalyzing input value??

# Nested Canalyzing Functions

## Definition (Nested canalyzing function)

Let $f(x_1, \ldots, x_k)$ be a Boolean function. For $\sigma \in S_k$, $f$ is a *nested canalyzing function* (NCF) in the variable order $x_{\sigma(1)}, \ldots, x_{\sigma(k)}$ with canalyzing values $a_1, \ldots, a_k$ and canalyzed values $b_1, \ldots, b_k$ if it can be expressed in the form

$$f = \begin{cases} b_1 & \text{if } x_{\sigma(1)} = a_1 \\ b_2 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} = a_2 \\ b_3 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} \neq a_2,\ x_{\sigma(3)} = a_3 \\ \vdots & \vdots \\ b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k-1)} \neq a_{k-1},\ x_{\sigma(k)} = a_k \\ \neg b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k)} \neq a_k \end{cases}$$

# Example: NCFs

$f(x_1, x_2, x_3) = x_2 \vee (\neg x_1 \wedge x_3)$ is a NCF with $b = [1, 0, 0]$, $a = [1, 1, 0]$, and $\sigma = [2, 1, 3]$:

$$f(x_1, x_2, x_3) = \begin{cases} 1 & \text{if } x_2 = 1 \\ 0 & \text{if } x_2 = 0,\ x_1 = 1 \\ 0 & \text{if } x_2 = 0,\ x_1 = 0,\ x_3 = 0 \\ 1 & \text{if } x_2 = 0,\ x_1 = 0,\ x_3 = 1. \end{cases}$$

# Biologically Relevant Properties of NCFs

- Introduced by Kauffman *et al.* in [KPST03]
- Networks made from NCFs have stable dynamic behavior and might be a good class of functions to express regulatory relationships in biochemical networks [KPST04]
- Nested (hierarchically) canalyzing functions show ordered behavior [NFW06]
- Systems of NCFs have a smaller average cycle length and average height (number of time steps it takes to converge to an attractor) of the state space graph compare to general Boolean networks

# Biologically Relevant Properties of NCFs

- Introduced by Kauffman *et al.* in [KPST03]
- Networks made from NCFs have stable dynamic behavior and might be a good class of functions to express regulatory relationships in biochemical networks [KPST04]
- Nested (hierarchically) canalyzing functions show ordered behavior [NFW06]
- Systems of NCFs have a smaller average cycle length and average height (number of time steps it takes to converge to an attractor) of the state space graph compare to general Boolean networks

# Biologically Relevant Properties of NCFs

- Introduced by Kauffman *et al.* in [KPST03]
- Networks made from NCFs have stable dynamic behavior and might be a good class of functions to express regulatory relationships in biochemical networks [KPST04]
- Nested (hierarchically) canalyzing functions show ordered behavior [NFW06]
- Systems of NCFs have a smaller average cycle length and average height (number of time steps it takes to converge to an attractor) of the state space graph compare to general Boolean networks

# Biologically Relevant Properties of NCFs

- Introduced by Kauffman *et al.* in [KPST03]
- Networks made from NCFs have stable dynamic behavior and might be a good class of functions to express regulatory relationships in biochemical networks [KPST04]
- Nested (hierarchically) canalyzing functions show ordered behavior [NFW06]
- Systems of NCFs have a smaller average cycle length and average height (number of time steps it takes to converge to an attractor) of the state space graph compare to general Boolean networks

- Recall the definition of NCFs:

$$f = \begin{cases} b_1 & \text{if } x_{\sigma(1)} = a_1 \\ b_2 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} = a_2 \\ b_3 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} \neq a_2,\ x_{\sigma(3)} = a_3 \\ \vdots & \vdots \\ b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k-1)} \neq a_{k-1},\ x_{\sigma(k)} = a_k \\ \neg b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k)} \neq a_k \end{cases}$$

- Quite restrictive!

- Recall the definition of NCFs:

$$f = \begin{cases} b_1 & \text{if } x_{\sigma(1)} = a_1 \\ b_2 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} = a_2 \\ b_3 & \text{if } x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} \neq a_2,\ x_{\sigma(3)} = a_3 \\ \vdots & \vdots \\ b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k-1)} \neq a_{k-1},\ x_{\sigma(k)} = a_k \\ \neg b_k & \text{if } x_{\sigma(1)} \neq a_1,\ \cdots,\ x_{\sigma(k)} \neq a_k \end{cases}$$

- Quite restrictive!

# Nested Canalyzing Depth

## Definition

We say that $f$ is a *partially nested canalyzing function* (or PNCF) if for some $d \in \mathbb{N}$ and Boolean function $g$,

$$f = \begin{cases} b_1 & x_{\sigma(1)} = a_1 \\ b_2 & x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} = a_2 \\ b_3 & x_{\sigma(1)} \neq a_1,\ x_{\sigma(2)} \neq a_2,\ x_{\sigma(3)} = a_3 \\ \vdots & \vdots \\ b_d & x_{\sigma(1)} \neq a_1, \ldots,\ x_{\sigma(d-1)} \neq a_{d-1},\ x_{\sigma(d)} = a_d \\ g(x_{\sigma(d+1)}, \ldots, x_{\sigma(k)}) & x_{\sigma(1)} \neq a_1, \ldots,\ x_{\sigma(d)} \neq a_d \end{cases}$$

where either $g$ is constant, or $g$ is non-constant and none of the variables $x_{\sigma(d+1)}, \ldots, x_{\sigma(k)}$ are canalyzing in $g$. The integer $d$ is called the *active depth* of $f$. The (full) *depth* of $f$ is $d$ if $g$ is non-constant, and $k$ otherwise.

# Example: Nested Canalyzing Depth

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| $x_2$ | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $x_3$ | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $x_4$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $f$ | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

- $x_2$ is canalyzing with canalyzing input 0 and output 0
- $\sigma(1) = 2$ and $b_1 = a_1 = 0$; $\sigma(2) = 3$, $b_2 = 1$, and $a_2 = 0$.
- The remaining function in $x_1$ and $x_4$ is neither canalyzing nor constant, so $f$ has depth 2

# Sensitivity

- The *sensitivity* of a function quantifies the sensitivity of the output to variations in the function inputs
- It is given by $s^f(\mathbf{x}) = \sum_{i=1}^{k} \chi\left[f(\mathbf{x} \oplus e_i) \neq f(\mathbf{x})\right]$, where $e_i$ denotes the $i$th unit vector and $\chi$ is an indicator function
- Essentially computing the number of inputs with Hamming distance one from an input $\mathbf{x}$ that gives a different function output than $\mathbf{x}$
- The *average sensitivity* of a function, $s^f$, is the expected value of $s^f(\mathbf{x})$, $s^f = E\left[s^f(\mathbf{x})\right]$

# Sensitivity

- The *sensitivity* of a function quantifies the sensitivity of the output to variations in the function inputs
- It is given by $s^f(\mathbf{x}) = \sum_{i=1}^{k} \chi\left[f(\mathbf{x} \oplus e_i) \neq f(\mathbf{x})\right]$, where $e_i$ denotes the $i$th unit vector and $\chi$ is an indicator function
- Essentially computing the number of inputs with Hamming distance one from an input $\mathbf{x}$ that gives a different function output than $\mathbf{x}$
- The *average sensitivity* of a function, $s^f$, is the expected value of $s^f(\mathbf{x})$, $s^f = E\left[s^f(\mathbf{x})\right]$

# Sensitivity

- The *sensitivity* of a function quantifies the sensitivity of the output to variations in the function inputs
- It is given by $s^f(\mathbf{x}) = \sum_{i=1}^{k} \chi \left[ f(\mathbf{x} \oplus e_i) \neq f(\mathbf{x}) \right]$, where $e_i$ denotes the $i$th unit vector and $\chi$ is an indicator function
- Essentially computing the number of inputs with Hamming distance one from an input $\mathbf{x}$ that gives a different function output than $\mathbf{x}$
- The *average sensitivity* of a function, $s^f$, is the expected value of $s^f(\mathbf{x})$, $s^f = E\left[s^f(\mathbf{x})\right]$

# Sensitivity

- The *sensitivity* of a function quantifies the sensitivity of the output to variations in the function inputs
- It is given by $s^f(\mathbf{x}) = \sum_{i=1}^{k} \chi\left[f(\mathbf{x} \oplus e_i) \neq f(\mathbf{x})\right]$, where $e_i$ denotes the $i$th unit vector and $\chi$ is an indicator function
- Essentially computing the number of inputs with Hamming distance one from an input $\mathbf{x}$ that gives a different function output than $\mathbf{x}$
- The *average sensitivity* of a function, $s^f$, is the expected value of $s^f(\mathbf{x})$, $s^f = E\left[s^f(\mathbf{x})\right]$

# Sensitivity

- Shmulevich and Kauffman suggest that networks created using functions that are less sensitive will be more "dynamically ordered" (stable) than those with higher sensitivity
- A random unbiased Boolean function in $k$ variables has average sensitivity $\frac{k}{2}$ [SK04]
- A random unbiased Boolean function with depth at least one has average sensitivity $\frac{k+1}{4}$

# Sensitivity

- Shmulevich and Kauffman suggest that networks created using functions that are less sensitive will be more "dynamically ordered" (stable) than those with higher sensitivity
- A random unbiased Boolean function in $k$ variables has average sensitivity $\frac{k}{2}$ [SK04]
- A random unbiased Boolean function with depth at least one has average sensitivity $\frac{k+1}{4}$

# Sensitivity

- Shmulevich and Kauffman suggest that networks created using functions that are less sensitive will be more "dynamically ordered" (stable) than those with higher sensitivity
- A random unbiased Boolean function in $k$ variables has average sensitivity $\frac{k}{2}$ [SK04]
- A random unbiased Boolean function with depth at least one has average sensitivity $\frac{k+1}{4}$

# Depth and Sensitivity

- When reverse engineering gene networks, a nested canalyzing structure may apply to some of the variables, while the behavior of the remaining variables may be ambiguous or unknown
- We say that a function has depth at least $d$ if $d$ of the variables are known to be nested canalyzing, while the remaining $k - d$ variables can take on any Boolean function, canalyzing or otherwise
- We'll show that the larger the value of $d$ is, the less sensitive the function will be on average

# Depth and Sensitivity

- When reverse engineering gene networks, a nested canalyzing structure may apply to some of the variables, while the behavior of the remaining variables may be ambiguous or unknown
- We say that a function has depth at least $d$ if $d$ of the variables are known to be nested canalyzing, while the remaining $k - d$ variables can take on any Boolean function, canalyzing or otherwise
- We'll show that the larger the value of $d$ is, the less sensitive the function will be on average

# Depth and Sensitivity

- When reverse engineering gene networks, a nested canalyzing structure may apply to some of the variables, while the behavior of the remaining variables may be ambiguous or unknown
- We say that a function has depth at least $d$ if $d$ of the variables are known to be nested canalyzing, while the remaining $k - d$ variables can take on any Boolean function, canalyzing or otherwise
- We'll show that the larger the value of $d$ is, the less sensitive the function will be on average

# Depth and Sensitivity

Extending Kauffman and Shmulevich's result to functions of depth at least $d$, we have the following theorem:

**Theorem**

*Let $f_d$ be a Boolean function in $k$ variables with nested canalyzing depth at least $d$. Then the average sensitivity of $f_d$ is*

$$E[s^{f_d}] = \frac{k-d}{2^{d+1}} + \sum_{i=1}^{d} \frac{1}{2^d} = \frac{k-d}{2^{d+1}} + 1 - \frac{1}{2^d}.$$

# Depth and Sensitivity

Now, we see that

$$\begin{aligned} E[s^{f_d}] - E[s^{f_{d+1}}] &= 1 - \frac{1}{2^d} + \frac{k-d}{2^{d+1}} - \left[1 - \frac{1}{2^{d+1}} + \frac{k-d-1}{2^{d+2}}\right] \\ &= \frac{k-d-1}{2^{d+2}} \\ &\geq 0 \text{ when } k-d \geq 1. \end{aligned}$$

Hence, the greater the depth of a function, the smaller its average sensitivity. Also, note *rapidly* diminishing returns in sensitivity as $d$ increases!

# Overview: Derrida Curves

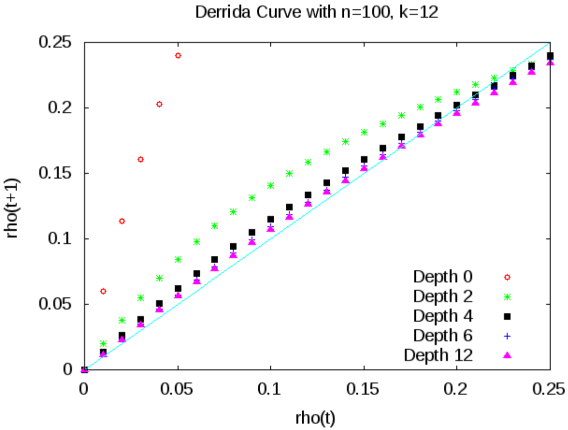
- $\rho(t)$ is the normalized Hamming distance between two randomly chosen points at time $t$
- A Derrida curve is a plot of $\rho(t+1)$ vs. $\rho(t)$, averaged over many pairs of points and many Boolean networks
- The behavior of the curve for small values of $\rho(t)$ indicates whether the network is sensitive to perturbations

# Assessing Network Dynamics

To assess network dynamics, we compare the Derrida curve at small values of $\rho(t)$ to the line $y = x$:

- Curves above the line indicate networks in the chaotic phase
- Curves below the line indicate networks in the frozen phase (reflect stability of biological systems)
- Curves near the line are in the critical phase (reflect adaptability of biological systems)

# Results


Derrida Curve with n=100, k=12
rho(t+1)
rho(t)
0
0.05
0.1
0.15
0.2
0.25
Depth 0
Depth 2
Depth 4
Depth 6
Depth 12

# Conclusions

- For fixed $k$, function sensitivity decreases as depth increases
- Boolean networks constructed using functions of increasing depth rapidly approach the critical regime and therefore have great potential as Boolean network models
- Diminishing returns dictate that the difference in stability between networks of NCF's and those of PNCF's of sufficient depth is almost negligible
- PNCF's have a much less restrictive structure than full NCF's, yet maintain their stability properties!

# Conclusions

- For fixed $k$, function sensitivity decreases as depth increases
- Boolean networks constructed using functions of increasing depth rapidly approach the critical regime and therefore have great potential as Boolean network models
- Diminishing returns dictate that the difference in stability between networks of NCF's and those of PNCF's of sufficient depth is almost negligible
- PNCF's have a much less restrictive structure than full NCF's, yet maintain their stability properties!

# Conclusions

- For fixed $k$, function sensitivity decreases as depth increases
- Boolean networks constructed using functions of increasing depth rapidly approach the critical regime and therefore have great potential as Boolean network models
- Diminishing returns dictate that the difference in stability between networks of NCF's and those of PNCF's of sufficient depth is almost negligible
- PNCF's have a much less restrictive structure than full NCF's, yet maintain their stability properties!

# Conclusions

- For fixed $k$, function sensitivity decreases as depth increases
- Boolean networks constructed using functions of increasing depth rapidly approach the critical regime and therefore have great potential as Boolean network models
- Diminishing returns dictate that the difference in stability between networks of NCF's and those of PNCF's of sufficient depth is almost negligible
- PNCF's have a much less restrictive structure than full NCF's, yet maintain their stability properties!

[Kau93] Stuart Kauffman.
*The Origins of Order: Self-Organization and Selection in Evolution*.
Oxford University Press, 1993.

[KPST03] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein.
Random boolean network models and the yeast transcriptional network.
*PNAS*, 100(25):14796–14799, 2003.

[KPST04] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein.
Genetic networks with canalyzing boolean rules are always stable.
*Proceedings of the National Academy of Sciences of the United States of America*, 101(49):17102–17107, December 2004.

[Lyn95] J. F. Lynch.
On the threshold of chaos in random boolean cellular automata.
*Random Struct. Algorithms*, 6(2–3):239–260, 1995.

[NFW06] S. Nikolajewa, M. Friedel, and T. Wilhelm.
Boolean networks with biologically relevant rules show ordered behavior.
*Biosystems*, 2006.

[SK04] I. Shmulevich and S.A. Kauffman.
Activities and sensitivities in boolean network models.
*Phys Rev Lett*, 93(4):048701, 2004.

[Sta87] D. Stauffer.
On forcing functions in kauffmans random boolean networks.
*Journal of Statistical Physics*, 46(3–4):789794, 1987.

[Ste99] M. D. Stern.
Emergence of homeostasis and noise imprinting in an evolution model.
*PNAS*, 96(19):10746–10751, 1999.

[Wad42] C. H. Waddington.
Canalisation of development and the inheritance of acquired characters.
*Nature*, 150:563–564, 1942.