A Combinatorial Problem in Gene Regulation

Winfried Just

Department Mathematics,
Ohio University

January 18, 2005

Genes and Genomes

- The *genome* of an organism contains the blueprint for its construction (development) and its workings. It is coded in DNA molecules and does not change from cell to cell or over the organism's lifetime.

- A *gene* is a stretch of DNA that codes the chemical composition of a single protein.

- A gene is *expressed* if its product (the protein coded by it) is manufactured. Most genes are expressed only some of the time in some cell types.

- The chemical state of a cell is characterized by which genes are expressed at which levels.

Regulation of Gene Expression

- Under the right conditions, a gene is being *transcribed* into *messenger RNA* (mRNA).

- The mRNA is then *translated* into protein.

- While gene (expression) regulation may occur at the stages of transcription, translation, or even *posttranslational modifications,* the first stage appears to be most decisive, and most mathematical models of gene regulation deal exclusively with *transcriptional regulation.*

- The rate at which a gene is transcribed depends on the concentrations of several *regulatory proteins* in the cell. The set of regulatory proteins and their action is specific to a given gene.

Continuous Models of Gene Regulation

Given that a set of n regulatory proteins determines the translation rate of a given gene, it is natural to model this rate as a differential equation that depends on n variables. However, the resulting models of systems of differential equations become quickly too complicated. When only a single input is varied, the regulatory function usually becomes a sigmoid function.

Boolean Models of Gene Regulation

- The continuous models can be simplified by assuming that transcription of a gene is either switched "on" or "off," and that the relevant regulatory proteins are either "present" or "absent" in the cell. Thus the state of transcription can be conceptualized as a Boolean function of the concentrations of the relevant regulatory proteins.

- The situation is usually further simplified by assuming that the system undergoes updating in discrete time steps, with the presence/absence of a given protein in the next time step entirely determined by whether transcription of the corresponding gene is "on" or "off" in the current time step.

Properties of Boolean (Gene) Nets

Biologists are interested in the *dynamics* of Boolean nets. Of particular importance are the following questions:

- How many *steady states* and *limit cycles* are there? These are thought to correspond to different cell types.

- What is the length of the limit cycles?

- How large are the basins of attraction of the steady states and limit cycles?

- How many steps does it take to reach a steady state or (short) limit cycle?

- How much disturbance does it take to move from one steady state or limit cycle to reach the basin of attraction of another one? In other words, how *robust* is the dynamics of the network?

How to Study Boolean Nets Without Complete Data?

The complete gene net of an organism contains anywhere from 500 (*M. genitalis*) to more than 25,000 (*H. sapiens*) genes. Although reliable data on gene regulation are becoming quickly available, we are still far from knowing the complete set of Boolean regulatory functions for any organism. How can we meaningfully study the dynamics of Boolean gene nets while we are waiting for the data?

Kauffman's NK-Model

The following approach, borrowed from statistical mechanics, was pioneered by Stuart Kauffman: Consider the ensemble of all Boolean networks of N nodes with (on average) K inputs. Treat this as a probability space, and investigate the steady states, limit cycles, and basins of attraction of a "typical" network in the ensemble.

This can be done either by using mathematical and statistical methods, or by computer simulations of networks with random regulatory functions.

Kauffman convincingly argues that evolution cannot entirely "escape" these typical properties, and thus the actual gene networks of organisms will have properties close to those of "typical" Boolean nets, *unless* there are important biological reasons for systematic deviations.

Some Caveats

- Kauffman implicitly assumed that all genes can regulate other genes. This is unrealistic. Only a small fraction of genes, called *transcription factors* regulate other genes.

- The number of genes with which any given gene interacts has a power-law distribution.

- Not all Boolean functions are equally likely to appear as regulatory functions for genes. In particular, empirical evidence shows that actual gene regulatory functions are predominantly *canalizing Boolean functions.*

How to Deal with these Caveats?

By refining the model, of course. For example, the predominance of canalizing function can be modeled by biasing the choice of regulatory function towards canalizing functions in a randomly generated network. For that, it is necessary to have:

- A formula for the number of canalizing Boolean functions of n variables.

- An efficient and unbiased procedure for randomly generating canalizing Boolean functions of n variables.

Neither of the above was known until recently.

Canalizing Boolean Functions

A *Boolean function* is a function $f : \{0,1\}^n \rightarrow \{0,1\}$. A *canalizing function* is a Boolean function in which at least one of the input variables is able to determine the function output regardless of the values of the other variables.

Example 1: $f(x_1, x_2, x_3) = x_1 + x_2 x_3$ is canalizing.

Setting x_1 to 1 guarantees that the function value is 1 regardless of the values of x_2 or x_3.

Example 2: $f(x_1, x_2) = x_1 \oplus x_2$, where $\oplus$ is addition modulo 2, is not canalizing.

The values of both variables always need to be known in order to determine the function output.

The Number of Canalizing Functions

Theorem 1: (Just, Shmulevich, Konvalina (2004), *Physica D* **197**, 211-221)

Let $|C|$ denote the number of canalizing Boolean functions of n variables. Then:

$$|C| = 2((-1)^n - n) + \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k+1} 2^{2^{n-k}}.$$

For large n this value asymptotically approaches the upper bound $4n \cdot 2^{2^{n-1}}$ given by Aldana *et al.* (2002).

The Number of Canalizing Functions

n	$\lvert C\rvert$
1	4
2	14
3	120
4	3514
5	1292 276
6	103 071 426 294
7	516 508 833 342 349 371 376
8	10 889 035 741 470 030 826 695 916 769 153 787 9
9	$4.168\,515\,213 \times 10^{78}$
10	$5.363\,123\,172 \times 10^{155}$

The Proportion of Canalizing Functions

The total number of Boolean functions of n variables is 2^{2^n}. By our result, about $4n \cdot 2^{2^{n-1}}$ of these functions are canalizing. Thus the probability that a Boolean function is canalizing is approximately

$$\frac{4n \cdot 2^{2^{n-1}}}{2^{2^n}} = \frac{4n}{2^{2^{n-1}}}$$

This calculation assumes a uniform distribution of Boolean function; corresponding results for a given bias p of functions taking the value 1 are proved in the paper.

About the Proof

A Boolean function f is canalizing if there exist a *canalizing variable* x_i (where $i \in \{1, \ldots, n\}$), a *canalizing value* $s \in \{0, 1\}$ and a *canalized value* $v \in \{0, 1\}$ such that:

$$\forall x \in \{0, 1\}^n \, (x_i = s \Rightarrow f(x_i) = v).$$

If $v = 1$, then we will say that f is *positively canalizing;* if $v = 0$, then we will say that f is *negatively canalizing.*

Where is the Difficulty?

Clearly, for any given choice of $i \in \{1, \ldots, n\}$, $s, v \in \{0, 1\}$ there are exactly 2^{n-1} canalizing Boolean functions with canalizing variable x_i, canalizing value s, and canalized value v. So why isn't the number of canalizing Boolean functions simply $4n \cdot 2^{2^{n-1}}$?

For starters, note that the two constant functions are both negatively and positively canalizing. More importantly, a Boolean function may have more than one canalizing variable.

Overcoming the Difficulty

The inclusion-exclusion principle allows us to overcome this difficulty. Details of the proof are somewhat delicate though. As an added bonus, the proof gives us also formulas for the number $c(k)$ of Boolean functions that are canalized by exactly k among the n variables. In particular, for $1 < k < n$ we get:

$$c(k) = \sum_{r=k}^{n} \binom{r}{k} (-1)^{r-k} \binom{n}{r} 2^{r+1} (2^{2^{n-r}} - 1).$$

An Algorithm for Randomly Generating Canalizing Functions

The problem is to find an algorithm that generates each of the canalizing functions with equal probability and is expected to terminate reasonably fast. Randomly picking Boolean functions until a canalizing one is found is not an option. The idea is to first pick $k \leq n$ and a subset of the variables of size k so that the function will be canalizing on exactly these variables; then to pick one of the four possibilities of canalizing value and canalized value for this subset, and finally to generate the rest of the function randomly while making sure that there are no additional canalizing variables. All this is straightforward, except determining the correct probability distribution for k. This crucial step becomes feasible thanks to our formula for $c(k)$.