Computational Systems Biology: Discrete Models of Gene Regulation Networks

Ana Martins[*], Paola Vera-Licona[*], and Reinhard Laubenbacher[1],
Virginia Bioinformatics Institute, Virginia Polytechnic Institute and State University

Name of Institution: Virginia Tech	
Size	About 28,000
Institution Type	Large research-intensive state university
Student Demographic	Regional, national and international students
Department Structure	Interdisciplinary research institute (Virginia Bioinformatics Institute) without a formal teaching mission

ABSTRACT

This article describes a 2-3 day workshop offered at regional undergraduate teaching institutions and high schools. Its goal is to use discrete dynamic models, in particular Boolean networks, to illustrate mathematical modeling of biological networks, such as gene regulatory networks, to a broad audience that can include undergraduate faculty, undergraduate students, high school teachers, and even high school students. The workshop covers the basics of biology, mathematical modeling, and model analysis, using the well-known *lac* operon network in *E coli* as a model system. The workshop materials can be used independently or as one or several modules in a college or high school class. Supplementary materials are available at admg.vbi.vt.edu/home/Outreach/Workshops/2.

COURSE STRUCTURE

- 2-3 days
- Average audience size: 5-15 participants
- Enrollment requirements: High school algebra and biology.
- Team-taught by one mathematics instructor and one biology instructor, with the mathematics instructor doing the lecture portion.
- Web site: http://admg.vbi.vt.edu/home/Outreach/Workshops/2

*these authors contributed equally to this work
[1]reinhard@vbi.vt.edu

INTRODUCTION

Mathematical biology uses theoretical and computational tools from mathematics to describe or analyze biological systems (Murray 1993). Biological problems are considered mathematically (such as effective drug targeting (Caplan and Rosca 2005) or inferring cancer-inducing genes (Ribba et al. 2006)). Mathematical models provide a language in which to encode the key features of a biological system, which can then be analyzed with mathematical tools to obtain insight into its structure and properties. Mathematical models can be designed for regulatory networks of genes and proteins, in which the expression of key units regulates the expression of other components in the network (deJong 2002). The modeling tools come from a broad range of mathematical fields. Most models of biological systems have been formulated as systems of differential equations, but other areas of mathematics have been used successfully to model and analyze biological systems, including algebra (Jarrah et al. 2007), control and optimization theory (Laubenbacher and Stigler 2004), graph theory (Barabasi and Oltvai 2004), logic (Albert and Othmer 2003), and statistics (Friedman *et al.* 2000).

The material presented in this paper is based on a workshop that was designed by us, researchers at Virginia Bioinformatics Institute at Virginia Tech (VBI), and conducted in collaboration with the Institute for Advanced Learning and Research (IALR) in Danville, Virginia, for high school teachers from the area. The aim was to provide background and materials for the teachers to introduce into their mathematics classes, in accordance with the Standards of Learning (SOL) curriculum (Virginia Department of Education 2012), and the NCTM standards (National Council of Teachers of Mathematics 2012). We introduced key concepts in biochemistry, biology, and discrete mathematics, which were applied using graphical modeling software to explore the regulation of the lactose (*lac*) operon, an example of gene transcription in prokaryotes (Jacob and Monod 1961). The participants completed the project and developed activities to show students the value of mathematical modeling in understanding biochemical network mechanisms and dynamics.

The bar to understanding and appreciating mathematical models of biological systems is high since students need to understand the mathematics and biology used. If differential

equations models are used, then students need to be familiar with some of the subtleties of the subject to appreciate topics like steady state analysis and bifurcation behavior. Therefore, we decided to use the simpler modeling tool of Boolean networks, which can be appreciated without sophisticated mathematical training. Boolean network models have been used in molecular biology since the 1960s (see Kauffman 1969) and have provided insights into the qualitative dynamical behavior of some important molecular networks, such as the cell cycle and the gene regulation mechanisms during embryonic development of organisms (Albert and Othmer 2003). The discrete analog of a continuous state space analysis is a graph-theoretic analysis of the state space graph (defined below). The material in this chapter can be used as examples in a variety of discrete mathematics courses.

We provide a basic introduction to genomics and a description of a much-studied model system, the *lac* operon in prokaryotic organisms, which regulates lactose metabolism. We also introduce Boolean networks and the tools for their analysis. We describe an example of a multi-component research project on Boolean network models of the *lac* operon and the biological insights that come from it. The project might be viewed as a case study of the utility of mathematical models in the discovery of new biology. While current molecular networks under study are substantially bigger and more complex than the *lac* operon, this simple example provides a template and interested readers can explore the recent literature.

There are, of course, other modeling frameworks that are being used successfully in systems biology, including ordinary differential equation models (see Veliz-Cuba et al. 2009), a classroom module that includes a guide to model analysis using the open source software package Copasi (Hoops et al. 2006), agent-based models, Petri net models, and Bayesian network models.

GENE REGULATION AND THE LAC OPERON

The *Escherichia coli lac* operon is one of the earliest and best understood examples of regulation of gene expression (Jacob and Monod 1961; Koolman and Röhm 1996; Lodish et al. 2000). Gene regulation in bacteria allows the cell to adjust to changes in the nutritional environment so that growth and division can be optimized. *E. coli* can use

glucose or lactose as energy and carbon sources, and when cells grow in a glucose-based medium, the activity of the enzymes involved in the metabolism of lactose is very low, even if lactose is available. When glucose is exhausted from the medium and lactose is present, there is an increase in the activity of enzymes involved in lactose metabolism (Lodish et al. 2000). Before describing in detail the molecular mechanisms, we need to introduce some of the fundamental concepts of gene regulation.

FUNDAMENTALS OF GENE REGULATION

The modern era of molecular biology began with the great discovery, by James Watson and Francis Crick, of the DNA structure (Watson and Crick 1953). Later, the central dogma of molecular biology revolutionized science. This was first enunciated by Francis Crick (1958):

> The central dogma of molecular biology deals with the detailed residue-by-residue transfer of sequential information. It states that such information cannot be transferred from protein to either protein or nucleic acid.

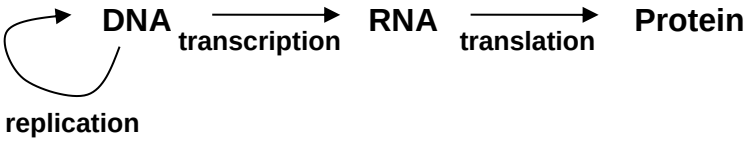

Figure 1. The central dogma of molecular biology.

The representation of the central dogma in Figure 1 shows the routes in the processing and transfer of information. DNA replication allows information to be passed from a cell to daughter cells, while transcription and translation pass the information through RNA to proteins, which serve to enact instructions coded in the DNA. The concept has been extended by the discovery of several additional processes, such as reverse transcription. A description of these processes can be found in any molecular cell biology textbook (see for example Lodish et al. 2000; Watson 2003) and will not be explained here. For our purpose,

it suffices to provide an overview of the transcription process, by which information is transferred from DNA to RNA.

Nucleic acids: DNA and RNA

Nucleic acids are macromolecules–polymers of small subunits called nucleotides. All nucleotides have a common structure: a *phosphate* group linked to a *pentose* (a five-carbon sugar molecule) that is linked to an organic *nitrogen base* (Figure 2). The pentose in RNA is ribose (hence the name ribonucleic acid) while the one in DNA is deoxyribose (hence the name deoxyribonucleic acid). There are two types of nitrogen bases: the one-ring pyrimidines, and the two-ring purines. Both DNA and RNA contain the bases adenine (A), guanine (G) and cytosine (C). Thymine (T) exists only in DNA, while uracil (U) is only present in RNA.

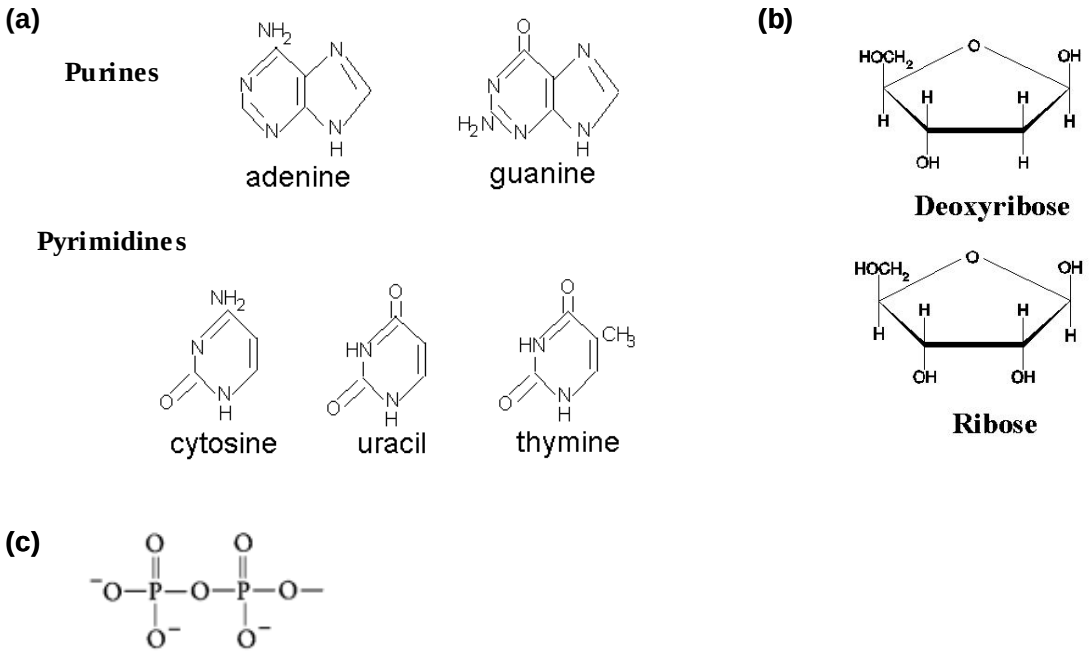

Figure 2. The constituents of nucleotides. (a) the nitrogen bases (purines and pyrimidines). Adenine, guanine and cytosine are common to RNA and DNA. Only DNA contains thymine, while only RNA contains uracil; (b) the sugars, ribose (constituent of RNA) and 2-deoxyribose (constituent of DNA); (c) a phosphate group.

The primary structures of RNA and DNA are similar, but the way polynucleotides twist and fold into stable three-dimensional conformations are different. DNA exists mainly as a single three-dimensional structure, the famous DNA double helix, while RNA can exist in several conformations. There are three main types of RNA: messenger RNA (mRNA), transfer RNA (tRNA), and ribosomal RNA (rRNA). Messenger RNA (mRNA) is involved in the transcription process, in which it serves as an information carrier from DNA to proteins. Transfer RNA (tRNA) is involved in translation, the building of proteins from its amino acid constituents. Ribosomal RNA (rRNA) is also involved in translation, being a constituent of the ribosomes, large ribonucleoprotein complexes where proteins are synthesized.

The genetic code

The DNA molecule contains four building blocks based on four nucleobases: adenine, cytosine, guanine, and thymine. Similarly, the RNA language is written in a four-letter alphabet, with uracil taking the place of thymine. Proteins may contain twenty different amino acids that are obtained from a genetic code in which three consecutive nucleobases function as a triplet called a *codon*. Of the sixty-four possible codons in the genetic code, sixty-one encode amino acids and three are called stop codons, which indicate that it is time to stop adding amino acids when building a protein. Most of the amino acids can be encoded by more than one codon (Table 1). This is why the genetic code is said to be degenerate; that is, there are synonyms.

Table 1. The genetic code. Each codon (triplet of three nucleotides) encodes an amino acid (except for the three stop codons). Most amino acids can be encoded by more than one codon. The DNA code is equivalent, with T in place of U.

First position	Second position				Third position
	U	**C**	**A**	**G**	
	Phe	Ser	Tyr	Cys	**U**

U	Phe	Ser	Tyr	Cys	**C**
	Leu	Ser	STOP	STOP	**A**
	Leu	Ser	STOP	Trp	**G**
C	Leu	Pro	His	Arg	**U**
	Leu	Pro	His	Arg	**C**
	Leu	Pro	Gln	Arg	**A**
	Leu	Pro	Gln	Arg	**G**
A	Ile	Thr	Asn	Ser	**U**
	Ile	Thr	Asn	Ser	**C**
	Ile	Thr	Lys	Arg	**A**
	Met (Start)	Thr	Lys	Arg	**G**
G	Val	Ala	Asp	Gly	**U**
	Val	Ala	Asp	Gly	**C**
	Val	Ala	Glu	Gly	**A**
	Val	Ala	Glu	Gly	**G**

The nucleotides are A = adenine, C = cytosine, G = guanine, and U = uracil. The amino acids are Phe = phenylalanine, Leu = leucine, Ser = serine, Tyr = tyrosine, Cys = cysteine, Trp = tryptophan, Pro = proline, His = histidine, Gln = glutamine, Arg = arginine, Ile = isoleucine, Met = methionine, Thr = threonine, Asn = asparagine, Lys = lysine, Val = valine, Ala = alanine, Asp = aspartic acid, Glu = glutamic acid, and Gly = glycine. The proteins always begin with a methionine, encoded by AUG (start codon), and the codons UAA, UAG, and UGA do not encode any amino acid, indicating the termination of translation.

Transcription

The word "double" in the description of DNA as a double helix refers to the structure of DNA as two complementary strands that have bases that alternate according to the base-pair rule: G in one strand corresponds to C in the other, and vice versa, and A is similarly linked with T. One strand serves as the functional strand, which encodes an amino acid sequence, while the other is the template strand used to synthesize an RNA molecule in the transcription process through the action of enzymes called RNA *polymerases*. Each T, C, A, and G in the template strand results in a corresponding A, G, U, and C in the RNA molecule; hence, the resultant RNA molecule is complementary to the template strand of DNA and identical to the functional strand except that uracil replaces thymine (Figure 3). RNA polymerases find an initiation site on the DNA duplex, bind it, temporarily separate the two strands, and begin generating a new RNA strand. Transcription is controlled by regulatory proteins called *transcription factors* (TF) that bind to specific sequences in DNA and activate or inhibit the transcription of genes. A TF that inhibits the transcription is

called a *repressor*, while those that stimulate transcription are called *inducers*. The sequences of DNA to which the TF binds are called *control elements*; they are *promoters* when they are involved in induction of transcription (binding of RNA polymerase) and *operators* when they are involved in repression of gene expression. These concepts will be important when we discuss the *lac* operon.

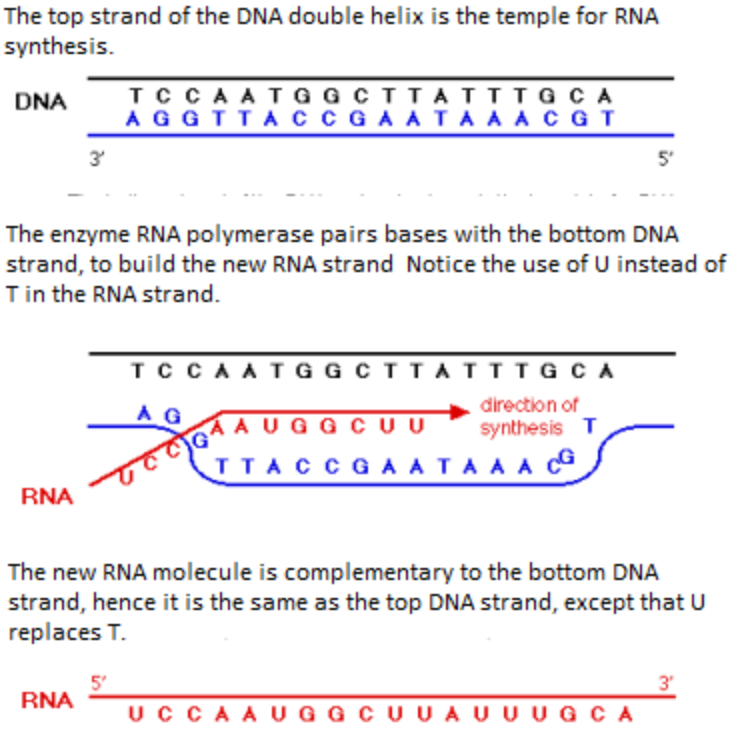

Figure 3. A simplified schematic view of the transcription process.

Operons

Prokaryotes are single-cell organisms, like bacteria, that consist of a single closed compartment of cytoplasm surrounded by a plasma membrane. In contrast with *eukaryotes*, such as yeast and all multi-celled organisms, prokaryotes do not possess internal organelles

surrounded by membranes. Prokaryotic DNA exists as large circular chromosomes, associated with polyamines and small proteins and folded into a compact structure. The most common arrangement of protein-coding genes in prokaryotes has a powerful and appealing logic: genes devoted to a single metabolic goal are most often found in a continuous array in DNA. The arrangement of genes in a functional group is called an *operon*. The full set of genes is transcribed into a single mRNA molecule. Ribosomes initiate translation at the beginning of each the genes in the mRNA produced from an operon and produce the polypeptides encoded in it.

THE LAC OPERON

Much of the pioneer work on the *lac* operon in *Escherichia coli* was done by François Jacob and Jacques Monod (1961). *E. coli* can regulate its gene expression depending on the carbon source used in the culture medium: when cells grow in glucose-based medium, the activity of the enzymes needed to metabolize lactose is very low, but in a lactose-containing medium there is an increase in the activity of the enzymes involved in lactose metabolism.

In *E. coli*, the enzymes induced in the presence of lactose are encoded by the *lac* operon, which contains structural genes for three enzymes involved in the metabolism of lactose (*LacZ, LacY, LacA*), one structural gene encoding a repressor protein (*LacR*), and three control elements involved in the regulation of transcription, P_R, P, and O (Figure 4). The *LacZ* gene encodes β-galactosidase, an enzyme that converts lactose into glucose and galactose, and the *LacY* gene encodes lactose permease, which is involved in the transport of lactose into the cell. The *LacR* gene encodes a control element called *lac*-repressor, involved in the regulation of the three structural genes in response to nutrient changes in the culture medium.

The structural genes *LacZ, LacY* and *LacA* are expressed only when lactose is present in the cell. In its absence, the *lac*-repressor (R) binds to the operator region O, and RNA polymerase, bound to the promoter P, is unable to move past this region. Hence, no

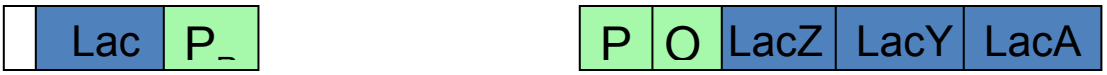

Figure 4. Schematic structure of the lac operon and the regions it contains**.** The operon contains regulatory regions and regions coding for proteins. The regulatory regions include P_R, a promoter for *lacR*; the operator O, binding site for the repressor R; and the promoter P, a binding site for RNA polymerase. The coding regions include the genes *LacR*, encoding the regulatory protein (repressor), and *LacZ*, *LacY*, and *LacA*, encoding proteins involved in the utilization of lactose by *E. coli* cells.

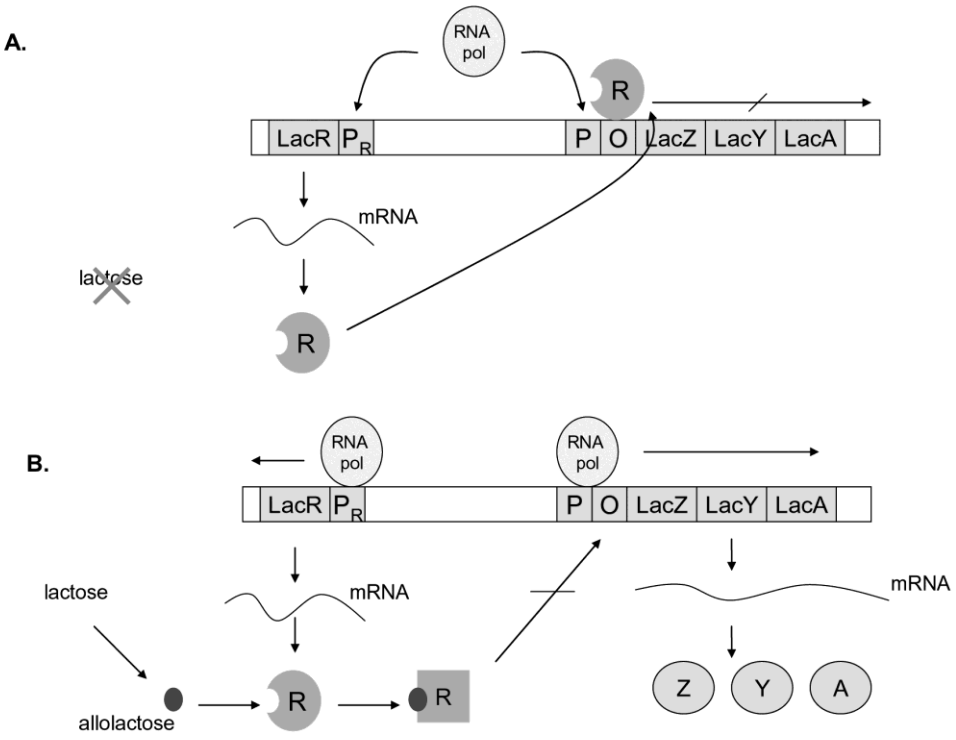

Figure 5. Regulation of gene expression in response to nutrients in *E. coli*: the *lac* operon (Koolman and Röhm 1996; Lodish et al. 2000). Details are in the text.

transcription of *LacZ, LacY* and *LacA* occurs (Figure 5A). When lactose enters the cell, it is converted by β-galactosidase into a similar molecule (isomer) called allolactose, which binds to the *lac*-repressor and induces a conformational change that prevents it from fitting into and binding to the operator region in the DNA. Without the *lac*-repressor blocking the

DNA, the RNA polymerase is able to move along the DNA, transcription of the three genes occurs, and lactose is metabolized (Figure 5B).

BOOLEAN NETWORKS

A *Boolean function* in n variables is a function that takes an n-bit string of 0s and 1s as input and produces a one bit output, using Boolean operators such as *and* ($\wedge$), *or* ($\vee$), and *not* ($\sim$). We call an n-bit string of 0s and 1s a binary *n-string*.

Example 1: A Boolean function in three variables is $f(x,y,z) = (x \wedge y) \vee (\sim z)$.
We observe:

$$f(0,1,0) = (0 \wedge 1) \vee (\sim 0) = 0 \vee 1 = 1$$
$$f(1,0,1) = (1 \wedge 0) \vee (\sim 1) = 0 \vee 0 = 0$$
$$f(1,1,1) = (1 \wedge 1) \vee (\sim 1) = 1 \vee 0 = 1$$

If $k = \boldsymbol{F}_2$ denotes the binary system $\{0,1\}$, then a Boolean function in n variables is a function $f\colon k^n \to k$. Here, k^n denotes the space of binary n-tuples. (It can be shown that any function $f\colon k^n \to k$ can be represented by a Boolean function).

Definition 1. A *Boolean network F* on n variables is a function $F = (f_1, \ldots, f_n)\colon k^n \to k^n$, where the f_i are Boolean functions. That is, F is a function that transforms binary n-strings into other binary n-strings, with the rule for transforming the i-th coordinate given by f_i.

Mathematically, we may view Boolean networks as time-discrete dynamical systems on a finite state space, where a state of the system is a binary n-tuple.

Example 2: Consider the Boolean network in 3 variables described by $F = (f_1, f_2, f_3)$,

where

$$f_1 = \sim (x_1 \veebar x_2) = \sim \{(x_1 \vee x_2) \wedge [\sim (x_1 \wedge x_2)]\},$$
$$f_2 = (x_1 \wedge x_2) \wedge x_3,$$
$$f_3 = x_1.$$

Note that since f_1 is the negative of the *exclusive or*, $f_1 = 1$ if $x_1 = x_2$ and $f_1 = 0$ otherwise.

There are two interesting directed graphs associated to a Boolean network: the dependency graph, or wiring diagram, and the state space graph. The *dependency graph* encodes the dependencies of a variable on the other variables. The nodes of the dependency graph correspond to the variables of the Boolean network. A directed edge from variable x to variable y indicates that x appears in the Boolean function of variable y. For the Boolean network in Example 2, the dependency graph is given in Figure 6.

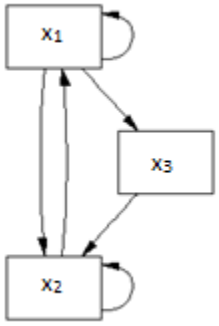

Figure 6. Boolean network of Example 2.

The dynamics of the network is given by the iterations of F:

$$F\,(1,0,1) = (0,0,1),\ \ F\,(0,0,1) = (1,0,0),\ \ F\,(1,0,0) = (0,0,1), \text{ etc.}$$

The dynamics of a Boolean network F on n variables can also be represented by a directed graph, the *state space* of F. It has 2^n vertices consisting of all binary n-strings, representing all possible states of the network. There is an edge from vertex a to vertex b if and only if

$F(a) = b$.

The state space of the Boolean network in Example 2 is given in Figure 7.

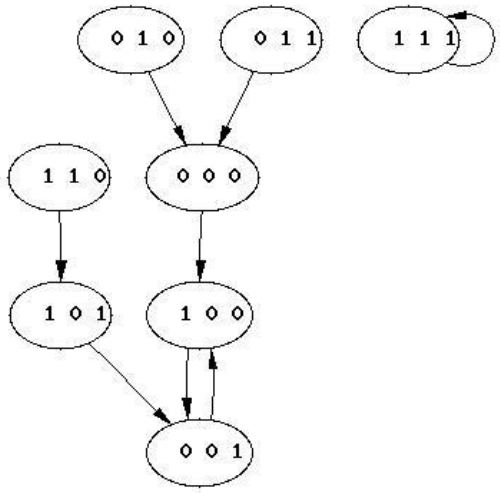

Figure 7. Dynamics of the network of Example 2

Definition 2. A node *a* in the phase space is called a *fixed point* if $F(a) = a$. A *limit cycle* in the phase space is a set of points $c_1, \ldots . c_t$ such that $F(c_1) = c_{i+1}$ and $F(c_t) = c_1$.

The state space of the Boolean network of Example 2 contains one fixed point $c = (1,1,1)$ and a limit cycle of length 2, consisting of the states (1,0,0) and (0,0,1).

STUDENT PROJECTS

The goal of the projects we designed is to let students experience modeling a molecular network with a minimum amount of preparation and prior knowledge. As mentioned earlier, this motivated our choice of Boolean networks as models. Molecular data describing the components of the *lac* operon are complicated to explain and to use, so we chose a modeling activity consisting of partial model validation based on the faithfulness of the model to basic biological features of the system.

Project 1

Based on the *lac* operon system described on the previous section, construct a Boolean network model *F* that contains the following as variables:

M = mRNA for *lac* genes, *Z* = beta-galactosidase, *S* = Allolactose (inducer), *L* = Lactose (intracellular), *Y* = Lactose permease

The dynamical system will be described as $F = (f_M, f_Z, f_S, f_L, f_Y)$, where each function indicates the presence or absence of the corresponding entity in terms of the state at the previous time step. For the model, we assume that each of transcription, translation, mRNA degradation, and protein degradation require one time unit and that extracellular lactose is always available.

One possible outcome of this activity is the Boolean model:

$$f_M = S$$
$$f_Z = M$$
$$f_S = S \vee (L \wedge Z)$$
$$f_L = Y \vee (L \wedge \sim Z)$$
$$f_Y = M$$

Each of the functions encodes a mechanism in the system that affects the corresponding molecular species. The first function, for instance, encodes the fact that the *lac* genes are expressed at time $t+1$ if and only if the inducer allolactose (S) is present at time t. The function f_S indicates that allolactose is present at time $t+1$ if it was present at time t or if lactose was present at time t together with β-galactosidase, which converts lactose into allolactose in one time step. We can assemble the functions into a Boolean network

$$F: \{0, 1\}^5 \rightarrow \{0, 1\}^5$$

that transforms a 5-tuple representing a system state into another 5-tuple representing another system state. Long-term dynamics are obtained by iteration of *F*.

Using the software package DVD (Jarrah et al. 2004), the participants can construct and visualize the topological and dynamical properties of the model. The model dynamics are

depicted in Figure 8. Each node of the directed graph represents one model state, including all $2^5 = 32$ states. A directed arrow from one state to another indicates a state transition. That is, if the functions in the model *F* are evaluated at the state at the origin of the arrow, then the resulting value is the node at the tip of the arrow.

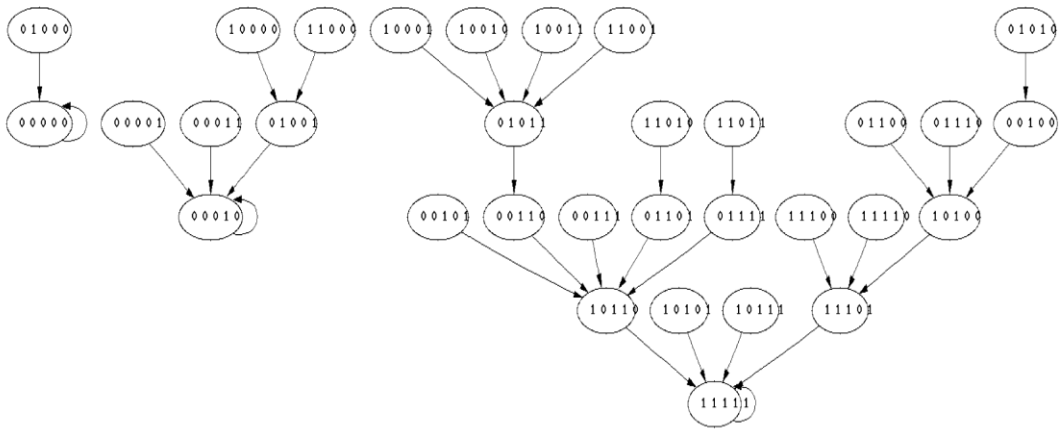

Figure 8. The topology and dynamics of the *lac* operon model**.** The figure was obtained using the software package DVD (Jarrah et al. 2004). The interpretation is described in the text.

Project 2

Based on the biological properties of the *lac* operon, analyze the model constructed in Project 1 and decide whether it is biologically realistic.

This project can be used to demonstrate how a mathematical model can be used to test and further understanding of the underlying biology. Assuming the Boolean model above as the outcome of Project 1, it has three possible long-term dynamic outcomes corresponding to the three fixed points of the state space graph. Since the *lac* operon is basically a bi-stable system which is either ON or OFF, only (0,0,0,0,0) and (1,1,1,1,1) should be fixed points; hence, the dynamics show that the model is not quite correct. Specifically, the additional

steady state of the model represents a situation in which lactose is present in the cell, but the machinery to metabolize it is turned off. Thus, the mathematical analysis points to a flaw in understanding the underlying biological mechanisms used to formulate the individual logical rules used in the model.

Project 3

Using additional biological insight and analysis of the model constructed in Project 1, modify it to better conform with biological knowledge.

In search of a way to modify the functions in the model so that the state (0,0,0,1,0) transitions to the steady state (0,0,0,0,0) participants need to understand more of the biology and reexamine the Boolean functions. One place to make a modification is the function for *S*. Its first term assures that *S* will be present at time $t+1$ if it was present at time *t*. Several modifications are possible, for instance deleting the first term or expanding it to include the presence of other variables. The process of model improvement leads to fruitful discussions that provide further insight into the biology, the modeling process, and the utility of models. The DVD software is a helpful, allowing easy visualization of the basic model properties (Jarrah et al. 2004). It also allows the participants to discuss whether the model constructed exhibits the expected properties of the biological system, how to test it, and how to improve the results obtained. This discussion is most useful if it is conducted in a team that contains different areas of expertise, e.g., math majors and biology or biochemistry majors.

DISCUSSION

Mathematical modeling is becoming an essential tool in the life sciences and in biomedicine, and several fields of expertise contribute to increasingly larger projects to understand the variety of biological networks that make organisms function. We believe that students should be exposed to this area at the interface of biology and mathematics as early as possible. We have designed a collection of projects that try to capture the essence

of mathematical modeling in biology, with a minimum of mathematical and biological background requirements. The projects are structured as open-ended hands-on team science activities that engage the students and encourage interaction.

While the *lac* operon has been studied for a long time, it continues to be an interesting and fruitful topic for ongoing research, as demonstrated by the recent literature on the subject. The projects thus bring students directly to a basic understanding of a topic at the forefront of current research. Depending on the setting, the projects can be expanded and extended in several directions, leading students to the intricacies of molecular data and mathematical models.

The projects are a case study for introducing real mathematical biology projects into the undergraduate and even high school curriculum. There are other biological topics that lend themselves to a similar approach, for example, the workshop introduced by Rivera-Marrero and Stigler (2004) applied to an epidemiology problem of viral epidemic prediction and prevention.

ACKNOWLEDGEMENTS

We thank the teachers who participated in the workshop for their feedback, which will allow us to make improvements. We thank Elena Dimitrova for her input into the workshop structure; Brandy Stigler and Olgamary Rivera-Marrero for sharing with us their workshop materials from the 2004 SEDI Workshop in Bioinformatics; and Raina Robeva and Jill Granger for their advice on strategies for a successful workshop. We additionally thank Raina Robeva for advising us on the publication of this manuscript. The Boolean network model used in the workshop was constructed by B. Stigler. We especially thank Susan Faulkner, the Education and Outreach Officer at VBI, for handling the paperwork between VBI and IALR that made this workshop possible. We also thank Morgan Maurer and Alana Manzini for their help in preparing the printed material for the workshop, Jim Walke for the critical revision of this manuscript, and Abdul Jarrah for providing references used in the manuscript preparation.

REFERENCES

Albert, R., and H.G. Othmer, 2003: The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*, *J. Theor. Biol.* **223**, 1-18.

Barabasi, A.L., and Z.N. Oltvai, 2004: Network biology: understanding the cell's functional organization, *Nat. Rev. Genet.* **5**, 101-113.

Caplan, M.R. and E.V. Rosca, 2005: Targeting drugs to combinations of receptors: a modeling analysis of potential specificity, *Ann. Biomed. Eng.* **33**, 1113-1124.

Crick, F.H., 1958: On protein synthesis, *Symp. Soc. Exp. Biol*. **XII**, 138-163.

de Jong, H., 2002: Modeling and simulation of genetic regulatory systems: a literature review, *J. Comput. Biol.* **9**, 67-103.

Friedman, N., M. Linial, I. Nachman and D. Pe'er, 2000: Using Bayesian networks to analyze expression data, *J. Comput. Biol.* **7**, 601-620.

Hoops, S., S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes and U. Kummer, 2006: COPASI–a COmplex PAthway SImulator, *Bioinformatics* **22**, 3067-74.

Jacob, F., and J. Monod, 1961: Genetic regulatory mechanisms in the synthesis of proteins, *J. Mol. Biol.* **3**, 318-356.

Jarrah, A., R. Laubenbacher and H. Vastani, 2004, cited in 2012: DVD: Discrete Visual Dynamics [Available online at http://dvd.vbi.vt.edu].

Jarrah, A., R. Laubenbacher, B. Stigler, and M. Stillman, 2007: Reverse-engineering of polynomial dynamical systems, *Adv. Appl. Math.* **39**, 477-489.

Kauffman, S.A., 1969: Metabolic stability and epigenesis in randomly constructed genetic nets, *J. Theor. Biol.* **22**, 437-467.

Koolman, J., and K.-H. Röhm, 1996: *Color Atlas of Biochemistry*, Thieme.

Laubenbacher, R., and B. Stigler, 2004: A computational algebra approach to the reverse engineering of gene regulatory networks, *J. Theor. Biol.* **229**, 523-537.

Lodish, H., L. Berk, L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, 2000: *Molecular Cell Biology*, W.H. Freeman.

Murray, J.D., 1993: *Mathematical Biology*, Springer-Verlag.

National Council of Teachers of Mathematics: Principles and Standards for School Mathematics, cited 2012: [Available online at http://standards.nctm.org/].

Ribba, B., T. Colin and S. Schnell, 2006: A multiscale mathematical model of cancer, and its use in analyzing irradiation therapies, *Theor. Biol. Med. Model.* **3**, 7-25.

Rivera-Marrero, O., and B. Stigler, 2004, cited 2012: Model Your Genes the Mathematical Way I [Available online at http://admg.vbi.vt.edu/home/Outreach/Workshops/1].

Veliz-Cuba, A., R. Laubenbacher, and M. Beeken, 2009, cited 2012: A Mathematics Classroom Module [Available online at http://admg.vbi.vt.edu/home/Outreach/OR/MathModule].

Virginia Department of Education, Standards of Learning Currently in Effect for Virginia Public Schools [http://www.pen.k12.va.us/VDOE/Superintendent/Sols/home.shtml].

Watson, J.D., 2003: *DNA–The Secret of Life*, Alfred A. Knopf.

Watson, J.D., and F.H. Crick, 1953: Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid, *Nature* **171**, 737-738.