Math. Model. Nat. Phenom.
Vol. 6, No. 6, 2011, pp. 39-60
DOI: 10.1051/mmnp/20116603

Boolean Biology: Introducing Boolean Networks and Finite Dynamical Systems Models to Biology and Mathematics Courses

R. Robeva[1*], B. Kirkwood[1] and R. Davies[2]

[1] Department of Mathematical Sciences, Sweet Briar College, Sweet Briar, VA 24595, USA
[2] Department of Biology, Sweet Briar College, Sweet Briar, VA 24595, USA

Abstract. Since the release of the *Bio 2010* report in 2003, significant emphasis has been placed on initiating changes in the way undergraduate biology and mathematics courses are taught and on creating new educational materials to facilitate those changes. Quantitative approaches, including mathematical models, are now considered critical for the education of the next generation of biologists. In response, mathematics departments across the country have initiated changes to their introductory calculus sequence, adding models, projects, and examples from the life sciences, or offering specialized calculus courses for biology majors. However, calculus-based models and techniques from those courses have been slow to propagate into the traditional general biology courses. And although modern biology has generated exciting opportunities for applications of a broad spectrum of mathematical theories, the impact on the undergraduate mathematics courses outside of the calculus/ordinary differential equations sequence has been minimal at best. Thus, the limited interdisciplinary cross-over between the undergraduate mathematics and biology curricula has remained stubbornly stagnant despite ongoing calls for integrated approaches.

We think that this phenomenon is due primarily to a lack of appropriate non-calculus-based interdisciplinary educational materials rather than inaccessibility of the essential underlying mathematical and biology concepts. Here we present a module that uses Boolean network models of the lac operon regulatory mechanism as an introduction to the conceptual importance of mathematical models and their analysis. No mathematics background beyond high school algebra is necessary to construct the model, which makes the approach particularly appropriate for introductory biology and mathematics courses. Initially the module focuses on modeling via Boolean logic, Boolean algebra, discrete dynamical systems, and directed graphs. The analysis of the model, however, leads to interesting advanced mathematical questions involving polynomial ideals and algebraic

*Corresponding author. E-mail: robeva@sbc.edu

Article published by EDP Sciences and available at http://www.mmnp-journal.org or http://dx.doi.org/10.1051/mmnp/20116603
https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

varieties that are beyond the mathematical proficiency of most biology students but are of interest in advanced level abstract algebra courses. These questions can also be used to map a path toward further research-level problems. All computations are carried out by computational algebra systems and the advanced mathematical theory implemented by the software can be covered in detail in abstract algebra courses with mathematics students. Biology students and students in lower level mathematic courses, on the other hand, can view the implementation as a "black box" and focus on the interpretation and the implications of the output.

The module is available from the authors for classroom testing and adoption.

Key words: educational modules for mathematics and biology, gene regulation, finite dynamical systems, Boolean networks, Groebner bases, systems of polynomial equations
AMS subject classification: 92B05, 92E10, 93A30, 93C65, 97C90

1. Introduction

In its report *Bio 2010*, the National Research Council stated in 2003 that major changes in research compel major changes in undergraduate education [1]. Based on the remarkable breakthroughs made recently in the life sciences, the report recommended aggressive curriculum changes and charted the way for educating the "quantitative biologists" of the future to have them prepared to meet the challenges of 21st century biology.

Indeed, during the last decade the challenges have been huge but the advances have been nothing short of revolutionary. For instance, it has long been understood that life at any level from cell functioning to human behavior is defined by the dynamical interactions between its components, not by the properties of these components in isolation. Only quite recently, however, have the fundamental advances in molecular biology, computing, and mathematical modeling allowed us to attempt a coordinated approach to studying and understanding this phenomenon.

Systems biology, with its heavy mathematical underpinning, is the cross-disciplinary methodology behind the effort to understand the dynamics of life, aiming at determining how the individual components of a living system interact in space and time to form functional networks. The challenges are huge: in biological systems at all levels of organization from sub-cellular to the cell, tissue, organ, and human behavior, control and functional mechanisms are emergent properties of the network, not of its separate components [15].

A recent proposal for a new national initiative (towards "the New Biology") identifies health issues as one of four key areas where a systems biology approach and improvements in mathematical and statistical modeling will be prerequisites for progress: *"Although there are increasing efforts to apply quantitative approaches to biological questions, more must be done to transform biology from its origins as a descriptive science to a predictive science. We will ultimately be limited in our ability to deploy biological systems to solve large-scale problems unless we significantly deepen our fundamental understanding of the organizational principles of complex biological systems, a*

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

staggeringly difficult challenge. The growth of the New Biology will be dramatically accelerated by developing frameworks for systematically analyzing, predicting, and modulating the behavior of complex biological systems." [11].

The tools of modern biology alone are insufficient for seeking answers. Their insights into available data from molecular biology, genomic, proteomic, metabolomic, neuroendocrine, and behavioral research need to be merged with mathematical models, computational tools, and engineering systems analysis to ensure better understanding of the interplay between function and behavior. In this effort, tight links between network modeling and experimentation and their consistent iterative interaction are critical to understanding the network structure.

At the molecular level, keys to understanding the mechanisms of gene regulation lie in the control of gene expression (the specific regulation of mRNA and protein synthesis) and in the protein interactions in the cell. In any given cell, hundreds of thousands of protein molecules may be active at any one time. These proteins may bind with the DNA or with each other, leading to extremely complicated interaction networks. This complexity is also a source of experimental challenges. Since multiple feedback loops are present to control the mechanisms of molecular interactions, it may be difficult to decide which biomolecular species are fundamental to the system and therefore which should be measured to obtain data to validate the network structure.

Mathematics provides a formal framework for organizing the overwhelming amounts of disparate experimental data and for developing models that reflect the dependencies between the various systems' components. However, unlike physics and engineering, which have provided questions that stimulated mathematical theory and have driven mathematical advances to the benefit of all three disciplines for centuries, biology and the life sciences have only recently made a noticeable turn toward mathematical approaches. Thus, not only is it imperative that we prepare future biologists to build and use mathematical models but it is also critical to educate and prepare mathematicians capable of applying their mathematical skills in a changing interdisciplinary landscape where biology is now posing questions driving the discovery of new mathematics [10].

The concept of the mathematical model is central to this effort. Regardless of its type, a good mathematical model should be able to reproduce key properties of a system's behavior such as convergence to equilibria or limit cycles, robustness to small perturbations and noise, and responses to stimuli. Good models can then facilitate new advances in biology, acting as "virtual laboratories" that allow for *in silico* experiments. Such experiments may then lead to a deeper understanding of the system, help generate new hypotheses, and suggest ways for designing new, more informative laboratory experiments. In this sense, mathematical models can serve as "microscopes" for observing how the system's structure affects its properties [4].

The two types of mathematical models used successfully to organize insights from molecular biology and capture network structure and dynamics are: (i) continuous-time models built from difference equations or differential equations (DE models), which focus on the kinetics of biochemical reactions; (ii) discrete-time finite dynamical systems (FDS) models built from functions of finite-state variables (in particular Boolean networks), which focus on the logic of the network variables' interconnections. In addition, other approaches including hybrid models (containing both DE and FDS portions) and stochastic models (accounting for some inherent randomness of the network) have proved beneficial.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

Variables in a DE model can span a continuous range of biologically feasible values. Modelers need comprehensive knowledge of interactions between variables, which may include, for example, specific details of control mechanisms, rates of production and degradation, or highest and lowest biologically relevant concentrations. In an algebraic model only values from a finite set are allowed. The special case of a Boolean network allows only two states, *e.g.*, 0 and 1, representing (for example) the absence or presence of gene products in a model of gene regulation. In contrast to DE, the information necessary to construct a Boolean model requires only a conceptual understanding of the causal links of dependency. Thus continuous models are quantitative while Boolean models are qualitative in nature.

Historically, DE models have been the preferred type of mathematical models used in biology. This type of dynamical modeling has proved to be essential for problems in ecology, epidemiology, physiology, and endocrinology, among many others. Since the release of the *Bio 2010* report, the popularity of DE models has increased and some DE models have become mainstream examples populating the revised calculus curriculum and the newly developed bio-calculus courses. Many curricular materials with DE orientation are now available and many institutions offer DE-focused courses that include problems coming from the life sciences. Difference equations are used to model system dynamics in precalculus level courses and bio-calculus courses place special emphasis on developing DE models for various biological systems in place of or in addition to the traditional linkages with physics and engineering. Common examples include population dynamics and the spread of an epidemic [2, 3, 12].

It is important to note, however, that those changes have been initiated primarily on the mathematics side of the curriculum in courses at the pre-calculus or calculus levels. Since the vast majority of institutions still do not require those as prerequisites for their general and cell biology courses, the introductory level biology courses have remained generally unaffected by these trends. In reality many biology majors may still graduate without having taken a single course in which mathematical models have been used and discussed in a serious way in connection with the relevant biological questions.

Boolean models, a special case of FDS models, were introduced in 1969 to study the dynamic properties of gene regulatory networks [8]. They have proven useful in cases where network dynamics are determined by the logic of interactions rather than by finely tuned kinetics, the details of which often are not known. Today, many FDS models appear in the literature, including a model of the metabolic network in *E. coli* [17], the abscisic acid signaling pathway [22], and T-cell receptor signaling [16]. However, in contrast with the abundance of undergraduate textbooks and educational modules focusing on DE models, very few curricular materials focusing on Boolean models and FDS models have been created, regardless of their high educational potential [13].

In this article we outline a teaching module developed by mathematics and biology faculty at Sweet Briar College in collaboration with faculty from the Virginia Bioinformatics Institute (VBI) at Virginia Tech. The module implements the ideas that Boolean models can and should be used as an introduction to mathematical modeling in entry-level mathematics and biology courses for which calculus is not a prerequisite [13] and that discrete models can be used as a bridge between mathematical applications to biology and mathematical concepts appropriate for advanced undergraduate mathematics courses [9]. As a main example the module uses Boolean models of one

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

of the simplest and best understood mechanisms of gene regulation: the *lactose (lac) operon* that controls the transport and metabolism of lactose in *E. coli*. Since the seminal work by Jacob and Monod [6], the lac operon has become one of the most widely studied and best understood mechanisms of gene regulation. It has also been used as a test system for virtually any mathematical method for modeling of gene regulation, including DE and FDS models.

2. Outline of the Lac Operon Module

2.1. Boolean Network Models and FDS Models

Boolean network models are time discrete dynamical models for which the model variables and parameters can take only two possible values, 0 or 1. The parameters are considered fixed and do not vary with time while the variables change in a way determined by the rules of interactions. Variables and parameters are denoted as nodes of a directed graph, called a *wiring diagram*, and the interactions between the nodes are depicted by arrows. Assume that a model has n variables denoted by $x_1, x_2, \ldots x_n$ and that variable x_j influences variable x_i. The wiring diagram will then have a directed link from x_j to x_i. The dynamical behavior of the model is then described by a set of *transition functions* $f_1, f_2, \ldots, f_n$. Mathematically, the transition function f_j of each variable x_j is a Boolean expression of the variables influencing it, built from the Boolean operations AND (denoted by the mathematical symbol $\wedge$), OR (denoted by the mathematical symbol $\vee$) and NOT (denoted by the mathematical symbol $\neg$). In the context of Boolean networks the following intuitive definitions for the operations AND and OR are often helpful: if two variables, say x and y, of the system control a third variable z, $z = x \wedge y$ reflects the idea that x and y need to be simultaneously present (that is, have values 1) to affect z while $z = x \vee y$ represents the concept that x and y influence z independently and z is affected when either x OR y is present. A directed link from x_j to x_i in the wiring diagram means that the variable x_j appears in the definition of the transition function f_i. For instance the set of transition functions

$$\begin{aligned} f_1 &= f_1(x_1, x_2, x_3, x_4) = x_3 \\ f_2 &= f_1(x_1, x_2, x_3, x_4) = x_3 \wedge x_4 \\ f_3 &= f_1(x_1, x_2, x_3, x_4) = x_2 \wedge x_3 \\ f_4 &= f_1(x_1, x_2, x_3, x_4) = x_1 \wedge x_2 \wedge x_3 \end{aligned} \tag{2.1}$$

is consistent with the wiring diagram in Figure 1 (left panel) and thus represents a set of possible transition functions for that system. The actual expressions defining the transition functions will be developed from information about the known types of interactions between the variables. In this specific example, due to the fact that all dependencies are conjunctive, the transition function for each x_i is constructed by using the AND operation on all variables x_j with outgoing links into x_i.

The transition functions determine the dynamical evolution of the model: Starting from an initial condition $(x_1^0, x_2^0, \ldots, x_n^0)$ at time $t = 0$ the values of the variables at time $t = 1$, which we will denote by $(x_1^1, x_2^1, \ldots, x_n^1)$, are computed from the transition functions as $x_j^1 = f_j(x_1^0, x_2^0, \ldots, x_n^0)$

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

for $j = 1, 2, \ldots, n$. Continuing the same way, at time $t = 2$, we obtain $x_j^2 = f_j(x_1^1, x_2^1, \ldots, x_n^1)$ for $j = 1, 2, \ldots, n$ and so on. In general, the transition between time t and $t + 1$ is given by $x_j^{t+1} = f_j(x_1^t, x_2^t, \ldots, x_n^t)$. As an example, consider the set of transition functions defined in Eqs. (2.1), with the initial condition $(x_1^0, x_2^0, x_3^0, x_4^0) = (0, 0, 1, 1)$. Substituting these values into Eqs. (2.1), one obtains

$$\begin{aligned}
x_1^1 &= f_1(x_1^0, x_2^0, x_3^0, x_4^0) = f_1(0, 0, 1, 1) = 1 \\
x_2^1 &= f_2(x_1^0, x_2^0, x_3^0, x_4^0) = f_2(0, 0, 1, 1) = 1 \wedge 1 = 1 \\
x_3^1 &= f_3(x_1^0, x_2^0, x_3^0, x_4^0) = f_3(0, 0, 1, 1) = 0 \wedge 1 = 0 \\
x_4^1 &= f_4(x_1^0, x_2^0, x_3^0, x_4^0) = f_4(0, 0, 1, 1) = 0 \wedge 0 \wedge 1 = 0.
\end{aligned}$$

Next, the values $(x_1^1, x_2^1, x_3^1, x_4^1) = (1, 1, 0, 0)$ are used to evaluate the transition functions again, producing $(x_1^2, x_2^2, x_3^2, x_4^2) = (0, 0, 0, 0)$. Plugging these values into the functions f_j again now returns the same values $(x_1^3, x_2^3, x_3^3, x_4^3) = (0, 0, 0, 0)$. We say that we have computed the *trajectory* $(0, 0, 1, 1) \to (1, 1, 0, 0) \to (0, 0, 0, 0) \to (0, 0, 0, 0)$ and that $(0, 0, 0, 0)$ is a *fixed point* for the Boolean network. Similarly, $(1, 1, 1, 1)$ is also a fixed point. Considering all sixteen different sequences $(x_1^0, x_2^0, x_3^0, x_4^0)$ composed of 0s and 1s as initial states will generate all possible trajectories for the Boolean system, leading to the entire directed graph representing the *state space transition diagram* of the Boolean network. Loops of length larger than one on the space transition diagram correspond to limit cycles. For a network with a large number of variables, computing the trajectories requires appropriate software. The web-based Discrete Visualizer of Dynamics (DVD) is an application (available at http://dvd.vbi.vt.edu) that takes the transition functions as input and returns the wiring diagrams and the state space of the Boolean network. The right panel of Figure 1 depicts the output from our example. There are two fixed points $(0, 0, 0, 0)$ and $(1, 1, 1, 1)$ and no limit cycles. The state space transition diagram has two components.

When the model has too many variables and displaying the entire state space is not possible, DVD allows for computing the characteristics of single trajectories.

FDS models can be considered a generalization of Boolean network models where the model variables can take values from a finite set $S = \{0, 1, 2, p - 1\}$. When p is a prime number, the set S can be considered a finite field and the transition functions defined for $(x_1, x_2, \ldots, x_n) \in S^n$ can be considered polynomials of the variables $x_1, x_2, \ldots,$ and x_n with coefficients from the field S.[†] The dynamical evolution of the system is determined in the same way as for Boolean networks: $x_j^{t+1} = f_j(x_1^t, x_2^t, \ldots, x_n^t)$ for $j = 1, 2, \ldots, n$ and $t = 0, 1, 2, \ldots$ but the functions f_j are evaluated according to the appropriate field arithmetic for S.

Boolean models are a special case of FDS models with $S = \{0, 1\}$. In the Boolean case the representation of the transition functions as polynomials can be done following the rules:

1. $x_1 \wedge x_2 = x_1 x_2$
2. $x_1 \vee x_2 = x_1 + x_2 + x_1 x_2$
3. $\neg x_1 = x_1 + 1$

[†]These mathematical details are only discussed in the last part of the module developed for use with mathematics majors in abstract algebra courses.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

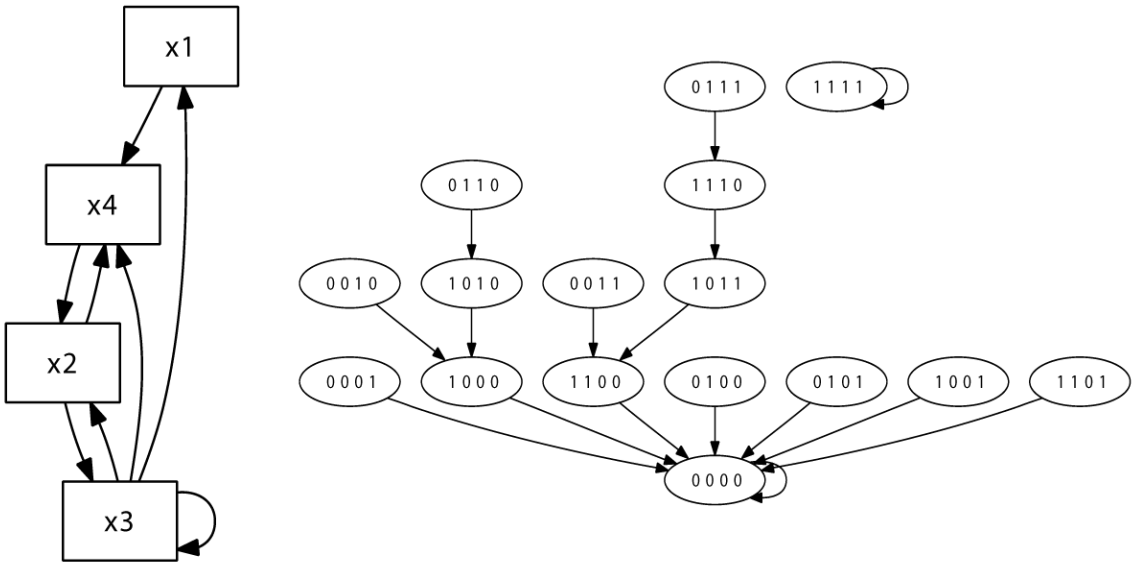


Figure 1: Wiring diagram (left panel) and the state space diagram (right panel) for the Boolean dynamical system in the example. Graphs produced with DVD (http://dvd.vbi.vt.edu).

2.2. The Lactose Operon

Here we present a very brief description of the lac operon mechanism in the bacterium *Escherichia coli (E. coli)*. A simplified Boolean model, reflecting only the negative regulation of the lac operon, is presented below. The module contains further details, including a description of both positive and negative control of the lac operon function and a discussion of a Boolean model that incorporates both.

Successful organisms must be able to make efficient use of available resources and to respond to changes in their environment, both of which are facilitated by effective control of gene expression. Given a choice, *E. coli* will use the most effective food molecule, glucose, first. If a mixture of glucose and lactose is present, *E. coli* will use the glucose first and only then will it use lactose. If only lactose is present, then *E. coli* will use lactose.

Lactose is a disaccharide composed of glucose and galactose. In order to be used by *E. coli*, lactose must first be taken into the cell through the action of lactose permease, a transport protein found in the plasma membrane. It must then be broken apart into glucose and galactose, through the action of the enzyme β-galactosidase. Making proteins is an energy-intensive process, so *E. coli* should only make lactose permease and β-galactosidase if lactose is present and glucose is not. Producing these proteins when they are not needed would be a waste of energy.

Interestingly, if lactose is the only sugar present, both proteins are produced simultaneously. This is due to the fact that they are found in a genetic organizing structure called the lac operon, described by Jacob and Monod [6]. In the lac operon, the genes for β-galactosidase (Lac Z) and lactose permease (Lac Y) are found adjacent to each other and under the control of a single promoter, so that the messenger RNA (mRNA) that is made from the operon contains the coding

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

information for both proteins. (See Figure 2.)

How does *E. coli* know when to turn on the lac operon and make lactose permease and β-galactosidase? If no lactose is present, the operon is off. This is due to the binding of a protein called the lac repressor which prevents the synthesis of mRNA. If lactose is present, some of it is converted to allolactose by β-galactosidase. The allolactose then binds the lac repressor and prevents it from binding to the DNA. The mRNA and then the lactose permease and β-galactosidase proteins can be made. (See Figure 2.)

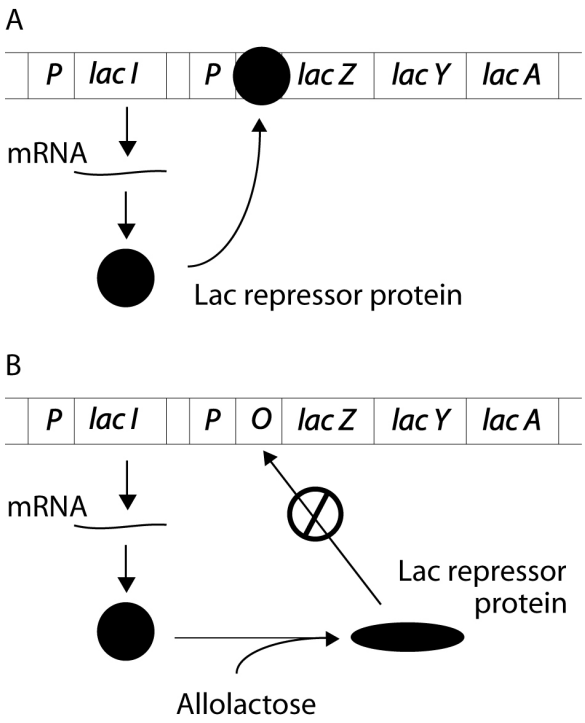


Figure 2: Panel A: The lac repressor protein in action. The lac repressor protein binds the lac operon at the operator, preventing transcription of the lac operon messenger RNA. The operon is OFF; Panel B: Binding of allolactose to the lac repressor causes a conformational change in the repressor, preventing it from binding at the operator. Transcription of the lac operon messenger RNA can proceed. The Operon is ON. Reproduced from *CBE-Life Sciences Education*, Vol. 9, Fall 2010, p.233 [14].

How does *E. coli* prevent the synthesis of these proteins when glucose is available? If glucose is present, the metabolism of glucose causes the intracellular levels of cyclic AMP (cAMP) to decline. If cAMP is present, it binds to the Catabolite Activator Protein (CAP), forming a complex which binds to the promoter of the lac operon and facilitates the attachment of RNA polymerase. If there is no cAMP, then there will be no CAP-cAMP, no polymerase binding, and the operon will remain off. If there is no glucose, the cAMP levels will rise, and then CAP-cAMP will form and facilitate polymerase binding. Figure 3 presents a schematic of the entire lac operon regulatory mechanism.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

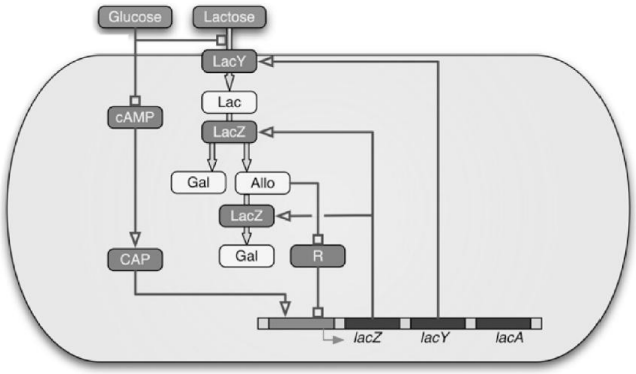


Figure 3: Schematic of the lac operon regulatory mechanism. LacY is a permease that transports external lactose into the cell. Protein LacZ polymerizes into a homotetramer named β-galactosidase. This enzyme transforms internal lactose (Lac) to allolactose (Allo) or to glucose and galactose (Gal). It also converts allolactose to glucose and galactose. Allolactose can bind to the repressor (R), inhibiting it. When not bound by allolactose, R can bind to a specific site upstream of the operon structural genes and thus avoid transcription initiation. External glucose inhibits the production of cAMP that, when bound to protein CAP to form the CAP-cAMP complex, acts as an activator of the lac operon. External glucose also inhibits lactose uptake by permease proteins. Reprinted from *Biophysics Journal*, Vol. 92, M. Santillan, M. C. Mackey, and E. S. Zeron, Origin of bistability in the lac operon, 3830 - 3842, Copyright (2007), with permission from Elsevier [18].

2.3. Boolean Models of the Lac Operon

The modeling process begins with choosing the model variables to represent the major dynamic elements of the system (quantities that change with time) and the model parameters that correspond to static descriptors. Different decisions regarding the exclusion or inclusion of any given component or part of the system will lead to different models. The next step is to define a wiring diagram, depicting the dependencies between variables and parameters, as described in Section 2.1. Some wiring diagrams provide additional information about the type of effect x exerts on y: a positive influence is given as an arrow and negative influences appear as links ending with circles or squares.

The models of the lac operon we will discuss initially follow a "minimal model" approach for choosing variables and parameters after Santillan *et al.* [18]. The model does not consider the CAP-cAMP positive control mechanism, which is essentially an amplifier for the transcription process. It focuses instead on the remaining part of the network interaction including mRNA, the lac operon proteins, and the presence or absence of lactose and glucose, inside and outside of the cell. It then further reduces the number of variables based on known dependences.

Thus, the model development begins with focusing on the following elements (the notation in the parentheses are the variable/parameter names used for the model) : mRNA (M), β-galactosidase (B), lac permease (P), intracellular lactose (L), allolactose (A), external lactose (L_e) and external glucose (G_e). Due to the fact that external conditions for the cell change slowly compared to the

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

lifespan of *E. coli*, we can assume that L_e and G_e remain relatively unchanged with time, assuming them to be constants and including them in the set of model parameters. The other quantities (M, B, P, L, and A) will be assumed to vary with time. However, some of these variables exhibit related dynamics due to similarities in the underlying biochemical structures and mechanisms.

Since β-galactosidase is a homo-tetramer made up of four identical lacZ polypeptides and the translation rate of the lacY transcript is assumed to be the same as the rate for the lacZ transcript, the following holds for the intracellular concentrations of β-galactocidase (B) and permease (P) : $P = E$ and $B = E/4$, where E denotes the LacZ polypeptide. Further, the model also assumes that the concentrations of internal lactose (L) and allolactose (A) are proportional, that is $A = pL$, where p indicates the fraction of lactose converted into allolactose and which can be determined experimentally. Thus, the model assumes three model variables – M, E, L, and two model parameters – L_e and G_e. Knowing the variables M, E, and L at any given time instance would allow for determining the values of P, B, and A at the same time instance, using $B = E/4$, $P = E$, and $A = pL$. The corresponding wiring diagram is depicted in Figure 4.

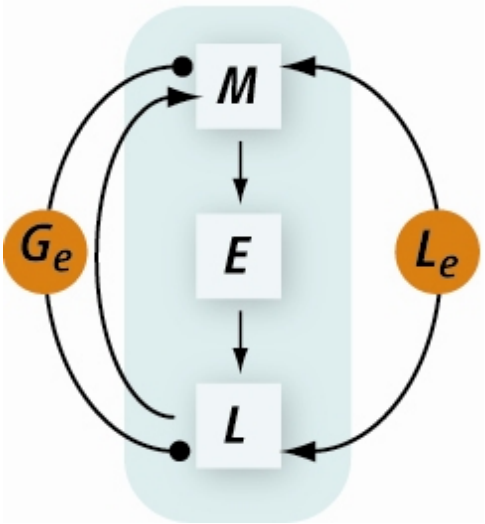


Figure 4: Wiring diagram for the minimal model. E denotes the lacZ polypeptide, M - the mRNA, L - internal lactose. L_e and G_e denote external lactose and glucose, respectively. The square nodes represent model variables while the round nodes represent model parameters. Directed links indicate influences: a positive influence is depicted by an arrow; a negative influence is depicted by a circle. Reproduced from *Science* 325 (2009), 542 – 543 [13].

We should note that the choice for variables and parameters discussed here is just one possibility among many. The model by Yildrim and Mackey [21], for instance, is based on assumptions leading to a wiring diagram including five nodes corresponding (in our notation) to the variables M, B, P, L, and A, and a node for external lactose as a parameter. This model and several others are discussed and analyzed in the module.

Once the model variables have been identified, the decision on the type of mathematical model should be made. As mentioned earlier, various types of mathematical models can be developed

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

(including DE, algebraic, stochastic, and simulation models among others) using the same wiring diagram. In this section, we will focus on developing a Boolean network model based on the wiring diagram in Figure 4.

The Boolean model is built under the assumptions below. Some of them may appear too strong but they are justified in the module by the qualitative nature of the model. 1) Transcription and translation require one unit of time. This means that if all necessary conditions for the activation of the molecular mechanism are present at time t, the protein production will be happening in time $t+1$; 2) Degradation of all mRNA and proteins occurs in one time step, and 3) Since trace levels (basal levels) of permease, and thus of polypeptide E, are present at all times, minimal, trace levels of lactose will be available in the cell when external lactose is available. This means that we assume that the values of L and E will be considered to be 1 when levels of lactose and lacZ polypeptide are measurably higher than the basal level.

The transition functions for the "minimal" lac operon model are given in Eqs. (2.2), followed by justification for each of the functions:

$$\begin{aligned} f_M(t+1) &= \neg G_e(t) \wedge (L(t) \vee L_e(t)) \\ f_E(t+1) &= M(t) \\ f_L(t+1) &= \neg G_e(t) \wedge (E(t) \wedge L_e(t)) \end{aligned} \tag{2.2}$$

Boolean function for M: The first equation states that for messenger RNA to be present at time $t+1$, there should be no external glucose at time t, and either internal or external lactose should be present. In other words, when external glucose is present ($G_e = 1$), no mRNA will be produced ($M = 0$). When there is no external glucose ($G_e = 0$) and there is lactose inside the cell ($L = 1$) or outside the cell ($L_e = 1$), there will be at least a small number of lactose molecules inside the cell. This will cause mRNA production at time $t+1$.

Boolean function for E: The production of mRNA ($M = 1$), will be followed by production of the lacZ polypeptide ($E = 1$).

Boolean function for L: If there is no external glucose ($G_e = 0$), external lactose is available ($L_e = 1$), and permease (as represented by the polypeptide E) is present ($E = 1$), the permease will bring extracellular lactose inside the cell, ensuring the presence of intracellular lactose.

The module continues with the analysis and initial validation of the model. As a model is just an approximation of the actual system, its validation has to be considered within the context of the general question the model is developed to help answer. In this example, due to the very simple nature of the model, it should be able to reflect at least the basic qualitative dynamic properties of the lac operon. Thus, at a minimum, the model should show that the operon has two steady states, ON and OFF. When extracellular glucose is available, the operon should be OFF. When extracellular glucose is not present and extracellular lactose is, the operon must be ON. We next demonstrate that our model satisfies these conditions.

The operon is ON when mRNA is being produced ($M = 1$). When mRNA is present, the production of permease and β-galactosidase is also turned on. This corresponds to the fixed-point state $(M, E, L) = (1, 1, 1)$. On the other hand, when mRNA is not made, the operon is OFF. This also means no production of lactose permease, and β-galactosidase. This corresponds to the fixed-point state $(M, E, L) = (0, 0, 0)$.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

For this Boolean model of the lac operon, there are four possible combinations for the values L_e and G_e of the model parameters: $L_e = 0, G_e = 0; L_e = 0, G_e = 1; L_e = 1, G_e = 0$; and $L_e = 1, G_e = 1$. For each one of these pairs of values we compute the state space using the transition functions of the model. The results are shown in Figure 5. Notice that according to the model, the operon is ON only when external glucose is unavailable and external lactose is present. In all other cases, the operon is OFF, as should be expected according to the underlying regulatory mechanisms. This shows that even a simple Boolean model as that described by Eqs. (2.2) is capable of capturing the main qualitative properties of the lac operon regulation.

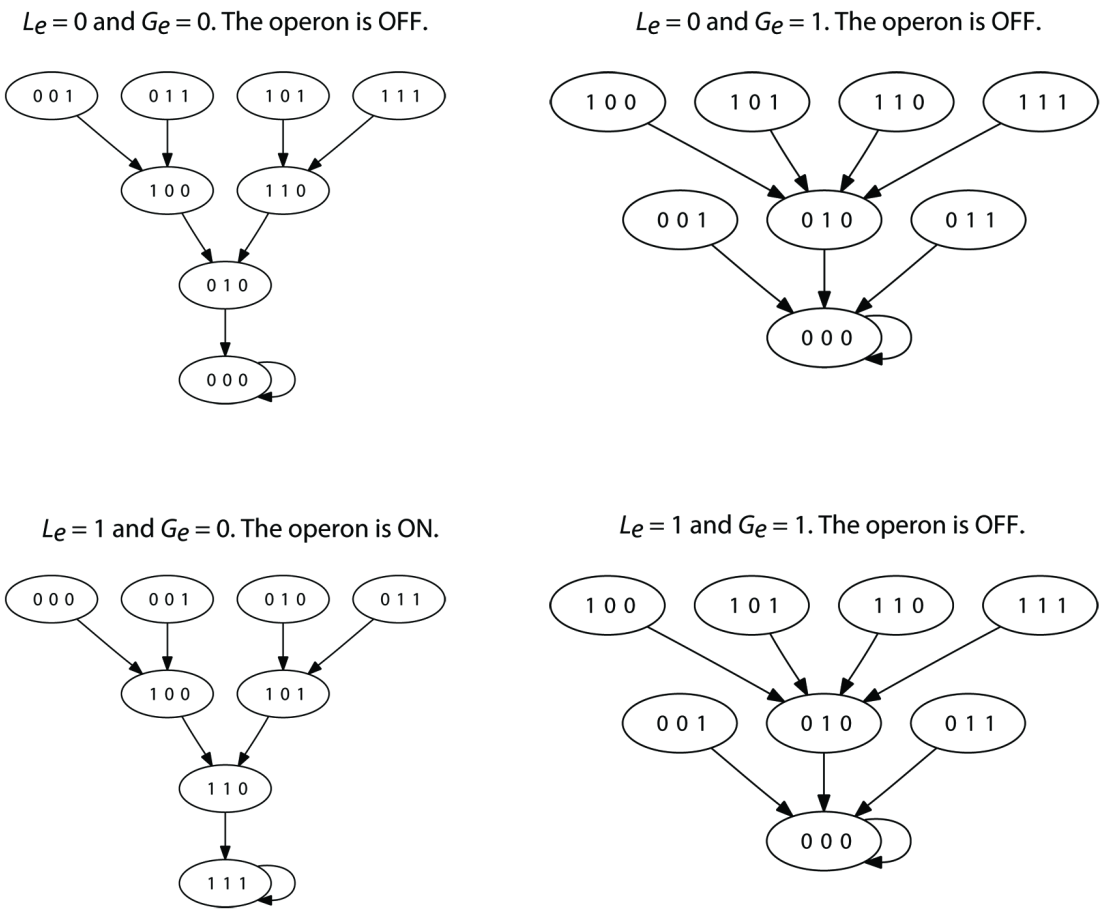


Figure 5: The state space transition diagram of triples (M, E, L) for the Boolean model of the lac operon for the four possible combinations of parameter values. When external glucose is present, the operon is OFF. When external glucose is unavailable and external lactose is present, the operon is ON. Graphs obtained using DVD (http://dvd.vbi.vt.edu). Reproduced from *CBE-Life Sciences Education*, Vol. 9, Fall 2010, p.236 [14].

In the module we go on to introduce two more examples of Boolean models of the lac operon.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

The first, containing five variables, is shown to be flawed, as it has a fixed point that is not biologically feasible. The second one, from Stigler and Veliz-Cuba [20], has nine variables and two parameters and explicitly includes the CAP-cAMP positive feedback loop. The analysis of this refined model is done at the end of Section 3 of the module. In Section 4 of the module the reader is guided to using Groebner Bases to find its fixed points.

Due to the simplicity of the minimal Boolean model, for each set of parameter values L_e and G_e there are only 8 possible states of the dynamical system (M, E, L) and the transition diagrams in Figure 5 can easily be computed explicitly. Mathematically a point $(p_1, p_2, \ldots, p_n)$ is a *fixed point* for the set of transition functions $f_1, f_2, \ldots, f_n$ when $p_j = f_j(p_1, p_2, \ldots, p_n)$ for $j = 1, 2, \ldots, n$. For such small systems as in the Boolean example discussed above, it is thus possible to check for fixed points by substitution of every possible state into the set of transition functions to see which inputs return the same outputs. For larger systems, however, this is as impractical as an attempt to compute the state space transition diagram. As an example, the Boolean model of T-cell receptor signaling [16] contains 94 nodes and, thus, 2^{94} different states. Clearly, as the number of states increases exponentially with the number of variables, a more computationally efficient approach than simple enumeration is needed. The mathematical concepts facilitating such an approach are described next. The material is appropriate for advanced abstract algebra courses at the undergraduate level.

2.4. Groebner Bases for Solving Systems of Polynomial Equations[‡]

The fixed points of a Boolean or an FDS model are solutions of the following system of equations determined by the transition functions:

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \ldots, x_n) \\ x_2 &= f_2(x_1, x_2, \ldots, x_n) \\ &\vdots \\ x_n &= f_n(x_1, x_2, \ldots, x_n) \end{aligned}$$

As already discussed, in case of FDS models the functions $f_1, f_2, \ldots, f_n$ are polynomials of the variables $x_1, x_2, \ldots, x_n$ and we seek the solutions of a polynomial system of equations. The method discussed in this section utilizes Groebner bases.

There is a rich and extensive body of theory that explores the relationships between algebra and geometry as related to Groebner bases. In our module, we try to select judiciously in order to keep the number of definitions and theorems to a minimum. A mathematics professor will find many options for developing ideas and introducing additional theory as course time permits or in case the material is used as an introduction to independent student research. Here we give an outline of an efficient introduction to Groebner Bases, and refer the reader to the full module for further details. We include some examples in order to provide some of the "flavor" of the subject. More details can be found in the monograph [5].

[‡]This part of the module is only appropriate for mathematics courses.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

2.4.1. Polynomial rings and ideals

This section of the module begins by defining the polynomial ring $K[x_1, x_2, \ldots, x_n]$ of all polynomials in $x_1, x_2, \ldots, x_n$ over a field K and defining an ideal. Recall that an ideal is a subset of the ring that contains 0, is closed under addition and is closed under multiplication by any polynomial in the ring.[§]

Example 1. *Suppose K is the field of real numbers, and let $I = \{f \in K[x_1, x_2] : f(2, -2) = 0\}$. That is, the set I consists of all polynomials in the ring that have $(x_1, x_2) = (2, -2)$ as a zero; $x_1^3 - x_1x_2^2$ and $4x_1 - 4x_1x_2 + 3x_2^3$ are examples of elements of I. It is easy to check that I is an ideal.*

It is convenient to define a particular ideal by specifying a generating set, as follows:

Definition 2. *Let $f_1, \ldots, f_s$ be polynomials in $K[x_1, x_2, \ldots, x_n]$. We set $\langle f_1, \ldots, f_s\rangle = \{\sum_{i=1}^{s} h_i f_i : h_i \in K[x_1, x_2, \ldots, x_n]\}$. This is the* ideal generated by $f_1, \ldots, f_s$.

The ideal $\langle f_1, f_2, \ldots, f_s\rangle$ thus comprises linear combinations of $f_1, \ldots, f_s$, where the coefficients are polynomials from the whole ring $K[x_1, x_2, \ldots, x_n]$. It has a nice interpretation in terms of polynomial equations. Given $f_1, \ldots, f_s \in K[x_1, x_2, \ldots, x_n]$, we get the system of equations

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_s(x_1, x_2, \ldots, x_n) &= 0 \end{aligned}$$

From these equations, one can derive others using algebra. For example, $3f_1 + 2x_3^5 f_2 = 0$, which is a consequence of the original system. Note that $3f_1 + 2x_3^5 f_2$ is a member of the ideal $\langle f_1, f_2, \ldots, f_s\rangle$. Thus $\langle f_1, f_2, \ldots, f_s\rangle$ consists of all "polynomial consequences" of the equations $f_1 = f_2 = \cdots = f_s = 0$.

The following important result shows that in fact, *every* ideal of a polynomial ring over a field K can be expressed in terms of a finite generating set:

Hilbert Basis Theorem. *Every ideal in the polynomial ring $K[x_1, x_2, \ldots, x_n]$ is finitely generated.*

Definition 3. *If $I = \langle f_1, \ldots, f_s\rangle$, then we say that $f_1, \ldots, f_s$ is a* basis *for I.*

Notice that the term "basis" in this definition does not have the same implications here that it does in linear algebra. The polynomials in a basis need not be linearly independent, and while a basis constitutes a spanning set for the ideal, a basis need not be a minimal spanning set.

Example 4. *Consider $K[x, y]$ and the ideal $I = \langle f_1, f_2\rangle$, where $f_1 = x^2 + 2xy^2$ and $f_2 = xy + 2y^3 - 1$. Observe that $x = yf_1 - xf_2$, so $x \in I$. Then $\{x, f_1, f_2\}$ is also a basis of I.*

[§]Gallian's Contemporary Abstract Algebra text [7] can be used as an excellent resource for reviewing the definitions and basic properties of the fundamental concepts from abstract algebra used here, including rings and fields.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

2.4.2. Ideal Membership and Division Algorithms

Given a polynomial g and an ideal $I = \langle f \rangle$ in a ring $K[x]$ of single variable polynomials, how can we tell whether g is in I or not? Since every polynomial in I is divisible by f, to determine if g is in I, we only need to divide g by f using the division algorithm for $K[x]$. When g divided by f has zero remainder, then $g \in \langle f \rangle$.

For polynomials of two or more variables the process is not so simple. Suppose K is the field of real numbers, and $K[x, y]$ is the ring of polynomials of the two variables x and y. Let $I = \langle f_1, f_2 \rangle$, and let g be another polynomial. We want to know if g is in I.

A logical approach to answering the question would be to divide f_1 into g, obtaining a quotient q_1 and remainder r_1. If $r_1 = 0$, then $g = q_1 f_1 \in I$. If $r_1 \neq 0$, divide f_2 into r_1, obtaining quotient q_2 and remainder r_2. Now $g = q_1 f_1 + q_2 f_2 + r_2$. If $r_2 = 0$, $g \in I$ and if $r_2 \neq 0$, then $g \notin I$.

This method obviously requires a division algorithm for $K[x, y]$. However, division in $K[x, y]$ is more complicated than division in $K[x]$. In $K[x]$, as we carry out the algorithm for dividing g by f, in order to verify that we have found the right quotient q, we check to see if the remainder r has $\deg(r) < \deg(f)$. If not, we look for a better choice of q.

To see how this fails in $K[x, y]$, try using the usual polynomial division to divide $g = x^3 + xy + 2y^2$ by $f = x + y$. You may obtain quotient $q = x^2 - xy + y + y^2$ and remainder $r = y^2 - y^3$ or quotient $q = 2y - x$ and remainder $r = x^3 + x^2$. Or you may end up in a loop, dividing $x + y$ into the current remainder, only to obtain the results of a previous step. To compare a remainder with the divisor, a monomial ordering is necessary, and it happens that there is more than one way to impose an order on the monomials of $K[x, y]$ and on monomials of $K[x_1, x_2, \ldots, x_n]$.

Definition 5. *A monomial order $<$ on the set of monomials in $K[x_1, x_2, \ldots, x_n]$ must have the following properties:*

1. *The ordering respects multiplication. That is, if $u < v$ and w is another monomial, then $uw < vw$.*

2. *If u and v are monomials that differ only in their coefficient, then u and v are equivalent under the ordering.*

3. *The constant monomial 1 is the smallest.*

In $K[x]$, there is only one monomial order: $1 < x < x^2 < x^3 < \ldots$. In $K[x, y, z]$, there are many monomial orders and there are many "lexicographic" orders which depend on the order of the indeterminates. We will use "lex order" determined by $x > y > z$. Then $x^3 > x^2 > x > 1$ and, for example, $x^3y^2 > x^3y > x^2y > xy^2 > xz > z > 1$.

Once a monomial ordering is specified, the terms of a polynomial can be ordered in an unambiguous way. For example, let $f = 4xy^2z + 4z^2 - 5x^3 + 7x^2z^2 \in K[x, y, x]$. Then with respect to our chosen lex order, putting the terms of f in decreasing order gives $f = -5x^3 + 7x^2z^2 + 4xy^2z + 4z^2$. (Powers of x dominate.)

In order to define Groebner Bases, we will need still more terminology.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

Definition 6. *Let f be a nonzero polynomial in $K[x_1, x_2, \ldots, x_n]$ and let $>$ be a monomial order. The* leading term *of f, denoted $\mathrm{LT}(f)$, is the term with nonzero coefficient which is greatest (with respect to $>$) of the terms of f. The* leading monomial *of f, $\mathrm{LM}(f)$ is the leading term with the coefficient set to 1. The* multidegree *of f, denoted $\mathrm{multideg}(f)$, is an n-vector of non-negative integers which lists the exponents of the indeterminates as they appear in the leading term.*

Example 7. *Let $f = 4xy^2z + 7x^2z^2 - 5x^3y + 4z^2$. With respect to lex order, $\mathrm{LT}(f) = -5x^3y$, $\mathrm{LM}(f) = x^3y$, and $\mathrm{multideg}(f) = (3, 1, 0)$.*

For our purposes, it is important to know that a division algorithm in $K[x_1, x_2, \ldots, x_n]$ exists but it is not important to develop facility with it, thanks to software. With this in mind, we give a statement of the algorithm because it reveals a key role played by the leading terms of the divisors.

Division Algorithm in $K[x_1, x_2, \ldots, x_n]$. Fix a monomial order and let $F = (f_1, \ldots, f_s)$ be an ordered s-tuple of polynomials in $K[x_1, x_2, \ldots, x_n]$. Then every $g \in K[x_1, x_2, \ldots, x_n]$ can be written as $g = a_1f_1 + \cdots + a_sf_s + r$, where $a_i, r \in K[x_1, x_2, \ldots, x_n]$ and either $r = 0$ or r is a linear combination, with coefficients in K, of monomials, none of which is divisible by any of $\mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s)$. We will call r a remainder of g on division by F. Furthermore, if $a_if_i \neq 0$, then we have $\mathrm{multideg}(g) \geq \mathrm{multideg}(a_if_i)$.

In contrast to the division algorithm in $K[x]$, here the quotients a_i and the remainder r need not be unique, and in fact they may depend on the order of the divisor polynomials. More examples and discussion of the Division Algorithm can be found in [5]. Here we move toward the goal of defining a Groebner Basis and showing its use for solving systems of polynomial equations.

2.4.3. Groebner Bases Defined

There are several (equivalent) ways to define Groebner Bases. Our preferred definition requires just a few more preliminaries.

Definition 8. *Let I be a non-zero ideal in $K[x_1, x_2, \ldots, x_n]$. Define $\langle \mathrm{LT}(I) \rangle$ to be the ideal generated by the set of leading terms of elements of I.*

Note that if $I = \langle f_1, \ldots, f_s \rangle$, then $\langle \mathrm{LT}(f_1), \mathrm{LT}(f_2), \ldots, \mathrm{LT}(f_s) \rangle \subset \langle \mathrm{LT}(I) \rangle$, but the two ideals need not be equal, as the following example shows:

Example 9. *Let $I = \langle f_1, f_2 \rangle$, where $f_1 = x^2 + 2xy^2$ and $f_2 = xy + 2y^3 - 1$. Using lex order on $K[x, y]$, we have $\mathrm{LT}(f_1) = x^2$, $\mathrm{LT}(f_2) = xy$. Since $y(x^2 + 2xy^2) - x(xy + 2y^3 - 1) = x$, we have $x \in I$. But $\mathrm{LT}(x) = x$ cannot be written as a linear combination of $\mathrm{LT}(f_1)$ and $\mathrm{LT}(f_2)$, so x is not in $\langle \mathrm{LT}(f_1), \mathrm{LT}(f_2) \rangle$.*

We are now ready to define a Groebner basis.

Definition 10. *Under a fixed monomial ordering, a subset $G = \{g_1, g_2, \ldots, g_t\}$ of an ideal I is said to be a Groebner basis if*

$$\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \mathrm{LT}(g_2), \ldots, \mathrm{LT}(g_t) \rangle$$

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

This means that a subset $G = \{g_1, g_2, \ldots, g_t\}$ of an ideal I is a Groebner basis if and only if the leading term of any element of I is divisible by one of the $\mathrm{LT}(g_i)$.

Groebner bases have many attractive properties. Among them:

- It can be shown that if G is a Groebner basis of I, then multivariate division of any polynomial in $K[x_1, x_2, \ldots, x_n]$ by G gives a unique remainder, and multivariate division of any polynomial in the ideal I by G has remainder of 0.
- It can be shown that every non-zero ideal I has a Groebner basis, and the basis can be effectively obtained for any ideal starting with a generating subset.

There are algorithms for finding Groebner bases and these are implemented in a variety of software[¶]. In our module we accept the algorithms without examination. Even to check that a given generating set is or is not a Groebner basis requires one to master additional definitions and theoretical machinery. In pursuit of our main goal, we rely on software to do the heavy computational work. With an admittedly limited grasp of the technical processes, we can nevertheless learn how to solve systems of polynomial equations. We need one more refinement.

Definition 11. *A reduced Groebner basis G for an ideal I satisfies:*

1. *For each g in G, the coefficient of* $\mathrm{LT}(g)$ *is* 1
2. *The set $\{\mathrm{LT}(g) : g \in G\}$ is a minimal spanning set of $\langle \mathrm{LT}(I) \rangle$ – nothing can be removed without losing its ability to span the ideal.*
3. *No trailing term of any g in G lies in $\langle \mathrm{LT}(I) \rangle$.*

Reduced Groebner bases have the very nice property that any nonzero polynomial ideal, with any given monomial ordering, has a unique reduced Groebner basis. See [5] for a proof.

2.4.4. Solving systems of polynomial equations.

Recall from linear algebra that every matrix can be put in reduced row echelon form in a unique way. This can be viewed as a special case of the uniqueness of reduced Groebner bases, as we will see in the following example.

Example 12. *Consider the system of linear equations*

$$2x + 3y + 4z = 5$$
$$3x + 4y + 5z = 2$$

[¶]We have found that SAGE (http://www.sagemath.org) suits our purposes very well. It is free and fast, and requires only a limited vocabulary of its command language in order to meet our needs. The module includes a brief tutorial on using SAGE for determining Groebner bases.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

Gaussian elimination gives the equivalent system

$$x - z + 14 = 0$$
$$y + 2z - 11 = 0$$

The first system of equations determines the polynomial ideal $I = \langle f_1, f_2 \rangle$ *where* $f_1 = 2x + 3y + 4z - 5$ *and* $f_2 = 3x + 4y + 5z - 2$. *A Groebner basis (under lex order, obtained from software) for this ideal is*

$$g_1 = x - z + 14,\ g_2 = y + 2z - 11.$$

Notice that these are the polynomials resulting from the Gaussian elimination. Here, as for other systems of polynomial equations, the obvious advantage of the Groebner basis is that it makes it easy to describe the solution set of the system of equations.

Example 13. *Now try to solve the following, nonlinear system of equations:*

$$x^2 + y^2 + z^2 - 1 = 0$$
$$x^2 + y^2 + z^2 - 2x = 0$$
$$2x - 3y - z = 0.$$

Gaussian elimination is not adequate for this task.

To see how a Groebner basis can help, let $J = \langle f_1, f_2, f_3 \rangle$ *with*

$$f_1 = x^2 + y^2 + z^2 - 1$$
$$f_2 = x^2 + y^2 + z^2 - 2x$$
$$f_3 = 2x - 3y - z.$$

A Groebner basis for J, *obtained from software, is* $\{g_1, g_2, g_3\}$ *where*

$$g_1 = y + \frac{1}{3}z - \frac{1}{3}$$
$$g_2 = z^2 - \frac{1}{5}z - \frac{23}{40}$$
$$g_3 = x - \frac{1}{2}.$$

Starting with $g_3 = 0$ and substituting back into $g_2 = 0$, then $g_1 = 0$, gives the solution set. This example illustrates the general result that finding a Groebner basis for an ideal with respect to the lex order simplifies the form of the equations considerably. A Groebner basis under the lex ordering is a triangular system, where the polynomial with fewest variables can be solved. Then solutions are back-substituted from one equation to the next until all solutions are produced.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

3. Discussion

In this paper we present an educational module developed in response to the need to incorporate Boolean and FDS models into the undergraduate mathematics and biology curricula. The module is part of a collection of mathematical biology modules based on modern molecular biology and modern discrete mathematics that is currently under development at Sweet Briar College and Western Michigan University (see [14]). The module illustrates the basic steps of developing, validating, and refining an FDS model, using the lactose utilization network of *E. coli* as an example.

Boolean models are important from an educational perspective since relatively few mathematics prerequisites are needed for the construction of such models. At the introductory level, the construction of the model is essentially a translation of the systems interactions represented by biology "cartoons" into directed graphs, followed by subsequent translation into logical expressions. This makes Boolean models ideal for an early (below-calculus level) introduction to mathematical models, removing the need for calculus or other mathematical prerequisites. For mathematics students, such models can be introduced in low-level finite mathematics or discrete mathematics courses and used to provide an early demonstration of the important link between mathematics and biology.

At the more advanced level, Boolean models and finite dynamical system models can provide an introduction to some serious theoretical mathematical questions and can also be used as a straightforward path to questions appropriate for student research projects. One such question included in the module is that of determining the fixed points for the FDS. A computationally efficient method uses (in the context of the question as it pertains to FDS) the theory of Groebner bases for ideals over finite fields. The algorithm is essentially a generalization of the well-known process of Gaussian elimination for solving systems of linear equations. In the case of FDS, the fixed points are found as solutions of systems of equations in which the functions are polynomials over a finite field. The mathematical theory behind the algorithm is appropriate for undergraduate mathematics courses in abstract algebra. For such courses, the original modeling problem presents a nice initial justification for the need of introducing Groebner bases and examining their properties.

The actual implementation of the algorithm requires the use of specialized software (as even the verification that a given set of polynomials is a Groebner basis for an ideal is labor intensive and unfeasible to do by hand). Although there are a number of open-source computational algebra systems that compute Groebner bases (e.g. Macauley 2, MAGMA, CoCoA, SINGULAR, and others), most such systems require download and installation. In the module, we use a web-based SAGE interface for computations that requires only a few straightforward commands and, thus, its use requires virtually no learning curve. The students can then focus on the output and its interpretation with regard to the question of solving polynomial systems of equations.

For mathematics students, we see the use of the module to be three-fold. On one hand, it introduces them to a new modeling approach that is currently not taught in any of the mainstream undergraduate mathematics courses. On the other hand, FDS models provide a link to important mathematical theory and results in abstract algebra and algebraic geometry that can be further pursued in advanced-level courses or as independent research projects. Some "natural" questions

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

arising from examining FDS (such as the question of determining the existence and number of limit cycles and their lengths for an FDS) are highly non-trivial, leading to an active area of research and some open questions. References to selected recent relevant papers are included in the module and can serve as starting points for the motivated student. Finally, the module provides evidence for the important connections between modern biology and modern mathematics, thus highlighting systems biology as a career path for mathematics majors.

For biology students the module provides an introduction to mathematical modeling without the need for a prior calculus background. Our experience indicates that the just-in-time approach for developing the necessary mathematical concepts as a way to formalize specific aspects of the biology works well for Boolean models. It allows students to focus on the logical links that determine the variable interactions instead of on the detailed kinetics needed for the DE models. Concurrent or subsequent introduction to DE models in calculus or differential equations courses will allow students to reinforce the conceptual framework, will further improve their mathematical sophistication, and will solidify the retention of basic ideas.

The detailed teaching module with student exercises is available from the authors by request. Parts of the module have already been tested in the courses Genetics (biology), Linear Algebra (mathematics), and Biomathematics (which students can take for either mathematics or biology credit) at Sweet Briar College. Future testing is planned in the mathematics course Topics in Abstract Algebra. The authors have also used the module material in the PREP faculty development workshops of the Mathematical Association of America "Mathematical Biology: Beyond Calculus" held June 13-18, 2010, and June 12-18, 2011 at Sweet Briar College, VA, and in a 90-min biomathematics workshop offered at the Symposium on Biomathematics and Ecology: Education and Research hosted by Illinois State University, September 4-5, 2010, Normal, IL.

Acknowledgements

The authors would like to thank Reinhard Laubenbacher from VBI for turning our attention to the educational potential of discrete models for both biology and mathematics students and for his generous assistance in identifying appropriate recent publications of discrete models describing gene regulation. We also gratefully acknowledge the support of the National Science Foundation under the Division of Undergraduate Education award 0737467.

References

[1] *BIO2010: Transforming undergraduate education for future research biologists*. The National Academies Press, Washington, DC, 2003.

[2] C. Neuhauser. Calculus for biology and medicine, 2nd ed. Prentice Hall, Upper Saddle River, NJ, 2003.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

[3] F. Adler. Modeling the dynamics of life: Calculus and Probability for life scientists, 2nd ed. Thompson, Belmont, CA, 2005.

[4] J. Cohen. *Mathematics is biology's next microscope, only better; Biology is mathematics' next physics, only better*. PLoS Biol., 2 (2004), No. 12, e439. doi:10.1371/journal.pbio.0020439.

[5] D. Cox, J. Little, D. O'Shea. Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra, 3rd edition. Springer, New York, 2007.

[6] F. Jacob, J. Monod. *Genetic regulatory mechanisms in the synthesis of proteins*. J. Mol. Biol., 3 (1961), 318–356.

[7] J. Gallian. Contemporary abstract algebra, 6th edition. New York, Houghton Mifflin Company, 2006.

[8] S. Kauffman. *Metabolic stability and epigenesis in randomly connected nets*. J. Theor. Biol., 22 (1969), 437–467.

[9] R. Laubenbacher, B. Sturmfels. *Computer algebra in systems biology*. The American Mathematical Monthly, 116 (2009), No. 10, 882–891.

[10] Mathematical Biosciences Institute (MBI) Current Topic Workshop: Mathematical Developments Arising from Biology. November 8-10, 2009.

[11] A New Biology for the 21st century. Committee on a New Biology for the 21st century: Ensuring the United States leads the coming biology revolution. The National Academies Press, Washington, DC, 2009.

[12] R. Robeva, J. Kirkwood, R. Davies, M. Johnson, L. Farhy, B. Kovatchev, M. Straume. An invitation to biomathematics. Academic Press, Burlington, MA, 2008.

[13] R. Robeva, R. Laubenbacher. *Mathematical biology education: beyond calculus*. Science, 325 (2009), 542–543.

[14] R. Robeva, R. Davies, T. Hodge, and A. Enyedy. *Mathematical biology modules based on modern molecular biology and modern discrete mathematics*. CBE - LSE, 9 (2010), Fall, 227–240.

[15] R. Robeva. *Systems Biology - old concepts, new science, new challenges*. Front. Psychiatry, 1 (2010), No. 1, 1–2. doi:10.3389/fpsyt.2010.00001

[16] J. Saez-Rodriguez, L. Simeoni, J. Lindquist, R. Hemenway, U. Bommhardt, B. Arndt, U. Haus, R. Weismantel, E. Gilles, S. Klamt, B. Schraven. *A logical model provides insights into T cell receptor signaling*, PLoS Comp. Biol., 3 (2007), No. 8, e163. doi:10.1371/journal.pcbi.0030163.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press

[17] A. Samal, S. Jain. *The regulatory network of E. coli metabolism as a Boolean dynamical system exhibits both homeostasis and flexibility of response*. BMC Systems Biology, 2, (2008), Article 21. doi:10.1186/1752-0509-2-21

[18] M. Santillan, M. Mackey, E. Zeron. *Origin of bistability in the lac operon*. Biophys. J., 92 (2007), 3830–3842.

[19] M. Santillan, M. Mackey. *Quantitative approaches to the study of bistability in the lac operon of Escherichia coli*. J. R. Soc. Interface, 5 (2008), S29-S39

[20] B. Stigler, A. Veliz-Cuba. *Network topology as a driver of bistability in the lac operon*. Available at http://arxiv.org/abs/0807.3995.

[21] N. Yildrim, M. Mackey. *Feedback regulation in the lactose operon: a mathematical modeling study and comparison with experimental data*. Biophys. J., 84 (2003), 2841–2851.

[22] R. Zhang, M. Shah, J. Yang, S. Nyland, X. Liu, J. Yun, R. Albert, T. Loughran. *Network model of survival signaling in large granular lymphocyte leukemia*. Proc. Natl. Acad. Sci. USA, 105 (2008), 16308–16313.

https://doi.org/10.1051/mmnp/20116603 Published online by Cambridge University Press