A Computer Algebra Approach To Biological Systems

Reinhard Laubenbacher
Virginia Bioinformatics Institute
1880 Pratt Drive
Blacksburg, VA 24061, USA
reinhard@vbi.vt.edu

ABSTRACT

This paper focuses on dynamic networks over finite fields and applications to the modeling and analysis of biological networks using tools from computer algebra, in particular gene regulatory networks, and agent-based simulations of processes in computational immunology.

Categories and Subject Descriptors: I.6.1 [Simulation Theory]: Types of Simulation

General Terms: Algorithms.

Keywords: computer algebra, biological system, dynamic network.

1. INTRODUCTION

Increasingly, research in many areas of biology focuses on the study of whole systems, beyond the analysis of their parts. In many cases this has been made possible by recent technological advances that allow experimental measurements at the systems level. Several computational frameworks have been proposed to model and simulate biological systems based on such large-scale measurements. In [10] the concept of dynamic networks over finite fields was introduced. These can be described via polynomial functions. Several examples of biological networks are presented that can be analyzed in this framework, using computational and conceptual tools from computational algebra and algebraic geometry. In particular, models of gene regulatory networks, and the problem of reverse-engineering of dynamics in computational immunology can be treated with methods from computer algebra.

2. DYNAMIC NETWORKS

We first recall a special case of a definition from [9].

Definition 1. An *n-dimensional dynamic network over a finite field k* is a function $f = (f_1, \ldots, f_n) : k^n \longrightarrow k^n$, with dynamics given by iteration.

The concept of a dynamic network is very general and encompasses several important classes of time-discrete dynamical systems, including cellular automata, Boolean networks, and sequential dynamical systems [1]. Sequential dynamical systems have been studied extensively [3], [6], and [7], and form one possible mathematical environment for the analysis of agent-based simulations.

EXAMPLE 1. Let $\mathcal{C}$ be a one-dimensional cellular automaton with n nodes, and Boolean functions $f_1, \ldots, f_n$. Let k be the field with two elements. Each Boolean function is built using the three logical operations $\wedge, \vee$ and $\neg$. These can be represented as polynomial functions over k as follows: $x \wedge y = xy, x \vee y = x + y + xy, \neg x = x + 1$. Using these representations, any Boolean function can be represented as a polynomial function over k, and $\mathcal{C}$ can be represented as a dynamic network with polynomial coordinate functions f_i.

In general, the connection between dynamic networks over a finite field k and polynomial algebra is based on the well-known fact that every function $k^n \longrightarrow k$ can be represented by polynomials in $k[x_1, \ldots, x_n]$. From this point of view, a dynamic network over a finite field is simply a transformation of affine n-space over k.

3. GENE NETWORKS

Reverse-engineering of dynamic networks from experimental data is an important problem in several areas, in particular in computational biology. The goal is to identify a dynamic network that interpolates one or more given time series of data points and which satisfies specified additional criteria. If the time series and the additional criteria do not identify the network uniquely, then one commonly chooses a network that is optimal in a specified way. An important instance of this problem is the reverse-engineering of gene regulatory networks from time courses of DNA microarray data and, possibly, other experimental data. We briefly discuss here an algorithm for this problem, as a first application of the algebraic viewpoint on dynamic networks. Details can be found in [10].

Microarray data measure concentrations of mRNA in cell extracts, and indicate activity levels of the corresponding genes. If we consider n genes, then a time point corresponds to an n-tuple of real numbers. There are several methods to discretize such data to give vectors over a suitable finite field k. (See [5] for results on the choice of k.) We model the regulatory network as a dynamic network

$$f = (f_1, \ldots, f_n) : k^n \longrightarrow k^n,$$

where the f_i are polynomial functions in n variables. Our

WOODSTOCK '97 El Paso, Texas USA

problem then is to determine the f_i, based on the given time course of data,using prior biological knowledge about the network to be modeled. Generally, the given data greatly underdetermine the system, so a variety of approaches have been suggested to choose a particular network. Our approach is to first compute all possible such functions and then choose the unique function that is reduced with respect to the ideal of all functions that vanish on the time series. The problem can therefore be solved in a way similar to finding all solutions to a nonhomogeneous system of linear equations: first find all solutions of the corresponding homogeneous system, then find a particular solution to the nonhomogeneous system. It is easily seen that each such f_i is of the form $h + g$, where h is a particular solution to the problem (computable using, e.g., a Chinese Remainder Theorem algorithm), and g is in the ideal of functions vanishing on the time series. This ideal can be computed by taking the intersection of the maximal ideals corresponding to each of the points.

It is very interesting to note that a similar problem appears in algebraic statistics, related to the design of experiments. There the goal is to find all polynomial models for a factorial design. For a detailed description see [11]. A very efficient algorithm for this problem is outlined in [12], using the Buchberger-Möller algorithm. It computes the ideal of functions vanishing on the collection of points and a particular solution at the same time. For our purposes it is more useful to compute the two separately, since the particular solutions vary for each node of the network.

Another approach to reverse-engineering of gene regulatory networks utilizes causal Bayesian network methods. In the recent paper [4] a very promising computer algebra approach to Bayesian networks has been developed, which is potentially useful for reverse-engineering problems.

4. DYNAMICS

Another problem that is accessible with the tools of computational algebra is the reverse-engineering of dynamics by modifying the local update functions of a dynamic network. We illustrate this approach with an example. For details see [8].

EXAMPLE 2. Let $X = \{-1, 0, 1\}$, and suppose we are given the following system $f = (f_1, \ldots, f_5)$ with five nodes:

$$f_1 = -x_2 + x_4 + 1, f_2 = x_2 + x_4,$$
$$f_3 = -x_2 - x_4 + x_5 - 1, f_4 = f_3, f_5 = x_4 + x_5 - 1.$$

Beginning with the state $\mathbf{s}_1 = (1, 1, 1, 1, 1)$, f produces the time series $(1, -1, 1, 1, 1), (0, 0, 0, 0, 1), (1, 0, 0, 0, 0)$. Suppose now that we want to modify f to obtain a new network $g = (g_1, \ldots, g_5)$ so that from the beginning state $(1, 1, 1, 1, 1)$ we reach the state $(1, 0, 1, -1, 0)$ in three steps, rather than the state $(1, 0, 0, 0, 0)$. Furthermore, we require that g should contain exactly the same variables as f. Using computer algebra algorithms similar to those in the previous section, together with elimination of variables, we obtain all possible such networks and can choose a particular one based on given criteria.

We briefly describe two applications. The first one is an *in vitro* virus competition model, explored by Karen Duca [2]. Canine liver cells in the center of a petri dish were infected with two different strains of influenza virus. Subsequent spread of the infection throughout the dish showed a surprising compartmentalization of infection with the two different strains, rather than the expected outcome of widespread dual infection. A simple “marbles-in-boxes” stochastic simulation of the experiments showed similar compartmentalization behavior, throwing doubt on an explanation of the pattern via biochemical mechanisms. In an ongoing project, we have constructed deterministic versions of the stochastic simulation, which can be analyzed as a dynamical system, described by polynomial functions. The goal is to explore the limit behavior of the system for different initial infection patterns and infection geometries.

The second project to which we are applying computational algebra methods is an agent-based simulation of immune system response to infection with the Epstein-Barr virus. The goal of the project is to develop mathematical tools to systematically reverse-engineer desired infection outcomes, e.g., complete viral clearance or entrainment of a more robust adaptive immune response, by modifying select rules of agents, locations, or populations of agents, or adding new agents representing, e.g., drugs. At present we are developing a mathematical specification for parts of the simulation that allows the application of the reverse-engineering methods described earlier.

Acknowledgements. The work described here is a joint effort of the Discrete Mathematics Group at the Virginia Bioinformatics Institute, together with several collaborators, in particular Karen Duca, Edward Green, Abdul Jarrah, Pedro Mendes, Michael Stillman, and Bernd Sturmfels. This research has been partially supported by NSF Grants DMS-0138323 and DMS-0083595.

5. REFERENCES

[1] C. Barrett, H. Mortveit, and C. Reidys. Elements of a theory of simulation II: Sequential dynamical systems. *Appl. Math. Comp.*, 107(2-3):121–136, 1999.

[2] K. Duca, A. Blumer, M. Locuta, and J. Fleming. MHV recombination in vitro and in silico. *Proc. Second Intl. Conf. Systems Biology*, pages 49–50, 2001.

[3] L. Garcia, A. Jarrah, , and R. Laubenbacher. Classification of finite dynamical systems. *preprint*.

[4] L. Garcia, M. Stillman, and B. Sturmfels. The algebraic geometry of bayesian networks. *Proc. of MEGA 2003*, in press.

[5] E. Green. On polynomial solutions to reverse-engineering problems. *preprint*.

[6] R. Laubenbacher and B. Pareigis. Equivalence relations on finite dynamical systems. *Adv. Appl. Math*, 26:237–251, 2001.

[7] R. Laubenbacher and B. Pareigis. Decomposition and simulation of sequential dynamical systems. *Adv. Appl. Math.*, in press.

[8] R. Laubenbacher, J. Shah, and B. Stigler. Simulation of polynomial systems. *in J.G. Anderson and M. Katzper (eds.) Simulation in the Health and Medical Sciences*, San Diego, in press.

[9] R. Laubenbacher and B. Stigler. Dynamic networks. *preprint*.

[10] R. Laubenbacher and B. Stigler. Polynomial models for biochemical networks. *preprint*.

[11] G. Pistone, E. Riccomagno, and H. Wynn. 2000.

[12] L. Robbiano. Groebner bases and statistics. *in Groebner bases and applications*, 1998.