Barycentric Algebras and Gene Expression

Anna B. Romanowska[1] and Jonathan D.H. Smith[2]

[1] Faculty of Mathematics and Information Sciences
Warsaw University of Technology
00-661 Warsaw, Poland
aroman@alpha.mini.pw.edu.pl
[2] Department of Mathematics
Iowa State University
Ames, Iowa, 50011, USA
jdhsmith@math.iastate.edu

Abstract. Barycentric algebras have seen widespread application in the modeling of convex sets, semilattices, and quantum mechanics. Recently, they were developed further to encompass Boolean logic and if-then-else algebras. This paper discusses an application of barycentric algebras to systems biology. Here, they provide a calculus for the conversion from simplified Boolean models of gene transcription to fuzzy models that give a more realistic tracking of the biochemistry. Indeed, it appears that logic gates experimentally observed in cells actually follow the barycentric algebra format.

1 Introduction

Barycentric algebras (as defined in §2.3 below) are universal algebras used for modeling convex sets, semilattices, geometry, hierarchical statistical mechanics, and quantum mechanics [5,6,12,13,14,15,16,17,18]. Recently [17], they have been developed further (as *abstract barycentric algebras*) by use of the LΠ-algebras of fuzzy logic [3,10,11], incorporating B-sets [2,20,21] and if-then-else algebras [8,9]. The aim of the current paper is to show how the calculus of barycentric algebras may be used in systems biology, to provide a virtually automatic translation from simplified Boolean models of gene expression to continuous, fuzzy logic models that give a much more realistic picture of the biochemical processes involved. Experimentally observed logic gates in cells do not follow the pattern directly suggested by standard Boolean models, but their features concur exactly with the models obtained using the barycentric algebra approach [19, Fig. 3b].

The bulk of the paper comprises two parts. Section 2 gives a direct account of the algebra required. For readers who may be unfamiliar with universal algebra, §2.2 discusses concatenations of binary operations. The two key incarnations of abstract barycentric algebras, namely classic "fuzzy" barycentric algebras and their crisp Boolean counterparts, are described in §2.3.

Section 3 then focusses on the systems biology. For readers unfamiliar with molecular biology, §3.1 gives a brief account of the way cells use transcription

V. Di Gesù, S.K. Pal, and A. Petrosino (Eds.): WILF 2009, LNAI 5571, pp. 20–27, 2009.

factors to respond to signals and regulate gene expression. Subsequent paragraphs formulate the crisp and fuzzy models of gene regulation in the language of barycentric algebras. Once this formulation is established, Eqn. (12) provides the automatic conversion from Boolean models to fuzzy models. In §3.4, the conversion process is illustrated by the example of the AND gate. The final paragraph explains how fuzzy logic gates that have been observed experimentally in cells actually follow the barycentric algebra format.

2 Algebra

2.1 Operations on Real Numbers and Binary Digits

Although the algebra of real numbers is traditionally performed in terms of field operations such as the addition $p+q$ and product pq of real numbers p and q, the algebra discussed in this paper requires different operations, which specialize to more familiar Boolean operations on the subset $\{0,1\}$ of the reals. In fact, this specialization will also work in any field. In particular, it works if the set $\{0,1\}$ of binary digits is interpreted as the two-element (Galois) field GF(2) or field of integers modulo 2.

For a real number p, define the *complementation* $p' = 1 - p$ specializing to the Boolean $\neg p$ or NOT p on the set $\{0.1\}$ of binary digits. Note that the complementation is *involutive*: $p'' = p$. For real numbers p and q, define the *product*

$$p \cdot q = pq \tag{1}$$

specializing to the Boolean $\wedge$ or AND on $\{0,1\}$. Define the *dual product*

$$p \circ q = p + q - pq \tag{2}$$

specializing to the Boolean $\vee$ or OR on $\{0,1\}$. Note that the dual product may be defined in terms of the product and complementation using *de Morgan's law* $p \circ q = (p'q')'$ or $(p \circ q)' = p'q'$. Define the *implication*

$$p \to q = \textbf{if } (p = 0) \textbf{ then } 1 \textbf{ else } q/p \tag{3}$$

specializing to the Boolean implication $p \to q = (\neg p) \vee q$ on $\{0,1\}$. Note that the implication (3) is always defined in any field, while the division q/p is not defined for $p = 0$.

2.2 Binary Operations

If x and y are elements of a real vector space, and p is a real number, it is convenient to define

$$xy\,\underline{p} = x(1-p) + yp = xp' + yp\,, \tag{4}$$

so that $\underline{p}$ is understood as a binary operation combining the arguments x and y. Schematically, the binary operation may be understood as a circuit element or

"black box" combining the inputs x and y to produce the output $xy\,\underline{p}$. For a second real number q and vector z, one may concatenate circuit elements to yield

$$xy\,\underline{p}\,z\,\underline{q} = (xp' + yp)z\underline{q} = xp'q' + ypq' + zq\,. \tag{5}$$

Alternatively one may concatenate the circuit elements to yield

$$x\,yz\,\underline{p}\;\underline{q} = x(yp' + zp)\underline{q} = xq' + yp'q + zpq\,. \tag{6}$$

Note that the parsing of the left hand sides of (5) and (6) is unique, even without the insertion of any brackets. This is one of the many advantages of the algebraic notation (4) for binary operations.

If p is an element of the closed real unit interval $I = [0,1] = \{p \mid 0 \leq p \leq 1\}$, then the operation (4) makes sense when the inputs x and y lie in some convex set C, for example some interval on the real line.

If p is a binary digit 0 or 1, the operation (4) makes sense when the inputs x and y are elements of some arbitrary set S, with

$$xy\,\underline{p} = \textbf{if } (p = 1) \textbf{ then } y \textbf{ else } x\,.$$

Recalling that the *truth value* $[\![P]\!]$ of a proposition P is 0 if P is false, and 1 if P is true, one obtains

$$xy\,\underline{[\![P]\!]} = \textbf{if } P \textbf{ then } y \textbf{ else } x\,. \tag{7}$$

Given an arbitrary set S, consider the convex set C of all finite probability distributions on S, identifying each element x of S with the distribution putting weight 1 on x. For elements x and y of S, and p in I, the operation (4) produces the distribution selecting y with probability p and x with probability p'.

2.3 Barycentric Algebras and If-Then-Else Algebras

An *abstract barycentric algebra* is defined as a set A that is equipped with binary operations $xy\,\underline{p}$ satisfying *idempotence* $xx\,\underline{p} = x$ for x in A, *skew-commutativity*

$$xy\,\underline{p} = yx\,\underline{p'} \tag{8}$$

for x, y in A, and *skew-associativity* $xy\,\underline{p}\,z\,\underline{q} = x\,yz\,\big(\underline{p \circ q \to q}\big)\,\underline{p \circ q}$ for x, y, z in A. There are two classical interpretations:

- Taking the operators p, q from the open real unit interval

$$I^\circ =]0,1[= \{p \mid 0 < p < 1\}$$

  yields a *barycentric algebra* [14,15,16].
- Taking Boolean operators p, q — elements of a Boolean ring such as GF(2) or its powers — yields *B-sets* [2,20], including certain types of *if-then-else algebras* [8,20].

Within abstract barycentric algebras, concatenations of the type (6) serve to implement the "AND" product (1) as

$$x\,xy\,\underline{p}\;\underline{q} = xy\,\underline{p\cdot q}\,, \tag{9}$$

while concatenations of the type (5) implement the dual "OR" product (2) as $xy\,\underline{p}\,y\,\underline{q} = xy\,\underline{p\circ q}$. Of course, skew-commutativity gives a direct implementation of the complement.

3 Systems Biology

3.1 Transcription Factors

Cells survive and develop by producing proteins in response to various signals that they receive. We describe a simplified model that will be adequate for the purposes of this paper. For fuller details, see [1,7]. A specific protein Y is produced by the expression of a corresponding part of the cell's DNA, namely the gene that encodes for protein Y. The gene is first *transcribed* to messenger RNA (mRNA). The mRNA is then *translated* into the required protein. The transcription process, synthesis of the mRNA, is facilitated by the enzyme RNA polymerase (RNAp). The enzyme binds itself to a regulatory region of the DNA, adjacent to the gene, known as the *promoter site*.

Signals that are of importance to a cell may be physical, such as a change in temperature, or chemical, such as the presence of a nutrient like glucose. Received signals switch proteins known as *transcription factors* from a dormant to an active state. Active transcription factors attach themselves to the promoter site, where they change the binding probability of the RNAp. If a transcription factor is an *activator*, it will increase the binding probability of the RNAp, thereby increasing the rate of transcription and protein production. Other transcription factors, known as *repressors*, have the opposite effect of inhibiting the expression of certain genes.

3.2 Crisp Logic

Fig. 1 displays sample dependencies of the transcription rate for production of a protein on the relative concentration x/k of an activator X. In the absence of the activator, the transcription rate assumes a *residual base level* v_0, in this case 0.1. (Often, a value of $v_0 = 0$ is appropriate.) If the activator is present in high concentrations, the transcription rate assumes a *maximal expression level* v_1, in this case 1.0.

The step function displays a crisp logical dependence of the transcription rate on the dimensionless ratio x/k between the actual concentration x of the activator X, and a critical threshold concentration level k. The transcription rate may be written as

$$v_0 v_1 \underline{\left[\!\left[\; 1 > \frac{k}{x} \;\right]\!\right]} \tag{10}$$

in the Boolean notation of (7). If the transcription factor X were a repressor rather than an activator, the corresponding transcription rate would appear in any of the forms

$$v_0 v_1 \underline{\left[\!\left[\; 1 > \frac{k}{x} \;\right]\!\right]}' = v_1 v_0 \underline{\left[\!\left[\; 1 > \frac{k}{x} \;\right]\!\right]} = v_0 v_1 \underline{\left[\!\left[\; 1 > \frac{x}{k} \;\right]\!\right]} \tag{11}$$

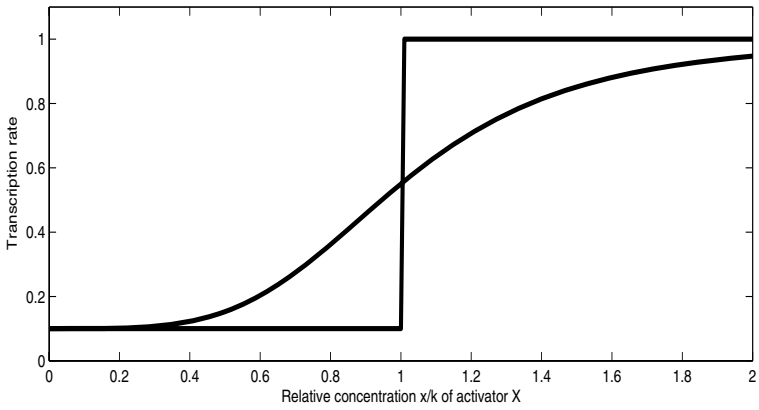

Fig. 1. Dependence of transcription rate on activator X

that are equivalent by virtue of the skew-commutativity (8) that implements complementation. (The last form neglects the improbable equality $x = k$.)

3.3 Fuzzy Logic

Because of its convenience, crisp logic has been used widely for the construction of network models in systems biology [4,22]. However, the curved graph of Fig. 1 displays a more realistic description of the dependence of transcription rates on the relative concentrations of transcription factors. One of the main theses of this paper is the way that the formalism of abstract barycentric algebras allows one to convert easily from the crisp functions of §3.2 to more realistic fuzzy functions. The crisp activator dependence (10) is replaced by the classic barycentric-algebraic expression

$$v_0 v_1 \underline{\left[\, 1 + \left(\frac{k}{x}\right)^n \,\right]^{-1}}$$

— a so-called *hill function* in the terminology of [1] — interpreted in the closed interval $[v_0, v_1]$, a convex set. Fig. 1 illustrates the case $n = 4$. For $n = 1$ (and $v_0 = 0$), the hill function implements Michaelis-Menten kinetics [1, A.7]. The case $n > 1$ corresponds to *cooperative reactions.* The crisp repressor dependencies (11) are replaced by either of the equivalent forms

$$v_1 v_0 \underline{\left[\, 1 + \left(\frac{k}{x}\right)^n \,\right]^{-1}} = v_0 v_1 \underline{\left[\, 1 + \left(\frac{x}{k}\right)^n \,\right]^{-1}}.$$

From these expressions, it is clear that the passage from crisp to fuzzy logic is formally achieved by the replacement

$$[\![1 > \lambda]\!] \qquad \longrightarrow \qquad [1 + \lambda^n]^{-1} \,. \tag{12}$$

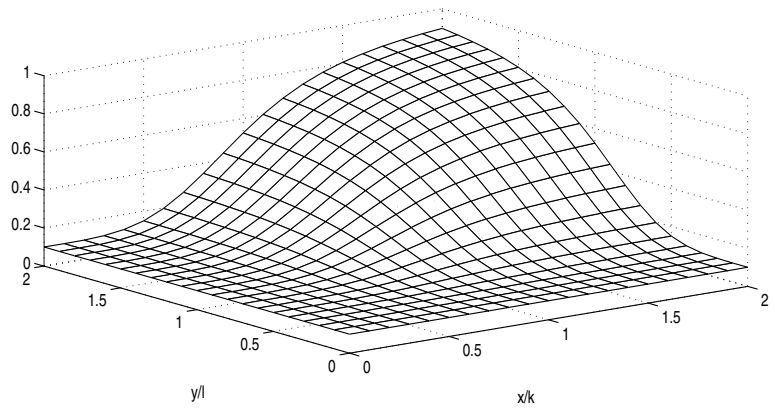

Fig. 2. Fuzzy AND gate

Here, the dimensionless quantity λ is taken as k/x for activators and x/k for repressors. The conversion process is illustrated in the following paragraph.

3.4 Logic Gates

Transcription rates may depend on logical combinations of different transcription factors. For example, the dependence

$$v_0 v_1 \underline{\left[\!\left[\, 1 > \frac{k}{x} \,\right]\!\right] \cdot \left[\!\left[\, 1 > \frac{l}{y} \,\right]\!\right]} \tag{13}$$

requires high concentrations of each of two transcription factors X and Y. The concentration x of X must exceed the critical threshold k; the concentration y of Y must exceed the critical threshold l. Using (9), the crisp logical expression (13) may be rewritten as the concatenation

$$v_0\, v_0 v_1 \underline{\left[\!\left[\, 1 > \frac{k}{x} \,\right]\!\right]}\ \underline{\left[\!\left[\, 1 > \frac{l}{y} \,\right]\!\right]}$$

which then translates to

$$v_0\, v_0 v_1 \underline{\left[1 + \left(\frac{k}{x}\right)^n\right]^{-1}}\ \underline{\left[1 + \left(\frac{l}{y}\right)^n\right]^{-1}} \tag{14}$$

under the replacement (12). With the previously used parameter values $v_0 = 0.1$, $v_1 = 1$, $k = 1$, $n = 4$, along with $l = 1$, this fuzzy AND gate is displayed in Fig. 2.

3.5 Some Experimental Observations

The fuzzy AND gate presented in (14) has the format $v_0\, v_0 v_1\, \underline{p}\ \underline{q}$ of (9). Here, the concatenated barycentric algebra operations have arguments (corresponding

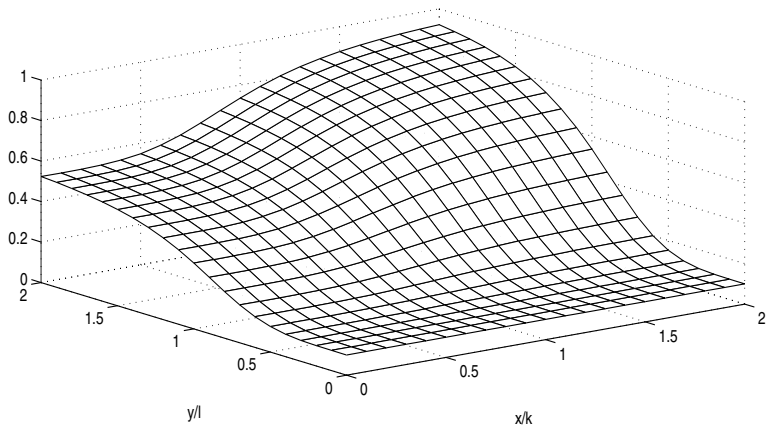

Fig. 3. Modified fuzzy AND gate

to transcription rates) that are repeated exactly. Exact repeats of this kind are improbable in biology. At first glance, it might appear that this would argue against the barycentric algebra approach. However, it turns out that real fuzzy AND gates as observed experimentally [19, Fig. 3b] actually have the format $v_0\, v_{01} v_1\, \underline{p}\ \underline{q}$ of (6) with distinct transcription rates $v_0 < v_{01} < v_1$, as illustrated in Fig. 3 using an intermediate expression level $v_{01} = 0.55$. It thus emerges that the barycentric algebra formulation gives a natural framework for the dependence of expression levels on transcription factor concentrations.

References

1. Alon, U.: An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC, Boca Raton (2006)
2. Bergman, G.M.: Actions of Boolean rings on sets. Algebra Universalis 28, 153–187 (1991)
3. Esteva, F., Godo, L., Montagna, F.: The LΠ and L$\Pi 1/2$ logics: two complete fuzzy systems joining Łukasiewicz and product logics. Arch. Math. Logic 40, 39–67 (2001)
4. Glass, L., Kauffman, S.A.: The logical analysis of continuous, non-linear biochemical control networks. J. Theor. Biol. 39, 103–129 (1973)
5. Gudder, S.P.: Convex structures and operational quantum mechanics. Comm. Math. Phys. 29, 249–264 (1973)
6. Ignatov, V.V.: Quasivarieties of convexors (in Russian). Izv. Vyssh. Uchebn. Zaved. Mat. 29, 12–14 (1985)
7. Klipp, E., Herwig, R., Kowald, C., Wierling, C., Lehrach, H.: Systems Biology in Practice. Wiley, New York (2005)
8. Manes, E.G.: Adas and the equational theory of if-then-else. Algebra Universalis 30, 373–394 (1993)
9. McCarthy, J.A.: A basis for a mathematical theory of computation. In: Braffort, P., Hirschberg, D. (eds.) Computer Programming and Formal Systems, pp. 33–70. North-Holland, Amsterdam (1993)

10. Montagna, F.: An algebraic approach to propositional fuzzy logic. Journal of Logic, Language and Information 9, 91–124 (2000)
11. Montagna, F.: Subreducts of MV-algebras with product and product residuation. Algebra Universalis 53, 109–137 (2005)
12. Neumann, W.D.: On the quasivariety of convex subsets of affine spaces. Arch. Math. 21, 11–16 (1970)
13. Ostermann, F., Schmidt, J.: Der baryzentrische Kalkül als axiomatische Grundlage der affinen Geometrie. J. Reine Angew. Math. 224, 44–57 (1966)
14. Romanowska, A.B., Smith, J.D.H.: Modal Theory. Heldermann, Berlin (1985)
15. Romanowska, A.B., Smith, J.D.H.: On the structure of barycentric algebras. Houston J. Math. 16, 431–448 (1990)
16. Romanowska, A.B., Smith, J.D.H.: Modes. World Scientific, Singapore (2002)
17. Orłowska, E., Romanowska, A.B., Smith, J.D.H.: Abstract barycentric algebras. Fund. Informaticae 81, 257–273 (2007)
18. Skornyakov, L.A.: Stochastic algebras. Izv. Vyssh. Uchebn. Zaved. Mat. 29, 3–11 (1985)
19. Setty, Y., Mayo, A.E., Surette, M.G., Alon, U.: Detailed map of a cis-regulatory input function. Proc. Nat. Acad. Sci. 100, 7702–7707 (2003)
20. Stokes, T.: Sets with B-action and linear algebra. Algebra Universalis 39, 31–43 (1998)
21. Stokes, T.: Radical classes of algebras with B-action. Algebra Universalis 40, 73–85 (1998)
22. Thieffry, D., Thomas, R.: Qualitative analysis of gene networks. In: Pac. Symp. Biocomput., vol. 3, pp. 77–88 (1998)