To appear in *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*, Boston, MA (2004).

A Model for Exploring Genetic Control of Artificial Amoebae

Barry Drennan[1] and Randall D. Beer[1,2]

[1]Department of Electrical Engineering and Computer Science
[2]Department of Biology
Case Western Reserve University
btd@po.cwru.edu, rxb9@po.cwru.edu

Abstract

We develop a computer simulation of several cellular processes seen in amoebae, including the production and regulation of proteins via a genome; the production, release, and destruction of diffusible chemicals; and regulated chemotaxis through a lattice environment facilitated by the interactions of proteins and diffusible chemicals. We also test this model by adapting biological situations to this model to evaluate its ability to model genetic networks and genetically regulated chemotaxis. The model will be used to simulate evolution in artificial amoebae to produce behavior seen in biological organisms such as *Dictyostelium discoideum*.

Introduction

The amoeba *Dictyostelium discoideum*, or soil-dwelling slime mold, is of particular interest to developmental biologists due to the unusual cooperative behavior it demonstrates. When its supply of food runs low, individual *Dictyostelium* amoebae collaborate in order to move to a new location where more food may be found. This process involves intercellular signaling (Söderbom and Loomis 1998) and cell differentiation (Brown and Firtel 1999), which are critical tools in the development of multicellular organisms.

Under normal circumstances, *Dictyostelium* exists as numerous individual amoebae. When food becomes scarce, these cells begin emitting pulses of a chemical signal, cyclic AMP (cAMP). Neighboring cells are attracted to this signal, and they exhibit chemotaxis along the upward gradient of cAMP. The cells thus aggregate into a small multicellular "slug" which subsequently propagates as one unit for some distance. During this time, the cells differentiate, forming two types of cells: a large number which will eventually become spores; and a small number that instead form a stalk to better disperse the spores, and then die.

Of interest to us is a model which can simulate these and other similar behaviors in the context of a genetic and protein framework. Savill and Hogeweg (1997) developed a model of *Dictyostelium* which exhibited its life cycle behaviors of aggregation, motility, and stalk formation. However, this model based its behavior on fixed numeric constants controlling various factors such as cell adhesivity and the release of cAMP, rather than placing them under genetic control. Dellaert and Beer (1994) developed a model for evolving morphology of an artificial multicellular organism on a grid, using Boolean networks for genetic regulation. Goel and Thompson devised models of bacteriophage self-assembly and operation (1988a), and protein synthesis (1988b) based on movable finite automata. The E-Cell project (Takahashi *et al.* 2003, Tomita *et al.* 1999) provides an open framework for modeling of cellular and biochemical processes, in an effort to create a highly detailed simulation of an entire cell.

A combination of some of these concepts is used here, with enhancements made to biological realism by modeling genetic regulation and protein production. In addition, various behaviors, such as chemical production, chemoattraction, and membrane channels are placed under the domain of these proteins rather than governing those behaviors with predefined constants. Our goal is to develop a model that can simulate arbitrary cell behavior, on the level of cell motility and intercellular signaling, and that, by providing a genome as an interface to these behaviors, permits the artificial evolution of these behaviors. We will ultimately use the behaviors observed in *Dictyostelium* as test cases for examining the evolutionary trajectory by which these behaviors are generated in the model.

Methods

The model developed herein combines simulations of the basic biological principles of cells, genes, and proteins (Kimball 2003) acting within a latticed environment with a genetic algorithm (GA) to evolve behavior of these cells. The model can also be used separately from the GA to test manually crafted genomes or to examine the behavior of previously evolved genomes.

Cells

The representation of a cell in this model is inspired by amoebae and other single-cell organisms which do not have a nominally-fixed shape. In other words, there are no structural features providing rigidity to the cell. Instead, a cell is free to expand, move, and contract throughout the environment, obtaining any contiguous shape at any time through means of minimizing its calculated free energy.

A simulated cell occupies any number of lattice points, or cellular automata (von Neumann 1966), in the three-dimensional cellular automaton (CA) grid. During each time step, a number of automata are selected for evaluation, to determine whether the occupation of that automaton by a new cell will create a lower free energy condition (Glazier and Graner 1992, 1993; Savill and Hogeweg 1997; Marée and Hogeweg 1999, 2002). For example, a cell has a nominal volume and surface area, and free energy is minimized by occupying or releasing automata to maintain that perimeter. Another factor affecting free energy is the satisfaction of membrane-bound receptors which are attractive or repulsive to other membrane-bound receptors or to diffusible chemicals.

Contrary to eukaryotes such as amoebae, however, the cells represented here do not have discrete organelles. Genetic material, proteins, and diffusible chemicals are considered to be available everywhere within the cell. While the inclusion of intracellular transport mechanisms may prove important for the evolution of more complex organisms, this is beyond the scope of this project.

Cells can also produce, destroy, absorb, or release any of a number of chemicals which diffuse throughout the environment. These diffusible agents can thus function as chemical messengers between cells.

Genome

In biological cells, DNA is the encoding chemical of genetic information, consisting of the four bases adenine, guanine, cytosine, and thymine (A, G, C, and T). Each base is paired with a corresponding base (A-T, C-G). In this simulation, however, the gene-to-protein translation process does not correspond to the biological process. We therefore use different "names" for our four bases - W, X, Y, and Z.

Genetic strings in DNA are represented by the two matching halves of the DNA: the sense and antisense strands. The presence of two different coding strands could be important to the process of evolution; however, implementing both strands only serves to place constraints on the evolution process, in the event that both the sense and antisense strands had overlapping portions of genetic material which were translated into proteins (mutation of one base might thus alter two proteins). Thus, only one strand is implemented in this system.

Eukaryotes (organisms with discrete organelles) often have sections of noncoding DNA, called introns, interspersed with the sections of DNA which actually code for proteins. Prokaryotes, however, do not make use of introns, and as the process by which introns are removed during transcription is unclear, introns are not implemented in this system. In other words, every gene is contiguous on the genetic strand.

Transcription and Translation

Transcription is the process whereby DNA is used as a blueprint for forming messenger RNA (mRNA), which is then transported to ribosomes for translation into proteins. The mRNA strand is the base-pair complement of the DNA strand. As intracellular transport mechanisms are not modeled in this system, the process of transcription retains only one significant feature of biological transcription: the regulatory effect of promoters and operons (we use the scheme seen in prokaryotes). Promoters indicate the binding site of RNA polymerase, an enzyme which binds to DNA and begins the transcription process. Frequently, the activity of RNA polymerase is enhanced by the binding of a catabolite activator protein (CAP) to a region nearby on the DNA strand. Operons are regions of DNA located just downstream of the promoter which permit the binding of proteins to inhibit transcription of a particular gene. Through this process, the inhibitive property of operons inhibits any enhancements provided by CAP. The model developed here includes a region of noncoding DNA which indicates the start of transcription for each gene, and may also include regions which allow proteins to bind to the DNA for enhancing or inhibitive effects.

In the model, mRNA is not actually generated, and the effect of producing a protein through transcription and translation occurs all at once. That is, the intermediate stages of gene-to-protein translation are not tabulated, but rather, once a gene is selected for representation, the protein is subsequently produced.

Translation produces proteins from mRNA blueprints based on an encoding scheme which divides the mRNA strand into pieces three base-pairs long, called codons. Each codon encodes a separate amino acid using tRNA, which includes an antisense codon at one end and the corresponding amino acid at the other. The process for production of tRNA is genetically encoded in an organism, but in this model, sufficient tRNA is assumed to be present in this model without explicitly encoding a synthesis mechanism. In the model developed here, translation occurs virtually simultaneously with transcription; once a gene is selected for transcription, a protein is translated from the gene and is introduced into the cell.

One difference introduced here is that one strand of mRNA is capable of being a translation instruction for only one protein. However, mRNA manipulation strategies add yet another layer of complexity, and are thus beyond the scope of this model.

Amino Acids

Each codon represents one amino acid for a protein encoded by a gene. In biological organisms, there are 20 common amino acids translated from mRNA. As the chemical basis for protein behavior is not modeled accurately, there is no need to be faithful to the number of amino acids present in real organisms, and so we use 16. The amino acids, numbered 0-9 and A-F, have functions determined in part by their locations within the protein.

The representation indicates with what amino acid or DNA sequences a protein can bind, and by what diffusible agents a protein is affected. In addition, one amino acid shares a duty indicating the start of a protein for transcription/translation purposes, and another flags the end of a "field" in the protein's functional representation.

Proteins

Proteins in this model have their function determined through the interpretation of certain fields (domains) in the protein. The lead domain of a protein indicates its function according to the encoding shown in Table 2; the lead domain will also always start with the amino acid 6 as this amino acid encodes the start of a protein transcription sequence. An amino acid 9 indicates the stop of the lead domain (and the stop of every subsequent domain), while the amino acid following 9 indicates the function of the next domain.

This representation of proteins is not chemically accurate, by any means. However, one might find some small justification for such a method by considering the existence of domains within real proteins. Each domain frequently has a separate function from the other domains of the protein, whether it is to embed a protein in the membrane of a cell, or to bind to another protein or other chemical, etc. Thus, the portions of proteins modeled here are referred to as domains as well. Additionally, the head end of some mRNA contains a short sequence which indicates the destination of the final protein; other small pieces of mRNA known as signal recognition particles detect these sequences and help to direct the protein to the transport mechanism which will bring it to its final destination where it can fill its particular designed role.

Proteins can serve one of several different duties in the model. They can serve as enzymes for the production or destruction of a diffusible agent; channels for the passage of a diffusible agent through the cell membrane; chemoreceptors for adjusting the free energy calculations used in determining chemotaxis; or ligands for binding to other proteins or to DNA in order to change that protein's function or inhibit that DNA's transcription (see Table 1). Thse functions are determined by the lead amino acid of the active domain of a protein, and the active domain is selected by the binding of a diffusible agent or another protein to a selector domain of the protein.

In real biological systems, protein folding occurs as a protein is synthesized. Either on its own or with the guidance of certain enzymes, a protein assumes a folded state where amino acids within the protein form hydrogen bonds with other amino acids in that same protein. The net result is a more compact protein which has a much lower free energy than the newly-synthesized, unfolded protein. The surface of the protein determines its function through its folded geometry; the folded geometry is determined by the sequence of proteins (which was determined by genetic information in DNA).

Protein folding is unfortunately not modeled here. While the process of protein folding largely determines the function of a protein, and provides a mechanism for the introduction of mutations which do not immediately affect protein function, and thus permits the evolution of accumulated genetic traits, that process is also NP-complete, even in a 3-d lattice model (Berger and Leighton 1998). Instead, we model protein function solely by the amino acid sequence, as described above.

While all of these mechanisms together represent a fraction of the processes seen in biological cells, they represent a “cut” through those processes to provide the most basic interactions needed for the behaviors to be studied (chemotaxis, multicellular interaction, or survival).

Genetic Algorithm

The model also includes a facility for evolving genomes via genetic algorithm. Supported are one method of crossover (two-point) and four mutation methods (single base mutation, block insertion, block deletion, and block copy). Fitness functions can be defined to describe an arbitrary desired behavior. This code is also being modified to parallelize evaluations among several computers.

Results and Discussion

The success of a model depends upon its ability to simulate the modeled phenomenon. In order to verify this model’s fidelity, we have devised a number of test situations, two of which are described here.

Three-state Genetic Oscillator / “Repressilator”

A genetic oscillator can be formed by linking the repressors of three genes in a loop (Elowitz and Leibler 2000), as shown in Figure 1. The proteins generated by

Amino			DNA
acid	Domain function	Codons	Bind
0	None	wxw wxx wxy	none
1	Unused	wyw wyx wyy wyz	w
2	Chem Producer	wzw wzx wzy wzz	x
3	Activator	xww xwx xwy xwz wxz	wx
4	Chem Channel	xxw xxx xxy xxz	y
5	Actin	xyw xyx xyy xyz wwy	wy
6	Start Protein	www wwx	xy
7	Chemical Pump	xzw xzx xzy xzz	z
8	Chem Selector	yww ywx ywy ywz	z
9	End Domain	zzw zzx	none
A	Chem Attractor	yxw yxx yxy yxz wwz	xz
B	DNA Operon	yyw yyx yyy yyz	y
C	Chem Repeller	yzw yzx yzy yzz zyz	yz
D	Protein Attractor	zww zwx zwy zwz	x
E	Protein Selector	zxw zxx zxy zxz	w
F	Protein Repeller	zyw zyx zyy	wz
--	End Protein	zzy zzz	----

Table 1. Shown are the model’s 16 amino acids, their functions when interpreted as domain function markers, the codons which represent them, and the DNA bases to which they can bind.

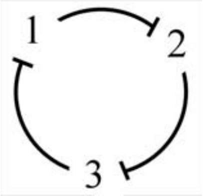
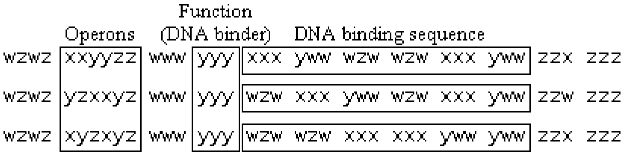


Figure 1. This diagram shows the general layout for the three-state repressilator. The links represent inhibition of the production of one protein by the previous protein in the sequence.

each gene serve as the repressors for the next gene in the sequence. As the concentration of that protein increases, its inhibitory effect increases as well, shutting down the next gene in the series. As the concentration of that protein decreases due to the inactivation of its gene, its inhibitory effect on the next gene decreases, and transcription of that gene begins anew. The result is a three-state oscillation of the concentrations of the three proteins where each state oscillates at a 120-degree phase difference from the other two. The behavior of the three-state "repressilator" is well-described, and forms a useful basis case for evaluating the fidelity of this model.

We recreate the repressilator within the model using the artificial genome shown in Figure 2. Each gene produces a protein which binds to the operon site of the next gene, thus preventing it from being transcribed. For example, the sequence in the first gene, "xxx yww wzw wzw xxx yww," creates a protein with amino acid sequence "482248," which can bind to the operon "yzxxyz" in the second gene.

The cell model, including the genome and the protein production facilities, is evaluated without the environment to obtain the effects of this genome. Figure 3a, a graph of the protein concentrations within the cell over time, indicates that the three protein concentrations in the model do in fact oscillate at about a 120-degree phase difference.



	Operons		Function (DNA binder)	DNA binding sequence	
wzwz	xxyyzz	www	yyy	xxx yww wzw wzw xxx yww	zzx zzz
wzwz	yzxxyz	www	yyy	wzw xxx yww wzw xxx yww	zzw zzz
wzwz	xyzxyz	www	yyy	wzw wzw xxx xxx yww yww	zzx zzz

Figure 2. The genome for the three-state repressilator is shown here. Each gene is shown in its own row, spaced to show functional units such as codons and operons. All three genes carry one domain, marked by the "yyy" codon, which indicates that the domain binds to DNA when active. The remainder of the domain, shown in the three boxed sequences to the right, specifies the operon sequence to which the resulting protein can bind.

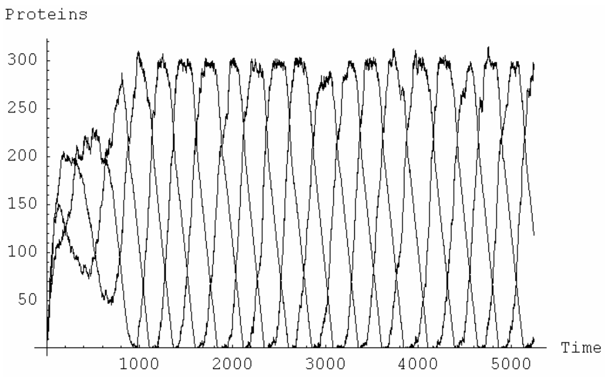


Figure 3. Shown are the model's resulting concentrations over time of the three proteins involved in generating the three-state "repressilator."

Genetically Regulated Chemotaxis

In order to exercise the environmental simulation and evaluate the ability of cells to signal each other, we devise a test setup where one cell is alternately attracted to two chemical signals produced by two other cells. In order to maintain state information regarding whether the active cell is attracted to the first or second chemical (C1 or C2, respectively), a bistable genetic switch is implemented. Such a switch has been implemented in living *E. coli* by Gardner *et al*. (2000). The network used here is an extension of that concept. A schematic of the genome implementing the active cell is shown in Figure 4.

The bistable genetic switch works through mutual inhibition of two genes. Under normal circumstances, the system falls toward a stable state where only one of these two genes is transcribed, since the resulting proteins inhibit transcription of the other gene. This can be counteracted by applying an external influence to overcome the inhibition and cause the other of the two genes to dominate.

In this system, the two mutually-inhibitive genes are *modeC1inh* and *modeC2inh*. They are named so because when the cell is in the mode where it is attracted to chemical C1, the gene *modeC1inh* is being transcribed. When a large amount of C1 or C2 enters the cell, the protein produced by *switchC2* or *switchC1*, respectively, is activated, enhancing the transcription of *modeC2inh* or *modeC1inh*. If the gene being enhanced is being inhibited by its counterpart currently, the enhancement causes transcription of the gene to occur anyway, which causes increasing inhibition of the counterpart gene. The continued enhancement leads the enhanced gene to become the dominating gene, and transcription of its counterpart falls to zero. In other words, the cell switches between modes "C1" and "C2".

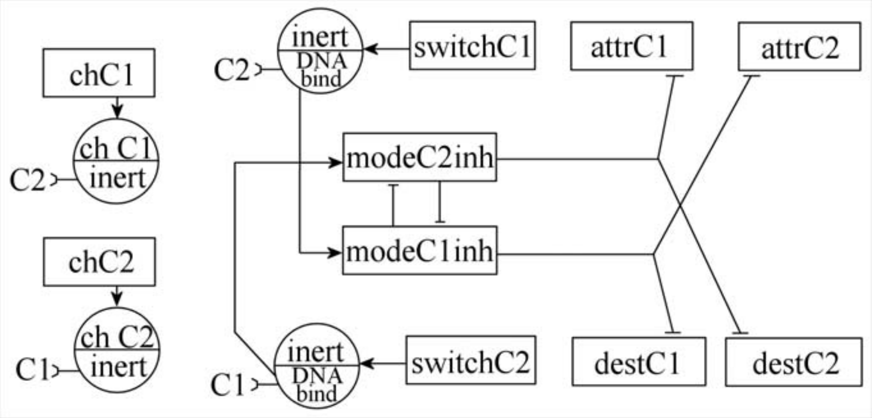


Figure 4. Shown is a diagram of the genetic network implemented in the oscillating attractor. Genes are shown in rectangles, and their proteins are denoted by circles. Above the bar in a circle is the primary function, and below is the secondary function when the protein is activated by the chemical noted adjacent to that region. At the left, the two genes *chC1* and *chC2* code for channel proteins that pass C1 and C2 through the cell membrane, but are inactivated in the presence of C2 and C1, respectively. In the center are the genes *modeC1inh* and *modeC2inh*. Their mutual inhibition permits only one of these proteins to be produced at one time. They also inhibit the production of proteins which metabolize the two chemicals inside the cell (*destC1* and *destC2*) and cause chemoattraction to chemicals outside the cell (*attrC1* and *attrC2*). The proteins generated by *switchC1* and *switchC2* cause a state switch to occur when the concentration of C2 or C1 increases, by enhancing the transcription of *modeC1inh* and *modeC2inh*.

In addition, the *modeC1inh* and *modeC2inh* genes affect the attraction of the cell to chemicals C1 and C2 outside the cell. They do this by inhibiting attractor genes. In mode C1, the cell inhibits the *attrC2* gene, and thus is not attracted to C2; likewise, the *attrC1* gene is inhibited in mode C2. These genes also inhibit metabolism genes (*destC1* and *destC2*) which eliminate the two chemicals C1 and C2. In mode C1, the metabolism of chemical C1 is inhibited, since the cell requires a large enough concentration of C1 in order to switch to mode C2; likewise, *destC2* is inhibited in mode C2.

Finally, the channels permitting C1 and C2 to flow through the cell membrane are directly influenced by the concentration of C1 and C2 inside the cell. A large concentration of C1 causes the C2 channels to close. This generally occurs in mode C1, where the cell is attracted to the C1 chemical. Once the cell switches modes – C1 also triggers the switch to mode C2 – the cell moves toward the chemical C2 and begins metabolizing C1. When the concentration of C1 drops, the C2 channels open and eventually cause the C1 channels to close.

The experimental setup also includes two other cells placed in the environment. Their genomes are much simpler – each of these passive cells produces one of the two chemicals, and has membrane channels to let that chemical pass out of the cell. They are otherwise not responsive to the environment.

Initial conditions place the active cell close to the passive cell producing C1 (Figure 5a, i.). At this point, the stable state which the cell chooses is still unaffected by the tiny concentrations of C1 and C2 in the cell; in this case, the cell randomly chooses mode C1, and is thus attracted to passive cell 1.

The active cell follows the gradient of C1 upwards, thus leading it to push passive cell 1 out of the way. In this way, it “follows” passive cell 1 around the environment (5a, ii.) until the concentration of C1 reaches a critical level (5b, iii.). Once this occurs, the cell switches modes (5c, iv.) and becomes attracted to C2. It then moves to passive cell 2 (5a, iv.) and pushes it around the environment until the mode switches back to C1. This process repeats indefinitely, though the frequency is in part determined by the amount of chemical in the environment.

Conclusions

The model developed here is able to reproduce some of the basic notions of engineered genetic networks, such as three-state oscillations and bistable switching. We plan next to apply a genetic algorithm to this model to evolve behaviors observed in *Dictyostelium*, such as aggregation, migration, and stalk formation, as well as the transitions between these behaviors. Ultimately, as computing power improves, this model could also be used to evolve open-ended simulations where survival is the only measure of fitness.

Acknowledgments

This work was made possible in part by the NSF IGERT program in Neuromechanics at Case Western Reserve University.

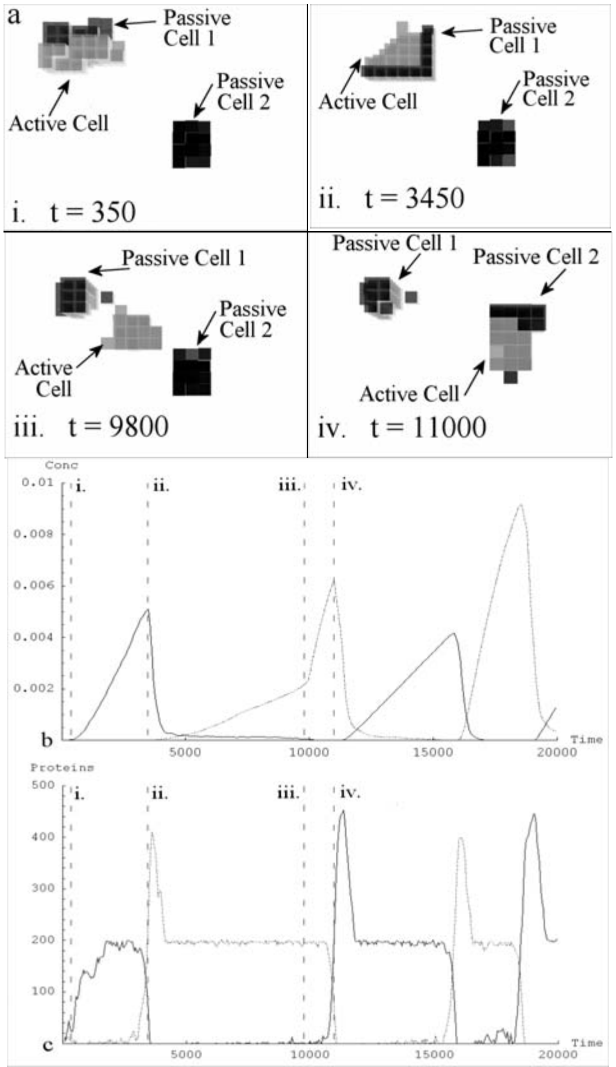


Figure 5. a. Four snapshots of the genetically-regulated chemotaxis test are shown here. **b.** Concentrations of C1 (solid) and C2 (gray). **c.** Amounts of proteins *modeC1inh* (solid) and *modeC2inh* (gray). In **b** and **c**, the dotted lines i-iv correspond to the times of the four snapshots in **a**.

References

Berger, B. and Leighton, T. (1998) Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. Proceedings of the Second Annual International Conference on Computational Molecular Biology 30-39.

Brown, J. and Firtel, R. (1999) Regulation of cell-fate determination in *Dictyostelium*. Developmental Biology 216:426–441.

Dellaert, F. and Beer, R. (1994) Toward an evolvable model of development for autonomous agent synthesis. Proceedings of Artificial Life IV 246-257.

Elowitz, M.B. and Leibler, S.A. (2000) Synthetic gene oscillatory network of transcriptional regulators. Nature 403:335-338.

Gartner, T.S., Cantor, C.R., and Collins, J.J. (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403:339-342.

Glazier, J. and Graner, F. (1993) Simulation of the differential adhesion driven rearrangement of biological cells. Physical Review E 47(3):2128-2154.

Goel, N.S. and Thompson, R.L. (1988b) Movable Finite Automata models for biological systems II: Protein Biosynthesis. Journal of Theoretical Biology 134:9-49.

Graner, F., and Glazier, J. (1992) Simulation of biological cell sorting using a two-dimensional extended Potts model. Physical Review Letters 69(13):2013-2016.

Kimball, John W. (2003) Kimball's Biology Pages. http://biology-pages.info/

Marée, A.F.M., Panfilov, A.V., and Hogeweg, P. (1999) Migration and thermotaxis of *Dictyostelium discoideum* slugs, a model study. Journal of Theoretical Biology 199:297-309.

Marée, A.F.M. and Hogeweg, P. (2002) Modelling *Dictyostelium discoideum* morphogenesis: the culmination. Bulletin of Mathematical Biol. 64:327-353.

Savill, N. and Hogeweg, P. (1997) Modelling morphogenesis: from single cells to crawling slugs. Journal of Theoretical Biology 184:229-235.

Söderbom, F. and Loomis, W. (1998) Cell–cell signaling during *Dictyostelium* development. Trends in Microbiology 6:402-406.

Takahashi, K., Ishikawa, N., *et al.* (2003) E-Cell 2: Multi-platform E-Cell simulation system. Bioinformatics 19:1727-1729.

Thompson, R.L. and Goel, N.S. (1988a) Movable Finite Automata models for biological systems I: Bacteriophage assembly and operation. Journal of Theoretical Biology 131:351-385.

Tomita, M., Hashimoto, K., *et al.* (1999) E-Cell: software environment for whole cell simulation. Bioinformatics 15:72-84.

von Neumann, J., and Burks, A.W. (ed.) (1966) Theory of Self-Reproducing Automata. University of Illinois Press.