Accepted Manuscript

Title: Use of Evolved Artificial Regulatory Networks to
Simulate 3D Cell Differentiation

Authors: Arturo Chavoya, Irma R. Andalon-Garcia,
Cuauhtemoc Lopez-Martin, M.E. Meda-Campaña

PII: S0303-2647(10)00119-X
DOI: doi:10.1016/j.biosystems.2010.07.011
Reference: BIO 3095

To appear in: *BioSystems*

Received date: 2-5-2010
Accepted date: 15-7-2010

Please cite this article as: Chavoya, A., Andalon-Garcia, I.R., Use of Evolved Artificial Regulatory Networks to Simulate 3D Cell Differentiation, *BioSystems* (2010), doi:10.1016/j.biosystems.2010.07.011

Use of Evolved Artificial Regulatory Networks to Simulate 3D Cell Differentiation

Arturo Chavoya [a,*] Irma R. Andalon-Garcia [a] Cuauhtemoc Lopez-Martin [a]
M.E. Meda-Campaña [a]

[a] *Universidad de Guadalajara, Periférico Norte 799-L308, Zapopan, Jal., Mexico CP 45100*

Abstract

Cell differentiation has a crucial role in both artificial and natural development. This paper presents results from simulations in which a genetic algorithm (GA) was used to evolve artificial regulatory networks (ARNs) to produce predefined 3D cellular structures through the selective activation and inhibition of genes. The ARNs used in this work are extensions of a model previously used to create 2D geometrical patterns. The GA worked by evolving the gene regulatory networks that were used to control cell reproduction, which took place in a testbed based on cellular automata (CA). After the final chromosomes were produced, a single cell in the middle of the CA lattice was allowed to replicate controlled by the ARN found by the GA, until the desired cellular structures were formed. Two simple cubic layered structures were first developed to test multiple gene synchronization. The model was then applied to the problem of generating a 3D French flag pattern using morphogenetic gradients to provide cells with positional information that constrained cellular replication.

Key words: Cell differentiation, Artificial regulatory network, Genetic algorithm, French flag problem, Cellular automata

1. Introduction

Cell differentiation is at the core of the processes that lead to the development of multicellular organisms. The initial undifferentiated cell —the zygote— undergoes a series of divisions that ultimately produce tissues and organs made of highly differentiated cells. It is now evident that gene regulatory networks play a central role in the development and metabolism of living organisms (Davidson, 2006). Furthermore, it has been found in recent years that the diverse body structures and patterns generated during the development of an organism are greatly influenced by the selective activation and inhibition of very specific regulatory genes. On the other hand, computational models known as Artificial Regulatory Networks (ARNs) have the objective of emulating the gene regulatory networks found in nature to some extent. ARNs have previously been used to study differential gene expression either as a computational paradigm or to solve particular problems (Eggenberger, 1997; Reil, 1999; Banzhaf, 2003; Joachimczak and Wróbel, 2008; Chavoya, 2009). In order to evolve ARNs to perform specific tasks, evolutionary computation techniques have been used in the past (Bongard, 2002; Kuo et al., 2004).

In this work we describe results on the use of a genetic algorithm (GA) to evolve ARNs in order to synchronize cell differentiation with the purpose of obtaining 3D structures by means of the selective

* Corresponding author.
Email addresses: achavoya@cucea.udg.mx (Arturo Chavoya), agi10073@cucea.udg.mx (Irma R. Andalon-Garcia), cuauhtemoc@cucea.udg.mx (Cuauhtemoc Lopez-Martin), emeda@cucea.udg.mx (M.E. Meda-Campaña).

activation and inhibition of genes. The ARNs used in this work were originally based on the model developed by Banzhaf (2003) and have been extended to overcome gene synchronization limits previously reported (Chavoya and Duthen, 2008). In order to test the functionality of the ARN found by the GA, chromosomes representing ARNs were applied to a cellular growth testbed that has been successfully used in the past to develop simple 2D and 3D geometrical shapes (Chavoya and Duthen, 2006). The cellular growth testbed is based on cellular automata (CA), as they provide a simple mathematical model that can be used to study self-organizing features of complex systems (Wolfram, 1983). Gene synchronization was first tested by developing genomes that generated two simple structures consisting of cubic structures with two and three concentric layers, respectively. Each layer corresponds to differentiated cells expressing distinct genes controlled by an ARN. The cell differentiation model was then applied to what is known as the *French flag problem*, using morphogenetic fields to spatially constrain cell reproduction.

The paper starts with a section describing the French flag problem with a brief description of models that have used it as a test case. The next section describes the cellular growth testbed developed to evaluate the evolved genomes, followed by a section presenting the morphogenetic gradients used. The next section introduces the artificial genomes containing the ARNs and how they were implemented. The following section describes the GA used and how it was applied to evolve the genomes. Results are presented next, followed by a section of conclusions.

2. The French Flag Problem

The problem of generating a French flag pattern was first introduced by Wolpert in the late 1960s when trying to formulate the problem of cell pattern development and regulation in living organisms (Wolpert, 1968). This formulation has been used since then by some authors to study the problem of artificial pattern development. More specifically, the problem deals with the creation of a pattern with three sharp bands of cells with the colors and order of the French flag stripes.

Lindenmayer and Rozenberg (1972) used the French flag problem to illustrate how a grammar-based L-System could be used to solve the generation of this particular pattern when enunciated as the production of a string of the type $a^n b^n c^n$ over the alphabet $\{a, b, c\}$ and with $n > 0$. On the other hand, Herman and Liu (1973) developed an extension of a simulator called CELIA (Baker and Herman, 1970) and applied it to generate a French flag pattern in order to study synchronization and symmetry breaking in cellular development.

Miller and Banzhaf (2003) used what they called Cartesian genetic programming to evolve a cell program that would build a French flag pattern. They tested the robustness of their programs by manually removing parts of the developing pattern. They found that several of their evolved programs could repair to some extent the damaged patterns. Bowers (2005) also used this problem to study the phenotypic robustness of his embryogeny model, which was based on cellular growth with diffusing chemicals as signaling molecules.

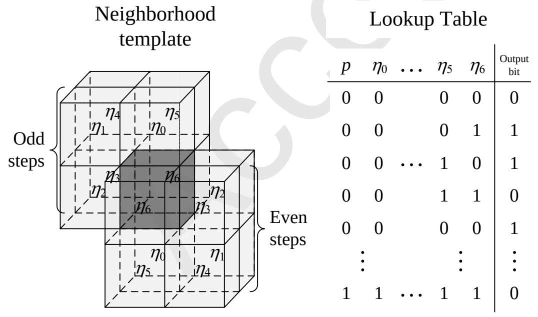
Gordon and Bentley (2005) proposed a development model based on a set of rules that described how development should proceed. A set of rules evolved by a GA was used to develop a French flag pattern. The morphogenic model based on a multiagent system developed by Beurier et al. (2006) also used an evolved set of agent rules to grow French and Japanese flag patterns. On the other hand, Devert et al. (2007) proposed a neural network model for multicellular development that grew French flag patterns. Even models for developing evolvable hardware have benefited from the French flag problem as a test case (Tyrrell and Greensted, 2007; Harding et al., 2007).

More recently, Knabe et al. (2008) developed a model based on the CompuCell3D package (Cickovski et al., 2007) combined with a genetic regulatory network that controlled cell parameters such as size, shape, adhesion, morphogen secretion and orientation. They were able to obtain final 2D patterns with matches of over 75% with respect to a 60×40 pixel target French flag pattern.

3. Cellular Growth Testbed

Cellular automata were chosen as models of cellular growth, since they provide a simple mathematical model that can be used to study self-organizing features of complex systems (Wolfram, 1983). CA are characterized by a regular lattice of N identical cells, an interaction neighborhood template η, a finite set of cell states Σ, and a space- and time-independent transition rule ϕ which is applied to every cell in the lattice at each time step.

In the cellular growth testbed used in this work, two regular lattices with non-periodic boundaries were defined. A $19 \times 19 \times 19$ lattice was used for the cubic patterns and a $13 \times 13 \times 13$ lattice was used to produce a solid 3D French flag structure. In the latter case a smaller lattice was used in order to reduce simulation times, since the introduction of morphogenetic fields required a higher number of GA experiments to obtain a genome that generated the desired pattern, as discussed in Section 7. In both cases, the set of cell states was defined as $\Sigma = \{0, 1\}$, where 0 can be interpreted as an empty cell and 1 as an occupied or active cell. The interaction neighborhood η considered for this work was a 3D Margolus template (Fig. 1), which has been previously used with success to model 3D shapes (Wu et al., 2004). In this template there is an alternation of the block of cells considered at each step of the CA algorithm. At odd steps, the seven cells shown to the left and the back in the figure constitute the interaction neighborhood, whereas at even steps the neighborhood is formed by the mirror cells of the previous block. The 3D Margolus neighborhood was chosen over, for example, a 3D Moore neighborhood, since in the latter case the interaction neighborhood would consist of the nearest 26 cells, giving a CA lookup table of $2^{26} \approx 6.7 \times 10^7$ rows, as opposed to the $2^7 \times 2 = 256$ rows required by the 3D Margolus neighborhood. In general, the number of rows in the CA lookup table grows exponentially with the number of cells in the neighborhood used. The 3D Margolus template covers a fair amount of the neighboring cells requiring relatively few rows in the CA lookup table.



p	η_0	…	η_5	η_6	Output bit
0	0		0	0	0
0	0		0	1	1
0	0	…	1	0	1
0	0		1	1	0
0	0		0	0	1
⋮			⋮		⋮
1	1	…	1	1	0

Fig. 1. Cellular automaton's 3D Margolus neighborhood template and the associated lookup table. The parity bit p in the lookup table determines which block of the neighborhood template is being considered for evaluation. The objective cell is depicted as a darker cube in the middle of the template.

The CA rule ϕ was defined as a lookup table that determined, for each local neighborhood, the state (empty or occupied) of the objective cell at the next time step. For a binary-state CA, these update states are termed the rule table's "output bits". The lookup table input was defined by the binary state value of cells in the local interaction neighborhood (η_0 to η_6), where 0 meant an empty cell and 1 meant an occupied cell and the parity bit p determined which of the two blocks of cells was being considered for evaluation (Chavoya and Duthen, 2006). When $p = 0$, the first 128 rows (2^7) in the lookup table represent the possible configurations of the block of cells to the left and back of the objective cell, whereas when $p = 1$ the last 128 rows code for the possible configurations of the mirror cells of the previous block. The output bit values shown in Fig. 1 are only for illustration purposes; the actual values for a predefined shape, such as a cube, are found by a GA.

In the testbed, a cell can become active only if there is already an active cell in the interaction neighborhood. Thus, a new active cell can only be derived (reproduced) from a previously activated cell in the interaction neighborhood. Starting with an active cell in the middle of the lattice, the CA algorithm is applied allowing active cells to reproduce for 60 time steps according to the CA rule table. This upper limit of 60 was chosen because it was found that the system was not sensitive to the number of time steps. In principle, allowing a higher upper limit in the number of time steps would give the system more time to evolve a desired solution. However, in most experiments performed, cells tended to reproduce rapidly at the beginning and then quickly fill up the available space for growth. Besides, given the exponential reproduction of cells, a higher number of time steps would mean longer simulation times.

During an iteration of the CA algorithm, the order of reproduction of active cells is randomly selected in order to avoid artifacts caused by a deterministic order of cell reproduction. For the sake of simplicity, cell death is not considered in the present model.

For all experiments, the CA were implemented as NetLogo models. NetLogo is a programmable modeling environment based on StarLogo that can be used to simulate natural and social phenomena (Wilensky, 1999). It works by giving instructions to hundreds or thousands of independent "agents" all operating concurrently. It is well suited to study emergent properties in complex systems that result from the interaction of simple but often numerous entities. For each of the cell structures studied, a NetLogo model was built.

4. Morphogenetic Gradients

Ever since Turing's seminal article on the theoretical influence of diffusing chemical substances on an organism's pattern development (Turing, 1952), the role of these molecules has been confirmed in a number of biological systems. These organizing substances were termed *morphogens*, given their involvement in driving morphogenetic processes. In the present model, morphogenetic gradients were generated similar to those found in the eggs of the fruit fly *Drosophila*, where orthogonal gradients offer a sort of Cartesian coordinate system (Carroll et al., 2004). These gradients provide reproducing cells with positional information in order to facilitate the spatial generation of patterns. The artificial morphogenetic gradients were set up as suggested in (Meinhardt, 1982), where morphogens diffuse from a source towards a sink, with uniform morphogen degradation throughout the gradient.

Before cells were allowed to reproduce in the cellular growth model, morphogenetic gradients were generated by diffusing the morphogens from one of the CA boundaries for 1000 time steps for each of the three orthogonal axes. Initial morphogen concentration level was set at $2^8 - 1 = 255$ arbitrary units, and the source was replenished to the same level at the beginning of each cycle. The diffusion factor was 0.20, i.e. at each time step every grid position diffused 20% of its morphogen content and all neighboring positions received an equal amount of this percentage. This factor was introduced to avoid rapid morphogen depletion at cell positions and its value was experimentally determined to render a smooth descending gradient. The sink was set up at the opposite boundary of the lattice, where the morphogen level was always set to zero. At the end of each time step, morphogens were degraded at a rate of 0.005 throughout the CA lattice. Three orthogonal gradients were defined in the CA lattice, one for each of the main Cartesian axes (Fig. 2).

5. Genomes

Two different artificial genomes are proposed in this work, depending on whether or not they constrain cell reproduction by means of morphogenetic fields. In both cases, genomes were defined as binary strings starting with a series of ten regulatory genes, followed by a series of structural genes, which contain the CA's lookup tables that control cell reproduction. The regulatory genes at the beginning of the genome constitute an ARN and they determine the concentration level of the regulatory proteins that control structural gene activation.

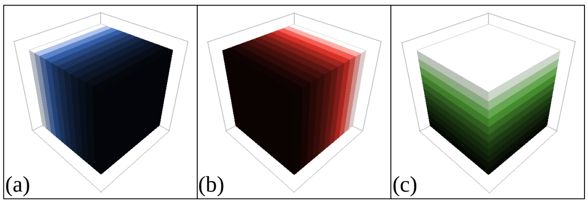


Fig. 2. Morphogenetic gradients. Positions with highest morphogen concentration are depicted in white; darker tones mean lower concentrations. (a) Left to right (x axis); (b) back to front (y axis); (c) top to bottom (z axis).

The genome that does not make use of morphogenetic fields is presented in Fig. 3

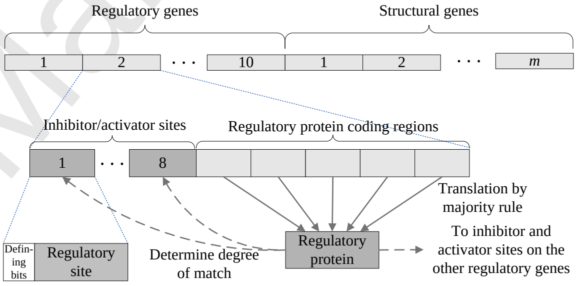

Regulatory genes
Structural genes
1
2
· · ·
10
1
2
· · ·
m
Inhibitor/activator sites
Regulatory protein coding regions
1
· · ·
8
Translation by majority rule
Defin-ing bits
Regulatory site
Determine degree of match
Regulatory protein
To inhibitor and activator sites on the other regulatory genes

Fig. 3. Genome structure and regulatory gene detail. Regulatory genes make up an artificial regulatory network, whereas structural genes contain the lookup tables that control cell reproduction.

Structural genes are always associated to the corresponding regulatory genes, that is, structural gene number 1 is associated to regulatory gene number 1 and its related translated protein, and so on. A structural gene was defined as being active if and only if the regulatory protein translated by the associated regulatory gene was above a certain concentration threshold. The value chosen for the threshold was 0.5 as the sum of all protein concentrations is always 1.0, making it impossible for two or more regulatory proteins to be with a concentration above 0.5 units at the same time. As a result, one structural gene at most can be expressed at a particular time step in a cell. A structural gene is interpreted as a CA rule table by reading its bits as output bits of the CA rule. If a structural gene is active, then the CA lookup table coded in it is used to control cell reproduction. If no protein concentration is above the 0.5 threshold, then cell reproduction cannot occur.

The gene regulatory networks implemented in this work are an extension of the model originally proposed by Banzhaf (2003). However, unlike the ARN developed by this author, genes implemented in the present models are not preceded by promoter sequences and there are no unused intergene regions. Promoters in biology indicate where regulatory binding sites begin. In Banzhaf's model, a genome is a randomly generated bit string where the beginning of a gene is signaled by a fixed arbitrary 8-bit sequence. As a result, this promoter sequence can occur with a probability of $2^{-8} \approx 0.0039 = 0.39\%$ and the number of genes is proportional to the length of the genome.

For the present model, it was decided that unused bit sequences between genes would be a waste of space and an additional source of variation in the evolution experiments, since the number of regulatory genes could vary from genome to genome. Furthermore, this approach would make relating a fixed number of structural genes to a varying number of regulatory genes difficult. As a result it would be possible to have less regulatory genes than structural genes, which went against the concept of one regulatory gene controlling one structural gene.

In the models proposed in the present work, all regulatory genes are adjacent and have predefined initial and end positions. Furthermore, the number of regulatory genes is fixed and the number of regulatory sites is more than two and they can behave either as an activator or an inhibitor, depending on the configuration of the function defining bits associated with the regulatory site (Chavoya and Duthen, 2008). If there are more 1's than 0's in the function defining region, then the site functions as an activator, but if there are more 0's than 1's, then the site behaves as an inhibitor. Finally, if there is an equal number of 1's and 0's, then the regulatory site is turned off. This means that the regulatory site role as an activator or as an inhibitor can be evolved by the GA. Furthermore, if the number of function defining bits is even, then the regulatory site can be turned on and off. The number of regulatory sites was extended with respect to the original model in order to more closely follow what happens in nature, where biological regulatory genes involved in development typically have several regulatory sites associated with them (Davidson, 2006).

Besides the inhibitor/activator sites, each regulatory gene contains a series of five regulatory protein coding regions which "translate" a protein using the majority rule, i.e. for each bit position in the protein coding regions, the number of 1's and 0's is counted and the bit that is in majority is translated into the regulatory protein. An odd number of regulatory protein coding regions sites was chosen by Banzhaf in order to avoid ties when applying the majority rule.

The regulatory sites and the individual protein coding regions all have the same size in bits. Thus the protein translated from the coding regions can be compared on a bit by bit basis with the regulatory sites from the inhibitors and activators, and the degree of matching can be measured. As in (Banzhaf, 2003), the comparison was implemented by an XOR operation, which results in a "1" if the corresponding bits are complementary.

Each translated protein is compared with the inhibitor and activator sites of all the regulatory genes in order to determine the degree of interaction in the regulatory network. The influence of a protein on an activator or inhibitor site is exponential with the number of matching bits. The strength of excitation en or inhibition in for gene i with $i = 1, ..., n$ is defined as

$$en_i = \frac{1}{v}\sum_{j=1}^{v} c_j e^{\beta\left(u_{ij}^{+} - u_{\max}^{+}\right)} \text{ and} \qquad (1)$$

$$in_i = \frac{1}{w}\sum_{j=1}^{w} c_j e^{\beta\left(u_{ij}^{-} - u_{\max}^{-}\right)} \ , \qquad (2)$$

where n is the total number of regulatory genes, v and w are the total number of activator and inhibitor sites, respectively, c_j is the concentration of protein j, β is a constant that fine-tunes the strength of matching, u_{ij}^{+} and u_{ij}^{-} are the number of matches between protein j and the activator and inhibitor sites of gene i, respectively, and $u_{\max}^{+}$ and $u_{\max}^{-}$ are the maximum matches achievable between a protein and an activator or inhibitor site, respectively (Banzhaf, 2003).

Once the en and in values are obtained for all regulatory genes, the corresponding change in concentration c for protein i in one time step is found using

$$\frac{dc_i}{dt} = \delta\left(en_i - in_i\right) c_i \ , \qquad (3)$$

where δ is a constant that regulates the degree of protein concentration change. Parameters β and δ

were set to 1.0 and 1.0×10^6, respectively, as previously reported (Chavoya and Duthen, 2007b).

Protein concentrations are updated and if a new protein concentration results in a negative value, the protein concentration is set to zero. Protein concentrations are then normalized so that total protein concentration is always the unity. At time step 0, all ten proteins start out with the same concentration level, i.e. with a value of 0.1 units.

As for structural genes, they code for the particular shape grown by the reproducing cells and were obtained using the methodology presented in (Chavoya and Duthen, 2006). Briefly, a gene was evolved by a GA in the cellular growth testbed described in Section 3 in order to produce predefined 3D shapes. The GA worked by evolving the CA rule table's output bits.

In the series of experiments presented in this work, the number of structural genes is always less than the number of regulatory genes. Thus, some proteins both regulate concentration for other proteins and directly control structural gene expression, while others only have a regulatory role. Structural gene expression is visualized in the cellular growth testbed as a distinct external color for the cell. Thus, cells with different external color represent differentiated cells that express a specific structural gene. The color associated with a structural gene is assigned to a cell when it is created as a result of the activation of that particular structural gene.

The second genome used in this work is an extension of the first genome presented and its structure is shown in Fig. 4. The difference lies in the addition of three morphogen activation sites at the end of the regulatory gene which are shown encircled in the figure. A similar genome with two activation sites was previously proposed to develop 2D patterns (Chavoya, 2008).

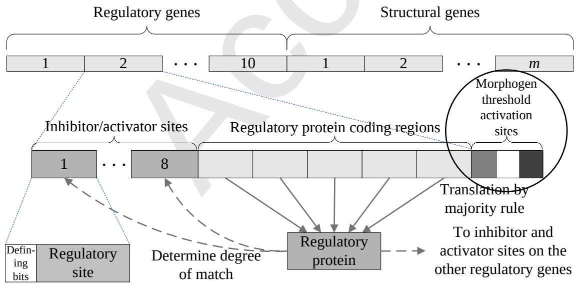

Regulatory genes
Structural genes
1
2
. . .
10
1
2
. . .
m
Morphogen threshold activation sites
Inhibitor/activator sites
Regulatory protein coding regions
1
. . .
8
Translation by majority rule
Defin-ing bits
Regulatory site
Determine degree of match
Regulatory protein
To inhibitor and activator sites on the other regulatory genes

Fig. 4. Genome structure and extended regulatory gene detail. Three fields were added to each regulatory gene to control cell reproduction through the definition of morphogen activation thresholds.

The morphogen threshold activation sites can provide reproducing cells with positional information as to where they are allowed to grow in the CA lattice. There is one site for each of the three orthogonal morphogenetic gradients described in Section 4. These sites are 9 bits in length, where the first bit defines the allowed direction (above or below the threshold) of cellular growth, and the next 8 bits code for the morphogen threshold activation level, which ranges from 0 to $2^8 - 1 = 255$. If the site's high order bit is 0, then cells are allowed to replicate below the morphogen threshold level coded in the lower order eight bits; if the value is 1, then cells are allowed to reproduce above the threshold level. Since in a regulatory gene there is one site for each of the three orthogonal morphogenetic gradients, for each set of three morphogen threshold activation levels, the three high order bits define in which of the eight relative octants cells expressing the associated structural gene can reproduce.

6. Genetic Algorithm

Genetic algorithms are search and optimization methods based on ideas borrowed from natural genetics and evolution (Holland, 1992). A GA starts with a population of chromosomes representing vectors in search space. Each chromosome is evaluated according to a fitness function and the best individuals are selected. A new generation of chromosomes is created by applying genetic operators on selected individuals from the previous generation. The process is repeated until the desired number of generations is reached or until the desired individual is found.

The GA in this paper uses tournament selection as described in (Mitchell, 1996) with single-point crossover and mutation as genetic operators. Single-point crossover consists of randomly selecting two chromosomes with a certain probability called crossover rate, and then randomly selecting a single bit position in the chromosome structure. From this point on, the remaining fragments of the two chromosomes are exchanged. The resulting chromosomes then replace the original ones in the chromosome population. On the other hand, mutation consists of randomly flipping one bit in a chromosome from 0 to 1 or vice versa. The probability of each bit to be flipped is called the mutation rate.

As in a previous report, we used the following parameter values (Chavoya and Duthen, 2007a). The initial population consisted of 1000 binary chromosomes chosen at random. Tournaments were run with sets of 3 individuals randomly selected from the population. Crossover and mutation rates were 0.60 and 0.15, respectively. Finally, the number of generations was set at 50, since there was no significant improvement after this number of generations.

Single-point crossover with a rate of 0.60 was chosen because it was reported to give the best results when trying to evolve a binary string representing a CA using a GA (Breukelaar and Bäck, 2005). As for the mutation rate, we decided to use a value one order of magnitude higher than the one used in the same report, since it was found that single bits could have a considerable influence on the final behavior of the ARN. In particular, in one experiment the flipping of a single bit almost doubled the fitness value of an evolving genome (Chavoya and Duthen, 2007b).

The fitness function used by the GA was defined as

$$Fitness = \frac{1}{k}\sum_{i=1}^{k}\frac{ins_i - \frac{1}{2}outs_i}{des_i} \ , \qquad (4)$$

where k is the number of different colored shapes inside a pattern, each corresponding to an expressed structural gene, ins_i is the number of active cells inside the desired shape i with the correct color, $outs_i$ is the number of active cells outside the desired shape i, but with the correct color, and des_i is the total number of cells inside the desired shape i. This fitness function is an extension of the one used in (de Garis, 1999), where the shape produced by only one "gene" was considered. To account for the expression of several structural genes, the combined fitness values of all structural gene products were introduced in the fitness function used.

During the course of a GA experiment, each chromosome produced in a generation was fed to the corresponding NetLogo model, where the structural genes were attached and cells were allowed to reproduce controlled by the ARN found by the GA. Fitness was evaluated after the simulation stopped and a colored pattern was formed. This process continued until the maximum number of generations was reached or the optimal solution was found.

7. Results

The GA described in Section 6 was used in all cases to obtain de CA's rule tables to conform the structural genes for specific simple patterns and to evolve the ARNs for the desired multicolored patterns. After an evolved genome was obtained, an initial active cell containing it was placed in the center of the CA lattice and was allowed to reproduce for 60 time steps in the cellular growth testbed described in Section 3, controlled by the gene activation sequence found by the GA. In order to grow the desired structure with a predefined color and position for each cell, the regulatory genes in the ARN had to evolve to be activated in a precise sequence and for a specific number of iterations. Not all GA experiments produced a genome capable of generating the desired pattern.

In the figures shown next the following conventions are used: in the 3D insets the positive x axis extends to right, the positive y axis is towards de back of the page, the positive z axis is at the top, and the axes are rotated 45 degrees to the left to show a better perspective. The graphs presented in the figures correspond to some of those experiments where ARNs with fitness function values equal to 1.0 were found by the GA.

A two-layer cubic structure grown from the expression of two structural genes that produce concentric cubes is shown in Fig. 5. Expression of the first gene generates the inner cube, while the second gene drives cells to grow the outer layer to complete the structure. The graph of the corresponding regulatory protein concentration change over time is shown in 5(e). Starting with an initial white cell (a), the inner white cube is formed from the expression of the first gene (b), followed by the outer cube resulting from gene 2 being expressed (c); the finished cube structure is shown in (d) with a cutout to exhibit the layered structure.

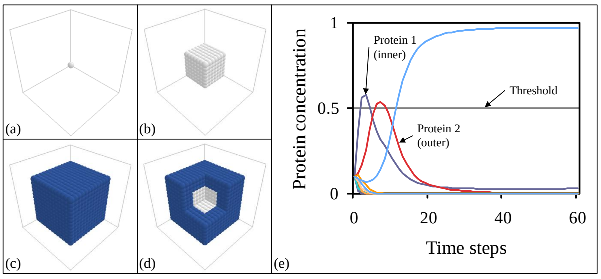


Fig. 5. Growth of a two-layer cube structure. (a) Initial cell; (b) inner cube; (c) outer layer with inner cube inside; (d) finished cube structure with a cutout made to show the two layers; (e) graph of protein concentration change from the genome expressing the two-layer cube structure.

Figure 6 shows a three-layer cubic structure grown from the expression of three structural genes, each controlling the expression of cells forming a cube layer of a different color. The graph of the corresponding regulatory protein concentration change over time is shown in 6(e). Starting with an initial white cell (a), a white central cube is formed from the expression of gene number 1 (b), the middle cubic layer is then grown (c), followed by the outer layer that completes the desired 3D structure (d).

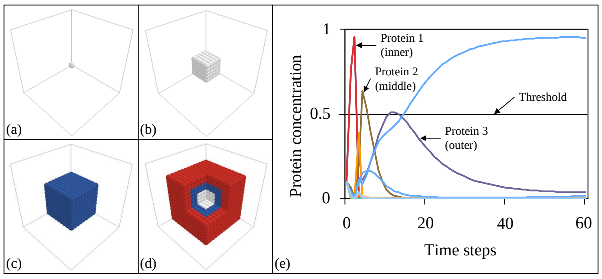


Fig. 6. Growth of a three-layer cube structure. (a) Initial cell; (b) inner cube; (c) middle layer with inner cube inside; (d) finished cube structure with a cutout made to show the three layers; (e) graph of protein concentration change from the genome expressing the three-layer cube structure.

The previous structures were chosen so that structural genes were expressed for the same number of time steps in the cellular growth testbed. For the two-color cubic pattern, each structural gene is expressed for exactly three time steps, whereas for the three-color cube structure, each of the three genes involved is activated for two time steps.

In order to explore the result of combining different structural genes that are expressed for a different number of time steps and to test the use of morphogenetic fields to constrain cellular growth, three different structural genes were used to grow a solid 3D French flag pattern. Expression of the first gene creates the white central cube, while the other two genes drive cells to extend the lateral walls to the left and to the right simultaneously, expressing the blue and the red color, respectively. These two last genes do not necessarily code for a cube, since they only extend a wall of cells to the left and to the right for as many time steps as they are activated and when unconstrained, they produce a symmetrical pattern along the x axis. In order to produce the desired French flag pattern, cells expressing one of these two genes should only be allowed to reproduce on each side of the white central cube (left for the blue cube and red for the right cube). This behavior was achieved through the use of genomes where the morphogen threshold activation sites evolved to allow growth only in the desired portions of the 3D CA lattice.

Figure 7 shows a 9 × 3 × 3 solid French flag pattern grown from the expression of the three structural genes mentioned above. The graph of the corresponding ARN protein concentration change is shown in Fig. 7(e). Starting with an initial white cell (a), a white central cube is formed from the expression of gene number 1 (b), the right red cube is then grown (c), followed by the left blue cube (d). The evolved morphogenetic fields where cells are allowed to grow are depicted in the figure as a translucent volume for each of the three structural genes. Note that for the genes that extend the wall of cells to the sides, the corresponding morphogenetic fields limited growth to the desired direction (red to the right and blue to the left) and produced the desired French flag pattern.

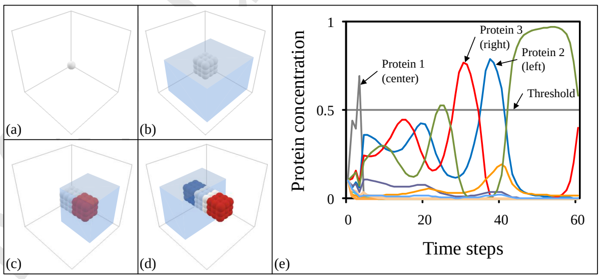


Fig. 7. Growth of a 3D French flag pattern. (a) Initial cell; (b) central white cube with morphogenetic field for gene 1 (cube); (c) central white cube and right red cube with morphogenetic field for gene 3 (extend red lateral walls); (d) finished flag pattern with morphogenetic field for gene 2 (extend blue lateral walls); (e) graph of protein concentration change from the genome expressing the French flag pattern.

Unlike the problem of growing the two- or three-layer cube structure, where one gene had to finish forming the corresponding pattern before the next gene could become activated, there is more flexibility in the activation sequence needed to grow the French flag structure. In particular, after the central white cube is fully formed, the genes that extend this cube to either side can be activated in any order, and their corresponding activations can even alternate before either one has finished growing (Chavoya and Duthen, 2007b). However, the requirement of simultaneously evolving both the correct gene activation sequence and the appropriate morphogen threshold activation values, caused that the genetic algorithm could not easily evolve the genomes to produce the desired pattern. In fact, the result presented corresponded to the only successful case that produced the desired French flag structure out of over 500 experiments, whereas the experiments that produced genomes to grow the three-layer cube had an 18% success rate and the experiments that generated the two-layer cube were always successful.

8. Conclusions

The results presented in this paper show that a GA can give reproducible results in evolving an ARN to grow predefined simple 3D cellular structures starting with a single cell. In particular, simulations showed that the combination of a GA and CA with a 3D Margolus interaction neighborhood was a feasible choice for modeling 3D pattern generation.

Morphogenetic fields should in principle assist in the creation of more complex patterns by providing positional constraints to cellular growth. However in the results obtained with the extended ARN with morphogenetic fields, it was apparently harder for the GA to find an activation sequence for the creation of the 3D French flag pattern. One possible explanation is that with the addition of the morphogen threshold activation sites to the ARN, the search space grew even larger than in the original ARN model, making it more difficult for the GA to find an appropriate activation sequence. Despite this apparent disadvantage, without the use of morphogens it would have been impossible to generate the 3D French flag form, as the genes that extend the lateral walls of cells produce a symmetrical pattern along the x-axis when not constrained.

On the other hand, there is evidence that the fitness landscape on which the GA performs the search to evolve the ARNs is very rugged (Chavoya and Duthen, 2007b). In particular this means that vectors that are adjacent in search space have very dissimilar values in fitness evaluation. It is conjectured that this behavior is widespread in the search spaces defined in the models developed, given the difficulties encountered when synchronizing more than two structural genes. Most likely, a change of representation would aid in the search process. Ideally, a representation should be associated with a smooth fitness landscape in which the search process could easily lead to the discovery of local or global optima in the fitness function. More work is needed to obtain a smoother fitness landscape. One other possibility is to modify the GA crossover operator to limit the exchange of bits at the regulatory gene level and not at the ARN level. In this manner, only one regulatory gene per chromosome would be modified at a time, leaving the other regulatory genes intact. In principle, this approach would limit the side effects caused by exchanging several regulatory genes at the same time when classical single-point crossover is used.

One restriction of the ARN models presented is that all cells synchronously follow the same genetic program, as a sort of biological clock. This has obvious advantages in the synchronization of the behavior of developing cells, but it would also be desirable that cells had an individual program —possibly a separate ARN— in order to react to local unexpected changes in their environment. After all, living organisms do contain a series of gene regulatory networks for development and metabolism control. One could even envision either a hierarchy of ARNs, where some ARNs could be used to regulate others ARNs, or a network of ARNs, where all ARNs could influence and regulate each other.

In general, the framework developed proved to be suitable for generating simple 3D differentiated structures, but more work is needed to explore generation of more complex forms. It is also desirable to study cellular structure formation allowing cell death and cell displacement, as in actual cellular growth. Furthermore, in order to build a more accurate model of the growth process, the use of a more realistic physical environment may be necessary.

One of the long-term goals of this work is to study the emergent properties of the artificial development process. It is our hope that one day it will be feasible to build complex structures composed of distinct parts resulting from the differential expression of artificial genes that interact in an intricate fashion.

References

Baker, R. W., Herman, G. T., 1970. Celia - a cellular linear iterative array simulator. In: Proceedings of the fourth annual conference on Applications of simulation. Winter Simulation Conference, pp. 64–73.

Banzhaf, W., 2003. Artificial regulatory networks and genetic programming. In: Riolo, R. L., Worzel, B. (Eds.), Genetic Programming Theory and Practice. Kluwer, Ch. 4, pp. 43–62.

Beurier, G., Michel, F., Ferber, J., 2006. A morphogenesis model for multiagent embryogeny. In: Rocha, L. M., Yaeger, L. S., Bedau, M. A., Floreano, D., Goldstone, R. L., Vespignani, A. (Eds.), Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems (ALife X). pp. 84–90.

Bongard, J., 2002. Evolving modular genetic regulatory networks. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002). IEEE Press, Piscataway, NJ, pp. 1872–1877.

Bowers, C., 2005. Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals. In: Capcarrere, M. S., Freitas, A. A., Bentley, P. J., Johnson, C. G., Timmons, J. (Eds.), Advances in Artificial Life, Proceeding of the 8th European Conference on Artificial Life: ECAL 2005. Springer, pp. 149–158.

Breukelaar, R., Bäck, T., 2005. Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior. In: GECCO '05. pp. 107–114.

Carroll, S. B., Grenier, J. K., Weatherbee, S. D., 2004. From DNA to Diversity: Molecular Genetics and the Evolution of Animal Design, 2nd Edition. Blackwell Science.

Chavoya, A., 2008. Cell pattern generation in artificial development. In: Rossi, C. (Ed.), Brain, Vision and AI. In-Teh, Croatia, Ch. 4, pp. 73–94.

Chavoya, A., 2009. Artificial development. In: Foundations of Computational Intelligence. Volume 1: Learning and Approximation (Studies in Computational Intelligence). Springer Berlin / Heidelberg, Ch. 8, pp. 185–215.

Chavoya, A., Duthen, Y., 2006. Using a genetic algorithm to evolve cellular automata for 2D/3D computational development. In: GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM Press, New York, NY, USA, pp. 231–232.

Chavoya, A., Duthen, Y., 2007a. An artificial development model for cell pattern generation. In: ACAL '07: Proceedings of the 3rd Australian Conference on Artificial Life. Springer, pp. 61–71.

Chavoya, A., Duthen, Y., 2007b. Evolving an artificial regulatory network for 2D cell patterning. In: Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife'07). IEEE Computational Intelligence Society, pp. 47–53.

Chavoya, A., Duthen, Y., 2008. A cell pattern generation model based on an extended artificial regulatory network. BioSystems 94 (1), 95–101.

Cickovski, T., Aras, K., Swat, M., Merks, R. M. H., Glimm, T., Hentschel, H. G. E., Alber, M. S., Glazier, J. A., Newman, S. A., Izaguirre, J. A., 2007. From genes to organisms via the cell: A problem-solving environment for multicellular development. Computing in Science and Eng. 9 (4), 50–60.

Davidson, E. H., 2006. The Regulatory Genome: Gene Regulatory Networks in Development And Evolution, 1st Edition. Academic Press.

de Garis, H., 1999. Artificial embryology and cellular differentiation. In: Bentley, P. J. (Ed.), Evolutionary Design by Computers. Morgan Kaufmann Publishers, Inc., San Francisco, USA, pp. 281–295.

Devert, A., Bredeche, N., Schoenauer, M., 2007. Robust multi-cellular developmental design. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, New York, NY, USA, pp. 982–989.

Eggenberger, P., 1997. Evolving morphologies of simulated 3D organisms based on differential gene expression. In: Harvey, I., Husbands, P. (Eds.), Proceedings of the 4th European Conference on Artificial Life. Springer, pp. 205–213.

Gordon, T. G. W., Bentley, P. J., 2005. Bias and scalability in evolutionary development. In: GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation. ACM, New York, NY, USA, pp. 83–90.

Harding, S. L., Miller, J. F., Banzhaf, W., 2007. Self-modifying cartesian genetic programming. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM, New York, NY, USA, pp. 1021–1028.

Herman, G. T., Liu, W. H., 1973. The daughter of celia, the french flag and the firing squad. In: WSC '73: Proceedings of the 6th conference on Winter simulation. ACM, New York, NY, USA, p. 870.

Holland, J. H., 1992. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. MIT Press, Cambridge, MA, USA.

Joachimczak, M., Wróbel, B., 2008. Evo-devo *in silico*: a model of a gene network regulating multicellular development in 3D space with artificial physics. In: Bullock, S., Noble, J., Watson, R., Bedau, M. A. (Eds.), Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press, Cambridge, MA, pp. 297–304.

Knabe, J. F., Nehaniv, C. L., Schilstra, M. J., 2008. Evolution and morphogenesis of differentiated multicellular organisms: autonomously generated diffusion gradients for positional information. In: Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems. MIT Press, pp. 321–328.

Kuo, P. D., Leier, A., Banzhaf, W., September 2004. Evolving dynamics in an artificial regulatory network model. In: Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tino, P., Kabán, A., Schwefel, H.-P. (Eds.), Proceedings of the Parallel Problem Solving from Nature Conference (PPSN-04). LNCS 3242. Springer, Berlin, Birmingham, UK, pp. 571–580.

Lindenmayer, A., Rozenberg, G., 1972. Developmental systems and languages. In: STOC '72: Proceedings of the fourth annual ACM symposium on Theory of computing. ACM, New York, NY, USA, pp. 214–221.

Meinhardt, H., 1982. Models of Biological Pattern Formation. Academic Press, London.

Miller, J. F., Banzhaf, W., Oct. 2003. Evolving the program for a cell: from French flags to Boolean circuits. In: Kumar, S., Bentley, P. J. (Eds.), On Growth, Form and Computers. Academic Press, pp. 278–301.

Mitchell, M., 1996. An introduction to genetic algorithms. MIT Press, Cambridge, MA, USA.

Reil, T., 1999. Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In: Proceedings of the 5th European Conference on Artificial Life (ECAL). Springer Verlag, New York, NY, pp. 457–466.

Turing, A. M., 1952. The chemical basis of morphogenesis. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 237 (641), 37–72.

Tyrrell, A. M., Greensted, A. J., 2007. Evolving dependability. J. Emerg. Technol. Comput. Syst. 3 (2), 7.

Wilensky, U., 1999. NetLogo. http://ccl.northwestern.edu/netlogo/, center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Wolfram, S., 1983. Statistical mechanics of cellular automata. Reviews of Modern Physics 55, 601–644.

Wolpert, L., 1968. The French flag problem: a contribution to the discussion on pattern development and regulation. In: Waddington, C. (Ed.), Towards a Theoretical Biology. Edinburgh University Press, New York, NY, USA, pp. 125–133.

Wu, P., Wu, X., Wainer, G. A., 2004. Applying cell-devs in 3D free-form shape modeling. In: Cellular Automata, 6th International Conference on Cellular Automata for Research and Industry, ACRI 2004. pp. 81–90.