4

Cell Pattern Generation in Artificial Development

Arturo Chavoya
Universidad de Guadalajara
Mexico

1. Introduction

In biological systems, development is a fascinating and very complex process that involves following an extremely intricate program coded in the organism's genome. One of the crucial stages in the development of an organism is that of pattern formation, where the fundamental body axes of the individual are outlined. It is now evident that gene regulatory networks play a central role in the development and metabolism of living organisms. Moreover, it has been discovered in recent years that the diverse cell patterns created during the developmental stages are mainly due to the selective activation and inhibition of very specific regulatory genes.

Over the years, artificial models of cellular development have been proposed with the objective of understanding how complex structures and patterns can emerge from one or a small group of initial undifferentiated cells. An artificial development model that generates cell patterns by means of the selective activation and inhibition of development genes under the constraints of morphogenetic gradients is proposed here. Cellular growth is achieved through the expression of structural genes, which are in turn controlled by an Artificial Regulatory Network (ARN) evolved by a Genetic Algorithm (GA). The ARN determines when cells are allowed to grow and which gene to use for reproduction, while morphogenetic gradients constrain the position at which cells can replicate. Both the ARN and the structural genes constitute the artificial cell's genome. In order to test the functionality of the development program found by the GA, the evolved genome was applied to a cellular growth testbed that has been successfully used in the past to develop simple 2D and 3D geometrical shapes (Chavoya & Duthen, 2006b).

The artificial development model for cell pattern generation was based on the cellular automata (CA) paradigm. CA have previously been used to study form generation, as they provide an excellent framework for modelling local interactions that give rise to emergent properties in complex systems. Morphogenetic gradients were used to provide cells with positional information that constrained cellular replication. After a genome was evolved, a single cell in the middle of the CA lattice was allowed to reproduce until a cell pattern was formed. The model was applied to the canonical problem in cellular development of growing a French flag pattern.

2. Artificial Development

This section covers the main research areas pertaining to artificial development with special emphasis on the work more directly related to the model presented in Section 4.

2.1 Reaction-Diffusion Systems

It is usually attributed to Turing the founding of modern research on artificial development. He suggested in his seminal article on the chemical basis of morphogenesis (Turing, 1952) that an initially homogeneous medium might develop a structured pattern due to an instability of the homogeneous equilibrium, triggered by small random perturbations.

Using a set of differential equations, Turing proposed a reaction-diffusion model where substances called morphogens, or form generators, would react together and diffuse through a medium, which could be a tissue. The system can be fine-tuned with the proper parameters such that at some point the slightest disruption in the equilibrium can be amplified and propagated through the medium generating unpredictable patterns.

Even though his model was based on an oversimplification of natural conditions, Turing succeeded in demonstrating how the emergence of a complex pattern could be explained in terms of a simple reaction and diffusion mechanism using well-known physical and chemical principles.

2.2 Self-Activation and Lateral Inhibition Model

Experiments with biological specimens have demonstrated that development is a very robust process. Development can continue normally even after a substantial amount of tissue from certain parts has been removed from an embryo. However, there are small specialized regions that play a crucial role in the organization of the development process.

In order to explain the long range effect of these small organizing regions on the larger surrounding tissue and the robustness of their influence even after induced interferences, Wolpert introduced the concept of "positional information", whereby a local source region produces a signalling chemical (Wolpert, 1969). This theoretical substance was supposed to diffuse and decay creating a concentration gradient that provided cells with information regarding their position in the tissue.

Nevertheless, the problem remained as to how a local differentiated source region could be generated from a seemingly homogeneous initial cluster of developing cells. Even though many eggs have some predefined structure, all the patterns developed after a number of cell divisions cannot initially be present in the egg. A mechanism must exist that allows the emergence of heterogeneous structures starting with a more or less homogeneous egg.

Gierer and Meinhardt proposed that pattern formation was the result of local self-activation coupled with lateral inhibition (Gierer & Meinhardt, 1972; Gierer, 1981; Meinhardt, 1982). In this model, which has some resemblance to Turing's model, a substance that diffuses slowly, called the activator, induces its own production (autocatalysis or self-activation) as well as that of a faster diffusing antagonist, the inhibitor. These authors suggest that pattern formation requires both a strong positive feedback (autocatalysis) and a long-ranging inhibitor to stop positive feedback from spreading indefinitely (lateral inhibition).

Their results suggest how a relatively simple mechanism of coupled biochemical interactions can account for the generation of very complex patterns. The components of the model are based on reasonable assumptions, since mutual activation and inhibition of

biochemical substances and molecular diffusion actually exist in the real world. In recent years, molecular biology and genetics experiments have given support to many elements of the model.

2.3 Lindenmayer Systems

Lindenmayer systems, or L-systems, were originally introduced as a mathematical formalism for modelling development of simple multicellular organisms (Lindenmayer, 1968). The organism is abstracted as an assembly of repeating discrete structures or modules. The formalism is independent of the nature of the module, which can be an individual cell or a whole functional structure such as a plant branch. An L-system is a formal grammar with a set of symbols and a set of rewriting rules. The rules are applied iteratively starting with the initial symbol. Unlike traditional formal grammars, rewriting rules are applied in parallel to simulate the simultaneous development of component parts of an organism.

One of the main applications of L-systems has been in the modelling of the development of higher plants (Prusinkiewicz & Lindenmayer, 1990). The modelling does not take place at the cellular level. Instead, it is based on a modular construction of discrete structural units that are repeated during the development of plants, such as branches, leaves and petals (Prusinkiewicz, 1993). Initial models did not consider the influence of the environment on development. However, as organisms in nature are an integral part of an ecosystem, an extension to the modelling framework that considered interaction with the environment was introduced (Mech & Prusinkiewicz, 1996).

The use of L-Systems has been extremely fruitful in modelling the development of organisms at a high structural level. Implemented models of plant development that use L-systems are visually striking because of their resemblance to growth seen in real-life plants and trees.

2.4 Biomorphs

Richard Dawkins' well-known Biomorphs were first introduced in his famous book "The Blind Watchmaker" to illustrate how evolution might induce the creation of complex designs by means of micro-mutations and cumulative selection (Dawkins, 1996). Dawkins intended to find a model to counteract the old argument in biology that a finished complex structure such as the human eye could not be accounted for by Darwin's evolution theory.

Biomorphs are the visible result of the instructions coded in a genome that can undergo evolution. Dawkins introduced a constraint of symmetry around an axis so that the resulting forms would show bilateral symmetry, as in many biological organisms. Initially Dawkins thought that the forms produced would be limited to tree-like structures. However, to his surprise, the forms generated were extremely varied in shape and detail. There were biomorphs that roughly resembled insects, crustaceans or even mammals.

This author proposed next an "interactive" evolutionary algorithm, where the user played the part of the selection force. Initially the user has to decide which form he/she wants to evolve, such as a spider or a pine tree, and in each step of the algorithm he/she chooses the biomorph that best resembles the target form (cumulative selection).

Dawkins showed with his models that the evolution of complex structures was indeed feasible in a step by step manner by means of the cumulative selection of the individual that best approached the final structure.

2.5 Artificial Embryogenesis

Hugo de Garis worked on the creation of a self-assembly process that he called "artificial embryogenesis". His motivation was that he believed that in the future, machines would have so many components that a sequential mechanical assembly would not be feasible. He theorized that highly complex machines should be self-assembled in a similar way as biological organisms are developed.

He worked on artificial "embryos" as 2D shapes formed by a colony of cells using the CA paradigm (de Garis, 1991). He developed a model that evolved reproduction rules for CA, with the goal that the final shape of a colony of cells was as close as possible to a predefined simple shape such as a square or a triangle (de Garis, 1992). In this model, cells can only reproduce if there is at least one adjacent empty cell, i.e. only edge cells are allowed to reproduce. Several target shapes, both convex and non-convex were tested. Results showed that convex shapes could be obtained with a fitness value around 95%, but non-convex shapes evolved poorly, with low fitness values.

After these initial results, de Garis concluded that evolving an artificial embryo implies a type of sequential, synchronized unfolding of shapes. For example, after the main body is grown, then the head and limbs can be grown, followed by the emergence of more detailed shapes, such as those corresponding to fingers and toes (de Garis, 1992).

Even though the approach used by de Garis proved the potential of the application of evolutionary techniques to the growth of artificial cells in order to generate desired shapes, his results were of limited success. However, he was one of the first researchers to use the concept of sequential gene activation for the production of artificial cellular structures using the CA paradigm.

2.6 Evolutionary Neurogenesis and Cell Differentiation

Kitano (1990) was another of the first researchers that conducted experiments towards evolving an artificial development system. This author was successful at evolving large neural networks using GAs. He encoded into the GA chromosome the neural network connectivity matrix using a graph generating grammar. Instead of using a direct encoding of the connectivity matrix, a set of rules was created by a grammar overcoming the scalability problem on the cases tested. Previous attempts saw how convergence performance was greatly degraded as the size of the neural network grew larger. The grammar used was an augmented version of Lindenmayer's L-System and used matrices as symbols.

Kitano (1994) later developed a model of neurogenesis and cell differentiation based on a simulation of metabolism. The idea was to see if artificial multicellular organisms could be created using GAs evolving the metabolic rules in the cell genome. Although all cells carry the same set of rules, individual cells can express different rules because of differences in their local environment, thus producing a sort of cell differentiation. Metabolic rules define which kind of metabolite can be transformed into another kind and under what conditions of metabolite concentration and enzyme presence.

2.7 Evolutionary 2D/3D Morphogenesis

Fleischer & Barr (1992) presented a simulation framework and computational testbed for the study of 2D multicellular pattern formation. Their initial motivation was the generation of neural networks using a developmental approach, but their interest soon shifted towards the study of the multiple mechanisms involved in morphogenesis.

Their approach combined several developmental mechanisms that they considered important for biological pattern formation. Previous work from other researchers had individually considered chemical factors, mechanical forces, and cell-lineage control of cell division to account for some aspects of morphogenesis. These authors decided to combine these factors into one modelling system in order to determine how the interactions between these components could affect cell pattern development. They emphasized that it was the interactions between the developmental mechanisms that were at the core of the determination of multicellular and developmental patterns, and not the individual elements of the model.

On the other hand, Eggenberger used an evolutionary approach for studying the creation of neural network and the simulated morphogenesis of 3D organisms based on differential gene expression (Eggenberger, 1997a; Eggenberger, 1997b). His model for simulating morphogenesis includes a genome with two types of elements: regulatory units and structural genes. The regulatory units act as switches to turn genes on and off, while structural genes code for specific substances that are used to modulate developmental processes. Eggenberger's models showed that a number of mechanisms central to development such as cellular growth, cell differentiation, axis definition, and dynamical changes in shape could be simulated using a framework not based on a direct mapping between a genome and the resulting cellular structure. The shapes that emerge in the models are the result of the interaction among cells and their environment.

2.8 METAMorph

METAMorph, which stands for *Model for Experimentation and Teaching in Artificial Morphogenesis*, is an open source software platform for the simulation of cellular development processes using genomes encoded as gene regulatory networks. The design is made by hand and it allows visualization of the resulting morphological cellular growth process (Stewart et al., 2005). As in higher organisms, cellular growth starts in METAMorph with a single cell (the zygote) and is regulated by gene regulatory networks in interaction with proteins. All cells have the same genome consisting of a series of genes. Each gene can produce exactly one protein, although the same protein can be produced by different genes. The main disadvantage of this simulation platform is that the cellular development model has to be designed through a trial and error process that is limited by the designer's ability to introduce the appropriate parameter values. By the authors' account, this trial and error process typically involves a considerable amount of time, since simulation times are usually high due to the parallel nature of the morphogenetic process. To compound the problem, small changes in design can have substantial consequences on the final shape caused by "the butterfly effect."

2.9 Random Boolean Networks

Random Boolean Networks (RBNs) are a type of discrete dynamical networks that consist of a set of Boolean variables whose state depends on other variables in the network. In RBNs, time and state values take only integer values. The first Boolean networks were proposed by Kauffman (1969) as a randomized model of a gene regulatory network. The connections between nodes are randomly selected and remain fixed thereafter. The dynamics of the RBN is determined by the particular network configuration and by a randomly generated binary function, defined as a lookup table for each node.

Depending on the behaviour of the network dynamics, three different phases or regimes can be distinguished: ordered, chaotic and critical (Kauffman, 2004). The critical type of behaviour is usually considered by researchers as the most interesting of the three types. The ordered type is too static to derive useful observations applicable to dynamic systems, whereas the chaotic type is too random to study any kind of reproducible property.

Kauffman suggested that biological entities could have originally been generated from random elements, with no absolute need of precisely programmed elements (Kauffman, 1969). This conjecture was derived from his observations of the complex behaviour of some of these randomly generated networks and the inherent robustness he found in them.

2.10 Artificial Regulatory Networks

Over the years, many models of ARNs have emerged in an attempt to emulate the gene networks found in nature. Reil (1999) was one of the first researchers to propose an artificial genome with biologically plausible properties based on template matching on a nucleotide-like sequence. The genome is defined as a string of digits and is randomly created. Genes in the genome are not predefined, but are identified by a "promoter" sequence that precedes them. As with RBNs and other dynamical systems, three basic types of behaviour were identified: ordered, chaotic, and complex. Gene expression was called ordered if genes were continuously active or inactive throughout the run. If gene expression seemed to be random with no apparent emerging pattern, it was called chaotic. If the expression of genes was considered to be between ordered and chaotic with the formation of identifiable patterns, then it was called complex.

Reil observed that even after manual perturbations in the model, gene expression usually returned to the attractors that emerged previously. It must be emphasized that the artificial genomes endured no evolution. The behaviours observed were the result of the properties of genomes entirely generated at random. Reil hypothesized that robustness in natural genomes might be an inherent property of the template matching system, rather than the result of the natural selection of the most robust nucleotide sequences (Reil, 1999).

An important advancement in the design of an artificial genome model was made by Banzhaf, who designed a genetic representation based on ARNs (Banzhaf, 2003). His genome consists of a randomly generated binary string where special sequences signal the beginning of genes. Each gene has an enhancer and an inhibitor region that regulate the expression of proteins. After a protein has been produced, it is then compared on a bit by bit basis with the enhancer and inhibitor sequences on all genes in the genome affecting their protein expression.

After observing the dynamics of proteins from genomes that had experienced no evolution, Banzhaf used Genetic Programming in an attempt to drive the dynamics of gene expression towards desired behaviours. He started by evolving the genome to obtain a target concentration of a particular protein. He found out that in general the evolutionary process quickly converged towards the target state.

Another author that evolved an ARN in order to perform a specific task was Bongard (2002). He designed virtual modular robots that were evaluated for how fast they could travel over an infinite horizontal plane during a time interval previously specified. The robots are composed of one or more morphological units and zero or more sensors, motors, neurons and synapses. Each morphological unit contains a genome, and at the beginning of the evolution a genome and a motor neuron are inserted into the initial unit. Using his model,

Bongard demonstrated that mobile units could be evolved in a virtual environment. His results suggest that a similar model might be applied in the design of physical robots.

Other authors have performed research on ARNs using a number of approaches. Willadsen & Wiles (2003) designed a genome based on the model proposed by Reil (1999). As in other models, the genome consists of a string of randomly generated integers where a promoter precedes a fixed-length gene. Gene products are generated, which can regulate expression of other genes. While their genome model offered no major improvement over previous models, these authors succeeding in showing that there was a strong relationship between gene network connectivity and the degree of inhibition with respect to generating a chaotic behaviour. Low connectivity gene networks were found to be very stable, while in higher connectivity networks there was a significantly elevated frequency of chaotic behaviour. Flann et al. (2005) used ARNs to construct 2D cellular patterns such as borders, patches and mosaics. They implemented the ARN as a graph, where each node represents a distinct expression level from a protein, and each edge corresponds to interactions between proteins. A protein is influenced when its production or inhibition is altered as the function of other protein concentration levels. A set of differential equations was used to define the rate of production or inhibition. These authors conjectured that complex ARNs in nature might have evolved by combining simpler ARNs. Finally, Nehaniv's research group has worked on ARNs aiming at evolving a biological clock model (Knabe et al., 2006). They studied the evolvability of ARNs as active control systems that responded with appropriate periodic behaviours to periodic environmental stimuli of several types.

2.11 Evolutionary Development Model

Kumar & Bentley (2003) designed a developmental testbed that they called the Evolutionary Development System (EDS). It was intended for the investigation of multicellular processes and mechanisms, and their potential application to computer science. The EDS contains the equivalent of many key elements involved in biological development. It implements concepts such as embryos, cells, cell cytoplasm, cell wall, proteins, receptors, transcription factors, genes and *cis*-regulatory regions.

Cells in the EDS are autonomous agents that have sensors in the form of surface receptors capable of binding to substances in the environment. Depending on their current state, cells can exhibit a number of activities such as division, differentiation shown as an external colour, and apoptosis or programmed cell death. A GA with tournament selection was used to evolve the genomes.

The design of the EDS was probably too ambitious by involving many elements that introduced more variables and interactions in the system than desired. Results obtained with the EDS are not as good as expected, considering the number of concepts involved. The system might prove its true potential with a more complex target cellular structure.

3. 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.

Lindenmayer & 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 & Liu (1973) developed an extension of a simulator called CELIA (Baker & Herman, 1970) and applied it to generate a French flag pattern in order to study synchronization and symmetry breaking in cellular development.

More recently, Miller & Banzhaf (2003) used what they called Cartesian genetic programming to evolve a cell program that would construct a French flag pattern. They tested the robustness of their programs by manually removing parts of the developing pattern. They found that some 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 signalling molecules.

Gordon & 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. Finally, even models for developing evolvable hardware have benefited from the French flag problem as a test case (Tyrrell & Greensted, 2007; Harding et al., 2007).

4. Cell Pattern Generation Model

In the proposed model of artificial development, cellular patterns are generated by means of the selective activation and inhibition of development genes under the constraints of morphogenetic gradients. Cellular growth is achieved through the expression of structural genes, which are in turn controlled by an ARN evolved by a GA. The ARN establishes the time at which cells can reproduce and determines which structural gene to use at each time step. At the same time, morphogenetic gradients constrain the position at which cells can replicate. The combination of the ARN and the structural genes constitutes the artificial cell's genome.

4.1 Cellular Growth Testbed
In order to evaluate the performance of the development program obtained with the model, their evolved genomes were applied to a cellular growth testbed designed to generate simple geometrical shapes (Chavoya & Duthen, 2006b). This growth model is based on the extensively studied CA paradigm.

Cellular automata are simple mathematical models that can be used to study self-organization in a wide variety of complex systems (Wolfram, 1983). CA are characterized by a regular lattice of N identical cells, an interaction neighbourhood template η, a finite set of cell states Σ, and a space- and time-independent transition rule ϕ which is applied to every cell on the lattice at each time step.

In the cellular growth model presented here, a 33×33 regular lattice with non-periodic boundaries was used. 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 template η

used was an outer Moore neighbourhood. The CA's rule ϕ was defined as a lookup table that determined, for each local neighbourhood, the state (empty or occupied) of the objective cell at the next time step. For a 2-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 neighbourhood, where 0 meant an empty cell and 1 meant an occupied cell (Chavoya & Duthen, 2006a).

Figure 1 shows an example of the relationship between a CA neighbourhood template and the corresponding lookup table. For each neighbourhood configuration, the output bit determines whether or not a cell is to be placed at the corresponding objective cell position. In this example, if there is only an active cell at the objective cell's right position, then the objective cell is to be filled with an active cell (second row of the lookup table in Fig. 1). The actual output bit values used have to be determined for each different shape and are found using a GA. For the sake of simplicity, the neighbourhood shown in the figure is an outer Von Neumann template, but as mentioned above the neighbourhood used in the testbed was an outer Moore template with the eight nearest cells surrounding the central objective cell.

Neighborhood template   Lookup Table

n_0	n_1	n_2	n_3	Output bit
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
⋮				⋮
1	1	1	1	0

Figure 1. Relationship between a cellular automaton neighbourhood template and the corresponding lookup table. The output bit values shown are used only as an example

A cell can become active only if there is already an active cell in the interaction neighbourhood. Starting with an active cell in the middle of the lattice, the CA algorithm is applied allowing active cells to reproduce for 100 time steps according to the rule table. During an iteration of the CA algorithm, the order of reproduction of active cells is randomly selected to avoid artifacts caused by a deterministic order of cell reproduction. Finally, cell death is not considered in the present model for the sake of simplicity.

4.2 Morphogenetic Gradients

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 have been termed *morphogens* due to their role in driving morphogenetic processes. In our proposed development 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. 2005). 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. Initial morphogen concentration level was set at 255 arbitrary units, and the source was replenished to the same level at the beginning of each cycle. 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. We defined two orthogonal gradients on the CA lattice, one generated from left to right and the other from top to bottom (Fig. 2).
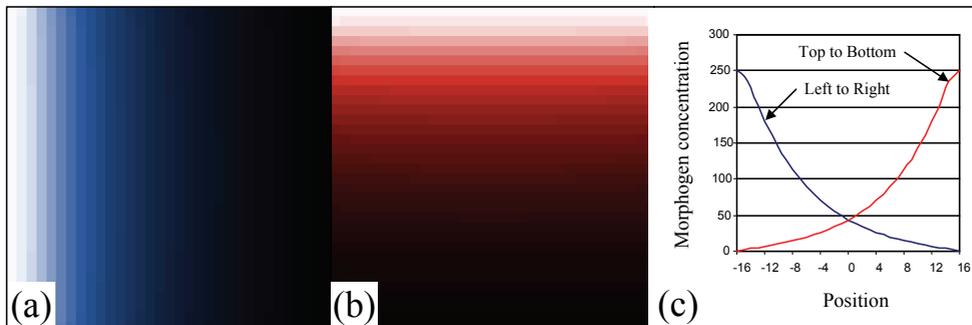

Figure 2. Morphogenetic gradients (a) Left to Right; (b) Top to Bottom; (c) Morphogen concentration graph

4.3 Genome

Genomes are the repository of genetic information in living organisms. They are encoded as one or more chains of DNA, and they regularly interact with other macromolecules, such as RNA and proteins. Artificial genomes are typically coded as strings of discrete data types. The genome used in the proposed model was defined as a binary string starting with a series of regulatory genes, followed by a number of structural genes.

The series of regulatory genes at the beginning of the artificial genome constitutes an ARN. For the sake of simplicity, the term "regulatory gene" is used in this model to comprise both the elements controlling protein expression and the regions coding for the regulatory protein. On the other hand, structural genes code for the particular shape grown by the reproducing cells and they will be described in more detail in Subsection 4.3.2.

4.3.1 Artificial Regulatory Networks

In nature, gene regulatory networks have been found to be a central component of an organism's genome. They actively participate in the regulation of development and in the control of metabolic functions in living organisms (Davidson, 2006). Artificial Regulatory Networks on the other hand are computer models whose objective is to emulate to some extent the gene regulatory networks found in nature. ARNs have previously been used to

study differential gene expression either as a computational paradigm or to solve particular problems.

The ARN model presented here (shown as the series of regulatory genes of the genome in Fig. 3) was originally based on the ARN proposed by Banzhaf (Banzhaf, 2003). However, unlike the ARN model developed by this author, the ARN implemented in the present work does not have promoter sequences and there are no unused intergene regions. All regulatory genes are adjacent and have predefined initial and end positions. Furthermore, the number of regulatory genes is fixed and their internal structure has been modified by adding more inhibitor/enhancer sites and by allowing their role to evolve. 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). Another addition was the incorporation of morphogen threshold activation sites in the regulatory gene.
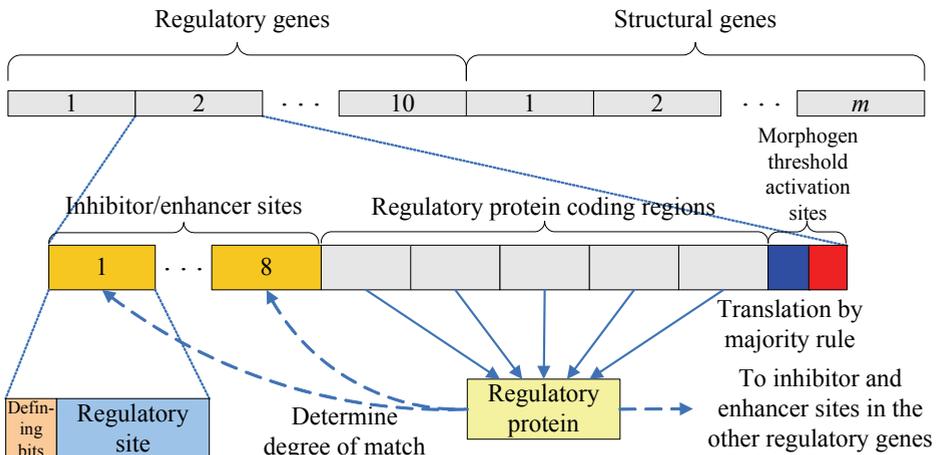

Figure 3. Genome structure and regulatory gene detail

Each regulatory gene consists of a series of eight inhibitor/enhancer sites, a series of five regulatory protein coding regions, and two morphogen threshold activation sites that determine the allowed positions for cell reproduction (Fig. 3). Inhibitor/enhancer sites are composed of a 12-bit function defining region and a regulatory site. The values used for the number of inhibitor/enhancer sites and the number of function defining bits are those that previously gave the best results under the conditions tested (Chavoya & Duthen, 2007c). Regulatory sites can behave either as an enhancer or an inhibitor, depending on the configuration of the function defining bits associated with them. If there are more 1's than 0's in the defining bits region, then the regulatory site functions as an enhancer, 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 (Chavoya & Duthen, 2007b).

Regulatory protein coding regions "translate" a protein using the majority rule, i.e. for each bit position in these regions, the number of 1's and 0's is counted and the bit that is in majority is translated into the regulatory protein. The regulatory sites and the individual protein coding regions all have the same size of 32 bits. Thus the protein translated from the

coding regions can be compared on a bit by bit basis with the regulatory site of the inhibitor and enhancer sites, 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 enhancer 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 enhancer 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

$$en_i = \frac{1}{v} \sum_{j=1}^{v} c_j e^{\beta(u_{ij}^+ - u_{max}^+)} \tag{1}$$

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

where *n* is the total number of regulatory genes, *v* and *w* are the total number of active enhancer 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 enhancer and inhibitor sites of gene *i*, respectively, and u_{max}^+ and u_{max}^- are the maximum matches achievable (32 bits) between a protein and an enhancer 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 calculated using

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

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

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. Parameters β and δ were set to 1.0 and 1.0×10^6, respectively, as previously reported (Chavoya & Duthen, 2007a).

The morphogen threshold activation sites provide reproducing cells with positional information as to where they are allowed to grow on the CA lattice. There is one site for each of the two orthogonal morphogenetic gradients described in Subsection 4.2. 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 two orthogonal morphogenetic gradients, for each pair of morphogen threshold activation levels, the pair of high order bits defines in

which of the four relative quadrants cells expressing the associated structural gene can reproduce. Quadrants can have irregular edges because morphogenetic gradients are not perfectly generated due to local morphogen accumulation close to the non-periodic boundaries of the CA lattice.

Genome size in bits is dependent on the number and size of its component genes. For all simulations the following parameter values were used: The number of structural genes took values from 3, 4 or 8, depending on the experiment performed, as explained in Section 5. The number of regulatory genes was chosen as 10 because this figure was within the range of values previously reported for this kind of ARN (Banzhaf, 2003), and it was found that this value gave a desirable behaviour in the protein concentration variations needed to control cell reproduction. Parameter values for the number of regulatory protein coding regions and the region size in bits are 5 and 32, respectively, and are equal to those used in (Banzhaf, 2003). Finally, structural genes are always 256 bits in length, which results from the use of an outer Moore neighbourhood with its eight cells surrounding the central objective cell. Since each cell in the template can take a value of 1 or 0, the lookup table coding for the structural gene has $2^8 = 256$ rows (Chavoya & Duthen, 2006a).

4.3.2 Structural Genes

Structural genes code for the particular shape grown by the reproducing cells (Chavoya & Duthen, 2006a) and they correspond to the CA rule table's output bits from the cellular growth testbed presented in Section 4.1. Previously to being attached to the regulatory genes to constitute the genome, structural genes were evolved by a GA in order to produce predefined simple 2D shapes, such a square or a line.

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, since the sum of all protein concentrations is always 1.0, and there can only be a protein at a time with a concentration above 0.5. As a result, only one structural gene can be expressed at a particular time step in a cell. If a structural gene is active, then the CA lookup table coded in it is used to control cell reproduction.

In the series of simulations presented in Section 5, the number of structural genes used in the genome depended on the particular pattern grown and this number was always less than the number of regulatory genes. Thus, some regulatory proteins both regulated concentration for other proteins and directly controlled structural gene expression, while other proteins only had a regulatory role. Structural gene expression is visualized in the cellular growth testbed as a distinct external colour for the cell.

4.4 Genetic Algorithm

A simple GA was chosen in this work for evolving the genomes due to the discrete and fixed-size nature of the artificial genome used. Moreover, it was considered that the GA was the evolutionary computation paradigm that resembled the most the actual evolutionary mechanism seen in nature. GAs 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 work uses tournament selection with single-point crossover and mutation as genetic operators. Single-point crossover consists in 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 in 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.

After several calibration experiments, the parameter values described next were considered to be appropriate. The initial population consisted of either 500 binary chromosomes chosen at random for evolving the form generating genes, or 1000 chromosomes for the simulations involving the ARN models. Tournaments were run with sets of 3 individuals randomly selected from the population. Crossover rate was 0.60 in all cases, whereas the mutation was 0.015 for the evolution of structural genes, and 0.15 for the evolution of ARNs. The crossover 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 & Bäck, 2005). As for the mutation rate, it was decided to use a value one order of magnitude higher in the evolution of the ARN models than the one used in the same report, due to the great influence that single bits can have in the convergence towards optimal solutions (Chavoya & Duthen, 2007a). Finally, the number of generations was set at 50 in all cases, since there was no significant improvement after this number of generations.

When evolving the ARNs with the goal of synchronizing the expression of structural genes, the chromosomes used for the GA runs were simply the ARN chains themselves. Chromosome size in this case depended on the values of the parameters chosen. Under the conditions tested, the ARN binary string has a size of 6560 bits, which represents a search space of $2^{6560} \approx 5.7 \times 10^{1974}$ vectors. Evidently, search space grows exponentially with the number of regulatory genes. But even for the simplest of ARNs, the one consisting of only two regulatory genes, the search space has a size of $2^{1312} \approx 8.9 \times 10^{394}$, which is still too large to be explored deterministically. It should be evident that the search space for the ARN model is far too large for any method of exhaustive assessment. Therefore, the use of an evolutionary search algorithm for finding an appropriate synchronization of gene expression is amply justified.

For evolving the ARNs that synchronized the expression of structural genes, the fitness function used by the GA was defined as

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

where c is the number of different coloured shapes, each corresponding to an expressed structural gene, ins_i is the number of filled cells inside the desired shape i with the correct colour, $outs_i$ is the number of filled cells outside the desired shape i, but with the correct

colour, and des_i is the total number of cells inside the desired shape i. In consequence, a fitness value of 1 represents a perfect match. This fitness function is an extension of the one used in (de Garis, 1992), 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 a GA run, each chromosome produced in a generation was fed to the corresponding CA model, where the previously evolved structural genes were attached and the cells were allowed to reproduce controlled by the ARN found by the GA. Fitness was evaluated at the end of 100 time steps in the cellular growth testbed, where a coloured pattern could develop. This process continued until the maximum number of generations was reached or when a fitness value of 1 was obtained.

5. Results

For all experiments, the GA previously described was used to evolve the ARN for the desired coloured patterns. The goal was to combine different coloured shapes expressed by structural genes in order to generate a predefined pattern. After an ARN was obtained and the previously evolved structural genes were attached to constitute the artificial genome, an initial active cell in the middle of the CA lattice was allowed to reproduce controlled by the structural gene activation sequence found by the GA. In order to achieve the desired pattern with a predefined colour for each cell, the genes in the ARN had to evolve to be activated in a precise sequence and for a specific number of iterations. It should be mentioned that not all GA experiments rendered an ARN capable of forming the desired pattern. Furthermore, some difficulties were found when trying to evolve appropriate ARNs for developing patterns involving four structural genes.

In order to explore the result of combining different structural genes that are expressed for a different number of time steps, three different genes were used to grow a French flag pattern. One gene drove the creation of the central white square, while the other two genes extended the central square to the left and to the right, expressing the blue and the red colour, respectively. The last two structural genes do not code specifically for a square, instead they extend a vertical line of cells to the left or to the right for as many time steps as they are activated.

For the generation of the French flag pattern, the central square could be extended to the left or to the right in any of the two orders, that is, first extend to the left and then to the right, or vice versa. This endowed the GA with flexibility to find an appropriate ARN. Figure 4 shows a 27x9 French flag pattern grown from the expression of the three structural genes mentioned above. The graph of the corresponding regulatory protein concentration change over time is shown in 4(e). Starting with a single white cell (a), a white central square is formed from the expression of gene number 1 (b), the left blue square is then grown (c), followed by the right red square (d). The evolved morphogenetic fields are shown for each of the three structural genes. Since the pattern obtained was exactly as desired, the fitness value assigned to the corresponding ARN was the unity (Chavoya & Duthen, 2007d).

In order to explore once again the result of combining different structural genes that are expressed for a different number of time steps, four structural genes were used to grow a French flag with a flagpole pattern. Unlike previous reports where only the French flag itself was produced, the flagpole was added in order to increase the complexity of the pattern generated. The same three structural genes used previously for growing the French flag

pattern were used. The fourth gene added created the brown flagpole by means of growing a single line of cells downward from the lower left corner of a rectangle.
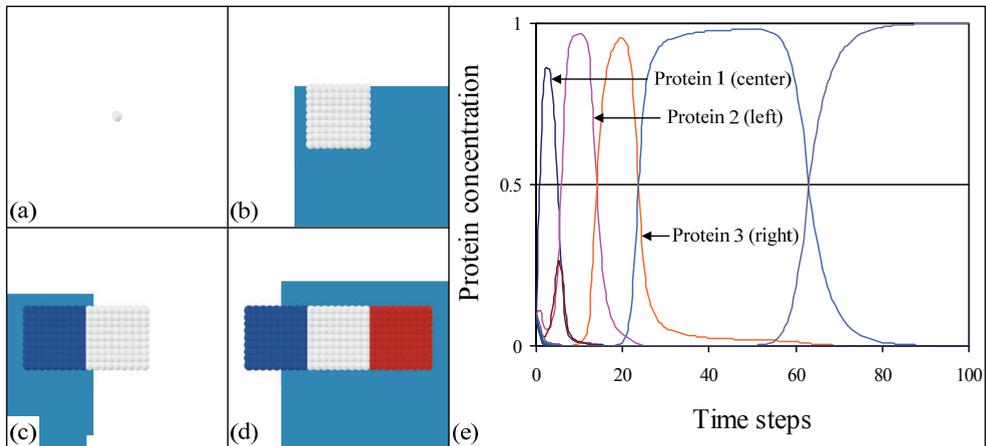

Figure 4. Growth of a French flag pattern. (a) Initial cell; (b) Central white square with morphogenetic field for gene 1 (square); (c) White central square and left blue square with morphogenetic field for gene 2 (extend to left); (d) Final flag pattern with morphogenetic field for gene 3 (extend to right); (e) Graph of protein concentration change from the genome expressing the French flag pattern

When trying to evolve an ARN to produce the French flag with a flagpole pattern, it was found that the GA could not easily evolve an activation sequence that produced the desired pattern. In consequence, it was decided to use the approach of setting a tandem of two identical series of the four structural genes that could produce the desired pattern. In that manner, for creating the white central square, the ARN could express either structural gene number 1 or gene number 5, for the left blue and right red squares it could use genes 2 or 6, or genes 3 or 7, respectively, and finally for the flagpole it could express structural genes 4 or 8. In this way the probability of finding an ARN that could express a French flag with a flagpole pattern was significantly increased.

The 21x7 French flag with a flagpole pattern produced by the expression of this configuration of structural genes is shown in Fig. 5. The graph for the corresponding regulatory protein concentration change is shown in 5(e). After the white central square is formed (a), a right red pattern (b) and the left blue square (c) are sequentially grown, followed by the creation of the flagpole (d). The evolved morphogenetic fields are shown for each of the four structural genes expressed. Note that the white central square is formed from the activation of the first gene from the second series of structural genes, while the other three genes are expressed from the first series of the tandem. It should also be noted that the last column of cells is missing from the red right square, since the morphogenetic field for the gene that extends the red cells to the right precluded growth from that point on (Fig. 5(b)). On the other hand, from the protein concentration graph in 5(e), it is clear that this morphogenetic field prevented the growth of red cells all the way to the right boundary, as gene 3 was active for more time steps than those required to grow the appropriate red square pattern. The fitness value assigned to this pattern was 0.96, which corresponded to

the most successful simulation obtained when trying to grow this particular pattern (Chavoya & Duthen, 2007d).
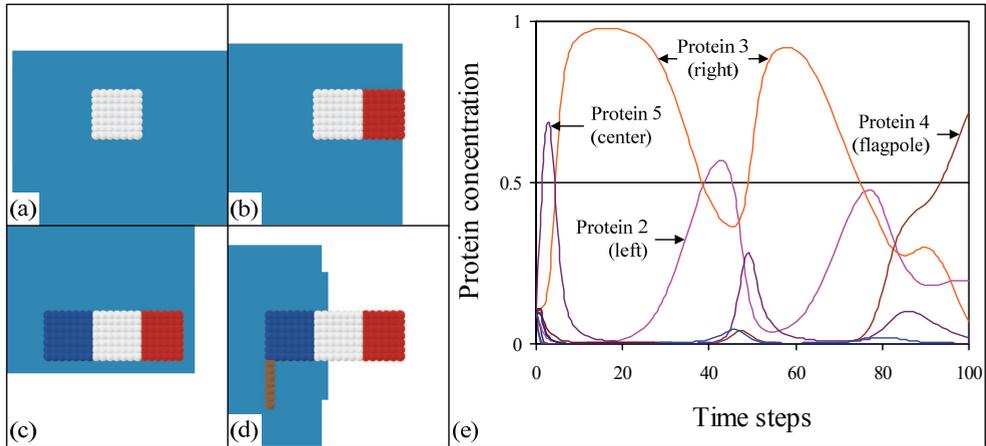

Figure 5. Growth of a French flag with a flagpole pattern. (a) Central white square with morphogenetic field for gene 5 (square); (b) White central square and right red pattern with morphogenetic field for gene 3 (extend to right); (c) White central square, right red pattern and left blue square with morphogenetic field for gene 2 (extend to left); (d) Finished flag with a flagpole pattern with morphogenetic field for gene 4 (flagpole); (e) Graph of protein concentration change from the genome expressing the French flag with a flagpole pattern

Unlike the problem of growing a sequential pattern, where one gene had to finish forming the corresponding shape before the next gene could become activated, there is a certain amount of flexibility in the activation sequence needed to grow a French flag pattern. In particular, after the white central square is fully formed, the genes that extend the central square to either side can be activated in any order, and their corresponding activations can even alternate before either one has finished growing (Chavoya & Duthen, 2007a). However, in the case of the French flag with a flagpole pattern, unless the morphogenetic fields preclude growth of cells at undesired locations, it is essential that the flag is fully formed before the flagpole can begin to grow. It is evident that the left blue square has to be complete in order to start growing the flagpole at the correct position, but consider the case where the right red square is not fully formed after the flagpole, or part of it, was grown. In this case, if the gene that extends a vertical line of cells to the right is activated, it would not only produce the cells required to finish the red right square, but it would equally start to extend the flagpole to the right if allowed by the corresponding morphogenetic field, since the flagpole also consists of a vertical line of cells.

6. Conclusions

As is often the case, by studying how nature works, insight can be gained that aid in proposing approaches for solving a particular problem. In this case, it was decided that the number of enhancer and inhibitor sites in the regulatory network could be increased with respect to the original ARN model, as biological gene regulatory networks usually contain a number of such sites. Likewise, the role as enhancer or inhibitor of the regulatory sites was

allowed to be evolved, as is the case in biological genomes, where the role of regulatory sites depends on the particular nucleotide sequence present at the appropriate places.

Simulations involving the artificial development model proposed show that a GA can give reproducible results in evolving a genome to grow predefined simple 2D cell patterns starting with a single cell. In particular, it was found that using this model it was feasible to reliably synchronize up to three structural genes. However, some problems were encountered when trying to synchronize the activation of more than three structural genes in a precise sequence. Despite its limitations, this model demonstrated that the synchronization of structural genes similar to the gene expression regulation found in nature was feasible.

In a previous model, apart from the gene activation sequence coded in the genome, cells only had local information to determine whether or not to reproduce. In particular, cells had no global positional information, since the shape grown was mainly due to a self-organizing mechanism driven by the ARN (Chavoya & Duthen, 2007b). However, in order to achieve more complex shapes, it was considered necessary to allow cells to extract information from their environment through the use of diffusing morphogens.

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 present model, it was apparently harder for the GA to find an activation sequence for the creation of the French flag with a flagpole 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 previous ARN model, making it more difficult for the GA to find an appropriate activation sequence. However, since individual simulation times usually took several hours to complete, it could be that the number of simulations essayed was not high enough to draw an unambiguous conclusion.

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. This has been illustrated previously with the influence of single bits on the fitness values of an evolving model. In one of the simulations, it took the shift of one bit value in the genome string of the basic ARN model to go from a fitness value of 0.50 to 0.93, and one additional single bit shift led the fitness value to a perfect match (Chavoya & Duthen, 2007a). In this particular case, that meant that adjacent vectors in the search space had very dissimilar values in fitness evaluation. It is conjectured that this behaviour is widespread in the search spaces defined in the model developed, given the difficulties encountered in synchronizing what could be considered just a handful of structural genes.

One restriction of the model presented is that all cells synchronously follow the same genetic program, as a sort of biological clock. This has obvious advantages for synchronizing the behaviour of developing cells, but it would also be desirable that cells had an individual program —possibly a separate ARN— for reacting to local unexpected changes in their environment. Morphogenetic fields provide a means to extract information from the environment, but an independent program would lend more flexibility and robustness to a developing organism. 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.

Additional work is needed in order to explore pattern formation of more complex forms, both in 2D and 3D. It is also desirable to search for a development model that can reliably synchronize the activation of more than four genes. In order to achieve the activation sequence of five or more structural genes using the approach presented of ARN synchronization, it is probably necessary to change the representation of the model, so that a smoother fitness landscape could be obtained. Furthermore, in order to increase the usefulness of the model, interaction with other artificial entities and extraction of information from a more physically realistic environment may be necessary. Until now this work has been devoted to generating predefined patterns in a kind of directed evolution. However, it would be desirable to let cells evolve into a functional pattern under environmental constraints without any preconceived notion of the final outcome.

The approach used in the model proposed was used to shed light on the problem of determining how the physical arrangement of cells in body structures is achieved. However, it is not difficult to see that the spatial distribution of cells can have a decisive role in determining aspects of biological function. As an example, the distribution of neurons in the developing brain can constrain the creation of synapses and hence have an influence on the patterns of electrical and chemical signals that can travel through the neural paths.

The long-term goal of this work is to study the emergent properties of the artificial development process. It can be envisioned that one day it will be feasible to build highly complex structures arising mainly from the interaction of myriads of simpler entities.

7. References

Baker, R.W. & Herman, G.T. (1970). Celia - a cellular linear iterative array simulator, *Proceedings of the Fourth Annual Conference on Applications of Simulation*, pp. 64-73, Winter Simulation Conference

Banzhaf, W. (2003). Artificial regulatory networks and genetic programming. In: *Genetic Programming Theory and Practice,* Riolo, R.L. & Worzel, B. (Ed.), 43-62, Kluwer

Beurier, G.; Michel, F. & Ferber, J. (2006). A morphogenesis model for multiagent embryogeny, *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, *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, pp. 1872-1877, Honolulu, USA, May 2002, IEEE Press, Piscataway, NJ

Bowers, C.P. (2005). Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals, *Proceedings of the 8th European Conference on Artificial Life (ECAL 2005)*, pp. 149-158, Canterbury, UK, September 2005, Springer

Breukelaar, R. & Bäck, T. (2005). Using a genetic algorithm to evolve behavior in multi dimensional cellular automata: emergence of behavior, *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05)*, pp. 107-114, Washington, D.C. USA, June 2005, ACM Press

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

Chavoya, A. & Duthen, Y. (2006a). Evolving cellular automata for 2D form generation, *Proceedings of the Ninth International Conference on Computer Graphics and Artificial Intelligence 3IA'2006*, pp. 129-137, Limoges, France, May 2006

Chavoya, A. & Duthen, Y. (2006b). Using a genetic algorithm to evolve cellular automata for 2D/3D computational development, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pp. 231-232, Seattle, WA, USA, July 2006, ACM Press, New York, NY, USA

Chavoya, A. & Duthen, Y. (2007a). Evolving an artificial regulatory network for 2D cell patterning, *Proceedings of the 2007 IEEE Symposium on Artificial Life (CI-ALife'07)*, pp. 47-53, Honolulu, USA, April 2007, IEEE Computational Intelligence Society

Chavoya, A. & Duthen, Y. (2007b). Use of a genetic algorithm to evolve an extended artificial regulatory network for cell pattern generation, *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07)*, pp. 1062, London, UK, July 2007, ACM Press, New York, NY, USA

Chavoya, A. & Duthen, Y. (2007c). A cell pattern generation model based on an extended artificial regulatory network, *Proceedings of the 7th International Workshop on Information Processing in Cells and Tissues (IPCAT'07)*, pp. 149-158, Oxford, UK, August 2007

Chavoya, A. & Duthen, Y. (2007d). An artificial development model for cell pattern generation, *Proceedings of the 3rd Australian Conference on Artificial Life (ACAL'07)*, pp. 61-71, Gold Coast, Australia, December 2007

Davidson., E.H. (2006). *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution,* Academic Press

Dawkins., R. (1996). *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe without Design,* W. W. Norton

de Garis, H. (1991). Genetic programming: artificial nervous systems artificial embryos and embryological electronics, *Proceedings of the First Workshop on Parallel Problem Solving from Nature*, pp. 117-123, Dortmund, Germany, Springer-Verlag, Berlin, Germany

de Garis, H.; Iba, H. & Furuya, T. (1992). Differentiable chromosomes: The genetic programming of switchable shape-genes, *Proceedings of the Second Conference on Parallel Problem Solving from Nature*, pp. 489-498, Brussels, Belgium, September 1992

Devert, A.; Bredeche, N. & Schoenauer, M. (2007). Robust multi-cellular developmental design, *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07)*, pp. 982-989, ISBN, London, UK, July 2007, ACM Press, New York, NY, USA

Eggenberger, P. (1997a). Creation of neural networks based on developmental and evolutionary principles, *Proceedings of the Seventh International Conference of Artificial Neural Networks (ICANN'97)*, pp. 337-342, Springer

Eggenberger, P. (1997b). Evolving morphologies of simulated 3D organisms based on differential gene expression, *Proceedings of the 4th European Conference on Artificial Life (ECAL)*, pp. 205-213, Springer

Flann, N.; Hu, J. ; Bansal, M.; Patel, V. & Podgorski, G. (2005). Biological development of cell patterns: Characterizing the space of cell chemistry genetic regulatory networks, *Proceedings of the 8th European Conference on Artificial Life (ECAL'05)*, pp. 57-66, Canterbury, UK, September 2005, Springer

Fleischer K. & Barr, A.H. (1992). A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis, *Proceedings of the Workshop on Artificial Life (ALIFE'92)*, pp. 389-416, Addison-Wesley

Gierer, A. (1981). Generation of biological patterns and form: Some physical, mathematical, and logical aspects. *Prog. Biophys. Molec. Biol.,* Vol. 37, pp. 1-47

Gierer, A. & Meinhardt, H. (1972). A theory of biological pattern formation. *Kybernetik,* Vol. 12, pp. 30-39

Gordon, T.G.W. & Bentley, P.J. (2005). Bias and scalability in evolutionary development, *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO'05)*, pp. 83-90, Washington, D.C., USA, June 2005, ACM Press, New York, NY, USA

Harding, S.L.; Miller, J.F. & Banzhaf, W. (2007). Self-modifying Cartesian genetic programming, *Proceedings of 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07)*, pp. 1021-1028, ISBN, ACM Press, New York, NY, USA

Herman, G.T. & Liu, W.H. (1973). The daughter of Celia, the French flag and the firing squad, *Proceedings of the 6th Conference on Winter Simulation*, pp. 870, ACM Press, New York, NY, USA

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

Kauffman, S.A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology,* Vol. 22, pp. 437-467

Kauffman, S.A. (2004). *Investigations,* Oxford University Press

Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems,* Vol.4, pp. 461-476

Kitano, H. (1994). A simple model of neurogenesis and cell differentiation based on evolutionary large-scale chaos. *Artificial Life,* Vol. 2, No. 1, pp. 79-99

Knabe, J.F.; Nehaniv, C.L.; Schilstra, M.J. & Quick, T. (2006). Evolving biological clocks using genetic regulatory networks, *Proceedings of the Artificial Life X Conference (ALife 10)*, pp. 15-21, MIT Press

Kumar, S. & Bentley, P.J. (2003). An introduction to computational development, In: *On Growth, Form and Computers,* Kumar, S. & Bentley, P.J., (Ed.), 1-44, Academic Press, New York, NY, USA

Lindenmayer, A. (1968). Mathematical models for cellular interaction in development Parts I and II. *Journal of Theoretical Biology,* Vol. 18, pp. 280-315

Lindenmayer, A. & Rozenberg, G. (1972). Developmental systems and languages, *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, pp. 214-221, ACM Press, New York, NY, USA

Mech, R. & Prusinkiewicz, P. (1996). Visual models of plants interacting with their environment, *Proceedings of SIGGRAPH 96*, pp. 397-410

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

Miller, J.F. & Banzhaf, W. (2003). Evolving the program for a cell: from French flags to Boolean circuits, In: *On Growth, Form and Computers,* Kumar, S. & Bentley, P.J., (Ed.), 278-301, Academic Press, New York, NY, USA

Prusinkiewicz, P. (1993). Modeling and Vizualization of Biological Structures, *Proceedings of Graphics Interface '93*, pp. 128-137, ISBN, May 1993

Prusinkiewicz, P. & Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants,* Springer-Verlag

Reil, T. (1999). Dynamics of gene expression in an artificial genome – implications for biological and artificial ontogeny, *Proceedings of the 5th European Conference on Artificial Life (ECAL'99)*, pp. 457-466, Lausanne, Switzerland, Springer Verlag, New York, NY, USA

Stewart, F.; Taylor, T. & Konidaris, G. (2005). METAMorph: Experimenting with genetic regulatory networks for artificial development, *Proceedings of the 8th European Conference on Artificial Life (ECAL'05)*, pp. 108-117, Canterbury, UK, September 2005, Springer

Turing, A.M. (1952). The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences,* Vol. 237, No. 641, August 1952, pp. 37-72

Tyrrell, A.M. & Greensted, A.J. (2007). Evolving dependability. *J. Emerg. Technol. Comput. Syst.,* Vol. 3, No. 2, pp. 7

Willadsen, K. & Wiles, J. (2003). Dynamics of gene expression in an artificial genome, *Proceedings of the IEEE 2003 Congress on Evolutionary Computation*, pp. 199-206, IEEE Press

Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics,* Vol. 55, pp. 601-644

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

Wolpert, L. (1969). Positional information and the spatial pattern of cellular differentiation. *J. Theor. Biol.,* Vol. 25, pp. 1-47

Brain, Vision and AI

Edited by Cesare Rossi

The aim of this book is to provide new ideas, original results and practical experiences regarding service robotics. This book provides only a small example of this research activity, but it covers a great deal of what has been done in the field recently. Furthermore, it works as a valuable resource for researchers interested in this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Arturo Chavoya (2008). Cell Pattern Generation in Artificial Development, Brain, Vision and AI, Cesare Rossi (Ed.), ISBN: 978-953-7619-04-6, InTech, Available from:
http://www.intechopen.com/books/brain_vision_and_ai/cell_pattern_generation_in_artificial_development

INTECH
open science | open minds