

# The Evolution of Genetic Regulatory Networks for Single and Multicellular Development

Sanjeev Kumar

Krasnow Institute for Advanced Study  
George Mason University  
Fairfax, VA, 22030, USA  
[kumars@cs.gmu.edu](mailto:kumars@cs.gmu.edu)

**Abstract.** Recently, researchers have recognised the benefits of learning from biological development in order to engineer self-organizing solutions to problems. Building upon previous work, this paper explores the application of the developmental metaphor to the problem of controlling single and multicellular development. In this paper, a summary of experiments performed using a multicellular test-bed model of biological development, the Evolutionary Developmental System (EDS), is presented. The EDS is shown to successfully evolve genetic regulatory networks that specify and control the behaviour of single cells and the construction of 3D multicellular geometric morphologies to explore self-organization and phenomena akin to biological cell differentiation in multicellular development.

## 1 Introduction

Artificial life and developmental biology overlap on some quite important topics. One obvious topic is that of construction, another is control. Controlling the robust construction of complex adaptive systems in a self-organizing manner is a very difficult problem that highlights fundamental issues of scalability, modularity, self-organization, and self-repair. Nature has, however, seemingly solved these problems through the evolution of development—the process or set of processes responsible for constructing organisms [8]. How does the genome control the transformation of a single cell into a complex multicellular system with well defined structures and form? Inspired by biological development, computational development is seen as a potential solution to such problems [5].

Development offers an oft-forgotten alternative route to problems of control [7] that has much in common with biology—Genetic Regulatory Networks (GRNs). In this paper, I build upon previous work [4] and continue the application of the developmental metaphor to the problem domains of self-organization and control.

This work uses a software test-bed, the Evolutionary Developmental System (EDS), to evolve genetic regulatory networks that control the synthesis and decay of proteins, while in turn specifying elaborate developmental programs that construct varied 3D morphologies. Section 2 presents an overview of a novel biologically plausible model

of development. Section 3 details two experiments performed to study the evolution of genetic regulatory networks for two tasks: a) controlling concentration levels of multiple proteins within a single cell and b) controlling the development of multicellular 3D geometric morphologies. The emergence of phenomena akin to biological multicellular differentiation is also shown.

## 2 The Evolutionary Developmental System (EDS)

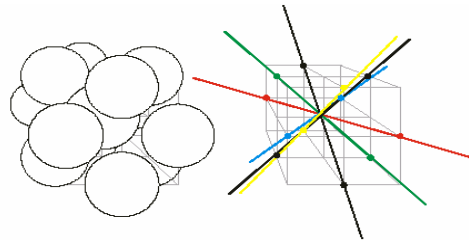
The Evolutionary Developmental System is an object oriented computer model of many of the natural processes of development [4]. At the heart of the EDS lies the developmental core. This implements concepts such as embryos, cells, cell cytoplasm, cell wall, proteins, receptors, transcription factors (TFs), genes, and cis-regulatory regions. Genes and proteins form the atomic elements of the system. A cell stores proteins within its cytoplasm and its genome (which comprises rules that collectively define the developmental program) in the nucleus. The overall embryo is the entire collection of cells (and proteins emitted by them) in some final conformation attained after a period of development. A genetic algorithm is wrapped around the developmental core. This provides the system with the ability to evolve genomes for the developmental machinery to execute. The following sections describe the main components of the developmental model: proteins, genes and cells.

### Proteins

In the EDS, the concept of a protein is captured as an object. In total there are forty proteins (see [4] for more details), each protein having five member variables:

- an ID tag (simply an integer number denoting one of forty six predefined proteins the EDS uses to control cellular behaviour)
- source concentration (storing the concentration of the protein)
- two sets of co-ordinates (isospacial [2] see Fig. 1, and Cartesian)
- a bound variable (storing whether or not a receptor has bound a protein).

(The latter is only used in receptor proteins.)



**Fig. 1.** Isospacial coordinates permit twelve equidistant neighbours for each cell

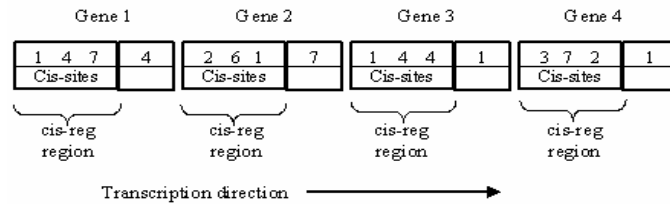
A protein's source concentration variable is responsible for storing the protein's

current concentration. In order to calculate concentration levels for a protein at a distance (during cell signaling, for example) or the creation of a new protein, the appropriate diffusion, production and decay rates are required. Proteins are able to diffuse within an embryo through an implementation that uses a Gaussian function centred on the protein source [4]. All coefficients are evolved and in order to access them the protein's ID tag serves as an index into the genome (which acts as a lookup table).

## Genes

In nature, genes can be viewed as comprising two main regions: the cis-regulatory region [1] and the coding region [6]. Cis-regulatory regions are located just before (upstream of) their associated coding regions and effectively serve as switches that integrate signals received (in the form of proteins) from both the extracellular environment and the cytoplasm. Coding regions specify a protein to be transcribed upon successful occupation of the cis-regulatory region by assembling transcription machinery.

The EDS uses a novel genetic representation termed the cis-trans architecture (fig. 2), based on new empirical genetics data emerging from experimental biology labs [1].



**Fig. 2.** An arbitrary genome created by hand. Genes consist of two objects: a cis-regulatory region and a coding region. Each number denotes a protein

The first portion of the genome contains protein specific values (e.g., protein production, decay, diffusion rates). These are encoded as floating-point numbers. The remaining portion of the genome describes the architecture of the genome to be used for development; it describes which proteins are to play a part in the regulation of different genes. It is this latter portion of the genome that is employed by each cell for development.

Currently, the EDS's underlying genetic model assumes a "one gene, one protein" simplification rule [6, 8] to aid in the analysis of resulting genetic regulatory networks. The genome is represented as an array of Gene objects (Fig. 2). Each gene object contains two members: a cis-regulatory region and a protein coding region. The cis-regulatory region contains an array of TF target sites; these sites bind TFs in order to regulate the activity of the gene. The gene then integrates these TFs and either switches the gene 'on' or 'off'. Integration is performed by summing the products of the concentration and interaction strength (weight) of each TF to find the total activity

#### 4 Sanjeev Kumar

of all TFs occupying a single gene's cis-regulatory region, see eqn. 1 table 1. This sum provides the input to eqn. 3, yielding a probability of the gene firing [3, 4].

**Table 1.** Equations used to calculate the activity of a single gene by summing the weighted product of all transcription factors regulating a single structural gene

<i>Equation</i>	<i>Explanation</i>
$\text{input}_j = \sum_{i=1}^d \text{conc}_i * \text{interaction\_strength}_i \quad (1)$	<p>Where <math>\text{input}_j</math> = total input of all TFs assembling upon the <math>j</math>th gene's cis regulatory region; <math>i</math>, = current TF;</p> <p><math>D</math> = total number of TF proteins visible to the current gene;</p> <p><math>\text{conc}_i</math> = concentration of the <math>i</math>th TF at the centre of the current cell;</p> <p><math>\text{weight}_{ij}</math> = interaction strength between TF <math>i</math> and gene <math>j</math>.</p>
$\text{activity}_j = \frac{\text{input}_j - \text{THRESHOLD\_CONSTANT}}{\text{SHARPNESS\_CONSTANT}} \quad (2)$	<p>Where <math>\text{activity}_j</math> = total activity of the <math>j</math>th gene; <math>\text{input}_j</math> = total input to the <math>j</math>th gene;</p> <p><math>\text{SHARPNESS\_CONSTANT}</math> = a constant taken from the range 0.1-0.001 and is typically set to 0.01.</p>
$\text{activation\_prob}_j = \frac{1 + \tanh(\text{activity}_j)}{2} \quad (3)$	<p>Gene activation probability. Where <math>\text{activation\_probability}_j</math> = activation probability for the <math>j</math>th gene;</p> <p><math>\text{activity}_j</math> = total activity of the <math>j</math>th gene.</p>

#### Cells

Cell objects in the EDS have two state objects: current and new. During development, the system examines the current state of each cell, depositing the results of the protein interactions on the cell's genome in that time step into the new state of the cell. After each developmental cycle the current and new states of each cell are swapped, ready for the next cycle.

The EDS supports a range of different cell behaviours, triggered by the expression of certain genes. The behaviours used for the experiments described in this work are:

- division (when an existing cell “divides”, a new cell object is created and placed in a neighbouring position)
- the creation and destruction of cell surface receptors

- and apoptosis (programmed cell death).

The EDS uses an n-ary tree data structure to store the cells of the embryo, the root of which is the zygote (initial cell). As development proceeds, cell multiplication occurs. The resulting cells are stored as child nodes of parents' nodes in the tree. Proteins are stored within each cell. When a cell needs to examine its local environment to determine which signals it is receiving, it traverses the tree, checks the state of the proteins in each cell against its own and integrates the information.

The decision for a cell to divide in the EDS is governed by the ratio of division activator protein to repressor; the direction (or isospatial axis) the daughter cell is to be placed is non-random and is specified by the position of the mitotic spindle within the cell see [4] for more details.

### 3. Experiments: Multicellular Development and Differentiation

This section details experiments to investigate the evolution of GRNs for both single and multicellular development. GRNs were evolved for two tasks: a) controlling the levels of multiple proteins at specific concentrations within a single cell and b) controlling the development of a multicellular 3D geometric shape, a cube. A common thread connecting both tasks is that of control.

#### 3.1. System Setup

For the single-cell protein control task, fitness was based on the sum of squared differences of the pattern to predefined targets (0.5 for each protein), in other words closeness-of-fit to a flat-line. Final fitness was the sum of repeated assessments of closeness-of-fit over a period of 60 iterations (from 40-100) throughout an individual's lifetime, which was set at 100 iterations.

The experiments in this section used the following parameter settings: a population size of 100 evolving for 250 generations was used. One-point crossover is applied all the time, while Gaussian mutation was applied at a rate of 0.01 per gene. Tournament selection was used with a tournament size of 33 (although 33 is regarded as high, informal experimentation with the system, not reported here, provided this value). 100 developmental iterations, six proteins and two cis-sites per gene were used for the single cell experiment, while the multicellular cube experiment used only 30 developmental iterations and 100 generations. Fitness for multicellular cubes was determined using the equation for a cube. This enabled the number of cells in and out the enclosed cube to be determined, resulting in a function to be minimised see eqn. 4.

$$\text{Fitness} = \left( \frac{1}{\text{cells\_inside}} \right) + \left( \frac{\text{cells\_outside}}{\text{SCALE}} \right) \quad (4)$$

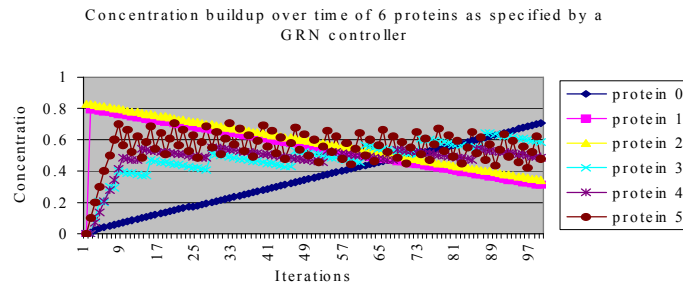
where SCALE refers to a shape dependant constant defining total number of cells in the shape, and was typically set at 512.

## 3.2 Results and Analysis

### 3.2.1 Single cells

The problem of controlling the flat-line protein arrangement requires a GRN capable of specifying the initial formation of the arrangement, and the maintenance of the arrangement throughout the remainder of the individual's lifetime.

In order to get an idea of the dynamics of the system, Fig. 3 shows the concentration buildup over time of the six different proteins for the best evolved GRN. The most salient feature of this graph is how the protein concentrations oscillate in a controlled manner within a range of around 0.4 (between approximately 0.3 and 0.7). This constrained oscillating behaviour is an evolved strategy to maintain protein levels around the required value of 0.5. The constrained behaviour is required by the task and is in stark contrast to the frenzied behaviour exhibited in the graphs of non-control static solutions described in previous work [4].



**Fig. 3.** Protein dynamics over time in a single cell for the flat-line control task of the best GRN

Proteins 4 and 5 buildup quickly and then hover up and down around the desired level of 0.5. Protein 4 is much more accurate than protein 5 maintaining a level of approximately 0.5 throughout. Protein 5 oscillates frantically between approximately 0.4 and 0.7. Proteins 2 and 3 gradually decrease in concentration, however, from iteration 40 they are within a range of approximately 0.3 of the target 0.5. Although protein 0 seems to be the worst, closer inspection reveals it is approximately as accurate as proteins 2 and 3. This is because everything after iteration 39 counts towards fitness and protein 0 is present at around 0.4 at iteration 39 and gradually builds up to around 0.7—a range of approximately 0.3.

Fig. 4 shows the resulting gene expression plot, which shows sub-routine like behaviour through the repeated expression of subsets of genes. Note, white and black boxes, in Fig. 4, denote activated and inactivated genes, respectively. For a fuller analysis see [4].

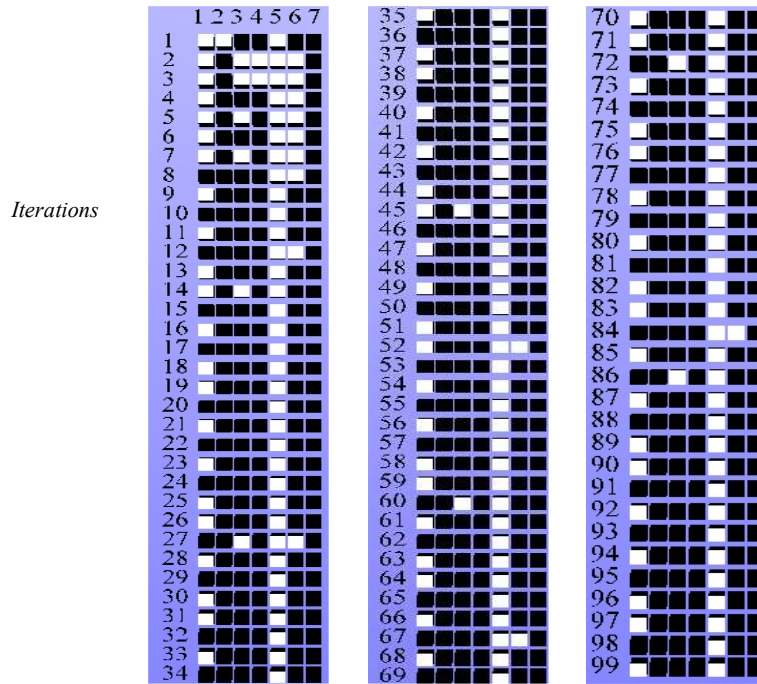
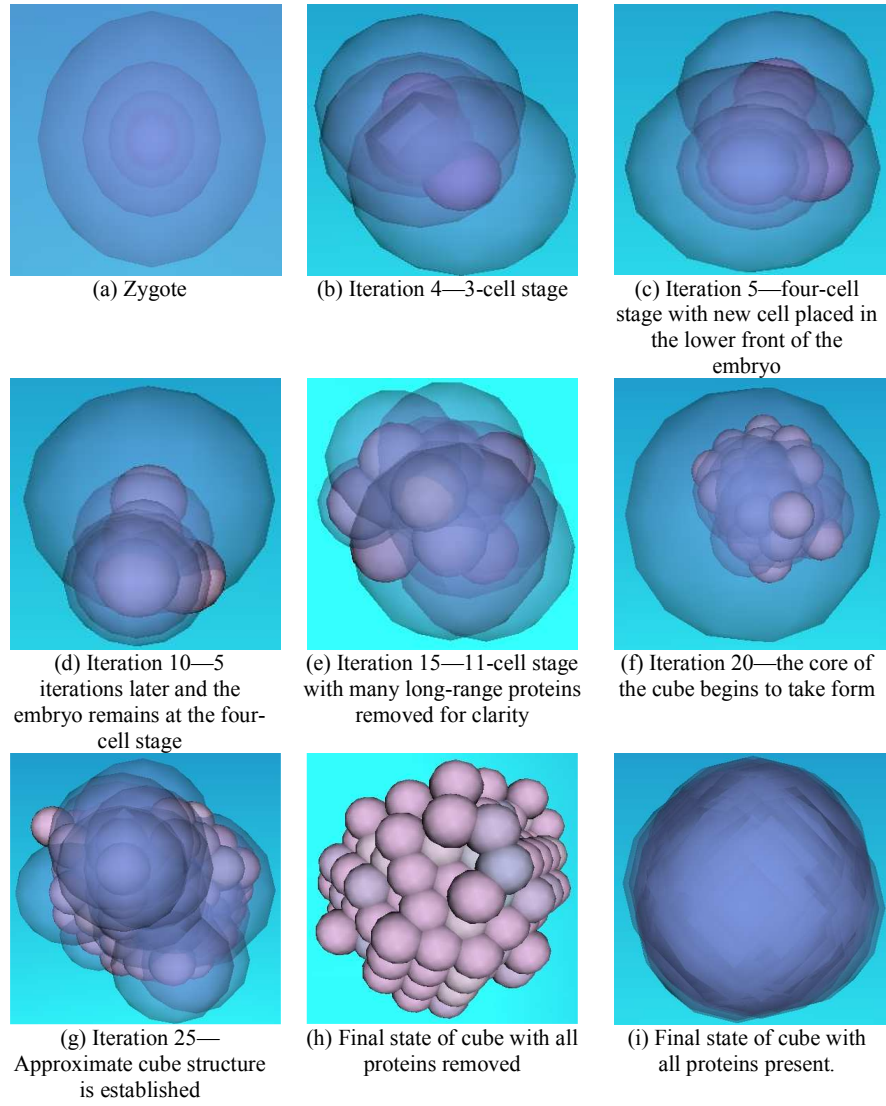


Fig. 4. Gene expression plot for the flat-line control program over 100 iterations

### 3.2.2 Multicellular 3D Morphology

Figure 5 shows snapshots of the development of the best cube, which attained a fitness score of 0.006173. Fig. 6 shows the evolved genome. Evolution has only evolved a single gene for directional control of cell division through gene 7, which emits protein 4. Protein 4 rotates the division spindle anti-clockwise by 1 direction (full analysis is beyond the scope of this paper, see [4]).

In the zygote, for example, two proteins control (more or less) the activation of gene 3: proteins 0 and 4, conferring inhibitory and excitatory stimuli, respectively. In the daughter cell, levels of both proteins 0 and 4 are low due to division, and so do not provide sufficient inhibition or activation (not shown, see [4]). Instead, it falls, not only to other proteins (such as proteins 24 and 37, not shown) to provide inhibition, but also to cell signalling, which initially delivers large inhibitory stimuli through the receptor 13-proteins 4 and 31 signal transduction pathways from the first division in iteration 3. Over time, as receptor 13 decays, so too does the inhibitory stimulus received through that pathway. Leaving the job of inhibiting gene 3 in the daughter



**Fig. 5.** The development of the best cube



[ 32, 3 | 28 ] [ 0, 35 | 20 ] [ 4, 30 | 28 ] [ 26, 28 | 0 ] [ 26, 36 | 30 ] [ 33, 32 | 31 ]  
 [ 25, 8 | 4 ] [ 27, 23 | 21 ] [ 22, 1 | 27 ] [ 37, 8 | 14 ]

Fig. 6. Evolved genome for the best cube

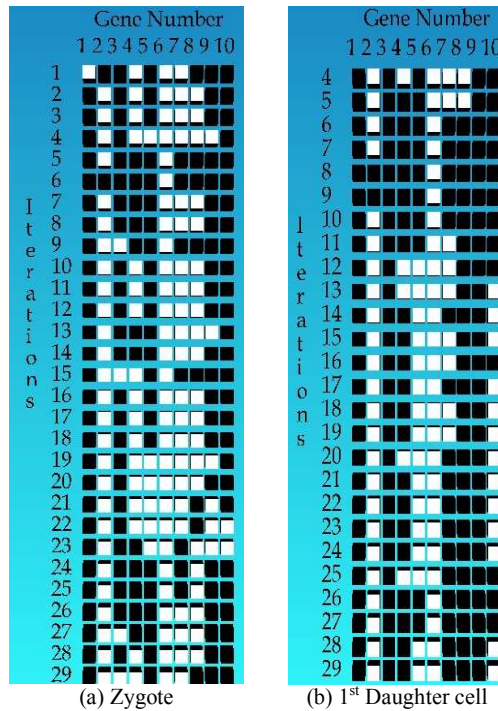


Fig. 7. Gene expression plot for the zygote (a) and the 1st daughter cell (b) of the best cube

cell to an alternative pathway. Note a full analysis is beyond the scope of this paper, see [4].

It must be noted that both cells by virtue of expressing a different subset of genes also have a different subset of active receptors. The zygote begins development with an assortment of receptors, while the daughter cell (and later progeny) inherit their state including receptors from their parent, and then begin to express different genes and consequently different receptors..

In addition, the behaviour of certain receptors (for example, 9 and 10) reflects the fact that different receptors may interact with exactly the same proteins, but the outcome of the interaction may be different. This type of differential receptor-protein interaction is important in development.

The gene expression plots of Fig. 7 reveal important differential gene expression patterns between the two cells, i.e. the cells have differentiated. Noticeably, genes 3 and 9 are expressed, albeit sparingly, in the zygote, but not at all in the daughter cell. Other important differences in gene expression between the two cells are the

expression of genes 7 and 8, which are both increasingly activated, in the zygote, over time, but are seldom activated in the daughter cell.

#### 4. Conclusions

This paper has described successful experiments to evolve genetic regulatory networks using the Evolutionary Developmental System (EDS), an object-oriented model of biological development.

The successful evolution of genetic regulatory networks that are able to control arrangements of proteins within a single cell and specify the construction of 3D, multicellular morphologies that exhibit phenomena such as cell differentiation and subroutining was shown.

#### Acknowledgments

Many thanks to Jeff Bassett, Michel Kerszberg, Paul Wiegand, and Lewis Wolpert for helpful advice and criticism. This research was funded by Science Applications International Corporation.

#### References

1. Davidson, E.H. (2001). *Genomic Regulatory Systems: Development and Evolution*. Academic Press.
2. Frazer, J. (1995) *An Evolutionary Architecture*. Architectural Assoc. London.
3. Kerszberg, M. and Changeux, J-P. (1998). A Simple Molecular Model of Neurulation. In *BioEssays* 20: 758-770.
4. Kumar, S. (2004). *Investigating Computational Models of Development for the Construction of Shape and Form*. Ph.D. Thesis, Department of Computer Science, University College London, UK.
5. Kumar, S. and Bentley, P. (eds) (2004). *On Growth, Form and Computers*. Elsevier Academic Press, London, UK.
6. Lewin, B. (1999). *Genes VI*. Oxford University Press, Oxford, UK.
7. Quick, T. et al. (2003). Evolving embodied genetic regulatory networks-driven control systems. In W. Banzhaf, et al., editors, *Proceedings of the 7<sup>th</sup> European Conference on Artificial Life (ECAL'2003)*. Springer Verlag, 2003.
8. Wolpert, L. et al. (2001). *The Principles of Development*. Oxford University Press, Oxford, UK.