Chapter 4

# ARTIFICIAL REGULATORY NETWORKS AND GENETIC PROGRAMMING

W. Banzhaf
*Department of Computer Science*
*University of Dortmund, D-44221 Dortmund, GERMANY*
banzhaf@cs.uni-dortmund.de

**Abstract** An artificial regulatory network able to reproduce a number of phenomena found in natural genetic regulatory networks (such as heterochrony, evolution, stability and variety of network behavior) is proposed. The connection to a new genetic representation for Genetic Programming is outlined.

**Keywords:** Regulatory Networks, Artificial Evolution, Evolutionary Algorithms, Genetic Programming, Development, Heterochrony

## 1. Introduction

Artificial evolution has proved useful in the optimization of numerical parameters in long-standing combinatorial problems, as well as in structural design problems. The former are known under labels such as Genetic Algorithms [15, 12], Evolutionary Strategies [26, 28] and Evolutionary Programming [10, 9], the latter are mostly known as Genetic Programming [20, 3].

One key problem of all search algorithms is the 'curse of dimensionality' [4]. This expression refers to the exponential growth of the search space's volume as a function of dimensionality. Thus even if there is an evolutionary computation solution to a problem of given size, the same problem with increased dimension might be completely unsolvable. The entire hope of Evolutionary Computation rests with the fact that methods can be implemented which are not volume-based but path-based. This means that history-, or otherwise constraint-dependent search operators scan only a small part of the search space around the present

areas of probing. It is expected that it will be possible to approach good solutions with this method.

In biology, developmental processes restrict the path of evolution to but a tiny fraction of the search space of possible forms and behaviors in Nature [13]. This fact was a point of debate among biologists for more than a century. Why would Nature restrict itself so radically, to the point of risking extinction of entire classes of organisms when changing conditions would dictate adaptations which were impossible to reach from a particular branch of the tree of life? We cannot but think that there is some advantage to this approach in natural search spaces. Perhaps the density of solutions is equal everywhere? Perhaps neutral variations[1] are easily achieved in natural evolutionary systems, so as to relax the constraints somewhat? Perhaps the pressure to achieve solutions quickly resulted in a somewhat stream-lined search process?

Evolutionary and developmental biology were separated for most of the 20th century. However, a recent convergence of principles in biology suggests the evolutionary computation community should also take a closer look at development. Since GP search spaces seem to be closer to natural search spaces (density of solutions, ubiquity of neutral variations), the time has come to explore developmental approaches, notably in connection with GP. A number of researchers have, over the last decade, explored some aspects of developmental approaches to GP, including the genotype-phenotype map [1, 2, 29, 17, 25, 21], the genetic code [11, 18], the mechanisms of gene expression and regulation (see, for example [16]).

Over recent years, a key insight of biologists studying development and evolution was that variety in the living world can be traced back to three genetic mechanisms, active both in development and evolution:

1 Interactions between the products of genes

2 Shifts in the timing of gene expression (heterochrony)

3 Shifts in the location of gene expression (spatial patterning)

If we take this conclusion seriously, and go to the root of its natural implementation, we arrive at regulatory networks. Nature uses regulatory networks as a means to set up and control these mechanisms. This way, Nature unfolds the patterns and shapes of organism morphologies and of their behavior. In addition, regulatory networks mediate between development and evolution, since many evolutionary effects can be followed through their regulatory causes.

[1] Neutral variations are changes in genotypes that do not lead to fitness changes of phenotypes.

How can we make use of these insights in artificial evolutionary systems? Previous work in the area is scattered. Eggenberger [8] has studied the patterning of artificial 3D-morphologies. Reil [27] has set up an artificial genome and studied some consequences for artificial ontogeny. Kennedy [19] examined a model of gene expression and regulation in an artificial cellular organism. Bongard and Pfeifer have considered the relation between evolving artificial organisms and behavior [5].

This chapter presents a new model of an artificial regulatory network which should be useful for algorithms like GP employing structural evolution. The model is a simplification and abstraction of the key elements of protein-genome interaction. It is not yet connected to a semantics of structures. That will be the next step. At this point, we can merely study the inner workings of the model and outline uses for it in GP. Section 2 explains the overall view of the regulatory network model, section 3 views it from the static (structural) perspective, while section 4 looks at the dynamic perspective. Section 5 explains the concept of heterochronic control. Section 6 exemplifies the plasticity of such systems to evolutionary pressure, section 7 discusses stability of steady states and means of communication. Finally, section 8 summarizes the discussion and outlines future steps. Our entire discussion here is qualitative, because only sample networks showing typical behavior will be presented.

## 2. A new genetic representation based on artificial regulatory networks

Our regulatory network model consists of a bit string, the 'genome', and mobile information-carrying molecules, 'proteins', which are equipped with bit patterns for interaction with the genome. Together, they represent a theoretically closed world with a network of interactions between genome and proteins, and a dynamics determined by this network.

A mechanism for reading off genes and for producing proteins with particular bit-patterns is used often called a 'genotype-phenotype mapping'. Proteins are able to wander about and to interact with any pattern on the genome, notably with 'regulatory sites' located upstream from genes. By attaching to these special sites, they can influence the production of (other) proteins. We observe the production of proteins and the dynamics of their concentration changes as a result of the interplay between all the interactions taking place simultaneously.

The genome is implemented as a sequence of 32-bit (integer) numbers. The length of the sequences, $L_G$, determines the length of the genome and is frequently used as a parameter. A particular START pattern, the 'promoter', is used to signal the beginning of a gene on

the bit string (analogous to an open reading frame (ORF) on DNA), starting at the next integer. The signal used is arbitrary and was chosen as 'XYZ01010101', with XYZ arbitrary bytes and the one-byte pattern which in a genome generated by randomly choosing '0's and '1's will appear with a probability of $2^{-8} \approx 0.0039 = 0.39\%$. Genes follow the promoter and have a fixed length of $l_g = 5$ 32-bit integers resulting in an expressed bit pattern of 160 bits for each gene.

Upstream from the promoter site there are two special sites. One enhancer site and one inhibitor site both are of length 32 bits. Attachment of proteins to these sites will result in changes in protein production of the corresponding gene. It is assumed that a very low production of proteins takes place if both sides are unoccupied. Usually, however, there will be proteins around to influence expression rate of a particular gene, and we shall look at that in more detail later. In this simple model, we restrict ourselves to just one regulatory site for expression and one for suppression of proteins. This is a radical simplification with regard to natural genomes, where 5-10 regulatory sites that might even be occupied by complexes of proteins are the rule.

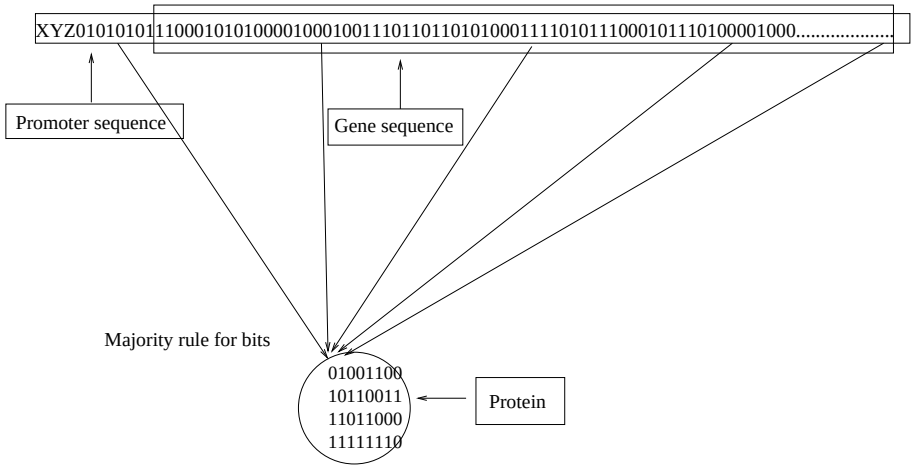


*Figure 4.1.* The genotype-phenotype mapping. Proteins are produced from genes via the genotype-phenotype mapping function.

In this model, we disregard the transcription process completely. Further, there are no introns, no RNA-like mobile elements and no translation procedure resulting in a different alphabet for proteins. Instead, proteins consist of bit patterns of a particular type: Each protein is a 32-bit number resulting from a many-to-one mapping of its gene: On each bit position in the gene's integers the majority rule is applied so

as to arrive at one bit for the protein. In the case of a tie (not possible with an odd number for $l_g$), it is resolved by chance (see Figure 4.1).

Proteins can now be examined as to how they 'match' the genome : Each bit pattern of a protein can be compared to the genome pattern. The comparison is implemented by an XOR operation which results in a "1" if both patterns are complementary. Thus, complementarity between genome and protein bit patterns - and therefore their match - is determined by the number of bits set resulting from this XOR operation. In general, it can be expected that a Gaussian distribution results when measuring the match between a protein and all the bit sequences of a random genome. Notably, there are a few high-matching and a few low-matching positions and many average-matching ones on the genome.

## 3. Static view

Table 4.1 gives three examples of genomes with increasing size. We list the number of genes which roughly follows the 0.39% rule, the maximum match between resulting proteins and their genome at any location, and the number of times such a maximum match has been found. As can be seen the number of proteins with maximum match remains about the same, but their specificity increases.

*Table 4.1.* Sample genomes of increasing size. Specificity of highest matches increase.

| Genome length $L_G$ | Number of genes | Max. match | Freq. max. match |
|---|---|---|---|
| 1,000 | 3 | 25 | 3 |
| 10,000 | 37 | 28 | 4 |
| 10,0000 | 409 | 30 | 3 |

Viewed from a protein's point of view, which scans the genome for good matching sites, Figure 4.2 depicts a typical situation: There is a wide variety of matching degrees, with very few high and low peaks. Average match is 16 bits just in the middle between 0 (min) and 32 bits (max), and statistically a Gaussian distribution emerges.

Let's change perspective and look from the genome's point of view, and more specifically, from the point of view of regulation sites. A number of proteins are produced and floating by, with some providing better matches to the site, other proteins providing worse matches. In principle, each protein has the potential to interact with each regulatory site, and the degree of matching will determine the probability of occupation of a certain site with a certain protein. The situation is depicted in Figure 4.3.

Because proteins are competing for attachment to regulatory sites, the probability of occupation with a particular protein is dependent on the degree of matching of all other proteins to this site. It is therefore necessary to normalize the degree of matching between the various proteins.

Under the simplifying assumption that the occupation of two regulatory sites per gene modulate the expression of the corresponding protein, a network of interactions between genes and proteins can be deduced, which can be parametrized by strength of match. Figure 4.4 shows a sample network, taken from an example genome with 32 genes / proteins.

Although no evolution has taken place (recall these genomes are generated by randomly drawing bits), the network of interactions shows a highly structured view of the resulting interactions. Despite this impression, however, the networks must be considered very complex (in terms of layers vs. participating nodes) and a deep hierarchy of interactions is visible.

A different picture emerges if we slightly change the strategy of generating our random genome. Figure 4.5 shows another genome which was generated by 'growing' a genome from a single 32-bit integer number. A series of length duplications, followed by mutations was applied to arrive at a genome of the same length $L_G = 10,000$. As can be seen from the figure, the result is a much shallower hierarchy, with particular master genes holding sway over an entire set of other genes, and other genes merely connected to their master gene.

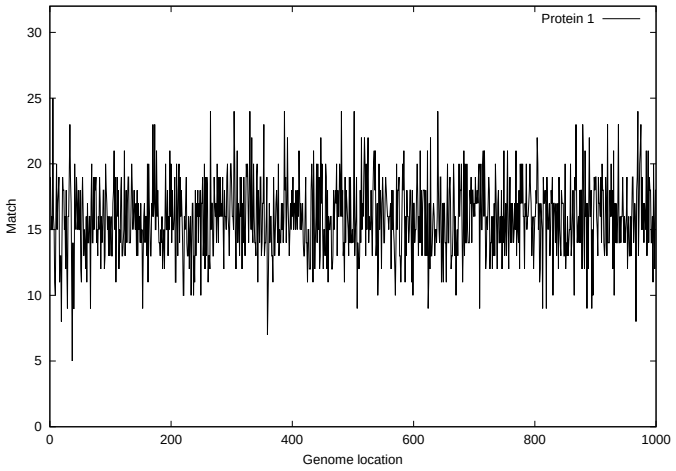


*Figure 4.2.* Matches of a sample protein with its genome. The genome was generated randomly, and matches are visibly distributed randomly as well.

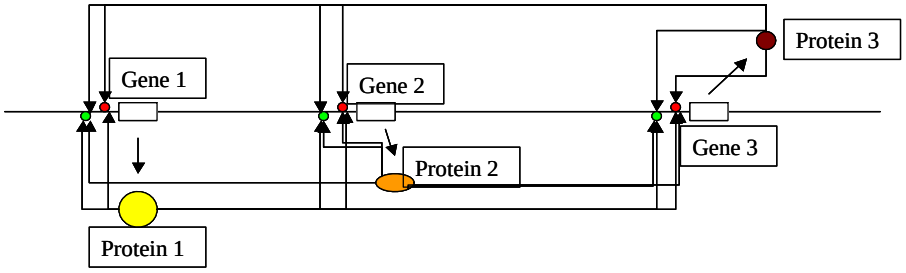


*Figure 4.3.* Interaction between genome and proteins, as seen from the regulatory sites. Each protein matches more or less to all regulatory sites.

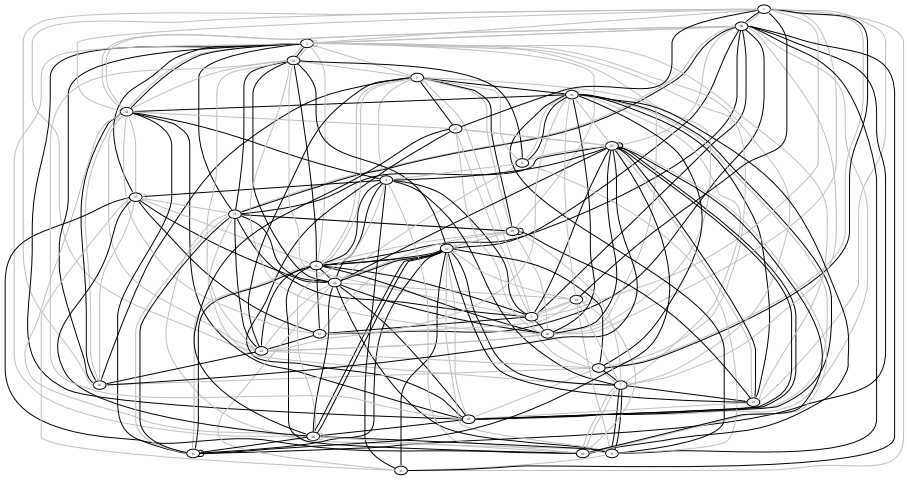

*Figure 4.4.* Network of interactions resulting from matches of proteins with the genome for a larger example. Depicted are 9 out of proteins and their matches with the regulatory sites of all 32 genes of this example ($L_G = 10,000$). Black: Enhancing interactions; Gray: Inhibiting interactions.

As has been observed in natural genome organization [6], shallow hierarchies, up to the point of modularity, are a hallmark of biological organisms. It is interesting to note that a simple process of duplication and divergence suffices to reach a similar state, even from a random genome.

## 4. Dynamic view

Our discussion will now leave the static picture and concentrate on dynamics of the interaction network. A match between protein and regulatory site of a gene leads to activation or inhibition of protein pro-

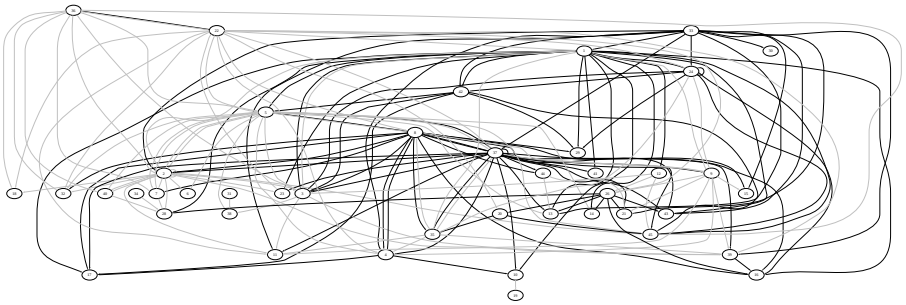

*Figure 4.5.* Network of interactions resulting from matches of proteins with a genome generated through a growth algorithm: A series of genome length duplications is followed by diversifying mutations ($L_G = 10,000$). Matching parameter is at higher values for arriving at a network complexity of approximately the same size as before. Black: Enhancing interactions; Gray: Inhibiting interactions.

duction of the corresponding gene. Generally, the influence of a protein $i$ with $i = 1, ..., n_p$ on an enhancer/inhibitor site is exponential in the number of matching bits, $exp(\beta(u_i - u_{max}))$ where $u_{max}$ is the maximum match achievable.

The concentration of protein molecules $c_j$ of protein $j$ modulates this strength to produce the following excitatory / inhibitory signals for the production of protein $i$:

$$e_i = \frac{1}{N} \sum_j c_j e^{\beta(u_j^+ - \bar{u}_{max}^+)} \tag{4.1}$$

$$in_i = \frac{1}{N} \sum_j c_j e^{\beta(u_j^- - \bar{u}_{max}^-)} \tag{4.2}$$

where a scaling was done as to have a maximum match for the best matching protein, both in excitatory and inhibitory signals.

Given these signals, protein $i$ is produced via the following differential equation

$$\frac{dc_i}{dt} = \delta(e_i - in_i)c_i - \Phi \tag{4.3}$$

A flow term assures that concentrations remain in the simplex, $\sum_i c_i = 1$, resulting in competition between sites for proteins.

If we look at the dynamics of concentration changes of proteins, starting from a state of equal concentration that reflects the native low-level expression of all genes, we can observe some proteins increase their level of concentration, then fall again, with usually one being left over. Thus, a typical dynamic system behavior can be seen, well known under the name 'point attractor' in dynamical systems theory [14].

For different random genomes (different number of genes, matching etc) the dynamics is remarkably different. There are cases of longer and shorter time scales, there are complicated and simple dynamics. Figures 4.6 to 4.9 show four different dynamics resulting form four different genomes.

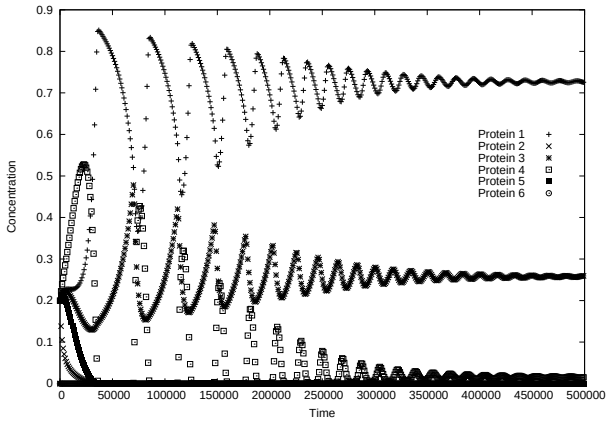


*Figure 4.6.* Time development of protein concentrations. Different dynamical systems are realized by different genomes (see later figures). Here, a dampened (nonlinear) oscillator type of dynamics is exhibited.

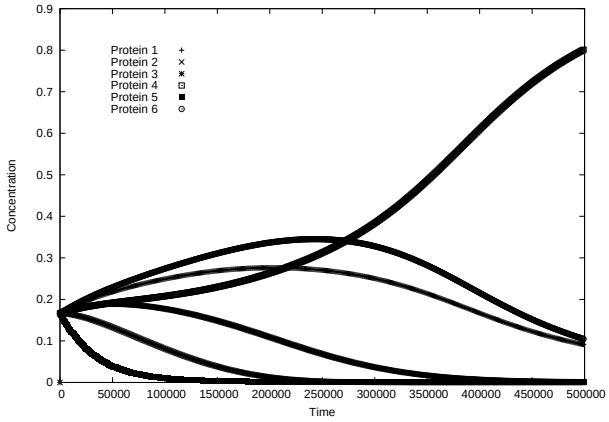


*Figure 4.7.* Time development of protein concentrations. Slow and smooth development of concentrations due to the particular bit patterns - and thus couplings strengths - between genes and proteins.

It should be noted that this richness of dynamics is merely a result of different genomes of the same length, with different patterns for proteins

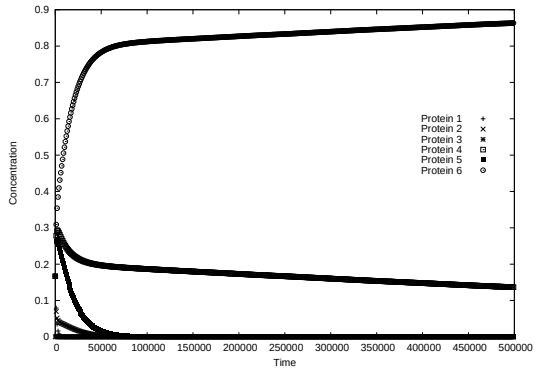


*Figure 4.8.* More examples of dynamics: Quick settlement into a point attractor.

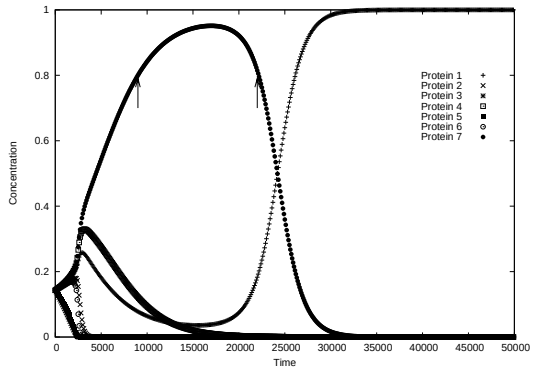


*Figure 4.9.* Extended transition phase with one protein achieving high values of concentration and a subsequent switch to expression of one other gene.

resulting in different matching and regulation results. No development or evolution has yet been put in place.

## 5. Heterochronic control

If we look at this from the perspective of how many proteins are above or below a certain production threshold we can observe the turning on or turning off of genes (on/off could be set equal to x 2 or x 1/2 of initial production, or it could be based on an absolute concentration value). This translates into a timing of onset/offset of gene production. Figure 4.7, right, for instance, shows the timing of onset and offset of concentrations above 0.8 for protein 7 (arrows), $t_{on} = 9,000, t_{off} = 22,000$.

Changing the degree of matching between regulatory sites and proteins by one or two bits can result in dramatic changes in the dynamics, but it need not. Sometimes there are no changes at all and we have a neutral variation. Sample changes that actually varied the expression are shown in Figure 4.10 and 4.11. It is interesting to note that variations in patterns are translated by the ARN into time variations, similar to what was observed in natural GRNs [7]

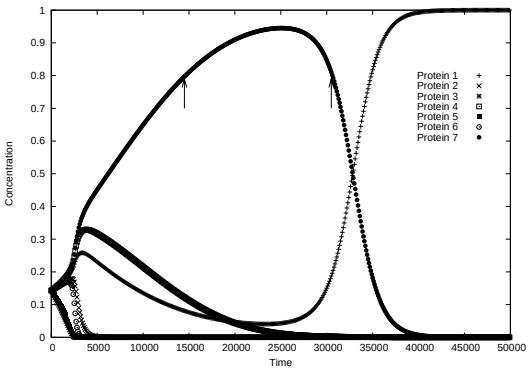


*Figure 4.10.* Genome of Figure 4.9. Degree of matching between protein 7 and inhibitory site to gene 4 changed by one bit. Timing of expression of protein 7 changes substantially: $t_{on} = 14,500, t_{off} = 30,500$.

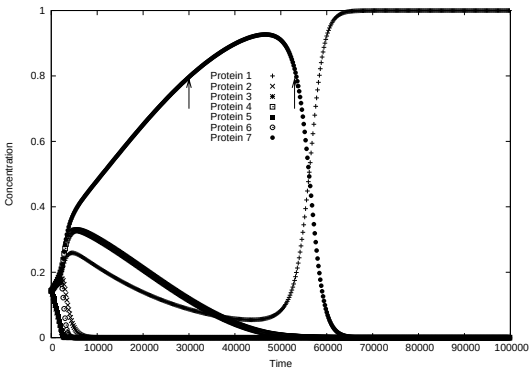


*Figure 4.11.* Genome of Figure 4.9. Degree of matching between protein 7 and inhibitory site to gene 4 increased by another bit, timing changes even further $t_{on} = 30,000, t_{off} = 53,000$.

Heterochrony, i.e. a variation in the timing of onset or offset of certain genes are heavily used in development for generating particular structural effects [23, 22]. As we can see by comparing Figures 4.10 and 4.11,

and 4.9, small changes cause small effects. The same principle could be also of use in physiological reactions, for instance under the control of external factors exceeding certain threshold values.

Interestingly, the range of possible changes is partitioned logarithmically, due to the change of occupation probability, that is depending on an exponentiated matching difference between proteins and DNA bitpatterns. This can be seen most easily, if we put all concentration curves of protein 1 and 7 into one plot, see Figure 4.12. We can clearly see the range of changes expanding with further additions of bit flips.

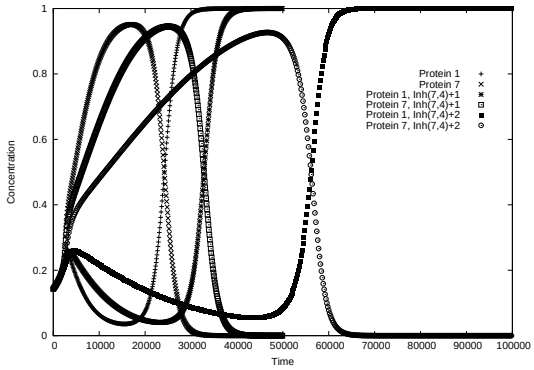


*Figure 4.12.* Genome of Figure 4.7. Degree of matching between protein 7 and inhibitory site to gene 4 changed progressively. Timing of expression of protein 7 changes in increasing step sizes.

## 6. Evolution

The most important question to be addressed with such a model is whether it would be possible to define arbitrary target states and evolve the genome / protein network toward this target state. Our first results in a typical simulation are shown in Figure 4.13. It shows the progress of a network in approaching the target concentration of a particular protein, here protein 6. As we can see, the evolutionary process quickly converges towards this target state. It must be emphasized, that the very simplest way of doing evolution was used here, a $(1 + \lambda)$ evolution strategy, with $\lambda = 1$ [26]. Various experiments were performed with the same genome (not shown here), allowing evolution of other concentration levels for other proteins. We can see from the figure, that steep declines in the deviation (error) curve are followed by apparent stagnation periods. These stagnation periods, are, however, accompanied by continued changes in the genome under evolution. It is merely the mapping of the genome that does not show many consequences of these

variations. By construction we designed a system with many neutral pathways. Evolutionary progress is thus interrupted superficially, but goes on in genomes due to neutral steps.

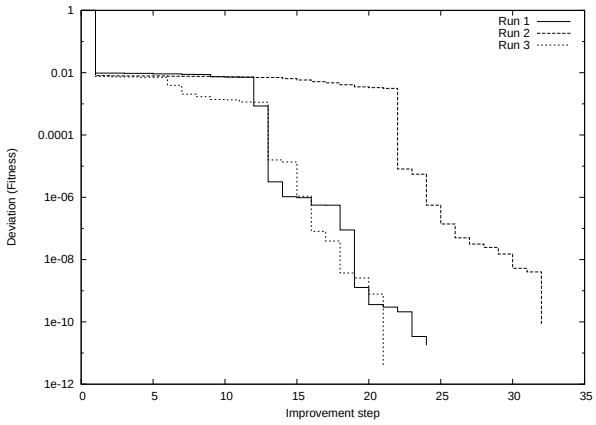


*Figure 4.13.* Evolution at work: 3 different runs of a $(1+\lambda)$ strategy to arrive at a prespecified concentration of one particular protein: $c_6 = 0.085$ at time $t = 100$.

This can be seen if we consider the changes in concentration levels of all proteins at $t = 100$ in Figure 4.14. Here we can discover that all protein concentrations change over time, with many stagnation periods for all proteins. Huge steps are sometimes shown by certain proteins, which are not reflected in the fitness of an individual, due to the focus on measuring only the deviation from $c_6 = 0.085$ for fitness.

When comparing the figures for heterochronic control one cannot but have the impression of rather small variations between the different expression patterns. Changes in exon content could bring about more effects. Figures 4.15 and 4.16 shows two 1-bit mutations in the expressed part of gene 4 and 1, respectively.

Differences are striking, though still some similiarity (particularly in the earlier iterations) remains. Around iteration 60,000, however, a radical switch occurs in the behavior that could not be observed without the mutation. This gives an indication how novelty might be generated in such a system: by slight changes in patterns, entirely new 'phases' appear in the phenotype.

Here we have not been concerned with diffusion and spatial variation, another fruitful area when examining development in biology. Much is already known about the early pre-structuring of organisms and the subsequent unfolding of these patterns into real organs and structures. This process, however, requires that cells are able to communicate with each other, which in turn manifests itself in a common language.

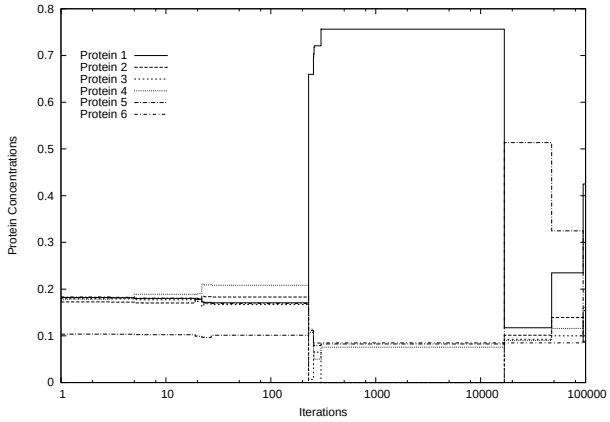


*Figure 4.14.* Evolution at work: Same run as in Figure 4.13, with all protein conentrations protocolled at $t = 100$. As can be seen, protein concentration of selected protein 6 meanders towards goal state $c_6 = 0.085$, whereas other protein concentrations pass through huge swings.

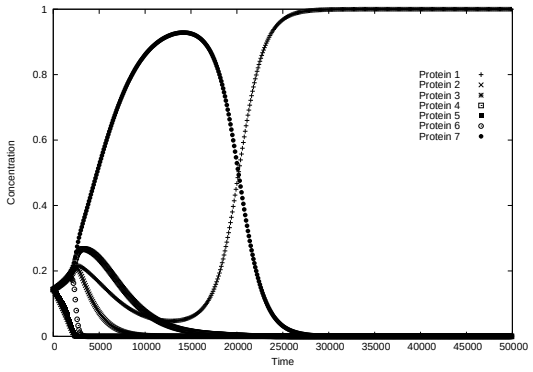


*Figure 4.15.* One-bit mutations in the expressed part of gene 4. Mutation does not change the phenotype.

## 7. Stability and Communication

What would be easier than using the mobile elements producable by cells, to let them carry meaning? This theme is the subject of the present section. We report on a few experiments with proteins that have been added or removed from a network, thus simulating the import from or export to other cells.

Two question will be asked: (i) Is the regulatory network providing a stable environment, so that the export of protein does not perturb the behavior of the network? (ii) Is the regulatory network sensitive

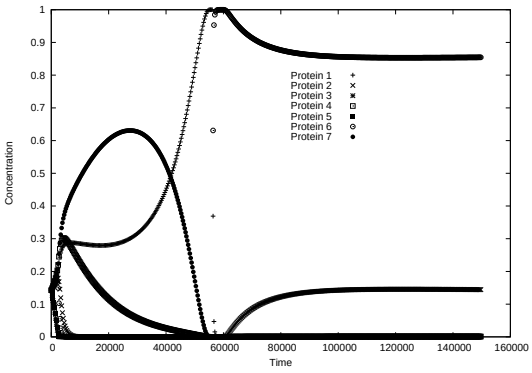


*Figure 4.16.* One-bit mutations in the expressed part of gene 1. Mutation radically changes the phenotype. A completely new behavior, including the dominance of another protein is visible in the right figure.

enough to change its behavior upon the impingement of protein from the outside?

It is clear that both requirements are somewhat contradictory. Nevertheless, both questions can be answered in the affirmative. Some proteins are very stable, regardless whether one adds or removes them, others are sensitive to addition, still others to removal (see Figures 4.17 - 4.22 for examples. Obviously, the network provides again a very rich behavioral environment, where various features can be selected upon.

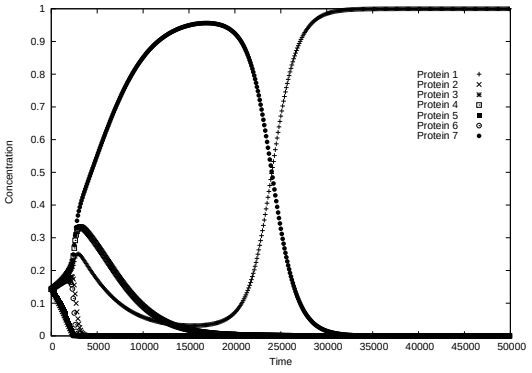


*Figure 4.17.* Removal of protein produced by the network. Removal of protein 1, 10 % . There is only a slight adaption of the dynamics, no real change in the behavior visible.

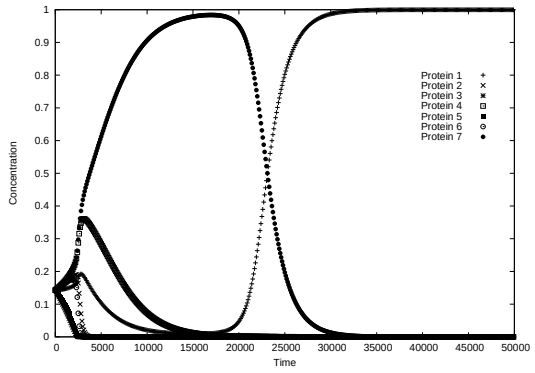


*Figure 4.18.* Removal of protein produced by the network. Removal of protein 1, 100 % . There is only a slight adaption of the dynamics, no real change in the behavior visible.

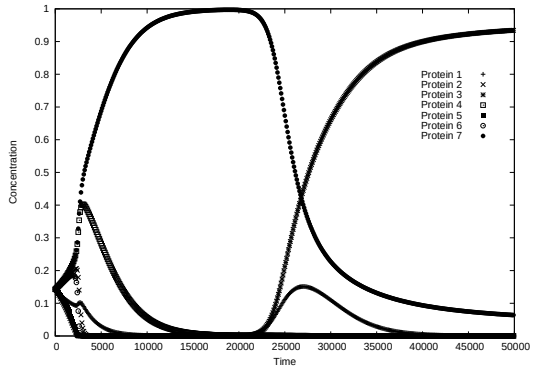


*Figure 4.19.* Removal of protein produced by the network. Removal of protein 1, 300 % . There is only a slight adaption of the dynamics when removing the protein.

## 8. Summary and Perspectives

In this contribution we have shown that a simple model for artificial regulatory networks can be formulated which captures essential features of natural genetic regulatory networks. Although we have only shown qualitative results, the difference in behavior of these networks from usual genetic representations can be seen already from the few examples shown here.

Our next steps are to move from an analyis of qualitative behavior to quantifying certain features like stability and evolvability. Even more interesting, however, is to press ahead and find a proper connection of this type of genetic representation to genetic programming. Here we can only outline our present thinking in this direction.

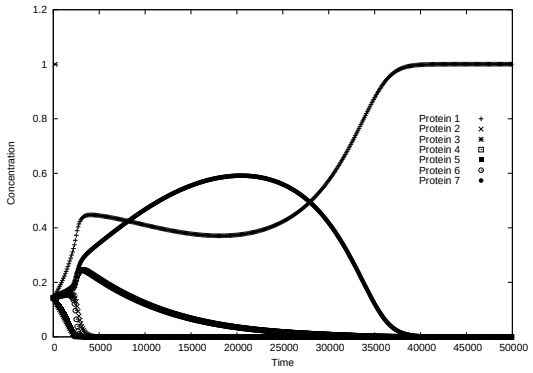


*Figure 4.20.* Addition of protein produced by the network. Addition of protein 1, 200 % . There is a stronger reaction of the dynamics when adding the protein.

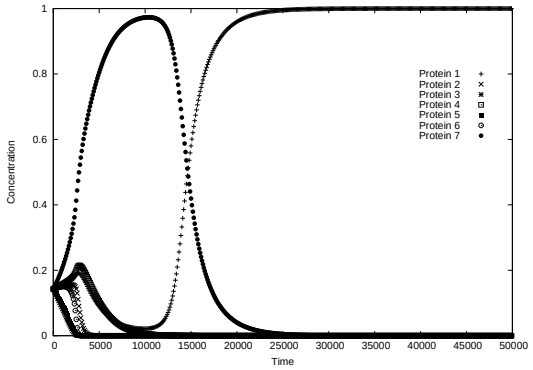


*Figure 4.21.* Addition of protein produced by the network. Addition of protein 7, 200 % . The opposite from Figure 4.20: There is only a slight adaption of the dynamics when adding the protein..

The proteins used here as a means for regulation might carry a second role, namely that of agents or objects behaving in the outside world. This would mean that a translational mechanism would be put into place that translates the protein pattern into another representation useful in the 'outside' world. A plain method, for instance, would be to interpret the 32 bits of the protein as an instruction for a 32-bit processor. There are some GP-systems able to digest bit patterns of arbitrary type and generating useful behavior from it (see [3] and [24]). Concentrations could be used to fix an order for the sequence of instructions to be executed. Much more complex relationships between the information carried by the protein and program constructs are reasonable and will be introduced in due time.

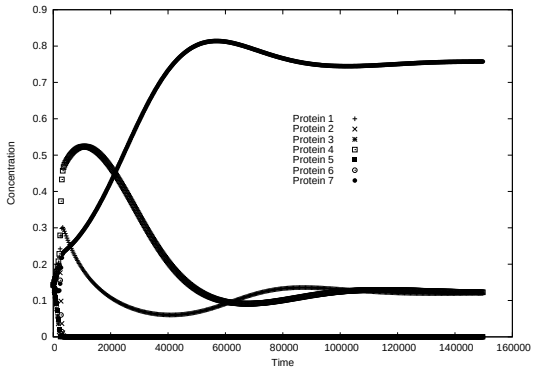


*Figure 4.22.* Removal of protein produced by the network. Removal of protein 7, 200 % . The opposite from Figure 4.19: There is a strong reaction when removing the protein.

## Acknowledgement

The author gratefully acknowledges a sabbatical stay at the Institute for Genomics and Bioinformatics at UC Irvine, where part of the ideas that lead to this work were born. Especially he wants to acknowledge the hospitality of its director, Prof. Pierre Baldi, and of its manager, Mrs. Ann Marie Walker.

## References

[1] Altenberg, L. Genome growth and the evolution of the genotype-phenotype map. In *Evolution as a Computational Process*, W. Banzhaf and F. H. Eeckman, Eds. Springer-Verlag, Berlin, Germany, 1995, pp. 205–259.

[2] Banzhaf, W. Genotype-phenotype-mapping and neutral variation – A case study in genetic programming. In *Parallel Problem Solving from Nature III* (Jerusalem, 9-14 Oct. 1994), Y. Davidor, H.-P. Schwefel, and R. Männer, Eds., vol. 866 of *LNCS*, Springer, pp. 322–332.

[3] Banzhaf, W., Nordin, P., Keller, R., and Francone, F. *Genetic Programming - An Introduction.* Morgan Kaufmann, San Francisco, CA, 1998.

[4] Bellman, R. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, NJ, 1961.

[5] Bongard, J., and Pfeifer, R. Behavioral selection pressure generates hierarchical genetic regulatory networks. In *Proceedings*

*of the Genetic and Evolutionary Computation Conference* (2002), W. B. L. et al., Ed., Morgan Kaufmann, p. 132.

[6] Dassow, G., Meir, E., Munro, E., and Odell, G. The segment polarity network is a robust developmental module. *Nature 406* (2000), 188 – 192.

[7] Davidson, E. *Genomic Regulatory Systems.* Academic Press, San Diego, CA, 2001.

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

[9] Fogel, D. *Evolutionary Computation.* IEEE Press, New York, 1995.

[10] Fogel, L., Owens, A., and Walsh, M. Artificial intelligence through a simulation of evolution. In *Biophysics and Cybernetic Systems* (1965), M. Maxfield, A. Callahan, and L. Fogel, Eds., pp. 131–155.

[11] Freeland, S. J. The darwinian genetic code: An adaptation for adapting? *Genetic Programming and Evolvable Machines 3* (2002), 113 – 127.

[12] Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, Reading/Mass., 1989.

[13] Gould, S. *The Structure of Evolutionary Theory.* Belknap Press of Harvard University Press, Cambridge, MA, 2002.

[14] Hirsch, M., and Smale, S. *Differential Equations, Dynamical Systems and Linear Algebra.* Academic Press, Reading, MA, 1997.

[15] Holland, J. *Adaptation in natural and artificial system.* MIT Press, Cambridge, MA, 1992.

[16] Kargupta, H. Editorial: Special Issue on Computation in Gene Expression. *Genetic Programming and Evolvable Machines 3* (2002), 111 – 112.

[17] Keller, R., and Banzhaf, W. Genetic programming using genotype-phenotype mapping from linear genomes into linear phenotypes. In *Genetic Programming 1996: Proceedings of the First Annual Conference* (Stanford University, CA, USA, 28–31 July 1996), J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds., MIT Press, pp. 116–122.

[18] Keller, R., and Banzhaf, W. The evolution of genetic code in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Orlando, Florida, USA, 13-17

July 1999), W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, Eds., vol. 2, Morgan Kaufmann, pp. 1077–1082.

[19] Kennedy, P., and Osborn, T. A model of gene expression and regulation in an artificial cellular organism. *Complex Systems 13* (2001).

[20] Koza, J. *Genetic Programming: On the Programming of Computers by Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.

[21] Koza, J. R., David Andre, Bennett III, F. H., and Keane, M. *Genetic Programming 3: Darwinian Invention and Problem Solving.* Morgan Kaufman, San Francisco, CA, 1999.

[22] McKinney, M. Heterochrony: Beyond words. *Paleobiology 25* (1999), 149 – 153.

[23] McKinney, M., and McNamara, K. *Heterochrony: The Evolution of Ontogeny.* Plenum Press, New York ,NY, 1991.

[24] Nordin, P., Banzhaf, W., and Francone, F. D. Efficient evolution of machine code for CISC architectures using instruction blocks and homologous crossover. In *Advances in Genetic Programming 3*, L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. J. Angeline, Eds. MIT Press, Cambridge, MA, USA, 1999, pp. 275–299.

[25] O'Neill, M., Ryan, C., Keijzer, M., and Cattolico, M. Crossover in grammatical evolution. *Genetic Programming and Evolvable Machines 4* (2003), 67 – 93.

[26] Rechenberg, I. *Evolutionsstrategie "93.* Frommann Verlag, Stuttgart, 1994.

[27] Reil, T. Dynamics of gene expression in an artificial genome - implications for biological and artificial ontogeny. In *Proceedings of the 5th European Conference on Artificial Life* (1999), D. F. et al., Ed., Springer, pp. 457–466.

[28] Schwefel, H.-P. *Evolution and Optimum Seeking.* Sixth-Generation Computer Technology Series. John Wiley & Sons, Inc., New York, 1995.

[29] Spector, L., and Stoffel, K. Ontogenetic programming. In *Genetic Programming 1996: Proceedings of the First Annual Conference* (Cambridge, MA, 28–31 July 1996), J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds., MIT Press, pp. 394–399.