Knabe, J. F., Nehaniv, C. L. and Schilstra, M. J. Evolutionary Robustness of Differentiation in Genetic Regulatory Networks. In *Proceedings of the 7th German Workshop on Artificial Life 2006 (GWAL-7)*, pages 75-84, Akademische Verlagsgesellschaft Aka, Berlin, 2006.

Evolutionary Robustness of Differentiation in Genetic Regulatory Networks

Johannes F. Knabe[1], Chrystopher L. Nehaniv[1,2], Maria J. Schilstra[2]

Adaptive Systems[1] and BioComputation[2] Research Groups
University of Hertfordshire
Hatfield AL10 9AB, UK
{j.f.knabe, c.l.nehaniv, m.1.schilstra}@herts.ac.uk

[A]ll cells of a given individual organism inherit the same set of blueprints in the form of DNA molecules. But as a higher organism develops from a fertilized egg a striking variety of different cell types emerges. Underlying the process of development is the selective use of genes, the phenomenon we call gene regulation. [... D]epending in part on environmental signals, cells choose to use one or another developmental pathway. - M. Ptashne [11, p. 1]

Abstract

We investigate the ability of artificial Genetic Regulatory Networks (GRNs) to evolve differentiation. The proposed GRN model supports non-linear interaction between regulating factors, thereby facilitating the realization of complex regulatory logics. As a proof of concept we evolve GRNs of this kind to follow different pathways, producing two kinds of periodic dynamics in response to minimal differences in external input. Furthermore we find that successive increases in environmental pressure for differentiation, allowing a lineage to adapt gradually, compared to an immediate requirement for a switch between behaviors, yields better results on average. Apart from better success there is also less variability in performance, the latter indicating an increase in evolutionary robustness.

1 Introduction

Typically in multicellular organisms, (almost) all of an individual's cells contain the same genome but still, depending on signals or differences in the internal environment, can take very different

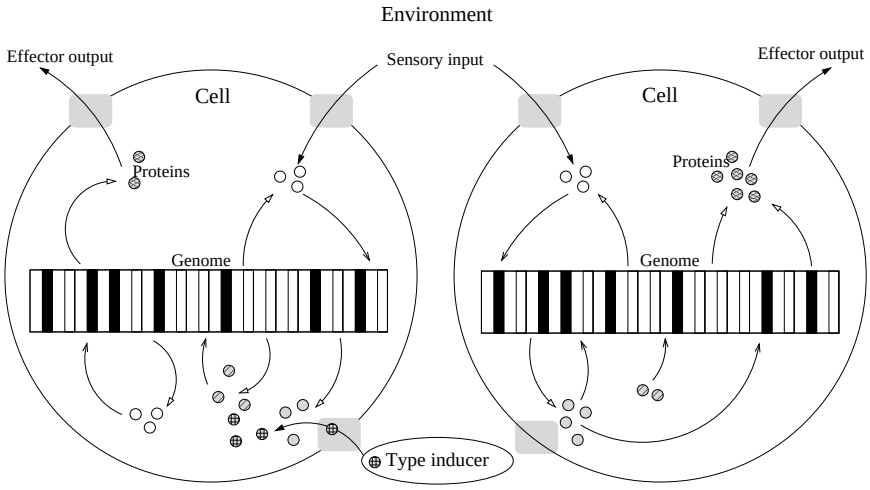


Figure 1: Schematic drawing of our model. The two cells have the same genome and thus the same regulatory network but can produce very different behavior, induced by a very simple signal which is here shown as external, but it could also be an internal gene that is always on due to cell division disparity.

functional roles. Crucial signals are believed to be induced by other cells or the environment early in development, e.g. turning on (a) *homeotic* gene(s), which "remain on through adult life and maintain particular aspects of the pattern of gene expression characteristic of that segment [they are part of.]" [11].

In biological Genetic Regulatory Networks (GRNs), genes encode proteins and proteins in turn regulate the expression (activation) level of genes. The dynamics of these interactions not only play a key role in development [4] but also in the ongoing metabolism of all cells during their lifetime [1]. Furthermore, cells do not exist in isolation but are embodied in an environment, which influences the cell; the cell can in turn influence its environment via internal regulatory dynamics; see fig. 1.

As an evolutionary and computational paradigm, GRNs support complex regulatory and evolutionary dynamics [2], which when combined with differentiated multicellularity represent a vast potential for massive adaptive parallel and distributed computation [9]. This is achieved by a continual coupling of internal and external dynamics as active, regulatory control systems [12].

Differentiation of cells into types has been investigated in artificial GRNs several times. The most famous example is from Kauffman [7], but this and other early models are usually based on random boolean networks. Newer non-boolean approaches mostly have a strong pre-specification of the network structure (e.g. [5]), in this work we start from randomly connected networks.

2 Methodology

Before complicating matters by modeling huge multicellular structures with a diversity of cell types we begin with evolving a system capable of showing two behaviors. In [8], where we first described the proposed GRN model, we used it to evolve biological clocks with the circadian rhythm abstracted to a sinusoidal wave. GRNs producing such cyclic behavior in response to various periodic environmental stimuli could easily be evolved. Mirroring the phase of their input as well as the production of the inverse phase was possible[1], however with every evolutionary

[1]For results from those experiments see also http://homepages.feis.herts.ac.uk/∼kj6an/GRNclocks/.

run having only one of these behaviors as its objective. So in the context of differentiation it was quite natural to ask whether it would be possible to integrate two or more functionalities into one GRN. We evolve populations of GRNs with two such functionalities in various settings and investigate the impact of the lineage's history on regulatory and evolutionary dynamics.
Cell cleavage and development are also victims of abstraction – from the start we have two identical cells receiving the same periodic external stimuli, cf. fig. 1. The expected difference in behavior is only signaled by a type inducer that raises a protein level, which in our model could be the result of either an internal gene turned on during cell division or externally generated. There is currently no diffusion or other kind of interaction between the cells.

2.1 GRN Model

The proposed GRN model makes locally smooth regulatory and evolutionary dynamics possible, and environmental interaction is explicitly considered. It has been first described in [8], where more details can be found.
Every cell consists of proteins and a genome with a fixed number of genes. Gene activation is controlled by regulatory sites (cis-sites or cis-modules), each composed of – possibly – several protein binding sites. Depending on the attachment of matching proteins to the binding sites the corresponding cis-modules positively or negatively influence the production of, not necessarily different, proteins. In molecular biology, proteins acting in such a way are called Transcription Factors (TFs). In our model all proteins are potentially regulatory. For simplicity in the regulatory dynamics we use template matching, i.e. a perfect match of binding site and the corresponding protein is required, unlike e.g. [2, 3]. The main difference to the `Biosys` model, described in [12], is that one can have any number of cis-modules per gene and every cis-module can have any number of protein binding sites. This is to allow for an additional level of protein regulation, as it is known to molecular biologists that TFs not only show additive behavior but might also interact with each other and thereby change their influence synergistically, see e.g. [13, and references therein]. This level could for example facilitate the advent of "master control genes", i.e. one active gene at the top of a control hierarchy that might start a cascade, turning on a huge number of other genes. For example [6] found that the out-of-place eye production in the fruit fly Drosophila can be triggered by a single signal. Such selectors can be thought of as choosing a particular pathway for the cell (as well as it's descendants) and are assumed to be involved in cell differentiation as well as developmental modularity.
In summary our approach facilitates the evolution of complex dynamics, coming a little closer to nature, where "5-10 regulatory sites are the rule that might even be occupied by complexes of proteins" [2].

2.1.1 Genetic Representation

The genome is represented as a string of integers, encoding the genes and some global parameters of the network. Digits 0 and 1 are *coding* digits that may be involved in regulation or protein coding. To differentiate between such a coding bit, a cis-module boundary and a gene boundary the genetic alphabet was increased to four digits, with 2 delimiting the end of a cis-module and 3 delimiting the end of a gene. There are eight different proteins in the version of the model used here, i.e. three bits encode a protein.
For this set of experiments we used a fixed number of genes, namely nine, as this had proven more than enough for coping with the single task described in our earlier paper [8]. After compartmentalizing the genome into genes, the last four coding digits of every gene determine

its output behavior, three bits for the protein produced and the last bit for the gene's activation type, which can be "default on" – even active when no activation is present or "default off" – only with positive activation.

For cis-modules the first coding bit determines its influence on the gene's activation level (*inhibitory/activatory*) and every following three coding digits are considered a protein binding site. For example the gene 010111021101020011113 will produce protein 7 (111) and is "off by default" (last bit is 1). It has two cis-modules, the first inhibitory (starting with 0) binding a combination of proteins 5 (101) and 6 (110), and an activatory cis-module (starting with 1) to which protein 5 (101) will bind. Note that the last zero of 110102 is ignored; we refer to such coding digits which are neither translated nor regulatory as *junk*.
The genome also encodes several evolvable variables global to the cell. These are the *protein-specific decay rates* (four bit for every protein, indexing into a fixed lookup table of values), the global *binding proportion* (also four bits indexing into a lookup table, but identical for all proteins), and finally the global *saturation value* (three bits indexing to a look up table, again identical for all proteins).

2.2 Regulatory Logic

The model is run over a series of discrete time steps, its lifetime. In each time step initially a fraction of the free proteins, determined by the global binding proportion parameter, are bound to matching sites; if there is more than one binding site competing for the same protein the fraction is equally distributed between all matching sites[2]. In this process all protein binding sites are treated equally, regardless of the cis-module to which they belong. Let b_i be the number of all binding sites matching protein i (there can be several for the same protein within and between cis-modules) and c_i^t denote the number of protein i being available for binding at time t. Then the amount p_{ijm}^t of protein i bound at time t to a given binding site in cis-module j of gene m and matching protein i is:

$$p_{ijm}^t = \frac{c_i^t}{b_i} + p_{ijm}^{t-1},$$

where p_{ijm}^{t-1} is the amount of protein i at the binding site in the previous timestep after saturation and protein-specific decay have been taken into account, with the initial condition $p_{ijm}^0 = 0$.
The activation level a_m of gene m with k cis-modules is calculated as:

$$a_m = \sum_{j=1}^{k} \pm_j \min_{i:\ \text{protein } i \text{ binds to cis-module } j} p_{ijm}^t,$$

where $\pm_j = \begin{cases} +1 & \text{if cis-module } j \text{ is activatory} \\ -1 & \text{if cis-module } j \text{ is inhibitory.} \end{cases}$

Note that this use of min is similar to a logical AND and results in non-additive effects ("synergy") in gene regulation.
So the calculation of every gene's activation level is done by adding (activatory) or subtracting (inhibitory) the values per cis-module but only the lowest value of bound protein per cis-module

[2]Note that all variables for protein amounts are continuous.

is used (min). The increase in protein concentration due to gene m is then $f_m(a_m)$, [3] where

$$f_m(x) = \begin{cases} \frac{r}{2}\,(\tanh(\frac{x-15}{s}) + 1) & \text{if gene } m \text{ is "default off"} \\ \frac{r}{2}\,(\tanh(\frac{x+5}{s}) + 1) & \text{if gene } m \text{ is "default on".} \end{cases}$$

The parameter $s = 5$ determines the steepness of the slope, with the function becoming more switch like as s gets smaller, and $r = 150$ determines the range of the function. The output of the gene's activation function is added to the unbound concentration of that gene's output protein type. After this calculation the concentrations of all unbound proteins are, if necessary, reduced to the global saturation value and then all proteins, free or bound, are decayed by the protein specific rate. Finally environmental input occurs by increasing the unbound concentration of certain proteins by some value and output by reading some protein concentration values. Simple scaling by r is used to map stimulus input levels from the signal range to a protein concentration, and *vice versa* for output protein levels.

2.3 Evolution

We use a fairly standard Genetic Algorithm with weak elitism, tournament selection and replacement. Every evolutionary condition was studied with ten repetitions; each lasting 500 generations of 250 individuals, where one individual consisted of two cells with the same genome and thus the same regulatory network. The initial population started with one cis-module per gene and one protein binding site per cis-module, all coding bit values being randomly assigned; in network terms the nodes are randomly connected, with at most one incoming arc.

2.3.1 Selection

Later generations are formed by carrying over the best-performing individual of the last generation automatically and, keeping population size constant, the other individuals are replaced by offspring. To generate each pair of offspring, 15 (not necessarily different) individuals of the prior generation are chosen randomly and of these the best two selected to be "parents".

2.3.2 Variability

A (single-point) crossover between the parent genomes occurred 90 percent of the times and every coding bit is flipped with a mutation probability of one percent. To generate a variable number of cis- and of protein binding sites per gene it is necessary to have variable length genomes. Note that despite this, the number of genes stays the same all the time. These properties are achieved by dividing the parent genomes into compartments: one compartment for every gene and one compartment for the global variables. Then (with a probability of 0.9) a single compartment is chosen for crossover and in this compartment a point allocated for crossover. However when crossing over from parent 1's genome to the second parent's genome copying does not necessarily continue at the same position of parent 2's genome but is shifted by an offset (see fig. 2), mimicing the unequal crossing-over observed in biology.

[3] For example, for the gene 010111021101020011113 from above this would mean that due to the first (inhibitory) cis-module, assuming a share of 20 type 5 proteins (101) and 1 type 6 protein (110) per binding site, the value -1 would go into the sum. The second (activatory) cis-module however would contribute $+20$ resulting in an overall activation of 19, which gives a protein output of about 125 type 7 proteins.

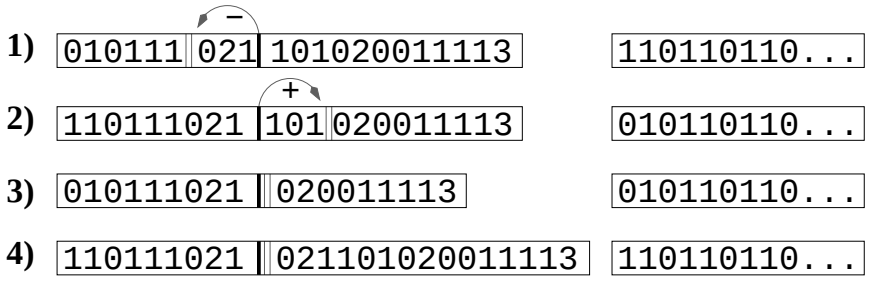


Figure 2: **Gaussian offset crossover.** Genomes of (1) parent 1, (2) parent 2, (3) offspring 1, (4) offspring 2. Only one gene and part of the global compartment shown. Both children get digits up to the crossover point from their respective parent, but then continue in the other parent's genome with opposite gaussian-distributed offsets (-3 and $+3$, respectively, here).

This offset is randomly drawn from a gaussian distributed random variable with mean 0 and standard deviation 4. The relatively large number four was chosen to increase the chance of duplicating genetic information, the importance of which was already pointed out by [10] for the evolution of biological complexity. Ohno put emphasis on whole-genome duplications while it is now, with better techniques, becoming ever clearer that "both small- and large-scale duplication events have played major roles" [14].
Note that the offset point is limited to stay within the boundaries of the compartment, hence if crossoverpoint + offset is smaller/larger than the left/right boundary it is set to the corresponding boundary value. So the number of 2s (cis-modules) might increase by crossover – mutation was only applied to coding digits (0s and 1s) – but not the number of 3s as these are the compartment boundaries. When crossover occurs in the part encoding for global parameters the offset is always set to 0 as more bits would be meaningless here.
These processes allow both neutral crossover and mutational changes, as 'half' cis-modules (i.e. less than three bit – one protein – long) are ignored. Additionally this means that, although the number of genes was constant over one evolutionary run, genes could become inactive, in a similar manner to the so called pseudo-genes found in nature, i.e. if there was not a single cis-module and the gene had an activation type of "off by default".

2.4 Environmental Coupling

We decided to systematically vary evolutionary conditions by varying the pattern of external signal received at the cellular level as well as the periodic output behavior expected.

2.4.1 Input stimuli

The basic idea was to have periodic environmental stimuli based on a sine curve (shifted to the interval $[0, 1]$). The wavelength w was set to 20 time steps, while the lifetime for every GRN was 400 steps. Variations included having only the positive part of sine, a periodic step function, and a brief pulse. The four functions used are depicted in fig. 3. As mentioned above, both cells of an individual always received the same periodic stimuli. However one cell additionally received an *inducing* signal with a continuous value of 1, realized as increasing the level of a protein type different from those used for periodic input and output.

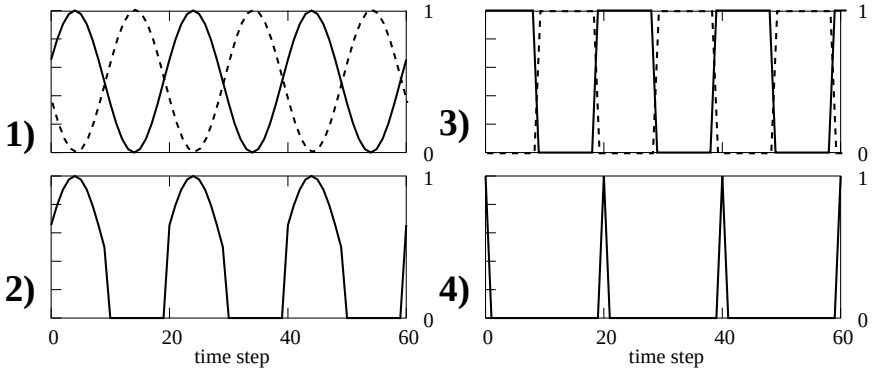


Figure 3: Periodic functions used: 1) sine (dashed the inverse/shifted wave), 2) positive part of sine, 3) step (dashed the inverse/shifted wave), 4) pulse.

2.4.2 Output behavior

Two periodic target functions were used to measure the performance of an individual and assign fitness: sine (fig. 3.1) and step (fig. 3.3). While the induced cell's desired output would be in the the same phase as the input, we ultimately want the other cell to produce the inverse of the input, which is equivalent to shifting the input's phase by one half. Fitness was measured as the deviation from this desired output, i.e. the smaller the value, the better adapted the GRN. Letting $c^t_{i_0}$ denote the (unbound) concentration of the induced GRN's output protein i_0 and d^t_p the desired output in phase p relative to that of the input at time t, the deviation is simply calculated as: $\sum_{t=1}^{L} |c^t_{i_0} - d^t_{0.0}|$ and similarly for the other cell, only with $d^t_{0.5}$ – a phase shift of one half which is equivalent to the inverse wave. Finally both deviations were added up and divided by 2. The lifetime L of every individual was set to 400 time steps; as a reference, over such a lifespan a random GRN achieved a deviation of approximately 200.
However in one set of experiments we did not immediately, i.e. from the first generation, expect individuals to fully differentiate and rate performance accordingly. Instead, the environment became *gradually* harder by increasing the relative shift in wavelength little by little from 0 to $w/2$ every 25 generations (writing g for the current generation we wanted $d^t_{p^*}$ with $p^* = \min(\frac{g}{\lfloor 25 \rfloor}, \frac{w}{2})/w)$ – so full differentiation was only required after 250 generations.

3 Results

Overall, 8 evolutionary scenarios were tested (two desired output types times four environmental stimulus input functions) and each scenario was run ten times. Additionally, the whole set of 8 scenarios was repeated for gradually increasing environmental pressure, as described above.

3.1 Evolutionary Dynamics

In every scenario most repetitions successfully produced well adapted individuals that had evolved a kind of switch, allowing them to behave very differently when an inducing stimulus was present. Not very surprisingly, the more sparse the input was the harder it was to reduce the deviation from the desired output wave. For the immediate full shifting set of experiments, when considering a deviation of 80 acceptable[4], in 30 (out of overall 80) repetitions no

[4]By experience we found that GRNs with a performance worse than 80 often were much better at one task

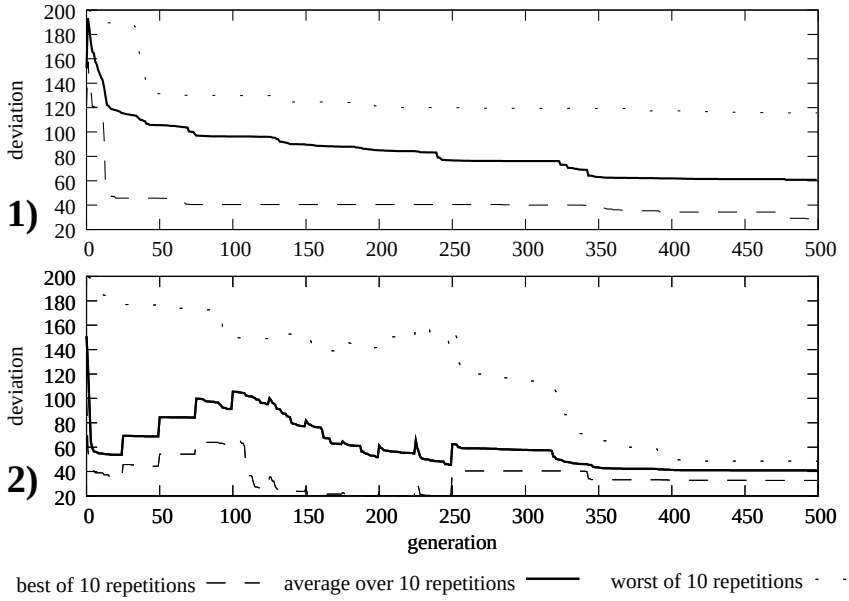


Figure 4: Exemplary evolutionary runs showing the best individual per generation (average over 10 repetitions); 1) with full differentiation pressure, 2) with gradually increasing differentiation pressure. For most experiments we found the best and worst repetitions to be closer together when the lineage's environment changed slowly.

GRN in the population could be considered to have achieved an acceptable performance level. For the gradual setting however, this failure happened only twice, and the superiority of this condition can also be seen from table 1. It seems that an evolutionary environment gradually introducing a requirement for a switch between behaviors facilitates differentiation, and the smaller standard errors suggest an increase in evolutionary robustness. This is also reflected by the finding that for most experiments the best and worst repetitions are closer together when the lineage's environment changed slowly; for an example see fig. 4.[5]

3.2 Evolved dynamics

In all the best evolved GRNs we found the use of AND-like regulatory logic with several binding sites bundled to a cis-module as described above, although the initial random nets started with only one site per module. Typically, the protein level being influenced by the type inducer, which might be considered as the output of a "master control gene" or an environmental stimulus, had a very prominent position (i.e. a high outdegree) in well adapted individuals. For example the one shown in fig. 5 participates in the regulation of 4 out of 7 functional genes. Finally, following are figures illustrating what is going on in an exemplary individual which was the best of its repetition in the scenario: sine input, sine output desired, gradually increasing differentiation pressure. Figures 6 and 7 show its dynamics, with the lower matrices each corresponding to the "inverse desired" cell.

than the other, i.e. no real differentiation had taken place.

[5]Additional results as well as the full source code will be made available at http://panmental.de/GWALdiff

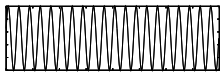
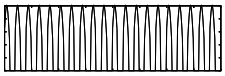
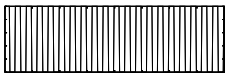
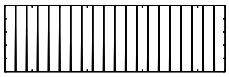
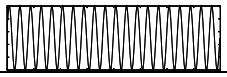
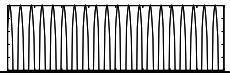
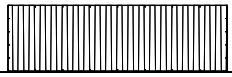
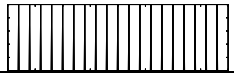
desired behavior / env. input	sine (inverse/mirror)	step (inverse/mirror)
sine 	**63.38** ±11.2 std. err. best evolved: 14.71 best rand.: 86.67/88.58	**76.19** ±12.0 std. err. best evolved: 27.90 best rand.: 126.2/92.01
pos. sine 	**50.14** ±6.29 std. err. best evolved: 21.78 best rand.: 80.18/75.37	**85.12** ±10.8 std. err. best evolved: 37.57 best rand.: 100.4/113.2
step 	**57.27** ±8.92 std. err. best evolved: 27.63 best rand.: 86.06/70.33	**60.75** ±9.40 std. err. best evolved: 28.90 best rand.: 72.17/70.84
pulse 	**74.34** ±6.25 std. err. best evolved: 27.93 best rand.: 86.44/89.68	**81.02** ±11.9 std. err. best evolved: 26.70 best rand.: 128.7/99.64
sine 	**29.52** ±3.62 std. err. best evolved: 18.87	**39.13** ±6.49 std. err. best evolved: 8.672
pos. sine 	**38.34** ±6.12 std. err. best evolved: 16.12	**56.34** ±6.83 std. err. best evolved: 31.04
step 	**37.15** ±3.10 std. err. best evolved: 24.78	**40.96** ±1.20 std. err. best evolved: 32.73
pulse 	**43.38** ±4.74 std. err. best evolved: 19.41	**63.59** ±6.66 std. err. best evolved: 23.39

Table 1: Outcomes of experiments with immediate (upper half), gradual (lower half) differentiation pressure, with the leftmost column depicting the environmental stimuli used and the topmost row the desired output behavior for every run. The data cells show the best final deviation averaged over 10 repetitions with 500 generations times 250 individuals each, ± the respective standard error. Additionally the best deviation achieved by evolution and, in the upper part, the best deviation found when testing the same number – 1.25 million – of random GRNs (one/two binding sites per gene are shown).

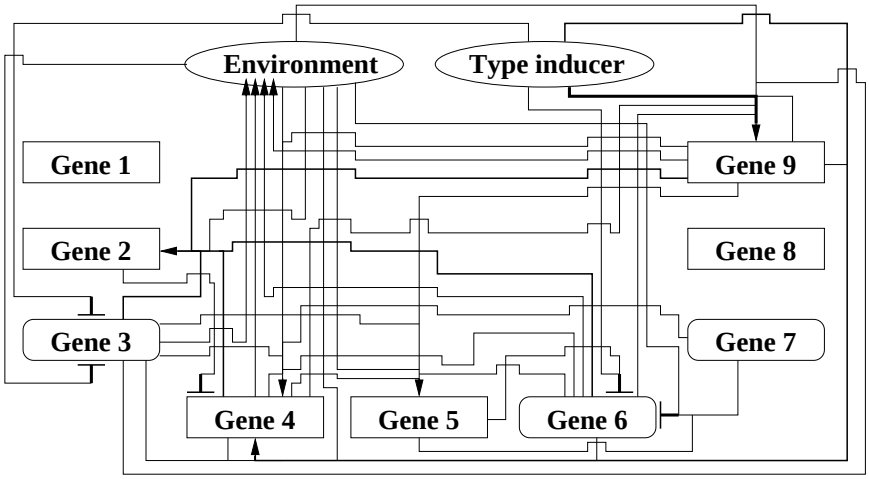


Figure 5: **Regulatory interaction diagram of an evolved 9-gene GRN.** Boxes denote genes (rounded corners indicating "default on" ones with the others being "default off"), connections ending in an arrow are for activatory influences and the T-like endings depict inhibitory ones. The bolder the connections the more binding sites the receiving gene has for the corresponding protein, resulting in a bigger share of the protein binding.

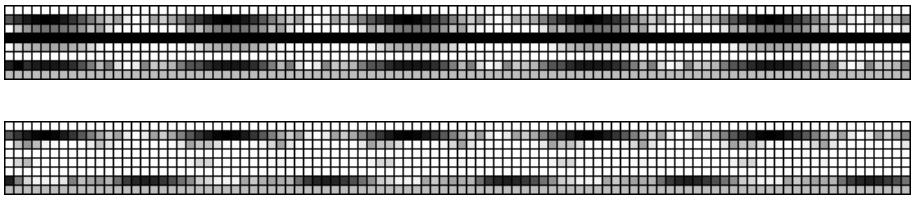


Figure 6: In these matrixes the **8 protein concentrations** of the GRN from fig. 5 over 100 time steps are depicted. Note that row 2 reflects the input protein level while row 7 corresponds to the GRN's output. In the lower matrix lack of activity in row 4 induces inversion of the input stimulus.

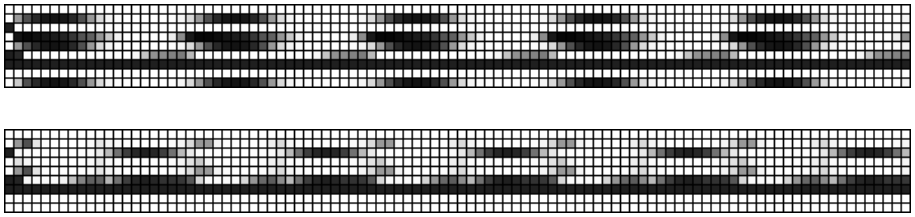


Figure 7: As in fig. 6, but here the **output activity of each of the 9 genes** is shown. Every row corresponds to one gene's protein output, where darker means more output. One can clearly see the distinct activation patterns. Note that genes 1 and 8 are inactive, i.e. generate no output ever, see fig. 5.

4 Discussion

The GRN model is clearly able to evolve functional differentiation. However the lineage's evolutionary history seems to be very important in determining the probability that a switch between two behaviors can be found. Comparing with the immediate requirement for a switch between behaviors we found that in the gradual case final GRNs usually showed better success with less variability in performance, the latter indicating an increase in evolutionary robustness. In the future we will analyze the properties of evolved networks further – what do those that show a switching behavior have in common as opposed to those with no switch? – and also: How did the switch evolve? Last but not least, it will be interesting to see how well these findings scale: can we evolve control hierarchies with levels of switching?

Acknowledgments

JFK wishes to thank Mark Robinson and Moritz Buck for useful discussions.

References

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland Science, 4th edition, 2002.

[2] Wolfgang Banzhaf. On the Dynamics of an Artificial Regulatory Network. In *Advances in Artificial Life, 7th European Conference, ECAL'03*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 217–227. Springer, 2003.

[3] Peter J. Bentley. Adaptive fractal gene reguatory networks for robot control. In J. Miller, editor, *Workshop on Regeneration and Learning in Developmental Systems, Genetic and Evolutionary Computation Conference (GECCO 2004)*, 2004.

[4] Eric H. Davidson. *Genomic Regulatory Systems: Development and Evolution*. Academic Press, 2001.

[5] Nicholas Geard and Janet Wiles. A gene network model for developing cell lineages. *Artificial Life*, 11(3):249–268, 2005.

[6] G. Halder, P. Callaerts, and W. J. Gehring. Induction of ectopic eyes by targeted expression of the eyeless gene in drosophila. *Science*, 267(5205):1788–92, 1995.

[7] Stuart A. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[8] Johannes F. Knabe, Chrystopher L. Nehaniv, Maria J. Schilstra, and Tom Quick. Evolving biological clocks using genetic regulatory networks. In *Proceedings of the Artificial Life 10 Conference (Alife X)*. MIT Press, 2006 (in press).

[9] Chrystopher L. Nehaniv. Self-replication, evolvability and asynchronicity in stochastic worlds. In *Stochastic Algorithms: Foundations and Applications*, volume 3777 of *Lecture Notes in Computer Science*, pages 126–169. Springer, 2005.

[10] Susumu Ohno. *Evolution by Gene Duplication*. Springer, 1970.

[11] Mark Ptashne. *A Genetic Switch*. Cell Press and Blackwell Science, 2nd edition, 1992.

[12] Tom Quick, Chrystopher L. Nehaniv, Kerstin Dautenhahn, and Graham Roberts. Evolving Embodied Genetic Regulatory Network-Driven Control Systems. In *Advances in Artificial Life, 7th European Conference, ECAL'03*, volume 2801 of *Lecture Notes in Artificial Intelligence*, pages 266–277. Springer, 2003.

[13] Maria J. Schilstra and Hamid Bolouri. Modelling the Regulation of Gene Expression in Genetic Regulatory Networks. Technical report, BioComputation group, University of Hertfordshire. http://strc.herts.ac.uk/bio/maria/NetBuilder/Theory/NetBuilderModelling.htm, 2002.

[14] John S. Taylor and Jeroen Raes. *The Evolution of the Genome*, chapter Small-Scale Gene Duplications. Elsevier Academic Press, 2005.