Knabe, J. F., Nehaniv, C. L., Schilstra, M. J. and Quick, T. Evolving Biological Clocks using Genetic Regulatory Networks. In *Proceedings of the Artificial Life X Conference (Alife 10)*, pages 15-21, MIT Press, 2006.

Evolving Biological Clocks using Genetic Regulatory Networks

Johannes F. Knabe[1], Chrystopher L. Nehaniv[1,2], Maria J. Schilstra[2] and Tom Quick[1,3]

Adaptive Systems[1] and BioComputation[2] Research Groups, University of Hertfordshire, Hatfield AL10 9AB, UK
Computer Science[3], University College London, Gower Street, London, WC1E 6BT, UK
j.f.knabe@herts.ac.uk

Abstract

We study the evolvability and dynamics of artificial genetic regulatory networks (GRNs), as active control systems, realizing simple models of biological clocks that have evolved to respond to periodic environmental stimuli of various kinds with appropriate periodic behaviors. GRN models may differ in the evolvability of expressive regulatory dynamics. A new class of artificial GRNs with an evolvable number of complex cis-regulatory control sites – each involving a finite number of inhibitory and excitatory binding factors – is introduced, allowing realization of complex regulatory logic. Previous work on biological clocks in nature has noted the capacity of clocks to oscillate in the absence of environmental stimuli, putting forth several candidate explanations for their observed behavior, related to anticipation of environmental conditions, compartmentation of activities in time, and robustness to perturbations of various kinds, or unselected accidents of neutral selection. Several of these hypotheses are explored by evolving GRNs with and without (gaussian) noise and "black out periods" for environmental stimulation. Robustness to environmental perturbation experienced by the lineage appears to account for some, but not all, dynamical properties of the evolved networks including unselected abilities such as capacity to adapt to shift in phase or frequency of environmental stimulus.

Biological Clocks

A characteristic of life on earth is its *incessant responsiveness* (West-Eberhard, 2003). Biological clocks provide one of the simplest yet most characteristic examples of such incessant responsiveness for life as it has evolved on the earth in that an organism's regulatory dynamics respond with periodic activity in close coupling with periodic cycles of environmental stimuli as experienced in the rhythm of light and dark, or in the effects of lunar gravitation of the ebb and flow of tides. Without the capacity to adjust to external signals, minute differences in timing period soon accumulate, leading to internal clocks being hopelessly out of step with the environment. A. T. Winfree (1986) surveys several (non-mutually exclusive) possible explanations for the advent and maintenance of biological clocks: (1) as a mechanism to anticipate the destructive effects of sunlight on cellular machinery, (2) optimization in transfer between metabolic modes, and (3) for compartmentalization of activities in time. Following Winfree, one may ask, How is it that do biological clocks still work when external stimuli are hidden (like the sun or other temporal cues) in isolation experiments on living organisms? How is it that they can adapt, within limits, to perturbations in cycle length, phase shift, and resetting? Why in isolation do they run at rates somewhat different from that of the external cycles? Are these accidents of neutral selective value, or do they have some adaptive significance at the individual (or lineage) level? In evolutionary and developmental biology, internalization of environment stimuli (Waddington's genetic assimilation, belt-and-suspenders phenomena, and the more general Baldwin effect (Baldwin, 1896; West-Eberhard, 2003)) provides robustness and adaptation to environmental perturbations experienced by a population over evolutionary time. Robustness to noise and periods of loss of signal from environmental stimulation may have played a role in the evolution of biological clock-like mechanisms under any of the hypotheses (1-3) above.

Methodology

Evolving artificial genetic regulatory networks that act as model biological clocks is a natural method to explore the above questions. As an evolutionary and computational paradigm, Genetic Regulatory Networks (GRNs) support complex regulatory and evolutionary dynamics (Banzhaf, 2003), which when combined with differentiated multicellularity represent a vast potential for massive adaptive parallel and distributed computation (Nehaniv, 2005), while achieving continual coupling of internal and external dynamics as active, regulatory control systems (Quick et al., 2003). Already early random boolean GRN models of (Kauffman, 1969; Kauffman, 1993) showed the potential of GRNs for capturing cyclic behaviors. In biological GRNs, genes encode proteins and proteins in turn regulate the expression (activation) of genes. The dynamics of these interactions

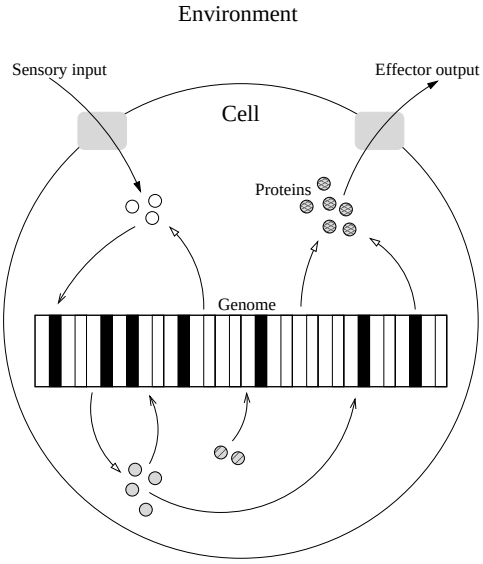


Figure 1: Schematic drawing of protein-genome-environment interaction; see text for details.

not only play a key role in development (Davidson, 2001) but also in the ongoing metabolism of all cells during their lifetime (Alberts et al., 2002). Furthermore, cells are not isolated but embodied in an environment, which influences the cell, and the cell can via internal regulatory dynamics react in turn; see fig. 1. We therefore evolve populations of GRNs in environments with periodic external stimuli of various types to exhibit periodic behaviors of various types under different conditions, and investigate the impact of stimuli experienced by the lineage on regulatory and evolutionary dynamics.

GRN Model

The GRN model we propose here makes locally smooth regulatory and evolutionary dynamics possible, and environmental interaction is explicitly considered. It is strongly inspired by `Biosys`, described in (Quick et al., 2003). As there we model a single cell, consisting of proteins and a genome with a fixed number of genes. Gene activation is controlled by regulatory sites (cis-sites or cis-modules), each having – possibly – several 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. (Banzhaf, 2003; Bentley, 2004). The main difference from the `Biosys` model is that one can have any number of cis-modules per gene and every cis-module can have

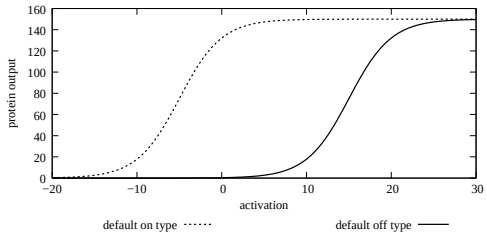


Figure 2: **Activation Types.** Every gene produces its proteins according the cumulative activation level of its cis-modules and its activation type: either even when no activation is present ("default on" - left) or only with positive activation ("default off" - right).

any number of protein binding sites. This is to allow for two levels of protein regulation, as it is known to molecular biologists that TFs might interact with each other and thereby change their influence non-linearly, i.e., as (Schilstra and Bolouri, 2002, see also references therein) put it: "[T]here is often significant synergism – defined as deviation from additive behavior – in the effect of multiple TFs on the expression of a single gene [references omitted]." This second level of regulation has previously not been taken into account by other similar GRN models (Reil, 1999; Banzhaf, 2003; Taylor, 2004). So 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" (Banzhaf, 2003).

Genetic Representation The genome is a string of integers, encoding a fixed number of 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. 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 *constitutive* ("default on") or *induced* ("default off"), see fig. 2. 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 01011102110102001113 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 look up table, same for all proteins).[1]

Regulatory Logic The model is run over a series of discrete time steps, its lifetime. In every 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. 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 instances 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 is used ($\min$). The increase in protein concentration due to gene m is then $f_m(a_m)$, [2] where

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

The parameter $s = 5$ determines the steepness of the slope and $r = 150$ the range of the function[3], see also fig. 2. The output of the gene's activation function is added to the unbound concentration of that gene's output protein type. Afterwards the concentrations of all unbound proteins are checked for being above the global saturation value and all proteins, free or bound, decayed by the protein specific rate. Finally environmental input might occur by increasing the unbound concentration of certain proteins by some value and output by reading some protein concentration values.[4]

Evolution

We use a fairly standard Genetic Algorithm with weak elitism, tournament selection and replacement. Every evolutionary condition was studied with ten repetitions (one run) lasting 250 generations with 250 individuals each. 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.

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. For every 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".

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 alloted for crossover. However when crossing over from parent 1's genome to the second parent's genome copying does not continue at the same position of parent 2's genome but is shifted by an offset (see fig. 3). 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 (Ohno, 1970) for the evolu-

[1] For full details, see http://homepages.feis.herts.ac.uk/∼kj6an/GRNclocks/.

[2] 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.

[3] The model seems to be quite robust against parameter choice as tests with different values for s, r and the inflection points of the activation functions (here, 15 resp. -5 for default on and off) produced qualitatively similar results.

[4] 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.

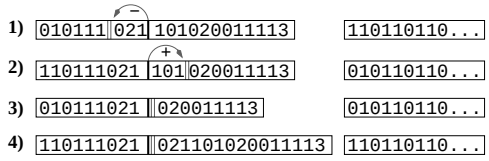


Figure 3: **Gaussian offset crossover.** Genomes of (1) parent 1, (2) parent 2, (3) offspring 1, (4) offspring 2. Only one gene and part of the global compartment shown. Both children get digits up to the crossover point from their respective parent, but then continue in the other parent's genome with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

tion of biological complexity. 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 – 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.
Thus we can have neutral crossover and mutation 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 be disabled completely if there was not a single cis-module and the gene had an activation type of "off by default". Beside that binding sites could require a protein never produced and thus be dysfunctional as well.

Environmental Coupling

As stated in the introduction, environmental cycles have a huge impact on the life of organisms on earth. But in what way these stimuli affect an active organism via its signal transduction pathways and what behavior is appropriate depends on the type of organism. Here we decided to systematically vary evolutionary conditions by varying the pattern of external signal received at the cellular level — in some scenarios distorted or interrupted or both — as well as the periodic output behavior expected.

Input stimuli The basic idea was to have periodic environmental stimuli based on a sine curve (shifted to the interval $[0, 1]$). The wavelength 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. 4. In addition, we varied whether gaussian noise or "black-outs", periods of no external signal, yielding four further conditions: $[\pm noise, \pm blackout]$. In $[-noise, -blackout]$ scenarios, the input signal was transduced to yield a corresponding input of a particular protein as described above, without any distortion. In a $[+noise]$ condition, gaussian white noise with a standard deviation of 0.1 was added to model imperfect signal transduction.[5] For $[+blackout]$ conditions, at random points in time the input stopped completely for an interval of time: every GRN experienced two periods without input, each lasting for 5 percent of its lifetime. In the $[+noise, +blackout]$ scenarios, these perturbations were combined (with the black-out being stronger than the noise, so there was no input – not even noise – during black-out periods).

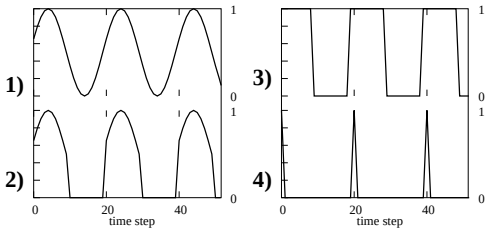


Figure 4: Periodic functions used: 1) sine, 2) positive part of sine, 3) step, 4) pulse.

Output behavior Two periodic target functions were used to measure the performance of a GRN and assign fitness: sine (fig. 4.1) and step (fig. 4.3). As the deviation from this desired output was measured, the smaller the value the better adapted the GRN. Letting c_i^t denote the (unbound) concentration of the GRN's output protein i_0 and d^t the desired output at time t the overall deviation is simply calculated as: $\sum_{t=1}^{L} |c_{i_0}^t - d^t|$, where the lifetime L of every individual was set to 400 time steps; as a reference, over such a lifespan a random GRN on average achieved a deviation of about 200.

Experimental Scenarios

Overall 32 evolutionary conditions were tested (two desired output types times four environmental stimulus input functions in four environmental coupling variations each, as described above) and every setting was run ten times. Additionally the number of genes was varied (see below). In almost every single repetition well adapted GRNs evolved, see Table 1. In one of the $[-noise, -blackout]$ conditions, several individuals even achieved deviation below 1. To test how the evolved GRNs were affected by their evolutionary history we put the best ones into environments not experienced by them or their ancestors before. At first, they got perturbed stimuli, i.e. variations of their usual input functions with noise and/or blackouts. Afterwards special new stimuli were used: constant input, phase shifted input, different wavelength input or very long blackout periods.

[5] Note however that values below zero are set to zero as negative protein input is not possible.

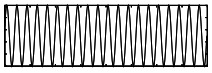
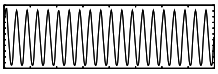
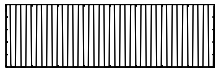
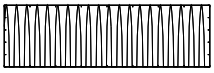
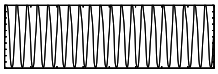
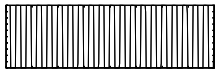
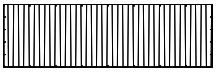
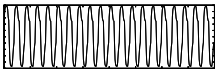
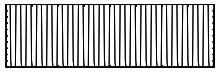
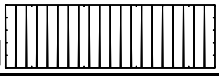
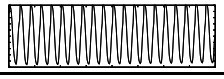
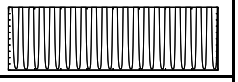
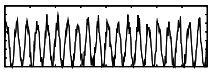
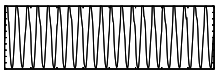
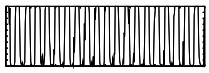
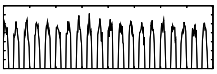
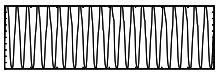
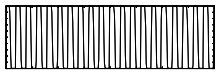
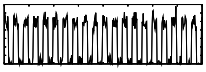
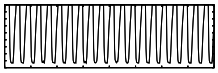
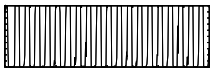
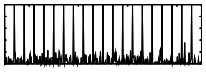
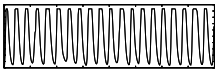
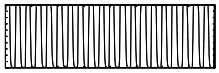
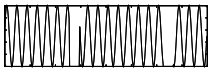
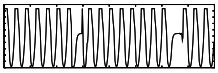
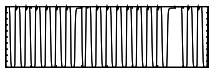
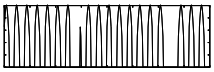
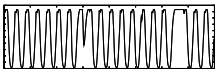
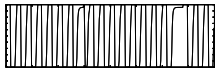
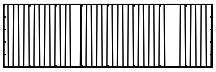
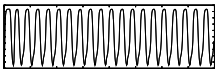
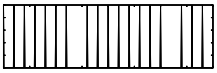
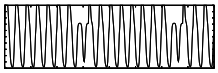
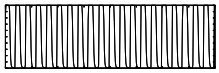
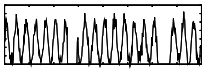
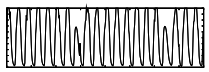
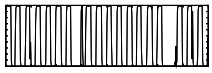
desired behavior / env. input	sine	step
sine $[-noise, -blackout]$ 	16.70 ±1.32 best: 10.40 (27.75 ±10.4) 	35.31 ±8.69 best: 0.762 (20.04 ±3.90) 
pos. sine $[-noise, -blackout]$ 	26.54 ±3.47 best: 12.09 (41.91 ±5.28) 	14.48 ±2.75 best: 1.145 (36.23 ±7.85) 
step $[-noise, -blackout]$ 	29.63 ±1.23 best: 21.51 (36.45 ±3.47) 	27.03 ±4.04 best: 2.931 (49.67 ±10.4) 
pulse $[-noise, -blackout]$ 	45.33 ±6.06 best: 13.10 (37.45 ±4.07) 	43.39 ±3.13 best: 26.23 (52.56 ±5.87) 
sine $[+noise, -blackout]$ 	27.59 ±2.01 best: 13.27 (30.44 ±1.81) 	54.78 ±12.6 best: 21.27 (41.40 ±6.16) 
pos. sine $[+noise, -blackout]$ 	32.12 ±3.08 best: 16.35 (37.16 ±4.44) 	35.00 ±5.88 best: 13.65 (45.30 ±9.35) 
step $[+noise, -blackout]$ 	34.96 ±1.29 best: 30.08 (38.83 ±1.85) 	30.34 ±2.55 best: 16.58 (70.11 ±14.4) 
pulse $[+noise, -blackout]$ 	43.74 ±2.62 best: 32.03 (59.79 ±7.38) 	59.07 ±6.93 best: 18.08 (65.87 ±3.32) 
sine $[-noise, +blackout]$ 	33.55 ±1.21 best: 28.92 (38.08 ±2.92) 	46.48 ±4.89 best: 26.27 (43.40 ±5.40) 
pos. sine $[-noise, +blackout]$ 	39.92 ±4.51 best: 29.29 (46.48 ±2.52) 	43.76 ±4.84 best: 25.38 (63.28 ±3.95) 
step $[-noise, +blackout]$ 	41.67 ±1.73 best: 29.10 (45.00 ±2.94) 	63.47 ±10.1 best: 38.20 (64.86 ±9.41) 
pulse $[-noise, +blackout]$ 	50.79 ±4.21 best: 24.29 (56.80 ±3.84) 	46.09 ±4.12 best: 17.75 (57.69 ±3.89) 
sine $[+noise, +blackout]$ 	39.50 ±1.87 best: 28.30 (55.14 ±7.60) 	53.05 ±3.31 best: 36.80 (63.08 ±4.34) 
pos. sine $[+noise, +blackout]$ 	46.48 ±3.39 best: 30.83 (54.92 ±5.71)	54.24 ±3.56 best: 35.77 (72.67 ±8.94)
step $[+noise, +blackout]$	44.35 ±1.55 best: 38.93 (49.55 ±2.74)	52.76 ±4.24 best: 37.61 (61.51 ±8.48)
pulse $[+noise, +blackout]$	53.27 ±4.34 best: 23.57 (71.81 ±6.31)	64.33 ±6.41 best: 27.48 (89.70 ±6.73)

Table 1: Outcomes of runs evolving 9-gene GRNs, 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 final deviation of the best individual, averaged over 10 repetitions with 250 generations each, ± its standard error, as well as the best ever observed GRN's deviation and its lifetime behavior graph. In brackets the average deviations ± std. error for the same condition only with 5 genes are given.

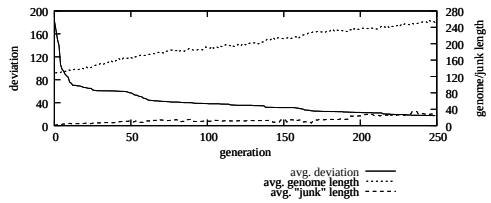


Figure 5: Typical course of an experimental scenario. Averages are over ten GRNs, each being the generation's best performing GRN in its respective repetition.

Results

Regulatory and Evolutionary Dynamics

Due to junk as well as dysfunctional genes and binding sites, we could observe neutral changes, i.e. despite of the fact that performance often stayed the same for some evolutionary period, genome length might change during crossover or bits without function might be flipped. Although when crossing over with an offset different from zero usually both a shorter and a longer descendant are produced, the average population genome length increases over evolutionary time. The amount of junk also increases, though at a slower rate, see. fig. 5.

Number of Genes

Running the same set of experiments with a fixed number of 5 genes and again with 9 genes it turned out that the ones with 9 genes in almost all cases ended up with a performance superior to their 5-gene equivalents (see table 1). Furthermore we found that, under environmental conditions different from those experienced during evolution, the best evolved 9-gene GRNs depended less on their environment. Having less internal complexity, the 5-gene GRNs more often produced fast oscillating or constant output in absence of environmental stimuli (see fig. 9 for an example).

Evolved Regulatory Dynamics

In all scenarios evolved GRNs exhibited a close match to the desired output profile and almost always relied on external signals to produce this behavior. As an exception, the best performing 9-gene GRN evolved with pulse input (fig. 4.4) distorted by noise and blackouts to produce a step output had no binding sites for the input protein, i.e. it did not rely on environmental stimuli at all. So the regulatory logic was in principle able to generate a close match to that desired output without any external stimuli. However the evolution of such dynamics was rare and it is probably not a coincidence that this happened under the evolutionary conditions with the least reliable input. Complex interaction networks evolved and almost all used the new regulatory logic with several binding sites on a cis-module (an example of a 5-gene GRN is shown in figure 6). Heterochronic control, i.e.

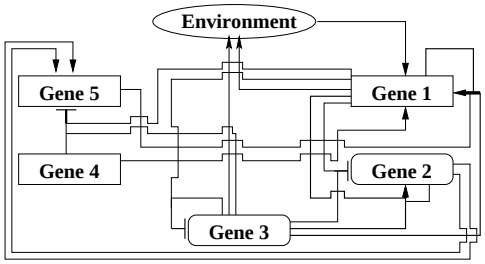


Figure 6: **Regulatory interaction diagram of a evolved 5-gene GRN.** Boxes denote genes (rounded corners indicating "default on" ones with the others being "default off"), connections ending in an arrow are for activatory influences and the T-like endings depict inhibitory ones.

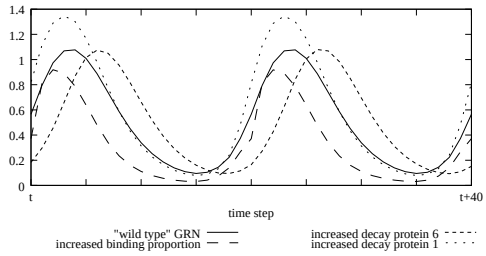


Figure 7: Heterochronic control is achieved through variable decay and binding rates. Shown is the behavior of a 9-gene GRN that is well adapted to producing sine waves (the evolved "wild type") and slight variations of it. The variations' genomes are at most two bit flips away from the wild type.

changes in the timing of gene expression without affecting the general dynamics, is achieved by varying protein decay rates and the binding proportion (cf. fig. 7).

Behavior in Evolutionarily New Conditions

In what way and how strongly the evolved GRNs relied on input from the environment turned out to depend strongly on the conditions under which they evolved (the evolutionary history of their lineage). In the absence of input, GRNs often exhibited an internalized output wavelength different from the one which was desired during evolution. Such behavior occurred mainly in GRNs evolved under pulse input, where the systems used the occasional input to stay synchronized, for an example see fig. 8. Apart from fast oscillations we observed systems with internal periods of varying length from 16 to almost 50 time steps (cf. fig. 9.1). Like most biological clocks studied in man and nature (Winfree, 1986), nearly all the best evolved GRNs – except those that completely ignored their input (which arose seldom and in only one scenario) – in the various scenarios were robust to the shifts in phase and limited shifts in wavelength of periodic environmental stimuli. This occurs despite their lin-

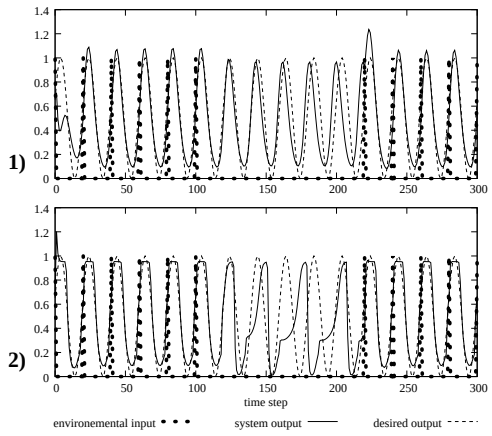


Figure 8: **Periodic Behavior during Blackout.** Plot of the output behavior of GRNs with 1) nine 2) five genes. There is a pulsed input every twenty steps, however from time step 100 to 200 environmental input is suppressed completely. Desired output refers to what behavior was required during evolution, here sine output. Note how the GRN in 1) gets slowly out of synchrony with the desired sine output during the blackout period while this happens quickly for the one in 2). Both GRNs are the result of evolutionary runs with shorter blackout periods of only 20 time steps and achieved a similar performance.

eage never having experienced such perturbations, i.e. without any selection for these capabilities. When GRNs evolved without noise (and/or blackouts) were placed in an environment with noise (and/or blackouts), performance was still good, however always worse than that of GRNs evolved for such environments.[6]

Discussion

The GRN model can easily evolve to produce cyclic behavior, generally in response to periodic stimuli like that of biological clocks in nature. Moreover, like natural biological clocks the evolved regulatory dynamics of artificial GRN clocks tend to be robust to non-selected perturbations such as phase shift, small period changes, and so on, as well as perturbations that have occurred in the lineage's evolutionary history (such as noise and blackouts). Especially when there is a sparse signal and lower reliability in environmental stimuli it pays for the GRNs to internalize the rhythm. How strongly the evolved GRN relies on environmental input depends on the coupling to the environment during evolution. We conclude that evolved artificial GRNs capture many characteristic properties of biological clocks and could serve as a useful model for further investigations.

References

Alberts, B. et al. (2002). *Molecular Biology of the Cell*. Garland Science, 4th edition.

Baldwin, J. M. (1896). A new factor in evolution. *American Naturalist*, 30:441–451, 536–553.

[6]Not all results shown here for reasons of space, please see http://homepages.feis.herts.ac.uk/∼kj6an/GRNclocks/.

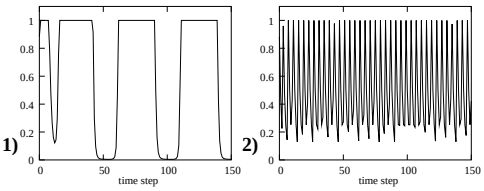


Figure 9: **Behavior without any external stimulus.** Plot of the dynamics of GRNs with 1) nine 2) five genes when there is no input ever. The internal period (if it exists) can be very different from the one the environment usually imposes. Both GRNs are the result of evolutionary runs using pulse input, $[-noise, -blackout]$, with step output desired and were the best of their runs.

Banzhaf, W. (2003). 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.

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

Davidson, E. H. (2001). *Genomic Regulatory Systems: Development and Evolution*. Academic Press.

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

Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.

Nehaniv, C. L. (2005). 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.

Ohno, S. (1970). *Evolution by Gene Duplication*. Springer.

Quick, T., Nehaniv, C. L., Dautenhahn, K., and Roberts, G. (2003). 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.

Reil, T. (1999). Dynamics of Gene Expression in an Artificial Genome - Implications for Biological and Artificial Ontogeny. In *Advances in Artificial Life, 5th European Conference, ECAL'99*, volume 1674 of *Lecture Notes in Artificial Intelligence*, pages 457–466. Springer.

Schilstra, M. J. and Bolouri, H. (2002). 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.

Taylor, T. (2004). A Genetic Regulatory Network-Inspired Real-Time Controller for a Group of Underwater Robots. In *Intelligent Autonomous Systems 8*, pages 403–412. IOS Press.

West-Eberhard, M. J. (2003). *Developmental Plasticity and Evolution*. Oxford.

Winfree, A. T. (1986). *The Timing of Biological Clocks*. Scientific American Books, Inc.