Proceedings of the 2007 IEEE Symposium on
Artificial Life (CI-ALife 2007)

The Essential Motif that wasn't there: Topological and Lesioning Analysis of Evolved Genetic Regulatory Networks

Johannes F. Knabe*, Chrystopher L. Nehaniv*,† and Maria J. Schilstra†
Adaptive Systems* and BioComputation† Research Groups
University of Hertfordshire
Hatfield AL10 9AB, UK

Abstract— **Networks that abstractly model natural Genetic Regulatory Networks (GRNs) are evolved to show a range of dynamical behaviors. Specifically one group was evolved to show differentiation, i.e. to be able to perform an additional behavior as compared to the original, single target behavior. These GRNs are then analyzed and compared with measures used in the biological sciences. Having huge numbers of GRNs available for not only analysis but also "metabolic" inspection, we find that evolutionary niches (target functions) do not necessarily mold network structure uniquely. Our results suggest that variability operators can have a stronger influence on network topologies than selection pressures, especially when many topologies can create similar dynamics. Furthermore, damaging the most significantly represented motif (whether in differentiating or non-differentiating GRNs) is found not to have a significantly bigger impact on function than random lesions, suggesting that particular motifs are not as important in the robust functioning of networks as might perhaps be expected.**

I. Introduction

In biological Genetic Regulatory Networks (GRNs), genes encode proteins and proteins in turn regulate the expression (activation) level of genes. These connections can conveniently be depicted as networks, with the genes being nodes and the regulating proteins the directed edges. The dynamics of these interactions not only play a key role in development [1] but also in the ongoing metabolism of all cells during their lifetime [2]. Understanding GRN dynamics is still a hard task and so methods of breaking their complexity down have been proposed. Very influential has been the static structural analysis method of searching for subgraph patterns – network motifs "are those patterns for which the probability P of appearing in a randomized network an equal or greater number of times than in the real network is lower than a cutoff value" [3].

The main research questions motivating our analysis were: Are there significant patterns that arise in the course of evolution in GRNs necessary for controlling the realization of particular functionality? Are some motifs more prevalent than others in evolution for particular functions? How unique are the networks that realize particular functionalities and how robust are they?

These questions render the use of current biological GRN data insufficient as many networks are needed that are evolved for particular functionalities and currently data is only available for networks in a few organisms. In addition metabolic analyses from observing GRN interactions *in vivo* is very difficult and techniques are still in their infancy. The model we use has been shown to exhibit some characteristics of natural GRNs [4]–[6] and has the substantial advantage that all variables can be controlled, also the GRNs are open to "metabolic" inspection.

This work is related to [7] which also employs a GRN model and analyzes static network topology. However the networks included in that analysis were randomly created (focusing on duplication) without being exposed to evolution under selective pressure – and therefore could not be lesioned to check for impact on function.

II. Methods

At first we describe the topological measure used on networks, then our GRN model follows. A description of the evolutionary algorithm and the two evolutionary settings we compare follows.

A. Network motifs

Motifs are the subgraph patterns that are found in networks in statistically significant numbers as compared to random networks [3]. Although they are a local, structural measure, the "basic idea is that patterns that occur in the real network much more often than in randomized networks must have been preserved over evolutionary timescales against mutations that randomly change edges" [8].

However one has to keep in mind that mutation is not the only variability operator, in nature (and many models) crossover and duplication of genome parts may take place. Instead of the randomization methods usually used (see [8]) we have the advantage of being able to create even better random networks. "Even better" as we do not only have to work with the data of one or two biological networks, due to the difficulty of determining GRN structure in natural systems, but can create as many networks as needed under conditions we want. So we can generate our random networks using the same creation process, i.e. using the same variability operators on them, only without selecting for any function (whereas the use of a randomization procedure starting from a given network of interest might itself bias results). This is termed

1-4244-0701-X/07/$20.00 ©2007 IEEE

random evolution in the following, i.e. normal variability but no selective pressure over the same number of generations. From every such run only one randomly created network was used for analysis to avoid sampling biases and to make sure that their developments over evolutionary time were independent.

For analysis, all subgraphs of a network are enumerated, their connection matrix brought into a canonical form and occurrences of each unique pattern counted. To keep analysis concise, in this paper we report only results for subgraphs of three nodes, as these are most commonly found in the literature. Additional results and Java code are available at http://panmental.de/GRNmotifs/. In the literature the pattern count within a single known network from biology is generally compared against the pattern distribution (average number and standard deviations) from a larger number of randomly evolved networks. As we have a lot of networks evolved for function we can additionally compare many against many.

B. GRN Model

The GRN model we use was first described in [4]. It allows for locally smooth regulatory and evolutionary dynamics, with environmental interaction being explicitly considered. Consideration of environmental interaction is not very surprising as the model is inspired by `Biosys`, described in [9], where GRNs were used as embodied control systems. 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 regions organized into *cis-modules* and these in turn contain – possibly – several *binding sites*. In every discrete time step, free proteins can attach to binding sites. Spatiality is not considered, but the attachment of proteins to binding sites is restricted by the match of site and protein type. For simplicity in the regulatory dynamics we currently use template matching, i.e. a perfect match of binding site and the corresponding protein is required, unlike real biological systems or other approaches (e.g. [10], [11]), where looser matchings are possible. 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.

Molecular biology terms proteins acting in such a way Transcription Factors (TFs). In our model all proteins are potentially regulatory. The main extension compared to the `Biosys` model is that a cell can have any number of cis-modules per gene and every cis-module can have any number of protein binding sites. So there are two levels of protein regulation, 1) interaction of binding sites within a cis-module and 2) among cis-modules (for details see the description in section II-B.2 below). Effects of protein regulation on gene expression are often assumed to be only *additive*, however it is known to molecular biologists that TFs might interact with each other and thereby change their influence non-linearly, i.e., as [12, see also references therein] puts 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.” This second level of regulation has previously not been taken into account by other similar GRN models [10], [13]–[15]. The additional control logic level might facilitate the advent of “master control genes”, i.e. active genes at the top of a hierarchy that might start a cascade, turning on a huge number of other genes. For example [16] found that ectopic eyes (out-of-place eye production) in the fruit fly *Drosophila* can be triggered by a single signal. Such activities can be thought of as choosing a particular pathway for the cell and are assumed to be involved in cell differentiation as well as developmental modularity. In another study [6] we show that GRNs using this model are able to differentiate in principle. For details on the genetic control of development see [1], [17]. Summarizing, 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” [10].

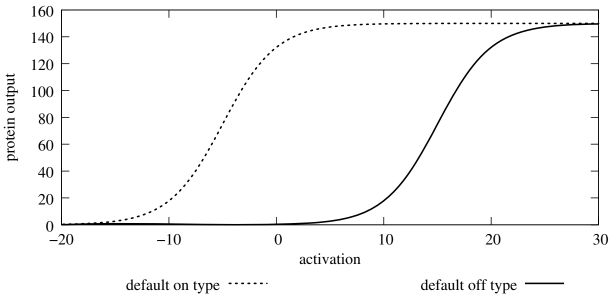

Fig. 1. **Activation Types.** Every gene produces its proteins according the cumulative activation level of its cis-modules and its activation type: either even when no activation is present (“default on” - left) or only with positive activation (“default off” - right).

1) Genetic Representation and Genotype-Phenotype mapping: Every GRN’s genotype is a string of integers, encoding a fixed number of genes and some global parameters of the corresponding phenotype’s 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 digit 2 delimiting the end of a cis-module and digit 3 delimiting the end of a gene. In the version of the model used here there are eight different proteins, i.e. three bits are sufficient to code for the protein type.

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 *constituitive* (“default on”) or *induced* (“default off”), see fig. 1. The first coding bit of a cis-module determines its influence on the gene’s activation level (*inhibitory/activatory*) and every following three coding digits are considered a protein binding site.

Note that, due to evolutionary operators explained below, there might be additional digits that are not meaningful. We refer to such digits which are neither translated nor regulatory as *junk*. See fig. 2 for an example gene representation.

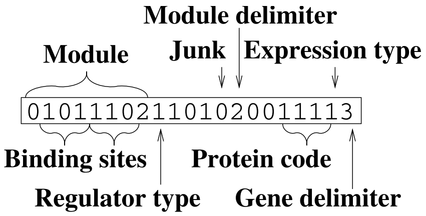

Fig. 2. **Example gene representation.** The gene 010111021101020011113 will produce protein 7 (111) and is "off by default" (last bit is 1). It has two cis-modules, the first inhibitory (starting with 0) binding a combination of proteins 5 (101) and 6 (110), and an activatory cis-module (starting with 1) to which protein 5 (101) will bind. The last zero of the cis-module 110102 as well as the following two zeros are all ignored, they are "junk".

The genome also encodes several evolvable variables global to the cell. These are 1) the *protein-specific decay rates*, four bits for each of the eight protein types, indexing into a fixed lookup table of values, 2) the global *binding proportion*, also four bits indexing into a lookup table, but identical for all proteins, and finally 3) the global *saturation value*, three bits indexing to look up table, same for all proteins. Look up table values can be found online at http://panmental.de/GRNmotifs/.

2) Regulatory Logic: A GRN 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. Should there be more than one binding site competing for the same protein type the fraction is equally distributed between all matching sites[1]. 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. Initially there are no proteins, i.e. $c_i^0 = 0$ and $p_{ijm}^0 = 0$.

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 an extension of logical AND (see table I) and results in non-additive effects ("synergy") in gene regulation. Furthermore this is a canalizing function in the sense of Kauffman [18], who underlines their importance for dynamical properties of boolean networks. For a function to be canalizing (at least) one input variable must be able to assume a value that enforces a certain output value, regardless of the other inputs – which is clearly given here as one low input to the min function suffices to ensure a low output.

TABLE I

min AND *and* FUNCTIONS FOR BOOLEAN $\{0,1\}$ INPUTS.

i_1	i_2	$\min(i_1, i_2)$	*and*(i_1, i_2)
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

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, with the smaller s is chosen the more switch like the function gets, and $r = 150$ the range of the function[3], see also fig. 1. 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 to the GRN-controlled cell can occur by increasing the unbound concentration of certain proteins by some value and output by reading some protein concentration values.[4]

C. Evolution

We use a standard Genetic Algorithm with elitism, tournament selection and replacement. Every evolutionary condition was studied with ten runs, lasting 750 generations each with each generation containing 250 individuals, where one individual consisted of a single cell with GRN-controlled interaction with its environment as determined by its genome and regulatory dynamics. The initial populations started with one cis-module per gene and one protein binding site per cis-module, all coding bit values being randomly assigned – with

[1] Note that all variables for protein amounts are treated as continuous.

[2] For example, for the gene 010111021101020011113 from fig. 2 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 moderate changes in 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.

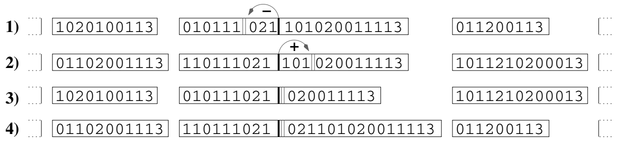

Fig. 3. **Gaussian offset crossover.** Genomes of (1) parent 1, (2) parent 2, (3) offspring 1, (4) offspring 2. Only the compartment chosen for crossover and two neighboring genes are shown. Both children get digits up to the crossover point (solid bar) from their respective parent, but then continue in the other parent's genome with opposite gaussian-distributed offsets (-3 and $+3$, respectively, here).

a fixed number of genes during evolution. In network terms, depicting genes as nodes and protein products of a gene that match a binding site of another gene as arcs between those nodes[5], the nodes are randomly connected, with at most one incoming arc.

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. 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".

2) Variability: A (single-point) crossover between the parent genomes occurred 90 percent of the times and there is mutation 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. 3), mimicking the unequal crossing-over observed in biology [19].

This offset is randomly drawn from a gaussian distributed random variable with mean zero and standard deviation four. The relatively large number four was chosen to allow for some shifts by three to occur. An offset of three is likely to add or remove exactly one binding site and at most disturb immediately adjoining sites. Other values are more probable to cause a change in the reading frame, i.e. all following binding sites change the protein type they are receptive for – very much like what biologists call a frameshift mutation. In network terms the latter can have a huge impact on the GRN's dynamics while the former might be a relatively smooth transition. The importance of duplicating genetic information was already pointed out by [20] for the evolution of biological complexity – see also [21], [22]. 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" [23, p. 320]. Experiments where the duplication of genes is possible, in network terms the addition of nodes with the same connection structure as an existing node, are under way.
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 digits 0 and 1 – 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 degenerate cis-modules (i.e. less than three bits – one protein encoding – long) are ignored. Additionally this means that, although the number of genes was constant over one evolutionary run, genes could become inactive, in a manner similar to the so-called pseudo-genes found in nature, i.e. if there is no non-degenerate cis-module and the gene had an activation type of "off by default".

D. 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. The set of functions was inspired by nature's circadian rhythms, as we first wanted one celled "simple models of biological clocks that have evolved to respond to periodic environmental stimuli of various kinds with appropriate periodic behaviors" [4], [5]. We will refer to this as the *original setting*. In other work we investigated evolving differentiation of cells [6]. There two-celled (both cells having the same genome) models had to respond very differently to an almost identical signal (see schema in fig. 5). This setting will be referred to as the *differentiation setting*.

1) Input stimuli: The basic idea for both settings was to have periodic environmental stimuli based on a sine curve (shifted to the interval $[0, 1]$). The wavelength was set to $w = 20$ time steps, while the lifetime L 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.

As mentioned above, in the differentiation setting both cells of an individual always received the same periodic stimuli, however one cell additionally received a constant *inducing* signal with a value of 1 (see fig. 5).

2) Output behavior: Two periodic target functions were used to measure the performance of an individual and assign fitness: *sine* (fig. 4.1) and *step* (fig. 4.3), with the first requiring smooth changes of protein levels and the latter a boolean step-function-like pattern. The desired output's shape and phase might differ from the input, however the wavelength was

[5] Another possibility would be to have protein types as nodes with arcs going from every binding site protein type a gene has to its output protein.

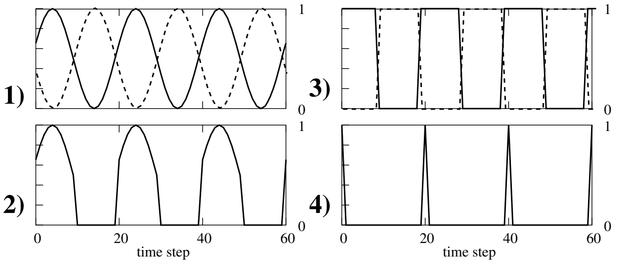

Fig. 4. Periodic functions used: 1) sine (dashed a shifted wave), 2) positive part of sine, 3) step (dashed a shifted wave), 4) pulse. Note that shifting phase by one half of the wavelength is equivalent to the inverse wave.

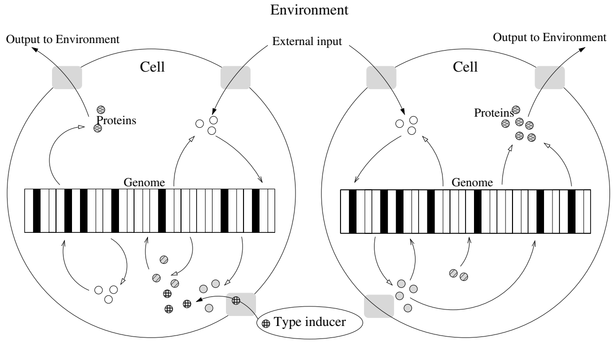

Fig. 5. Schematic drawing of our model; the two cells have the same genome and thus the same regulatory network but can produce very different behavior, induced by a very simple signal which is here shown as external, but it could also be an internal gene that is always on due to cell division disparity.

always the same. Fitness was measured based on 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 output protein i_0 and d^t_p the desired output in phase $p \in [0,1]$ relative to that of the input at time t the performance in the original setting is simply calculated as: $\sum_{t=1}^{L} |c^t_{i_0} - d^t_{0.0}|$.

For the differentiation setting, while the induced cell's desired output would still be in the the same phase as the input, we ultimately want the other cell to produce the mirror inverse of the input, which is equivalent to shifting the input's phase by one half. Fitness was measured as a funtion of the deviation from the corresponding desired output, i.e. the smaller the value, the better adapted the GRN.

With the denominations from above the deviation of the induced cell is calculated as: $\sum_{t=1}^{L} |c^t_{i_0} - d^t_{0.0}|$ and again for the other cell, only with $d^t_{0.5}$ – finally both values were added up and divided by 2 for comparability with the original setting. The lifetime L of every individual was set to 400 time steps; as a reference, over such a lifespan a random GRN typically achieved a deviation of approximately 200.

Because of results from our earlier research [6] we did not immediately, i.e. from the first generation, expect individuals to fully differentiate. Instead, the environment became *gradually* harder by increasing the relative shift in phase for two cells 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.

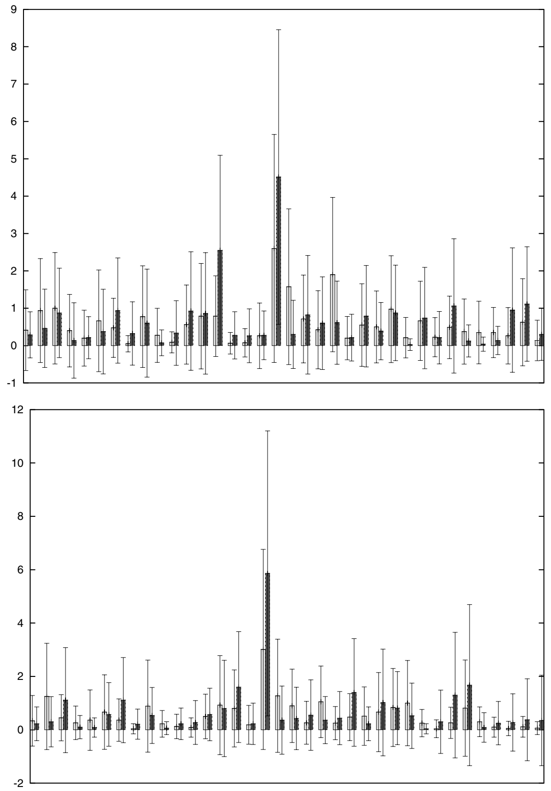

Fig. 6. **Evolved under evolutionary pressure (light gray) vs. randomly evolved (dark gray) average network pattern distributions.** On the x-axis every pair is for a subgraph pattern and on the y-axis the average within-population number of occurrences is shown. The upper plot shows results for the original and the lower for the differentiation setting. There are differences in the distributions however we find huge standard deviations and statistical significance levels are generally very low.

III. Results

In the following we compare the population subgraph patterns for the original, differentiation and random evolution settings. Every population consists of 80 individuals (four input stimuli times two expected outputs times 10 runs), each from generation 750 of one run. There are more three node patterns possible than appear in the plots, but for conciseness only those which had a number of occurences within the GRNs of a population at the end of a run exceeding at least 0.2 are shown (averaging over 10 runs, see also fig. 10 for all such patterns found in this study).

We find that subgraph patterns vary greatly between networks of all kinds (cf. standard deviations in figures 6, 7). If pattern occurrence differences are at all significant, significance levels are very low.

This huge diversity of network patterns might conceivably have been only due to the different starting populations used

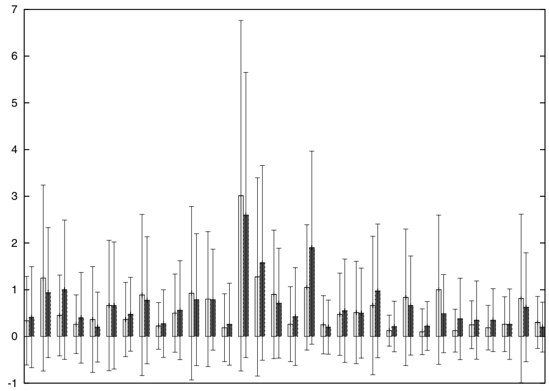

Fig. 7. **Evolved for differentiation (light gray) vs. evolved in the original setting for one task (dark gray) average network pattern distributions.** Subgraph patterns are shown on the x-axis and average within-population number of occurrences on the y-axis.

for every run (recall that we used 10 runs for every evolutionary condition). So we ran all experiments again, this time starting every run from the same initial population and only afterwards, for the variability operators, changing the random number generator seed. But even with the same set of initial networks and the same selective pressures and the same variability operators, different chance events in generating variation lead to a very big diversity in resulting networks (see fig. 8).

When comparing subgraph patterns of single evolved networks against the distributions of many random ones there was always at least one motif (significantly frequent pattern) found. This is not too surprising considering that network patterns were very likely (about 70 per cent of the cases) to be found more than once if found at all. Now we wanted to check if the most significant motifs – mostly different between networks – are functionally very important to their GRN.

A lesioning experiment should bring clarity: From every best evolved network we took away one binding site of a gene, either a) randomly any site or b) a random binding site from a subgraph of the most significant motif of this GRN. Results of running these disrupted networks and measuring their performance drop have huge standard deviations but there are on average no big differences between a) and b), cf. fig. 9. Fitness became on average worse by 47.32 ± 66.45 resp. 45.16 ± 67.73 for a) and b) in the original setting and 51.59 ± 65.88 resp. 58.19 ± 67.97 in the differentiation setting, revealing no significant robustness differences between leasioning a motif or a random location in either the original or differentiated settings. The higher complexity of the latter evolutionary setting can nicely be seen by the higher impact of the lesions on performance.

IV. Conclusions

The original intention for our analysis was to find a switch motif to control differentiation when the requirement to differentiate was the only difference between the two evolutionary

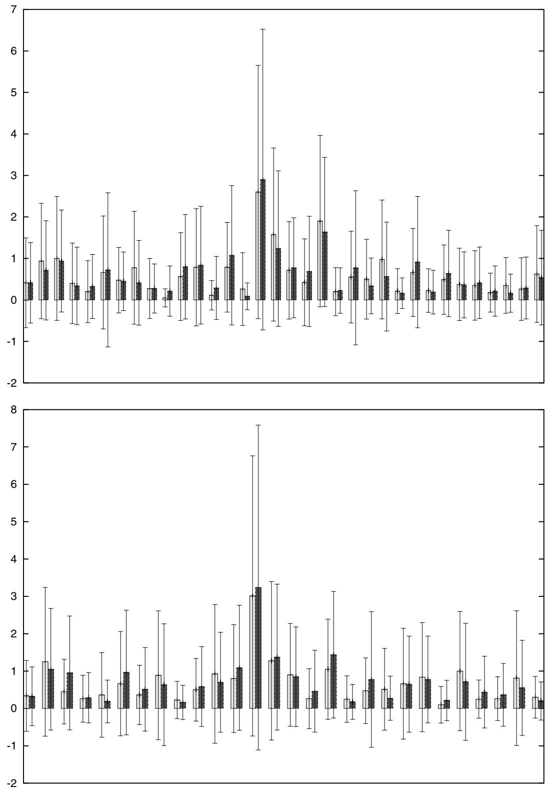

Fig. 8. **Independent initial populations (light gray) vs. same initial populations (dark gray) average network pattern distributions.** Subgraph patterns are shown on the x-axis and average within-population number of occurrences on the y-axis. The upper plot shows results for the original and the lower for the differentiation setting. Note that standard deviations are very large even when starting from the same initial populations.

environments. However this approach proved to be too naive – there was no convergence on the same single motif or a small set of switching motifs, and uniqueness of motifs was not observed. Instead we have found a wide variety of network patterns and topologies.

Although many evolutionary runs were compared, the ability of the GRNs to produce dynamics employing a wide range of different topologies might still be to a large degree due to the simplicity of the model and target functions used. Also in our model there is no cost for maintaining connections (producing proteins). Nevertheless the results warrant caution when topological measures like motif analysis are used to draw conclusions about functional properties.

We agree with [7] when they conclude that rather than investigating funtional impact of motifs "it may be more interesting to investigate transcriptional regulatory network topology with regard to the methods of network creation." Note that their GRN model as well as ours allow for large non-coding regions. For functional analysis it might be useful to focus not so much on the structure of networks, but on dynamical ("metabolic")

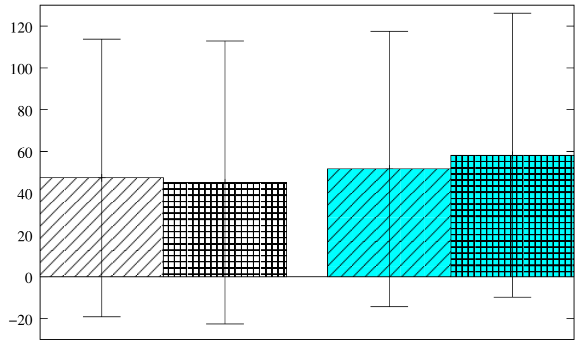

Fig. 9. **Impact of lesions:** Average reduction in fitness when one edge in a network is removed for the original (left) and differentiated GRNs (right). In each pair, the diagonally striped bar shows the impact when any edge can be chosen while for the grid patterned bar an edge from the nets most significant network motif was removed. On average the differences are very small and the standard deviations huge.

properties[6]. Currently of course it is hard to get data for biological systems; models might be at an advantage here.

"First, they [the gene motifs] may have come about through the random duplication and subsequent diversification of a few ancestral circuits. Given the high frequency at which genes and genomes undergo duplication, this is a plausible scenario. It is equally possible, however, that these circuits arose independently by recruitment of unrelated genes. If such convergent circuit evolution is prevalent, then these circuits owe their abundance to the action of natural selection" [24]. But another possibility is that we would see evolutionary convergence for complex problems where there is a clear optimum much preferred over all others. However for rather simple problems with many solutions that perform similarly well we might expect evolution to just take whatever variability generates first. That is the case in our experiments as standard deviations of network pattern occurrences are very big even when starting from the same initial populations, cf. fig. 8. Of course this has evolvability implications - for a more exact analysis lock-in effects should be taken into account; if one solution is readily created we might find it in most networks although there is a better but hard to find solution.

Summarizing, we checked for the functional importance of the network motifs found. This was done by removing one edge in the connection network, either randomly chosen from the whole network or only from the most significant motif of that network. On average there was no significant difference in fitness impact between these two settings. So in our experiments we ended up not only with many different motifs but also motifs seem not to be functionally especially important. Possbile shortcomings are the small size of networks used as well as the coarse level of detail in distinguishing network patterns, as no distinction was made between inhibitory and activatory connections, and interaction among TFs was disregarded.

[6] In addition, note that the motif measure does not treat nonlinear interactions between Transcription Factors at all, although these can have decisive impact on network dynamics.

References

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

[2] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 4th ed. Garland Science, 2002.

[3] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks." *Science*, vol. 298, no. 5594, pp. 824–827, October 2002.

[4] J. F. Knabe, C. L. Nehaniv, M. J. Schilstra, and T. Quick, "Evolving biological clocks using genetic regulatory networks," in *Proceedings of the Artificial Life X Conference*, L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, Eds. MIT Press, 2006, pp. 15–21.

[5] J. F. Knabe, C. L. Nehaniv, and M. J. Schilstra, "Genetic regulatory network models of biological clocks: Evolutionary history matters." *Artificial Life*, 2007 (in press).

[6] ——, "Evolutionary robustness of differentiation in genetic regulatory networks." in *Proceedings of the 7th German Workshop on Artificial Life 2006 (GWAL-7)*, S. Artman and P. Dittrich, Eds. Jena: Akademische Verlagsgesellschaft Aka, Berlin, 2006, pp. 75–84.

[7] P. Kuo, W. Banzhaf, and A. Leier, "Network topology and the evolution of dynamics in an artificial regulatory network model created by whole genome duplication and divergence." *BioSystems*, vol. 85, no. 3, pp. 177–200, 2006.

[8] U. Alon, *An Introduction to Systems Biology – Design Principles of Biologial Cicuits*. Chapman & Hall/CRC, July 2006.

[9] T. Quick, C. L. Nehaniv, K. Dautenhahn, and G. Roberts, "Evolving Embodied Genetic Regulatory Network-Driven Control Systems." in *Advances in Artificial Life, 7th European Conference, ECAL'03*, ser. Lecture Notes in Artificial Intelligence, vol. 2801. Springer, 2003.

[10] W. Banzhaf, "On the Dynamics of an Artificial Regulatory Network," in *Advances in Artificial Life, 7th European Conference, ECAL'03*, ser. Lecture Notes in Artificial Intelligence, vol. 2801. Springer, pp. 217–227.

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

[12] M. J. Schilstra and H. Bolouri, "Modelling the Regulation of Gene Expression in Genetic Regulatory Networks." BioComputation group, University of Hertfordshire., Tech. Rep., 2002. [Online]. Available: strc.herts.ac.uk/bio/maria/NetBuilder/Theory/NetBuilderModelling.htm

[13] T. Reil, "Dynamics of Gene Expression in an Artificial Genome - Implications for Biological and Artificial Ontogeny." in *Advances in Artificial Life, 5th European Conference, ECAL'99*, ser. Lecture Notes in Artificial Intelligence, vol. 1674. Springer, 1999.

[14] S. Kumar and P. J. Bentley, "Biologically inspired evolutionary development." in *Evolvable Systems: From Biology to Hardware, 5th International Conference, ICES 2003, Trondheim, Norway, March 17-20, 2003, Proceedings*, 2003, pp. 57–68.

[15] T. Taylor, "A Genetic Regulatory Network-Inspired Real-Time Controller for a Group of Underwater Robots." in *Intelligent Autonomous Systems 8*. IOS Press, 2004, pp. 403–412.

[16] G. Halder, P. Callaerts, and W. J. Gehring, "Induction of ectopic eyes by targeted expression of the eyeless gene in drosophila." *Science*, vol. 267, no. 5205, pp. 1788–92, 1995.

[17] W. Arthur, *The Origin of Animal Body Plans*, paperback edition ed. Cambridge University Press, 2000.

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

[19] R. T. Gregory, *The Evolution of the Genome*. Academic Press, December 2004.

[20] S. Ohno, *Evolution by Gene Duplication*. Springer, 1970.

[21] C. L. Nehaniv and J. L. Rhodes, "The evolution and understanding of hierarchical complexity in biology from an algebraic perspective." *Artificial Life*, vol. 6, no. 1, pp. 45–67, 2000.

[22] J. Maynard Smith and E. Szathmáry, *The Major Transitions in Evolution*. New York: W.H. Freeman, 1995.

[23] J. S. Taylor and J. Raes, *The Evolution of the Genome*. Elsevier Academic Press, 2005, ch. Small-Scale Gene Duplications.

[24] G. C. Conant and A. Wagner, "Convergent evolution of gene circuits," *Nature Genetics*, vol. 34, no. 3, pp. 264–266, 2003.

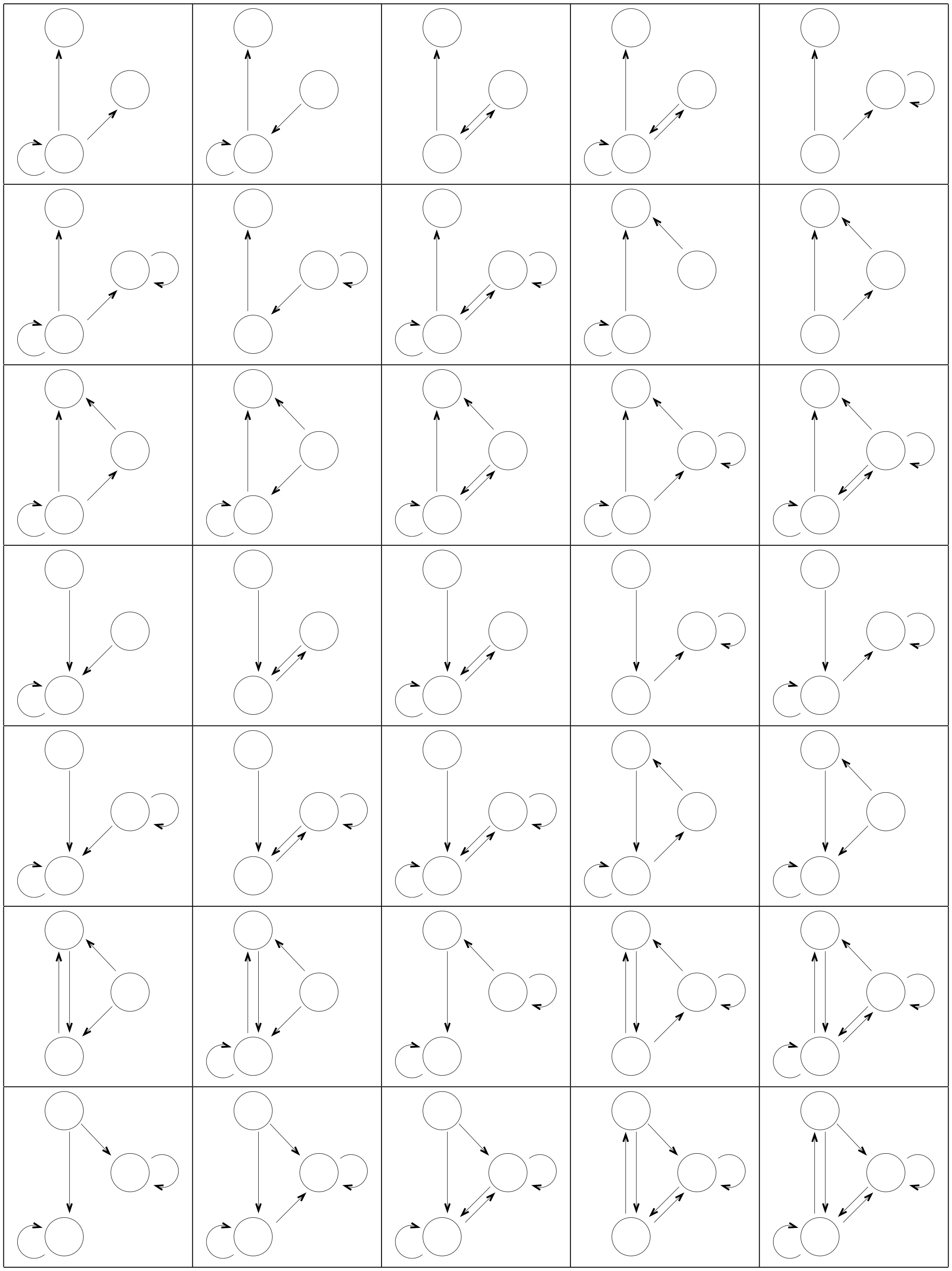

Fig. 10. 3-node subgraph network patterns occuring more often than 0.2 times within evolved GRN populations on average.