Regulation of Gene Regulation – Smooth Binding with Dynamic Affinity affects Evolvability

Johannes F. Knabe, Chrystopher L. Nehaniv and Maria J. Schilstra

Abstract— **Understanding the evolvability of simple differentiating multicellular systems is a fundamental problem in the biology of genetic regulatory networks and in computational applications inspired by the metaphor of growing and developing networks of cells. We compare the evolvability of a static network model to a more realistic regulatory model with dynamic structure. In the former model, each regulatory protein-binding site is always influenced by exactly one gene product. In the latter model, binding is only more likely to occur the better the match between site and gene product is (smooth binding) and, in addition, affinity dynamically changes under the action of *specificity factors* during a cell's lifetime. On evolutionary timescales, this means that often the strength of influences between nodes is perturbed instead of direct changes being made to network connectivity. A main result is that for evolutionary search spaces of increasing sizes evolved performance drops much more strongly in the classical network model as compared to the smooth binding model. This effect was even greater in the case of using smooth binding together with specificity factors.**

I. Introduction

In biological Genetic Regulatory Networks (GRNs), genes encode proteins and proteins in turn regulate the activation level of genes. 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]. Furthermore, cells do not exist in isolation but are embodied in an *environment*, which influences the cell, while the cell can via internal regulatory dynamics in turn influence its environment.

GRNs are often thought of as networks where nodes represent genes and arcs the influence of a gene product on a gene with a binding site which the product matches. These arcs are usually pictured as having a (static) weight representing the strength of the gene product's influence. But this is not the full story for biological GRNs, where proteins might also attach to areas they do not perfectly fit – however, the less site and protein match the smaller is the probability of binding in general. Additionally, nature has yet another level of regulation, as proteins exist that can alter the affinity of transcription factors to bind to non-perfect-match binding sites [3], [4]. Therefore these proteins are called *specificity factors* (SFs). Beside activators and repressors they are another mechanism for the regulation of gene expression that is active in biological organisms. These mechanisms have not been modeled fully in any work known to the authors. In the following we describe a model which uses smooth matching and specificity factors and their impact on evolutionary search is investigated. The target of the evolutionary algorithm (EA) is a simple two-celled model of differentiation in the sense of Jacob and Monod [5], who defined that "two cells are differentiated with respect from one another if, while they harbor the same genome, the pattern of proteins which they synthesize is different".

The other motivation for this work was the question of how to weight innovation against inheritance in successive generations in EAs. As we are concerned with networks, some small genetic changes might have profound consequences on a GRN's dynamics, leaving little resemblance to the predecessor's phenotype. In the worst case this makes an EA no better than random search, while on the other hand too much smoothness might restrict exploration and lead to premature convergence of the EA.

Related Work: Gerhart and Kirschner noted that "Eukaryotic transcription factors often have limited affinity and sequence specificity on their own and require the presence of other factors to confer stability and specificity in DNA binding" (they name Calmodulin as an example) and discuss some theoretical implications for evolution [4], [6].

Although most biologically inspired GRN models use regulatory dynamics with template matching, i.e. a perfect match of binding site and the corresponding protein is required, some approaches exist where looser matchings are possible. Banzhaf et.al. [7] proposed a bit-string model where the influence of a protein on an enhancer/inhibitor site is exponential in the number of matching bits u_i, $exp(u_i - u_{max})$, with scaling via u_{max}, the maximum match achievable, to have a full match for the best matching protein. The possible benefit of such a mechanism for evolutionary processes, namely that small changes to the genotype are more likely to have small effects on the phenotype, is pointed out. Regulation is restricted to one enhancer site and one inhibitor site per gene, if several regulating proteins are present at a site they are combined in an OR like fashion. Bentley [8] invented fractal proteins, gene products that are comprised of subsets of the Mandelbrot Set. In that model, multiple fractal protein 2D shapes are merged in an AND like fashion.

Both models allow a smooth matching between binding sites and regulatory gene products, but the affinities of gene products to sites do not change dynamically and only one kind of grouping, i.e. either logical OR or AND interaction between regulators, is possible. Furthermore no analysis of the evolutionary effects of smooth matching, e.g. by comparison to another model, is conducted in these papers.

The authors are with the Adaptive Systems (JFK, CLN) and Biological & Neural Computation (CLN, MJS) Research Groups, Centre for Computer Science & Informatics Research, University of Hertfordshire, Hatfield, AL10 9AB United Kingdom (email: j.f.knabe@herts.ac.uk; phone: +44 1707 286356).

978-1-4244-1823-7/08/$25.00©2008 IEEE

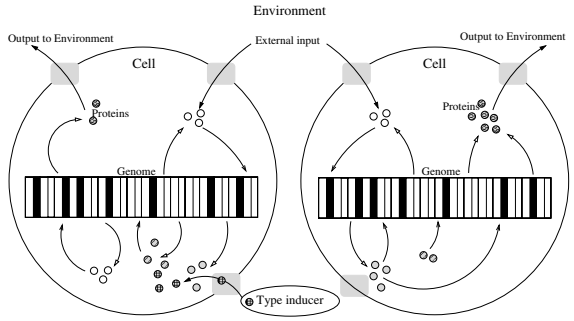

Fig. 1. **Schematic drawing of the differentiating genetic regulatory network model.** The two cells of the simple multicellular individual have the same genome and thus the same regulatory network but can produce very different behavior, induced by a very simple signal which is here shown as external, but it could also be an internal gene that is always active, e.g. due to cell division disparity resulting in an unequal distribution of proteins or other factors after cell cleavage.

II. Methodology and Model

In [9], where the proposed GRN model was first described in its basic form, we used it to evolve single-celled biological clocks with the circadian rhythm abstracted to a sinusoidal wave or other periodic function. GRNs producing such cyclic behavior in response to various periodic environmental stimuli could easily be evolved. Reproducing the phase of their input as well as the production of the inverse or shifted phase was possible[1], however in that investigation every evolutionary run had only one of these objectives. So in the context of differentiation it was quite natural to ask whether it would be possible to integrate two different functionalities in one GRN instantiated in different contexts in a multicellular entitiy. Cell cleavage and development are subject to abstraction; from the start there are two identical cells receiving the same periodic external stimuli, see fig. 1. The expected difference in behavior is only signaled by a type inducer (a raised protein level), which can be thought of as being the result of either an internal gene turned on in one cell only during cell division or an externally generated developmental signal.

Every cell consists of proteins and a genome with the number of genes fixed to nine. 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 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. A main difference from the `Biosys` model [10] is that there can be any number of cis-modules per gene and every cis-module can have any number of protein binding sites. This is to allow for a second level of protein regulation (absent from previous models), as molecular biologists have found TFs that not only show additive behavior but might also interact with each other and thereby change their influence synergistically, see e.g. [11], [12], [13]. In logical terms (but note that values are actually continuous) one can think of this grouping of inputs as an OR of ANDs. The AND level certainly constitutes a canalizing function in the sense of Kauffman [14] as one zero value there causes the whole term to be zero no matter what the other stimuli are. Such selectors can be thought of as choosing a particular pathway for the cell (and sometimes for its descendants too) and are thought to be involved in cell differentiation as well as developmental modularity. The importance of developmental modularity for evolution is pointed out by many of the articles in [15].

In summary the model, as compared to previous models, is designed to facilitate the evolution of complex dynamics, coming a little closer to nature than previous models in terms of regulatory logic, where "5-10 regulatory sites are the rule that might even be occupied by complexes of proteins" [7] and non-linear synergetic effects are possible [13].

A. Genetic Representation

The genome is represented as a string of base four digits, encoding several 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 a sequence of coding bits, a cis-module boundary and a gene boundary the genetic alphabet was increased to four values, 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 is a predefined number 2^n of different protein types, so that for example to have eight (2^3) types three bits encode a protein.

In the experiments described here we used a fixed number of genes, namely nine, as this had proven more than enough for coping with a simple single task [9], [16]. After parsing the genome into genes, the last four coding digits of every gene determine its output behavior, a number of bits for the protein type produced and the last bit for the gene's activation type, which can be "default on" – active unless repressed or "default off" – silent until activated by regulatory sites, see fig. 2.

For cis-modules the first coding bit determines its influence on the gene's activation level (*inhibitory/activatory*) and every following n coding digits are considered a protein binding site. As an example with $n = 3$ ($2^3 = 8$ protein types possible), the gene 010111021101020011113 will produce protein 7 (...200**111**13) and is "off by default" (...20011**1**3). It has two cis-modules, the first inhibitory (**0**1011102...) binding a combination of proteins 5 (0**101**1102...) and 6 (0101**110**2...), and an activatory cis-module (...2**1**10102...) to which protein 5 (...21**101**02...) will bind. Note that the last zero of 21101**0**2 is ignored; such coding digits which are neither translated nor regulatory are referred to as *junk*.

The genome also encodes several evolvable variables global to the cell. These are the *protein-specific decay rates* (four

[1]For results from those experiments see also http://panmental.de/GRNclocks/.

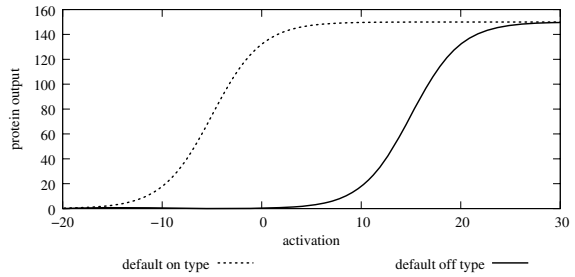

Fig. 2. **Activation Types.** Every gene produces proteins according to the cumulative activation level of its cis-modules and its activation type: either even when no activation is present ("default on" - left) or only with positive activation ("default off" - right).

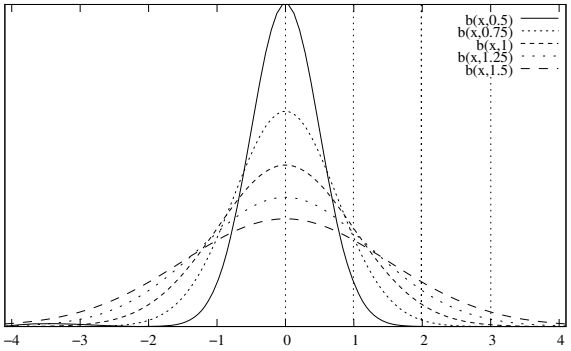

Fig. 3. **Distribution of available protein to binding sites.** The width of the curve is regulated by specificity factors and the dotted horizontal lines indicate the possible Hamming distances of the protein's bit representation to a perfect match. There are only four curves shown for clarity, but all intermediates are possible as well. See text for details.

bit for every protein, indexing into a fixed look-up table of values), the global *binding proportion* (also four bits indexing into a look-up 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). These latter variables especially facilitate changes in the strength and timing of gene expression without affecting the general dynamics important in the evolution of phenomena such as heterochrony (cf. [17], for a more detailed analysis in our model see [9], [16]).

B. Regulatory Logics

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. The next two subsections describe the two different matching mechanisms compared.

1) Template or Perfect Matching: Here the fraction of proteins available for binding is assigned to the binding site that has the same binary code as the protein. 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 time-step after saturation and protein-specific decay have been taken into account, with the initial condition $p_{ijm}^0 = 0$.

2) Smooth Matching and Specifictiy Factors (SFs): Above it was assumed that the binding sites on the cis-modules need to have exactly the same structure (bit pattern) as the protein that binds to them. In this variant proteins might also attach to areas they do not perfectly fit. As a protein's affinity for a binding site should depend on how well they match, a measure of closeness is needed. This is achieved by calculating the Hamming distance between their bit representations. The Hamming distance is simply the number of bits that are different between two strings, so for eight different protein types ($n = 3, 2^n = 8$) the maximum Hamming distance is 3. The distance values are then used as input to a bell shaped function, see fig. 3. The width of the curve, i.e. the protein's affinity for binding sites that it does not match perfectly, depends on SFs. SFs were incorporated as additional proteins, one for every of the original 2^n proteins, being produced by genes just as the old ones. So the number of proteins in the cell doubles to 2^{n+1}; however, SF proteins do not bind to binding sites themselves but are only used as modifiers when the other proteins are binding. Every SF influences the binding behavior of its corresponding protein (where the correspondence is given by SFnumber minus 2^n, i.e. the highest bit of a protein indicates normal protein or SF protein). The bell shaped curve (fig. 3) is given by:

$$b(x, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

The value for x is the distance of protein i (the one to be distributed) to a binding site and the value for σ is $0.5 + c_i^S$, where $c_i^{S,t}$ is the concentration of SF for protein i, $i \in \{0..2^n\}, c_i^{S,t} \in [0, 1]$. Let H_{ih} denote the Hamming distance between bit representations i and h. We can now formalize the number of protein types with a distance d as $N(d) = |\{x|H_{0x} = d\}|, x \in \{0..2^n\}$.
So for the example with $2^n = 8$ protein types we have $d \in \{0, 1, 2, 3\}$, and for every protein there is one binding site type matching perfectly ($N(d = 0) = 1$), three binding site types with $N(1) = 3$, three with $N(2) = 3$, and only one where all bits are different ($N(3) = 1$).
Combined, with the denominations as above, we have:

$$p_{ijm}^t = p_{ijm}^{t-1} + \sum_{h=1}^{2^n} c_h^t \left(\frac{b(H_{ih}, 0.5 + c_h^{S,t})}{\sum_{z=0}^{n} b(z, 0.5 + c_h^{S,t})}\right)/b_h N(H_{ih})$$

Note that a) the division by $\sum_{z=0}^{n} b(z, 0.5 + c_h^{S,t}))$ only occurs to normalize the sum of shares to 1, b) proteins decay with the decay–rate specific to the protein that perfectly matches the binding site they attach to, so one might speak of a binding-site-type specific decay rate now, and c) unlike concentrations of normal proteins c_i^t, SFs $c_i^{S,t}$ currently decay completely every time step (i.e. after the above calculations are done). Because of c) no decay rates for SFs need to be evolved and no saturation value applies, but $c_i^{S,t} \in [0, 1]$. When calculating the protein output of genes that produce SFs we simply do not multiply by r (limiting its output to values in $[0, 1]$) and if two genes are producing the same SF and their combined output is above 1 the value is just set back to 1.

Activation Levels: For both matching mechanisms 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, 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]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.

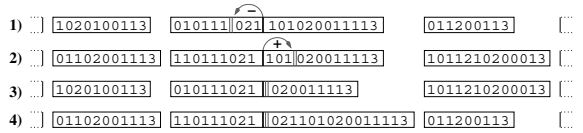

Fig. 4. **Gaussian offset crossover.** Genomes of (1) parent 1, (2) parent 2, (3) offspring 1, (4) offspring 2. Only the compartment chosen for crossover and two neighboring genes are shown. Both children get digits up to the crossover point (solid bar) from their respective parent, but then continue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

C. Evolution

A standard Genetic Algorithm with elitism, tournament selection and replacement is used. Every evolutionary condition was studied with ten runs, each lasting 500 generations containing 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.

Selection: Later generations are formed by carrying over the best-performing individual (the performance measure is described below) 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".

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. As there can be a variable number of cis- and of protein binding sites per gene their lengths will vary, so a standard bit-string crossover could change the number of genes drastically. To conserve all but (at most) one of the genes as basic building units, the genomes of the parents are divided 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.[3] This process is inspired by the biological mechanism known as synapsis, the pairing of homologous chromosomes where mostly "similar" sectors pool together. To achieve variable length genes, the unequal crossing-over observed in biology is mimicked: 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. 4).

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

[3]This is why 'at most' one gene is changed: The crossover point could be zero or equal to the gene's coding length.

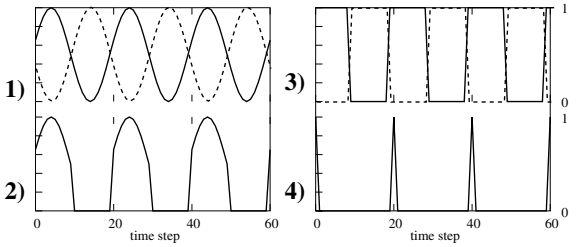

Fig. 5. **Periodic functions used**: 1) sine (dashed the inverse or shifted wave), 2) positive part of sine, 3) step (dashed the shifted wave), 4) pulse.

chance of duplicating genetic information, the importance of which was already pointed out by [18] 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" [19].

Note that the offset point is limited to stay within the boundaries of the compartment, hence if crossover point + 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 offsets would be meaningless here.

These processes allow both neutral crossover and mutational changes, as degenerate cis-modules (i.e. less than n bits – one protein – long) are ignored. Additionally this means that genes could become dysfunctional, in a similar manner to the so called pseudo-genes found in nature, e.g. if there were not a single cis-module and the gene had an activation type of "off by default".

D. Environmental Coupling

Evolutionary conditions are systematically varied by changing the pattern of external signal received at the cellular level 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 w was set to 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. 5. The impact of Gaussian noise and black-out periods in the input during evolution was investigated in earlier experiments published elsewhere [16].

As mentioned above, both cells of an individual always received the same periodic stimuli, however one cell additionally received an *inducing* signal with a value of 1.

Output behavior: Two periodic target functions were used to measure the performance of an individual and assign fitness: sine (fig. 5.1) and step (fig. 5.3), with the first requiring more smooth changes of protein levels and the latter a boolean like pattern. 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 using the deviation from the corresponding desired output, i.e. the smaller the deviation, 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 again for the other cell, only with $d^t_{0.5}$ – afterwards both values were added up and divided by 2.

However 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. Earlier experiments, published elsewhere [20], have shown that this gradual differentiation leads to better performance.

The lifetime L of every individual was set to 400 time steps. A randomly-generated initial GRN could typically achieve a deviation of approximately 200 over this time. Finally, we use this value to transform the deviation to a standard 0 to 100 performance scale: $(200-D)/2$, so zero deviation would result in a perfect performance value of 100.

III. Experimental Set-up and Results

Overall, 8 evolutionary scenarios were tested (two desired output types times four environmental stimulus input functions) and each scenario was run ten times. The whole experiment was repeated with the two different regulatory mechanisms: Perfect matching and smooth matching plus specificity factors[4].

Using smooth matching and specificity factors did not immediately lead to an increase in performance in the setting with $n = 3$ bits coding for a protein. The reason was probably that the small number of proteins created too many interferences for fine grained regulation, as the picture changed when n, the number 2^n of proteins and thereby the search space was increased. While results for the perfect matching condition became worse quickly as n increased, smooth matching had only slight losses, cf. table I [5]. In almost all evolved GRNs we found the use of SFs during lifetime, although there was not a clear trend towards less or more use over evolutionary time.

[4]Results for experiments with smooth matching but without specificity factors can be found online at http://panmental.de/CECdynAff. Generally these results were in-between the perfect matching condition and smooth matching plus specificity factors.

[5]Note that equal numbers of protein bits are not directly comparable as for the smooth matching condition one protein bit is used as a "specificity factor" flag.

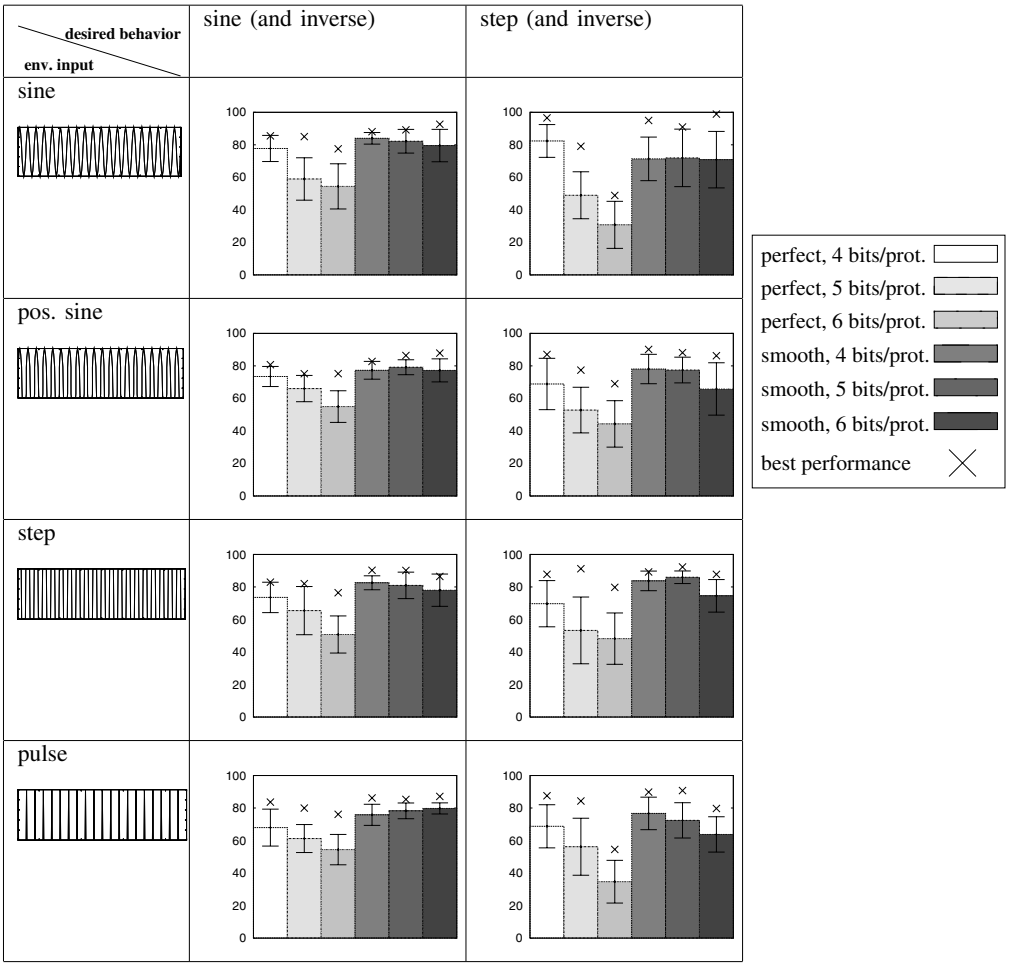

TABLE I

PERFECT VS. SMOOTH (PLUS SPECIFICITY FACTORS) PROTEIN MATCHING EXPERIMENT OUTCOMES, WITH THE LEFTMOST COLUMN DEPICTING THE ENVIRONMENTAL STIMULI USED AND THE TOPMOST ROW THE DESIRED OUTPUT BEHAVIOR FOR EVERY RUN. THE NUMBER OF GENES WAS FIXED TO 9 AND GRADUAL DIFFERENTIATION PRESSURE WAS USED. DATA CELLS SHOW THE BEST FINAL DEVIATION FOR RUNS VARYING THE NUMBER OF BITS USED TO ENCODE A PROTEIN. ALL VALUES ARE AVERAGED OVER 10 RUNS WITH 500 GENERATIONS TIMES 250 INDIVIDUALS EACH, $\pm$ THE RESPECTIVE STANDARD DEVIATION.

IV. DISCUSSION

The introduced genetic regulatory network (GRN) model with two layers of regulation is clearly able to evolve functional differentiation. Two such protein matching mechanisms were studied: Simple template matching (also referred to as perfect matching) and smooth matching complemented with specificity factors. For smooth matching a regulatory protein is only more likely to bind binding sites it exactly matches but with some probability it can also bind to sites of a different type, thus making the mutation operator less destructive. The more different site and protein type the less affinity for binding there is. The probabilty distribution that determines how strong the affinity towards perfect matches is can be changed by specificity factors during a GRN's "lifetime" thereby adding a level of regulation. Our experiments show that the use of smooth matching and specificity factors, compared to a static binding mechanism, can increase performance of evolutionary algorithms – especially for larger search spaces.

However, the use of smooth matching alone will also increase *pleiotropy* (one gene influences multiple phenotypic traits). A pleiotropic gene might cause problems when genetic changes to it improve one trait while doing the opposite to another trait. Here specificity factors might be advantageous: They can modify the level of the smoothness

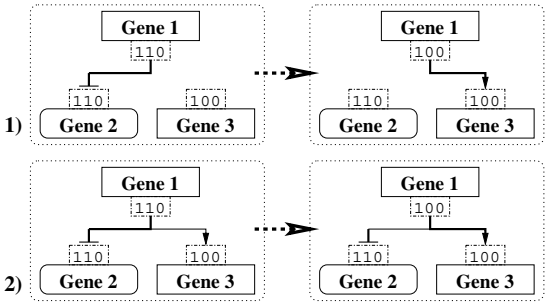

Fig. 6. **Schematic drawing of the change of the network structure after one single bit mutation occurs**, for 1) the perfect matching and 2) the smooth matching condition. Bolder lines represent stronger regulating influences. Note however that specificity factors can complicate this picture by dynamically changing affinities.

of the distribution of transcription factor proteins binding to potential sites but their activity also influences how genetic variability will change network dynamics. Depending on lifetime specificity factor levels, a mutation could have any effect between the "perfect matching" condition and almost no effect at all for very smooth distributions. This difference is schematically depicted in fig. 6.

As Altenberg [21] noted: "Genes are selected on for their organismal fitness effects but modify the variational properties of the genome as a systematic side effect". These effects and whether there is much interference between functionalities depends on the possible forms the distribution curve (see fig. 3) can take, which might be an interesting subject for further research. Also it would be interesting to see if the presented positive results still hold for more complex differentiation scenarios or other network formalism variants.

The source code and additional results are available at http://panmental.de/CECdynAff.

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] Raven and Johnson, *Biology*, 7th ed. New York: McGraw-Hill, 2004.

[4] J. C. Gerhart and M. Kirschner, *Cells, Embryos And Evolution*. Blackwell Publishing, June 1997.

[5] F. Jacob and J. Monod, "Genetic repression, allosteric inhibition, and cellular differentiation," in *Cytodifferentiation and macromolecular synthesis*, M. Locke, Ed. London: Academic Press, 1963, pp. 30–64.

[6] M. Kirschner and J. Gerhart, "Evolvability," *Proc Natl Acad Sci USA*, vol. 95, no. 15, pp. 8420–8427, July 1998.

[7] 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, 2003, pp. 217–227.

[8] 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.

[9] J. F. Knabe, C. L. Nehaniv, M. J. Schilstra, and T. Quick, "Evolving biological clocks using genetic regulatory networks," in *Artificial Life X: Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, L. M. Rocha, L. S. Yaeger, M. A. Bedau, D. Floreano, R. L. Goldstone, and A. Vespignani, Eds. MIT Press/Bradford Books, 2006, pp. 15–21.

[10] 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, pp. 266–277.

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

[12] M. J. Schilstra and C. L. Nehaniv, "Bio-logic: Gene expression and the laws of combinatorial logic," *Artificial Life (special issue on Systems Biology)*, vol. 14, no. 1, 2008 (in press).

[13] A. Cornish-Bowden, *Fundamentals of Enzyme Kinetics*. London: Portland Press, 2001.

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

[15] G. Schlosser and G. P. Wagner, Eds., *Modularity in Development and Evolution*. University Of Chicago Press, July 2004.

[16] J. F. Knabe, C. L. Nehaniv, and M. J. Schilstra, "Genetic regulatory network models of biological clocks: Evolutionary history matters." *Artificial Life*, vol. 14, no. 1, pp. 135–148, 2008.

[17] L. W. Buss, *The Evolution of Individuality*. New York: Columbia University Press, 1987.

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

[19] J. S. Taylor and J. Raes, "Small-scale gene duplications," in *The Evolution of the Genome*, T. R. Gregory, Ed. Elsevier Academic Press, 2005.

[20] J. F. Knabe, C. L. Nehaniv, and M. J. Schilstra, "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. [Online]. Available: http://panmental.de/GWALdiff/

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