

Emergence of Feedback in Artificial Gene Regulatory Networks

Till Steiner, Lisa Schramm, Yaochu Jin, Bernhard Sendhoff

2007

Preprint:

This is an accepted article published in IEEE Congress on Evolutionary Computation, CEC. The final authenticated version is available online at: [https://doi.org/\[DOI not available\]](https://doi.org/[DOI not available])

Emergence of Feedback in Artificial Gene Regulatory Networks

T. Steiner¹, L. Schramm², Y. Jin¹, B. Sendhoff¹

¹*Honda Research Institute Europe
Carl-Legien-Str. 30, 63073 Offenbach/Main, Germany
{till.steiner, yaochu.jin, bs}@honda-ri.de*

²*Technische Universität Darmstadt
Karolinenplatz 5, 64289 Darmstadt, Germany*

Abstract—In this paper, we present a model for simulating the evolution of development together with a method for the analysis of emergence of negative feedback inside the regulatory network. In order to record the development of feedback during evolution, we analyze both the static as well as the dynamic interactions between the transcription factors in the regulatory network. When perturbing the gene regulatory network using random mutations, we find that the evolved negative feedback is the main mechanism for robustness against such mutations. We argue that this robustness is the reason for the sustained emergence of negative feedback during evolution.

I. INTRODUCTION

Simulated evolutionary development combines evolutionary computation with simplified models of developmental biology. In this approach, an artificial genotype encodes the developmental process through Gene Regulatory Networks (GRNs) resulting in an indirect representation that is different from traditional evolutionary algorithms (EAs). Indirect representations are expected to be required for the evolution of complex systems. Apart from their scalability property, indirect representations (or encoding schemes) can help to understand and to exploit important principles of biological evolution, such as modularity [16], [18], robustness against mutations and environmental changes [11], and self-repair [2] to name but a few.

Several models have been proposed for simulating developmental processes, see [4] for a comprehensive overview. The majority of research focuses on the phenotypic features that the simulated developmental process results in. These features include the ability to grow two dimensional patterns like flags [13], [8], [9] and tessellating tiles [1], the ability to self-repair [3], or in general the ability to grow complex shapes and structures with varying notions of complexity [5], [10]. However, the mechanisms that cause these features, as well as the path that evolution follows to discover the representation that develops them, are not thoroughly investigated. This knowledge is required for achieving both a more fundamental understanding of biological systems and an improvement of the performance of evolutionary algorithms with indirect representations. Furthermore, in order to be able to construct artificial gene regulatory networks for simulated developmental processes dependent on the problem domain one wants to investigate, a deeper understanding of the complete system is required.

When modeling the developmental process on a computer, we have to carefully choose the appropriate abstraction level of the biological system. It is clear that the level of abstraction influences the behaviour and the result of the simulation and therefore, both its plausibility in a biological context as well as its applicability in a technical context. Although abstraction might limit the direct explanatory power of the results for biological observations, it helps us to focus on the most fundamental principles of the process because our view is not longer obstructed by the countless details of biology and biochemistry. Furthermore, in a computational framework, experiments can be re-run with exactly the same course of events, and virtually all processes can be altered by the experimenter without problems, limited only by the available computing power. This allows us for example to observe the dynamics of GRNs closely and investigate their evolution more thoroughly.

The dynamics of GRNs determine the developmental process and in turn strongly influence the characteristics of the phenotype including its shape, structure and behaviour. If a mutational change of the genotype affects the individual at all, it will cause a change in the dynamics of the GRN possibly without changing the static representation of the GRN, e.g. a heterochronic mutation may change the timing of regulatory events only.

In this paper, we investigate the evolution of simulated GRNs with a particular emphasis on their dynamics. We analyze the emergence of inhibitory feedback during evolution, which is, among others, believed to be responsible for robustness against mutation in biological systems [21]. A negative feedback loop in GRNs occurs when a gene directly or indirectly inhibits its own expression. Computational models of simple organisms have successfully identified feedback mechanisms that are used in heat shock response [6] and chemotaxis [22] of *E. Coli*. Here, we trace the emergence and persistency of such feedback mechanisms in evolving individuals and analyse the effect of evolved negative feedback on robustness against mutation. We argue that this kind of analysis can only reasonably be performed by investigating the dynamics on GRNs rather than the static regulatory interactions.

The paper is structured as follows. In Section II, we present the developmental model that we use for our simulations.

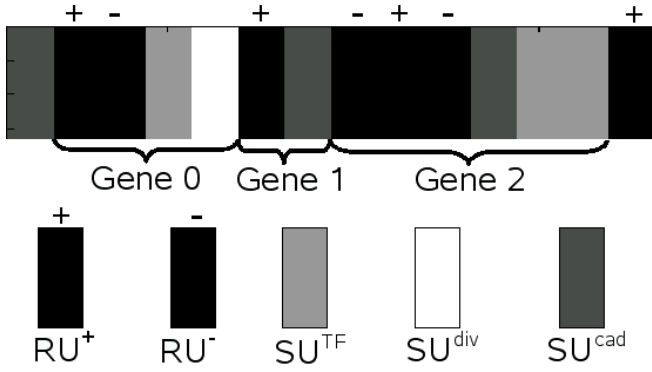


Fig. 1. An illustrative vDNA with three genes, each consisting of one or more structural subunits (SUs) and regulatory subunits (RUs). Two different kinds of RUs exist: inhibitor (RU^-) and activator (RU^+). A SU coding for the production of a transcription factor (TF) is denoted by SU^{TF} , a SU coding for a division by SU^{div} and a Cadherin producing SU by SU^{cad} .

Section III briefly describes the evolutionary algorithm used in this study. Methods for analyzing the resulting GRNs and the experimental results are given in Section IV and Section V, respectively. We conclude with a discussion of the results and an outlook in Section VI.

II. MODELING DEVELOPMENT

The model used in this work is based on the one we proposed earlier [19] with two major extensions: the implementation of physical interactions between cells and modifications to the genetic representation, so that the model is biologically more plausible. Firstly, we describe the genetics of our model, followed by a description of the cellular representation we use. Thereafter, we shortly illustrate the sequence of events during a simulation.

A. Genetics

In our model, cellular growth is controlled by a genome stored inside a virtual DNA (vDNA), of which an identical copy is available for translation to all cells in an individual. This genome consists of regulatory subunits (RUs) and structural subunits (SUs), which are initially lined up in a random order. A functional unit of this DNA, called a gene, is composed of a group of SUs and the preceding RUs. The SUs encode actions that a cell should perform, while the RUs determine whether a gene is active or not. The actions encoded in the gene will be performed only if it is active. Both RUs and SUs are represented by a set of double precision values. An illustrative example of a genome with three genes is given in Fig. 1. Note that the RUs behind the last SU and the SUs in front of the first RU are not taken into account for the developmental process.

- **Structural subunits:** A SU encodes the action to be performed, and contains the parameters that specify the action. Possible actions include cell division, production of a diffusing chemical, the transcription factor (TF) for cell-cell signaling, and production of Cadherin

molecules on the cell surface, which determine cell-cell adhesion forces.

Formally, a SU consists of a vector \mathbf{x} with five components $x_i \in [0..1]$, $i = 1, \dots, 5$. x_1 is used to determine the type t of action encoded by the SU:

$$t = \begin{cases} 1 & \forall \mathbf{x}: 0 \leq x_1 < \frac{1}{3} \\ 2 & \forall \mathbf{x}: \frac{1}{3} \leq x_1 < \frac{2}{3} \\ 3 & \forall \mathbf{x}: \frac{2}{3} \leq x_1 < 1 \end{cases}$$

If $t = 1$, cell division is encoded and x_2 is used to determine the division angle, while the values x_3 to x_5 are ignored. If $t = 2$, the production of a TF is encoded and x_2 represents an affinity value assigned to the TF (aff_{TF}), x_3 defines the amount of TF to be released, x_4 is a diffusion constant, and x_5 the decay rate.

In the case of $t = 3$, the production of Cadherin molecules is encoded by the gene and the type of Cadherin is determined by x_2 . In our model, cells containing the same type of Cadherin will adhere to each other. Note that for $t \neq 2$, not all x_i are used, but they are still kept as a part of the SU. Therefore, mutation affects them without being subject to selection pressure. If a mutation results in $t = 3$, all five values are required again as parameters for the production of a TF. So far we have not studied the effect of this intermediate random-walk of the redundant parameters on the evolution process.

- **Regulatory subunits:** Two types of RUs are used in our model, which either increase (activate) or decrease (inhibit) the expression of a gene. RUs can sense the presence of certain types of TFs in the vicinity of the cell. If the label of a TF is affine to a label associated with the RU, and if the concentration of the TF lies above a threshold, an activity value is determined for each RU. All activating (= positive sign) and inhibiting (= negative sign) activity values belonging to the same gene are summed up to determine the overall activity of the gene.

More formally, a RU consists of a vector \mathbf{y} with three components $y_i \in [0..1]$, $i = 1, \dots, 3$. y_1 codes for an affinity parameter, which is used to determine the affinity between that RU and the surrounding TFs. If the affinity decision variable γ , calculated by

$$\gamma = 0.2 - |\text{aff}_{TF} - y_1|$$

is greater than 0, the TF and the RU are affine to each other.

Let M be the number of RUs belonging to a certain gene. Let L_j be the number of TFs that are affine to RU_j , the j th RU of the gene, and y_i^j the i th entry of the vector \mathbf{y} of RU_j . We first determine the indices k_j , which denote the TFs that have a concentration c that is greater than y_2^j :

$$k_j = \{k \in \{1, \dots, L_j\} : c_k > y_2^j\}.$$

Here, y_2^j can be seen as a threshold of RU_j . The partial activity a_{RU_j} for the RU is given by

$$a_{RU_j} = \sum_{k_j} c_{k_j} - y_2^j$$

A sum over all partial activities, scaled by y_3 yields

$$\alpha = \sum_{j=1}^M a_{RU_j} \cdot (2 \cdot y_3^j - 1).$$

y_3^j can be interpreted as the sign of RU_j , because the term in brackets is negative for $y_3 < 0.5$. The overall activity A of the gene is finally determined by

$$A = \frac{2}{1 + \exp(-20 \cdot f \cdot \alpha)} - 1,$$

where f denotes the slope of the nonlinear function and is encoded in the vDNA for every gene. If A is greater than zero, the gene is active.

B. Cells and their interaction

The simulation area for cellular growth is defined by an equally spaced 26 by 26 grid with step size 0.5 on which the concentrations of the TFs are calculated. Cells are modelled as spheres with a radius of one. They interact with each other by reading and releasing TFs and by cellular motion through rigid body interactions coupled with adhesion forces, which are implemented in a similar way as described in [14]. We slightly modified the nonlinear force function and do not include deformations of the spheres. Instead, we allow for a small overlap between neighbouring cells. Note that cell positions are not fixed to the grid. Therefore, cells read the concentrations of TFs from the four nearest nodes of the diffusion grid and interpolate their actual value. The release of a TF by a cell is simulated by an increase of the concentration in the four nearest nodes on the diffusion grid. Each TF is simulated separately using the forward Euler method to discretize and solve the diffusion equation:

$$\frac{du}{dt} = D_f \nabla^2 u - D_c u + s(t),$$

where u is the concentration of the TF, D_f is the diffusion constant, D_c the decay constant and $s(t)$ is a source term, which contains the increase in concentration caused by the cells.

Since the evolutionary target of this study does not depend on cell sorting and only weakly on the shape of the individuals, the implemented mechanism for cell adhesion is fairly simple. If two cells contain the same type of Cadherins (which means that they express the same gene), they will adhere to each other.

There are several alternative models available for the simulation of cellular growth and their interaction. Frequently, cells are modelled as pixels on a fixed grid [20]. A spring-mass-damper system has been used in [15] to simulate the shape and physical behaviour of plant cells. As mentioned above, different cell models require the evolution of different

control mechanisms, resulting in different gene regulatory systems with varying properties. We have chosen our model mainly for two reasons. Firstly, it is easy to implement yet sufficiently biologically plausible. Secondly, in our model, a cell can always perform the actions that its genome activates. In contrast, it may happen in the pixel model that a division does not take place because the space for the new cell is already occupied by another cell. Therefore, the control of activation for such a gene would no longer evolve, since its function is automatically disabled.

C. Time scales and sequence of events

In the beginning of development, a single cell containing the vDNA is placed at the center of the simulation area. To start the growth process, an initial TF (maternal TF) is released, which maintains a constant concentration in the whole area over the entire developmental time. Contrary to most existing models, e.g., [3], [7], the initial TF concentration in our model does not provide any positional information. Rather, it fulfills the minimal requirement for starting a developmental process.

In each developmental step, the following events take place: Firstly, the translation of the DNA is initialised for all existing cells. Secondly, if the TFs in the vicinity of the cell activate a gene, the action that the gene encodes is executed. Finally, the position of all cells is updated and the diffusion of the released chemicals is simulated.

III. THE EVOLUTIONARY ALGORITHM

An evolutionary strategy (μ, λ)-ES with individual strategy parameter adaptation [17] is adopted in this work. The main variation operator in the ES is the mutation operator that adds a normally distributed zero-mean random number to each object parameter. Each design variable has its own variance, which self-adapts to the fitness landscape during evolution. Mutation and self-adaptation are shown in equation (1), where the x_i denote the design variables in our model as described in the last section, the σ_i the standard deviations of the normal distribution, and the index g is the generation counter.

$$\sigma_i^{g+1} = \sigma_i^g e^{\tau_0 w} e^{\tau w_i}, \quad w, w_i \sim N(0, 1) \quad (1)$$

$$x_i^{g+1} = x_i^g + z_i, \quad z_i \sim N(0, (\sigma_i^{g+1})^2) \quad (2)$$

$$\tau_0 = \frac{1}{\sqrt{2N}}, \quad \tau = \frac{1}{\sqrt{2\sqrt{N}}}.$$

We do not employ a recombination operator in this paper.

Different from conventional ESs, both gene transposition and gene duplication are implemented. During transposition two randomly chosen units (both SUs and RUs are possible) are marked and all units between these two marked units are cut out and pasted at another randomly chosen position. Gene duplication works similarly with the difference that the units are not cut out but copied to another randomly chosen location.

The probabilities for gene transposition and gene duplication are given by $p_t = p_m \cdot p_1$ for transposition and $p_d = p_m \cdot (1 - p_1)$ for duplication, respectively.

In this paper, the goal of evolution is to find GRNs that result in a stable developmental process before the maximum number of developmental time steps has been reached.

By stable development, we mean that the development must reach a state where cells do not move or divide anymore, i.e., the concentration of the TFs must either have decayed to a value below all activation thresholds, or reached a stable value, which indicates that no further change in gene activity will occur. Finally, the finite number of cells that make up the individual should be located inside a predefined diamond shape centered on the grid.

The evolution of finite growth is formulated as a minimization problem. The fitness f is given by the following equation:

$$\eta_i = \begin{cases} -1 & \forall \|\mathbf{p}_i\|_1 \leq 5 \\ 1 & \forall \|\mathbf{p}_i\|_1 > 5 \end{cases},$$

$$f = \sum_{i=1}^N \eta_i,$$

where \mathbf{p}_i is a two-element vector containing the position (x_i, y_i) of the i -th cell of the individual in the last timestep, N is the total number of cells, and $\|\cdot\|_1$ denotes the 1-norm. In other words, the fitness is expressed by the number of cells outside a diamond shape around the center of the calculation area, minus the number of the cells inside the diamond shape. If the constraints are violated, i.e., if the cells touch the border of the simulation area, or if the growth process does not reach a stable state within a maximum of T_D developmental time steps, a penalty term of +700 is added to the fitness function.

IV. THE ANALYSIS OF NETWORK DYNAMICS

In this section, we present the methods that we employ for the analysis of the computer experiments outlined in the next section. Firstly, a traditional, static representation of an evolved GRN is described. After demonstrating its weakness for the analysis of feedback, a description of the dynamic GRN and its usefulness follows.

A. Static GRN

Fig. 2 depicts the static interactions of the GRN belonging to an individual which results from generation 43 of an evolutionary run described in Section III. We used the Cytoscape-software¹ for this visualization. The static interactions can be directly derived from the vDNA of an individual in the following way: An arrow from a SU to a RU denotes that the RU takes part in the activation calculation for that SU. This is determined by the position of the RUs relative to the SUs inside the vDNA. An arrow from a TF-coding SU to a RU denotes that the label of the respective TF is affine to the label associated with the RU. Therefore, if that TF is produced, it will act on the RU if its concentration exceeds its threshold. If a gene consists of more than one SU, SUs are

grouped together (directly adjacent, or with an arrow directly between them).

This kind of representation can be useful for an overview over possible interactions, although the generally high number of interactions makes it hard to analyse them in detail.

However, the major drawback of this visualization method is that it does not become clear which interactions really become activated during development. The reason is that the interaction between a TF-coding gene and a RU depends on thresholds and the concentration of the TF. The concentration depends on the position of the cell which the gene belongs to, the expression rate of the TF and the actual developmental time step. In Fig. 2, we highlighted such a negative feedback loop, which is only one among many (in fact, the close-up reveals a direct negative feedback where the gene acts on its own inhibitory RU). However, the dynamic GRN analysis described in the next section reveals that none of these negative feedbacks are used during the development of the respective individual.

Therefore, to get an insight into the real interactions, the missing information – TF-concentrations and time steps – needs to be included.

B. Dynamic GRN

In Fig. 3, we depict a time series of network interactions as they take place in the first cell of an individual. Genes are represented by points and arranged in a circle. Since information about TF concentrations in the vicinity of the cell can be obtained for every timestep, the real interactions between genes can be shown. In each time step, the interactions are updated according to the changing TF concentrations. The top solid point in Fig. 3i) denotes the pre-diffused TF and therefore, exhibits initial interactions. From there, gene activation and inhibition can be tracked in each successive time step, from Fig. 3i) to Fig. 3vi).

Note that this dynamic representation of the GRN can differ from cell to cell. For our experiments, we checked that all cells of one individual reach the point where the GRN converged to the same stable state. Therefore, our analysis is performed only for the first cell of an individual.

We use the information provided by the dynamic GRN for negative feedback analysis. In every developmental timestep, we search for closed loops in the GRN and count the number of negative interactions which are part of the loop. This is achieved by transforming the network into a tree-graph and looking for the occurrence of already visited nodes by stepping along the tree. The method yields the number of negative feedback loops for all developmental time steps in one individual. By comparison, we can eliminate the occurrence of the same loop in successive time steps and thus find the number of unique inhibitory feedbacks used throughout the developmental process.

V. THE EMERGENCE OF FEEDBACK

A. The Experimental Setup

In the experiment, the parent and offspring population sizes are set to 400, and 2000 respectively. Evolution has

¹Cytoscape is a software for the visualization of biologic datasets, such as molecular interaction networks. <http://www.cytoscape.org>

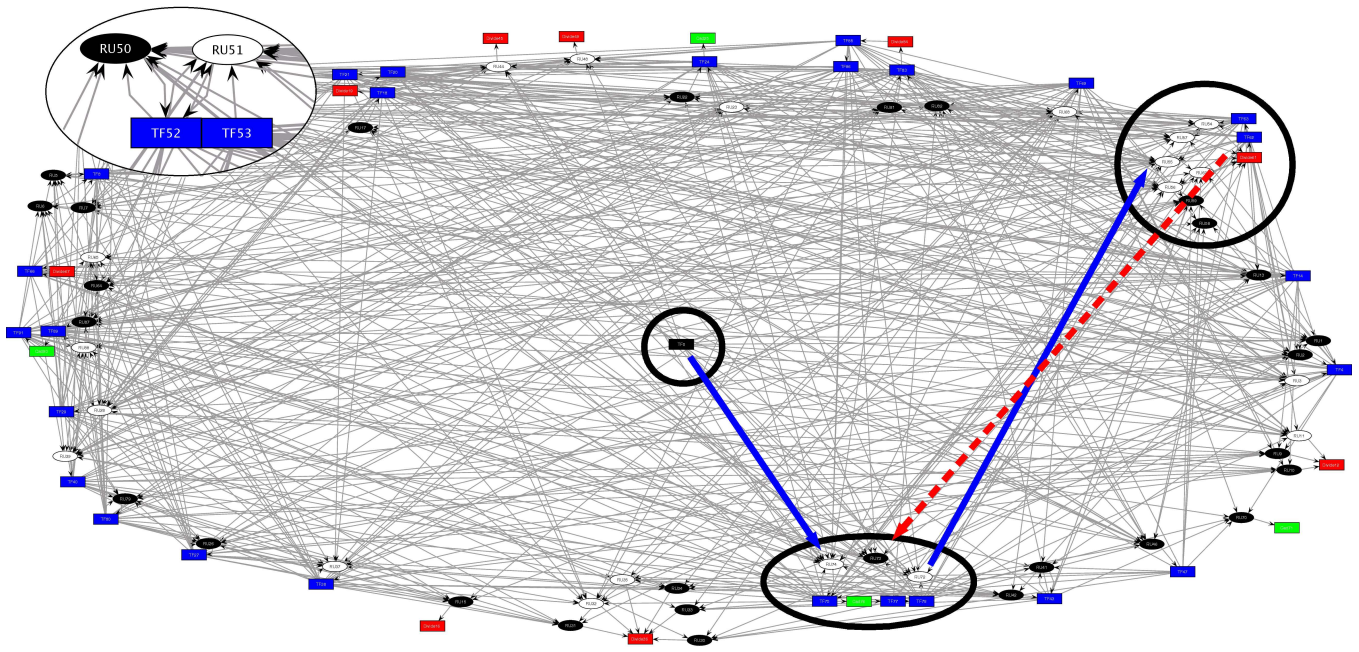


Fig. 2. The static interaction network of an individual from generation 43. The pre-diffused TF is placed in the center of the network. A close-up on one gene is depicted in the upper left corner: the gene consists of an inhibitory RU (black ellipses), an excitatory RU (white ellipses) and two TF-coding SUs (blue rectangles). Two interacting genes and the pre-diffused TF are emphasized by bold circles. A positive interaction (solid arrow) from the pre-diffused TF to the lower gene denotes an excitatory connection, which could be the starting point of a negative feedback loop between the two marked genes (the dashed arrow denotes a negative interaction). Note however, that the analysis of the dynamic GRN reveals that this feedback is not used, because the concentrations of the TFs do not exceed the threshold values.

converged after approximately 110 generations. The probability of the variation operations are given by $p_m = 0.1$ and $p_1 = 0.5$, and the maximum number of developmental time steps is set to $T_D = 100$.

B. Results

The result of a typical evolutionary run is presented in Fig. 4, where the fitness of the best individual, the average fitness and the fitness of the best individuals' ancestors are plotted. It can be seen from the figure that the population stagnates from time to time, before an innovation is found, which leads to a significant fitness increase. A much wider plateau has also been observed in some of the runs. Note that the goal of this paper is not to show how well the target shape can be realized. Instead, our model serves to analyse the emergence of feedback during evolution. Successful individuals exhibit the non-trivial behavior to grow towards a stable state during their development. This means that their shape and final state of the GRN remain constant after a certain developmental time step. Thus, the fitness of an individual is not coupled to a certain 'evaluation time step' as it is usually the case in simulated evolutionary development, but rather to the stable individual that is reached after the developmental process has converged.

The curve in Fig. 5 shows the emergence of feedback during the evolutionary run. Since the analysis is computationally expensive, we chose to test the 11 best individuals of each tenth generation for feedback. The curve shows clearly, that negative feedback starts to prevail between the 40th and 60th generation. After generation 60, all 11 best

individuals contain feedback loops. We are able to track the first occurrence of feedback back to the best individual of generation 44, whose dynamic GRN is depicted in Fig. 3. The negative feedback is visible in Fig. 3iii): an excitatory connection from the highlighted gene on the left side to the highlighted gene on the right side, and an inhibitory connection in the opposite direction.

We assume that the negative feedback stabilizes the development of individuals against mutation. With a negative feedback loop a TF is possibly self limiting. If the concentration increases beyond a defined threshold, the TF can decrease its own production. If the concentration decreases, the level of self influence is reduced resulting in a stable state. In comparison, a positive feedback loop could only cause a TF to increase its own production continuously without reaching a stable state. In general, a negative feedback loop in control engineering is a comparison between reference values and output values. Based on the difference the controller designed for this problem can minimize the deviation between reference and output values. Therefore, the system is stabilized towards that target value. In our case, the stabilizing effect is similar except that we do not pre-define a reference value. Instead such a value is system inherent. Therefore, a mutation may cause reference values to change, but as long as the negative feedback loop is not destroyed, a system can maintain its ability to stabilize. One possible effect is that offsprings of individuals with negative feedback will be less sensitive to mutations, i.e., fewer lethal mutations will occur. Here "lethal" means that individuals will not grow at all or

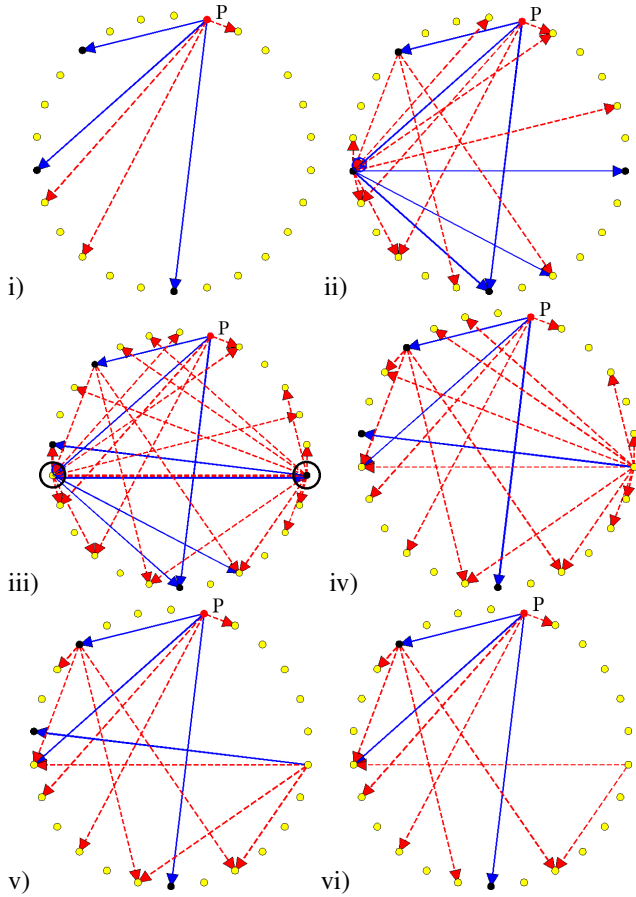


Fig. 3. A time series of interactions inside the dynamic GRN. Each gene is depicted as a small circle. The red point denotes the pre-diffused TF. Active genes are marked as filled circles. The interactions between the genes are either inhibitory (red, dashed arrows) or excitatory (blue, solid arrows). In iii) we highlighted two genes that form a negative feedback loop with an excitatory interaction from left to right and an inhibitory interaction in the opposite direction. Each figure represents the state of the GRN in one timestep. Note that the static condition for this individual is not yet reached after timestep vi).

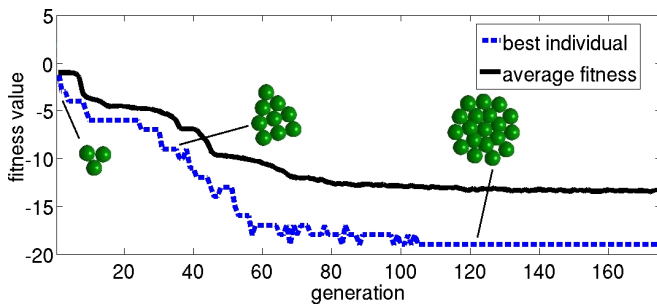


Fig. 4. The best (dashed line) and average (solid line) fitness of a typical evolutionary run. The shape of the best individual after convergence to a stable state is shown for three different generations. The average fitness is computed from those individuals only, that do not violate constraints.

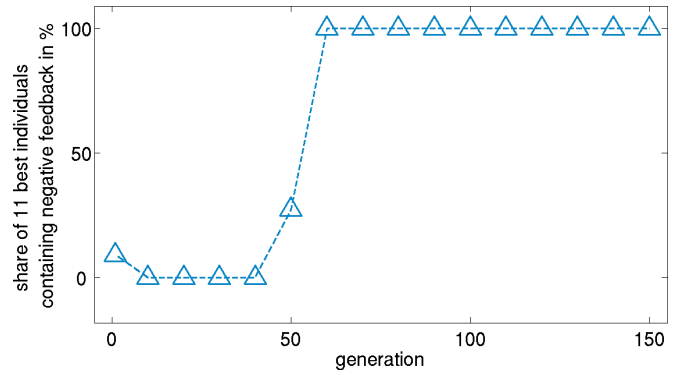


Fig. 5. The triangles mark the share of the 11 best individuals which possess one or more negative feedback-loops. The analysis is performed at every 10th generation.

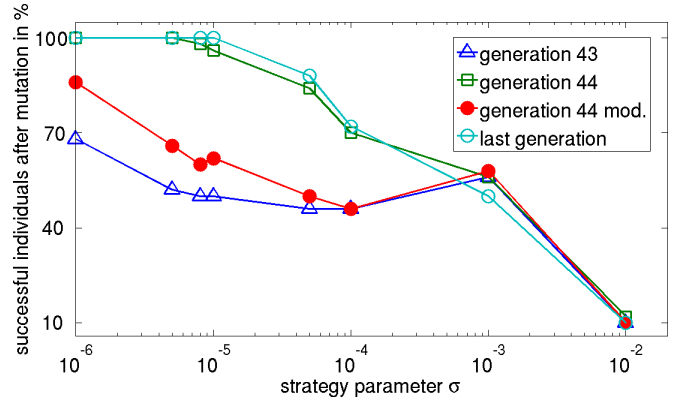


Fig. 6. The results of the mutation experiment: four individuals are mutated 50 times for each strategy parameter σ . The plot shows the percentage of individuals that survived mutation.

will not reach a stable state after the maximum number of allowed developmental time steps. In both cases, individuals are penalized, and not taken into account for further selection. Thus, the number of feasible offsprings from an ancestor containing negative feedback loops is higher than the number of feasible offsprings from an ancestor without negative feedback. If the fitness of individuals containing feedback is not worse than the fitness of those without feedback, the probability that a genome with feedback is passed on during evolution increases.

To verify this hypothesis, we perform a simple mutation experiment with four different individuals: The best individual from generation 44 which uses feedback, its direct ancestor from generation 43 which has no feedback (see the static GRN in Fig. 2), the best individual at the end of the evolutionary run and a modified version of the best individual from generation 44. The modification consists of removing the gene from the vDNA that causes the negative feedback (marked in Fig. 3iii), right circle). Note that these individuals still exhibit a stable, finite growth process, thus none of them violate the constraints. The four individuals are mutated 50 times each, for every sample point. Mutation is carried out by adding a random number generated from a zero-mean normal distribution with given standard deviation σ to each

value of the vDNA. Thereafter, we count the number of individuals that still produce stable growth without violating the constraints and denote them as successful. Note that feasible individuals with lower fitness than the unmodified ones, are also among them. Fig. 6 shows the results of this experiment.

It is clearly visible that mutations with σ smaller than 10^{-5} affect individuals without feedback much more severely than individuals containing feedback: 100% and 96% respectively of the individuals containing feedback survive, while only 62% and 50% respectively survive without feedback. At $\sigma = 10^{-4}$, feedback is still an advantage, although the percentage of successful individuals has been reduced significantly to 70%. The percentage of lethal mutations with a σ larger than 10^{-3} is similar for all individuals. This might be the result of mutation destroying the negative feedback loop, thus destroying the whole control mechanism that mainly set the different individuals apart. Note that during evolution, σ was in the range between 10^{-6} to 10^{-5} from generation 43 onwards and therefore, in a region where feedback seems to be a clearly advantageous.

VI. CONCLUSION

Using a simple model for the evolution of the developmental process of a multi-cellular individual with gene regulatory networks, we analysed the dynamics of the emerging networks with a particular emphasis on the occurrence and role of feedback in the GRN. In our model, the developmental process is not stopped after a pre-defined number of time steps, instead it must converge before a maximum number of time steps has been reached. This is rather uncommon compared to most approaches published so far, however, we believe it is an important step if we want to devise models that are useful for both the support of biological research and the structure optimization of engineering systems.

The results of our analysis indicate that negative feedback seems to prevail during evolution by having a stabilization effect against mutation. Individuals with negative feedback show a greater robustness in our mutation experiment than those without feedback. Thus, evolutionary success is a combination of fitness and robustness against mutations.

Furthermore, we have seen that static GRNs in computational models may contain interactions that are never used. Even in biology, when analysing GRNs, there are significant structural differences between static network interactions and dynamic interactions, see e.g. [12].

The emergence of negative feedback and its persistence during evolution supports the assumption by Wagner [21], that this kind of robustness is an evolved response to stabilizing selection.

Gathering knowledge about the relation between the evolution of feedback and robustness is also important for evolutionary design, because it could tell us something about the evolvability of the system. Indeed feedback might prove to be an important tool for stabilizing certain useful processes during evolution, while deliberately avoiding feedback

mechanisms might enable the evolutionary process to change features easily.

In general, computational models enable us to choose the level of abstraction for analysis that seems most suitable. In our case, for the analysis of feedback, this level is ideally chosen in between the analysis of the static GRN and the analysis of single gene activity functions. Thus, the appropriate level seems to be the analysis of the dynamics on GRNs. Compared to biologic research, this level is easy to achieve with a computational model, since all necessary parameters are accessible and all interactions can be evaluated at every time step.

We think that it is important to realize, how much information is accessible and unused in present models of evolutionary development. More analysis tools and means of visualization are necessary to acquire a more thorough understanding of processes in order to devise simulation systems that prove truly useful for both biological research and structure optimization.

REFERENCES

- [1] P. Bentley and S. Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 35–43, Orlando, Florida, USA, 1999. Morgan Kaufmann.
- [2] C. P. Bowers. Simulating evolution with a computational model of embryogeny: Obtaining robustness from evolved individuals. In *Proceedings of the 8th European Conference on Artificial Life*, pages 149–158, 2005.
- [3] C.P. Bowers. Formation of modules in a computational model of embryogeny. In *Proceedings of the 2005 Congress on Evolutionary Computation*, 2005.
- [4] H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. Comp. Biol.*, 9(1):67–103, 2002.
- [5] P. Eggenberger. Evolving morphologies of simulated 3d organisms based on differential gene expression. In *Proceedings of the 4th European Conference on Artificial Life*, 1997.
- [6] H. El-Samad, H. Kurata, J.C. Doyle, C. A. Gross, and M. Khammash. Surviving heat shock: Control strategies for robustness and performance. *PNAS*, 102(8):2736–2741, February 2005.
- [7] D. Federici. Increasing evolvability for developmental programs. In J. Miller, editor, *GECCO Workshop on Regeneration and Learning in Developmental Systems*, 2004.
- [8] D. Federici. Using embryonic stages to increase the evolvability of development. In J. Miller, editor, *proceeding of the Workshop on Regeneration and Learning in Developmental Systems, WORLDS 2004*, 2004.
- [9] D. Federici and T. Ziemke. Why are developing organisms also fault-tolerant. In *Proceedings of the 9th International Conference on Simulation of Adaptive Behavior*, pages 449–459, 2006.
- [10] N. Geard and J. Wiles. Investigating ontogenetic space with developmental cell lineages. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 56–62, 2006.
- [11] H. Kitano. Biological robustness. *Nature Reviews Genetics*, 5:826–837, 2004.
- [12] N. M. Luscombe, M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann, and M. Gerstein. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, 431(7006):308–312, 2004.
- [13] J. Miller. Evolving developmental programs for adaptation, morphogenesis and self-repair. In *Proceedings of the European Congress of Artificial Life, ECAL*, pages 256–265, 2003.
- [14] E. Palsson and H. G. Othmer. A model for individual and collective cell movement in dictyostelium discoideum. *PNAS*, 97(19):10448–10453, 2000.

- [15] T. Rudge and N. Geard. Evolving gene regulatory networks for cellular morphogenesis. *Recent Advances in Artificial Life*, pages 239–252, 2005.
- [16] G. Schlosser and G. P. Wagner, editors. *Modularity in Development and Evolution*. The University of Chicago Press, 2004.
- [17] H.-P. Schwefel. *Evolution and Optimum Search*. John Wiley, 1994.
- [18] B. Sendhoff and M. Kreutz. Variable encoding of modular neural networks for time series prediction. In V.W. Porto, editor, *Congress on Evolutionary Computation CEC*, pages 259–266. IEEE Press, 1999.
- [19] T. Steiner, M. Olhofer, and B. Sendhoff. Towards shape and structure optimization with evolutionary development. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, pages 70–76, 2006.
- [20] G. Tufte. Gene regulation mechanisms introduced in the evaluation criteria for a hardware cellular development system. In *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems*, 2006.
- [21] A. Wagner. Robustness against mutations in genetic networks of yeast. *Nature Genetics*, 24:355–361, 2000.
- [22] T. M. Yi, Y. Huang, M. I. Simon, and J. Doyle. Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *PNAS*, 97(9):4649–4653, April 2000.