Genome Parameters as Information to Forecast Emergent Developmental Behaviors

Stefano Nichele and Gunnar Tufte

Norwegian University of Science and Technology,
Department of Computer and Information Science,
Sem Selandsvei 7-9, 7491, Trondheim, Norway

{nichele, gunnart}@idi.ntnu.no

Abstract. In this paper we measure genomic properties in EvoDevo systems, to predict emergent phenotypic characteristic of artificial organisms. We describe and compare three parameters calculated out of the composition of the genome, to forecast the emergent behavior and structural properties of the developed organisms. The parameters are each calculated by including different genomic information. The genotypic information explored are: purely regulatory output, regulatory input and relative output considered independently and an overall parameter calculated out of genetic dependency properties. The goal of this work is to gain more knowledge on the relation between genotypes and the behavior of emergent phenotypes. Such knowledge will give information on genetic composition in relation to artificial developmental organisms, providing guidelines for construction of EvoDevo systems. A minimalistic developmental system based on Cellular Automata is chosen in the experimental work.

Keywords: Development, Cellular Computation, Emergence, Evolution, Parameterization of Rule Spaces.

1 Introduction

Artificial developmental systems are systems that can be used to grow artificial organisms, exploiting an indirect genotype to phenotype mapping [9]. Indirect mapping between genotype and phenotype enables two organisms with identical genes to develop to diverting phenotypes, caused by factors influencing the development process, e.g. interactions with the environment. Several artificial developmental systems take inspiration from cellular models [12, 19, 24, 27], where the key element is a cell. The way a cell behaves can be represented by gene regulation that encapsulates the rules and actions that a cell may perform, e.g. growth, differentiation, death. The result of such architecture is a system that can show different developmental behaviors from a single cell (zygote) to a multi-cellular organism.

Even if an artificial developmental process itself can be regulated by very simple mechanisms at cellular level, the whole emergent behavior of the system can show complex phenotypes with properties of stability or unpredictable self-reorganization.

Evolutionary Developmental (EvoDevo) systems [15] have been used in a wide variety of experiments with promising results (e.g. to reach a phenotypic target property [24], to execute a computational property emerging from the development of a machine structure [27] or to develop modular structures [9]), but at the theoretical level the amount of knowledge is limited. This work is focused on the underlying properties of EvoDevo systems and thus does not aim to develop specific organisms with specific properties. Rather, we want to gain more knowledge on the dynamics of developing organisms in relation with the information and representation of the genome and gene regulation.

In this work, the genotypes are represented as a transition rule table, where developmental actions are defined as function of the neighborhood configuration. In this way, it is possible to analyze the different developmental actions and calculate parameters obtained from the genome table. We investigate three different genome parameters. The first one takes inspiration from earlier work of Langton [21], using a developmental λ which is based only on the output of the genomic developmental table. The second parameter is the Majority parameter, a measure of regulatory input and relative output considered independently. This approach is similar to Neighborhood Dominance parameter described in [4]. The third is the Sensitivity parameter, an overall measure of the developmental table where table entries are dependent one to another, defined by Binder [1, 2]. Every parameter measures a different feature of the genome information and thus should be able to describe different phenotypic behaviors.

The article is laid out as follows: background information and motivation for the work is given in Section 2. In Section 3 the developmental model used in the experiments is described. Section 4 presents the used parameters in details. Section 5 shows the results of the experiments. The discussion of the results is given in Section 6 together with the conclusions.

2 Background and Motivation

Artificial developmental systems fall within the field of complex systems. In complex systems, the focus is on the global behavior rather than on the local behavior of the single parts from which the system is built. The interwoven interaction of the system components, without the action of a global controller, places such systems in the field of emergent computation [13].

A CA can be considered as a developing organism, where the genome specifications and the gene regulation information control the cells' growth and differentiation. The behavior of the CA is represented by the emergent phenotype, which is subject to size and shape modifications, according to the cellular changes along the developmental process. Such dynamic developmental systems can show adaptation, self-modification, plasticity [28] or self-replication [20] properties.

The works of Wolfram [29] and Langton [21] on the computation of cellular machines laid a foundation for further research on the possibility of "measuring" properties of the computation [26] and develop a better understanding of the emergent be-

havior of complex systems. The main idea is to find relations between properties of the genotype and the emergent phenotypes targeting specific characteristics, i.e. number of states in the transient length, organism growth speed, etc., in order to get an extended and detailed explanation of the underlying properties of developmental systems.

Several genome parameters have been previously proposed in order to measure genotype properties. Langton [21] studied a parameter λ as a measure of the activity level (the outcome) of the system. A similar parameter, neighborhood dependent, is Absolute Activity presented by De Oliveira [4]. Li [5] introduced Mean Field Parameters to monitor if the majority of the regulatory actions follow the "mean" configuration. De Oliveira [4] presented a very similar parameter called Neighborhood Dominance. Binder [1, 2] introduced the Sensitivity parameter which measures the number of changes in the output of the transition table based on a change in the neighborhood, one cell at a time, over all the possible neighborhoods of the rule being considered. This has also being studied by De Oliveira [3, 4] under the name of Context Dependence. However, all the proposed parameters are focused on helping to solve a specific computational task, e.g. synchronization task [25], rather than to exploit and understand developmental properties or to guide evolution.

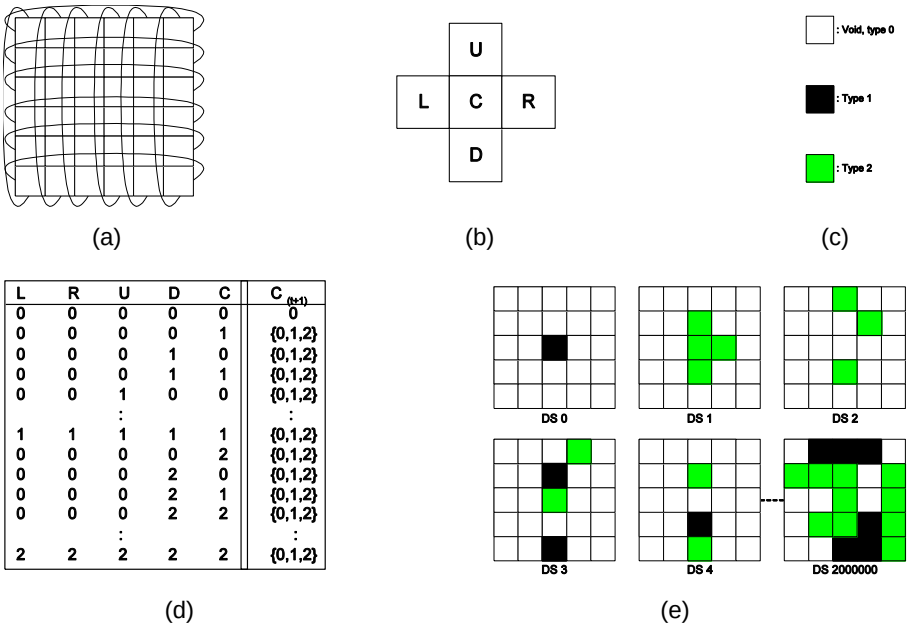
3 Developmental Model

The developmental model used in this work is a minimalistic cellular developmental model based on cellular automata, similar to cellular models used in [18, 24, 27]. The model is based on a two dimensional cellular automata with cyclic boundary conditions, as shown in Figure 1(a). The number of cell types is set to three instead of two in order to keep the property of multicellularity. In Figure 1(c) the three cell types are shown: two cells (type 1 and 2) plus the empty or dead cell (type 0). A single cell, placed in the centre of the development grid, develops according to a developmental table based on Von Neumann's neighborhood (five neighbors), as represented in Figure 1(b). All the possible regulatory input combinations are explicitly represented in a development table, i.e. 243 (3^5) neighborhood configurations. To ensure that cells will not materialize where there are no other cells around, a restriction has been set: if all the neighbors of an empty cell are empty, the cell will be empty also in the following development step. This is shown in Figure 1(d), where the first entry represents the growth restriction. A more detailed description of the development model is given in [6]. Figure 1(e) shows an example of a developing organism. In Development Step 0 there is only a single cell of type 1 (zygote). In DS 1 the cell has divided and differentiated. The following DSs from 2 to 4 show changes in phenotypic structure until the last DS 2 000 000 is reached.

Having all the 3^5 input combinations fully specified together with their respective cellular actions, makes it possible to use the genome information to calculate parameters (based only on the developmental table) that may describe different behaviors of the developing organisms or some specific characteristics of the developmental path, i.e. trajectory length (number of development steps until a state is reached for the

second time and an attractor is found), attractor length, growth speed (number of cells that become alive during the transient phase), change rate (number of cells that differentiate from development step to development step along the attractor).

Other more detailed non-minimalistic models [24, 27] make it very hard to specify all the possible regulatory combinations [6].



L	R	U	D	C	$C_{(t+1)}$
0	0	0	0	0	0
0	0	0	0	1	{0,1,2}
0	0	0	1	0	{0,1,2}
0	0	0	1	1	{0,1,2}
0	0	1	0	0	{0,1,2}
		:			:
1	1	1	1	1	{0,1,2}
0	0	0	0	2	{0,1,2}
0	0	0	2	0	{0,1,2}
0	0	0	2	1	{0,1,2}
0	0	0	2	2	{0,1,2}
		:			:
2	2	2	2	2	{0,1,2}

Fig. 1. Minimalistic cellular developmental model. (a): 2D grid world where the organism develops; (b): cellular neighborhood; (c): cell types; (d): genetic information, developmental table with regulatory input and cellular actions; (e): example of developing organism.

4 Genome Parameters

Parameters obtained from the genome information can be used to estimate the dynamic behavior of the system. Given a well-defined computational problem, e.g. synchronization task [25], it is possible to search for possible solutions/genotypes that are able to develop the target phenotype. A search algorithm, e.g. a genetic algorithm, may benefit from the usage of a parameter that guides the search in favorable areas of the search space. However, for a developmental approach, it may be better to have simple independent parameters, where each parameter indicate specific developmental behavior, e.g. long transient length or short attractor, and eventually combine several parameters together to "compose" the desired target behavior. The set of all the developmental characteristics may be seen as a multidimensional space, where every independent parameter represents a degree of freedom and allows moving on a specific axis. Figure 2 shows this idealized version of the genotype hyperspace. The differ-

ent parameters help to reach and keep sought phenotypic properties. Attractor length indicates stable or changing phenotype structure, growth speed differentiates fast and slow growing organisms and change rate indicates the ability for state change.

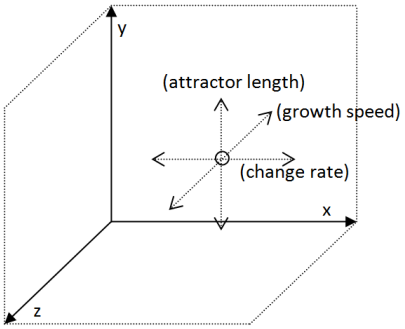


Fig. 2. Multidimensional representation of the genotype space, where each parameter may represent a different dimension.

Having defined the developmental genome in a transition table, as shown in Figure 1(d), makes it possible to simulate the development process of artificial organisms with cellular automata and relate their emergent behavior with genome parameters. The explored parameters are described in the following sections.

4.1 Lambda Parameter

Langton [21] tried to find a relation between CA behavior and a parameter λ. He observed that the basic functions required for computation (transmission, storage and modification of information) are more likely to be achieved in the vicinity of phase transitions between ordered and disordered dynamics (edge of chaos). He hypothesized that it is easier to find rules capable of complex computation in a region where the value of λ is critical. Since the developmental model is composed by 3^5 regulatory combinations, all the possible regulatory inputs and relative outputs (growth, differentiation or no action) are fully specified in the developmental table. In order to calculate λ, it is necessary to define a quiescent state, the void cell (type 0) in our case. Lambda is defined as follows:

$$\lambda = \frac{K^N - n}{K^N} \tag{1}$$

λ can be calculated according to Equation 1, where n represents the number of transitions to the quiescent state, K is the number of cells types (three in our case) and N is the neighborhood size (five in the Von Neumann neighborhood). In this way, the value of λ is based only on local properties of the neighborhood and in particular the cellular actions that are present in every cell. A restriction has been set in the transition table to prevent growth of cells surrounded by empty cells: if all the neighbors of an empty cell are empty, the cell will be empty in the next development step.

4.2 Majority Parameter

Li [5] studied Mean Field Parameters on one-dimensional cellular automata with two states, starting from random initial configurations. The goal was to capture if the cellular development was following the "mean value" of the other cells in the neighborhood. A generalization of those parameters could be what we call Majority parameter, i.e. how many neighborhood configurations in the rule table follow the majority state to determine the next state. This approach could be related to a structured development of multi-cellular organisms. Moreover, it can be calculated regardless of the number of cell types and neighborhood configurations. Majority parameter is a sum, over all the neighborhood configurations in the developmental table, of the number of cellular actions that are affected by the most present cell in the neighborhood. This is shown in Equation 2.

$$M = \sum_{(V1V2 \dots Vm)} [V(m+1) = maj(V1V2 \dots Vm)] \quad (2)$$

where m is number of cells in the neighborhood and V(m+1) is the value of the cell being considered, at the next time step. The function maj() retrieves the most present cell type (or the set of most present cell types) in the neighborhood. M is the count, over all possible neighborhoods, of the number of cellular actions in the developmental table, following the most present state among the neighbors. The parameter is normalized between 0 and 1, where the value 0 means that none of the cellular actions in the developmental table follow the most present state in the neighborhood. 1 represents a situation where the overall behavior is following the majority of the cells present in the neighborhood. Majority parameter is based on both neighborhood configuration and the relative cellular action, analyzed one by one, and measures the amount of change in respect to the neighborhood.

4.3 Sensitivity Parameter

Sensitivity parameter, introduced by Binder [1, 2] as μ, is a measure of the neighborhood and the output state as an overall (not entry by entry as for λ and M). μ captures the "dependency" of a single entry in the developmental table together with all the other entries with a similar neighborhood configuration. In details, it measures the number of changes in the output of the transition table based on a change in the neighborhood, one cell at a time, over all the possible neighborhoods of the rule being considered. The Sensitivity parameter is easy to calculate. However, it is much harder to generate a specific developmental table with a specific parameter value (in the following chapter we describe a Genetic Algorithm that is used for this purpose). μ is described only for one-dimensional cellular automata with two states. We generalized the formula to consider CAs that are representing the development of multi-cellular organisms with three cell types with five neighbors. This is described in Equation 3.

$$\mu = \frac{1}{n * m * (\mathrm{K}-1)} \sum_{(V1V2 \dots Vm)} \sum_{q=1}^{m} \frac{\delta \varphi}{\delta Vq} \quad (3)$$

In Equation 3, m is the number of cells in the neighborhood. n is the number of

possible neighborhood configurations (V1V2…Vm = 3^5 = 243). K is the number of cell types. The denominator is multiplied by K-1 because, if a specific cell state is being considered, there are K-1 other possible cell values to be checked. Equation 4 shows the core calculation, where every neighborhood configuration is compared with all the other neighborhood configurations with a single different value.

$$\frac{\delta\varphi}{\delta Vq} = \begin{cases} 1 \; if \; \varphi(V1 \dots Vq \dots Vm) \neq \varphi(V1 \dots \overline{Vq} \dots Vm) \\ 0 \; otherwise \end{cases} \tag{4}$$

The value of the derivate δφ / δVq is 1 if φ(V1 ... Vq ... Vm) ≠ φ(V1 ... $\overline{\text{Vq}}$ ... Vm). This happens when the value of the cell at the next time step is sensitive to the value of the neighbor in position q. The value of the derivate is 0 otherwise.

5 Experiments

In the experiments herein, the main idea is to generate genomes with a given property (a specific parameter value). In order to compare results for different parameters, the parameter intervals are normalized between 0 and 1. 1000 developmental tables are generated for each parameter value (from 0 to 1) with granularity 0.01. Each generated genotype is developed until a state is repeated twice and an attractor is found. The size of the CA grid is set to 4x4 cells.

For every parameter value, several measurements are performed and compared, i.e. attractor length, trajectory length, growth rate and change rate. Figure 3 summarizes the described experimental setup. Measurements are described in the last section of this chapter.

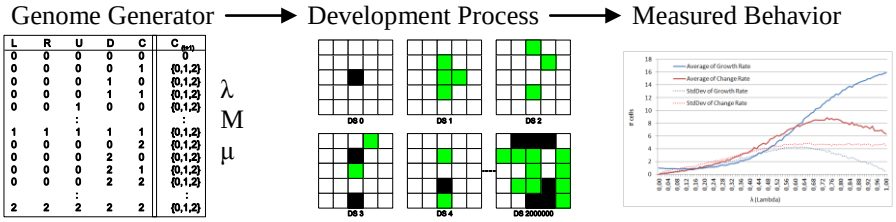


Fig. 3. Experimental setup: first genomes are generated according to a parameter, then artificial organisms are developed and finally phenotypic behaviors are measured.

5.1 Genomes generation with *λ* parameter

In the first experiment, genomes are generated with predefined values of λ. Test genomes were generated in a similar method to Langton's [21] random table method. For every entry in the development table, with probability (1- λ) the cell type at the next developmental step is quiescent (type 0). With probability (λ), the cell type at the next developmental step is generated by a uniform random distribution among the other cell types (type 1 or 2).

5.2 Genomes generation with *M* parameter

In the second experiment, the λ of the first experiment is substituted with M parameter generated genome. For this purpose, for each table entry, the procedure is:

- If there are more than 3 occurrences of a cell type, with probability M the cell type at the next developmental step follows the most present cell type in the neighborhood. With probability 1-M the cell type at the next developmental step is generated by a uniform random distribution among the other two cell types (the minority in the neighborhood);
- If there are 2 cell types with occurrence 2, with probability M/2 one of those 2 cell types is chosen. Otherwise, with probability 1–M the cell type at the next developmental step has the same type as the less present cell type in the neighborhood.

5.3 Genomes generation with μ parameter

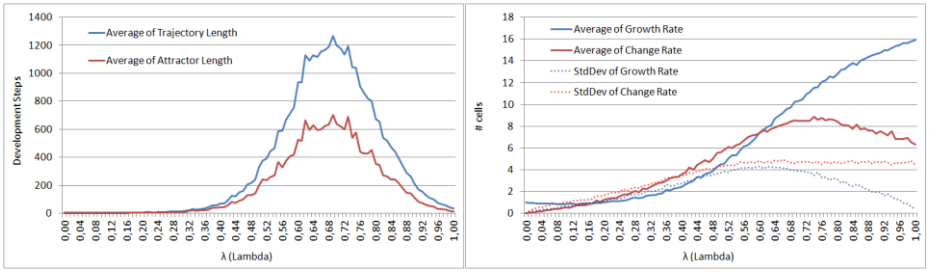
In the third set of experiments, genomes have to be generated with specific μ properties. Even if μ is easily computable for a specific development table, it is much harder to generate a development table with a target μ value. In order to generate 1000 samples for each value, a Genetic Algorithm has been implemented. The GA's fitness function is set to generate development table with target μ values in the sought range. It is important to highlight that the goal of this experiment is not to achieve good GA performances, whether to be able to generate the desired target genotypes.

5.4 Measurements

Having identified three parameters as an evaluation of the genetic information, measurements of the developmental organism have to be defined in order to find possible correlations between genotypes and emerging phenotypes. Such phenotypic measures should provide information regarding the development process as a whole and the phenotypic changes that occur during each development stage. Thus, it may be possible to differentiate distinct dynamic behavior of the developing organisms.

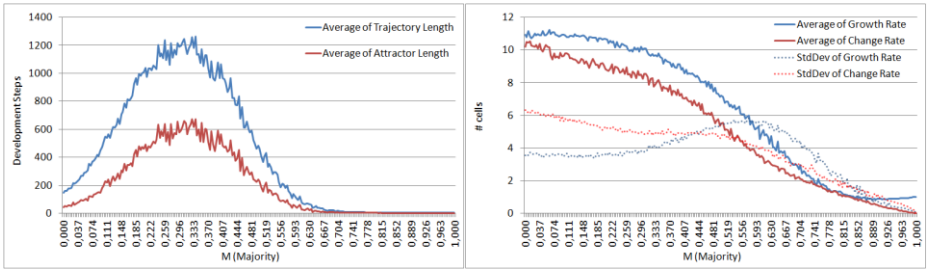
For a given organism, a trajectory starts from an initial cell (zygote) and follows the developmental path. Each state includes information on morphology, size, behavior etc. The trajectory describing the developmental path can end up in a final stable organism; a point attractor or as a self-reorganising organism; a cyclic attractor.
It may be argued that a stable final structure is important [24], i.e. development reaches a structure (or state) that is stable by self-regulation. On the other hand, it may be argued that a dynamic phenotypic structure with self-reorganizing possibilities may be an important part for computation and/or adaptation for developmental machines [27].

As such, the developmental trajectory with its transient part and attractor can represent a possible quantifiable measurement of the development of an artificial organism. Applying trajectory information to quantify developmental properties gives information regarding stability of the organism, does development create a stable



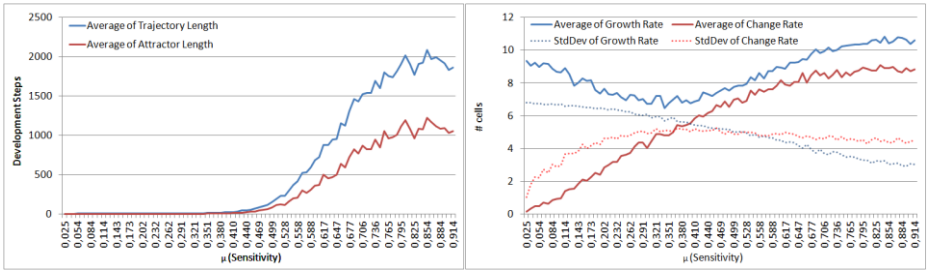
(a) Average trajectory and attractor length (b) Average growth and change rate

Fig. 4. Measurements in correlation to λ. Average over 1000 tests for each λ value.



(a) Average trajectory and attractor length (b) Average growth and change rate

Fig. 5. Measurements in correlation to M. Average over 1000 tests for each M value.



(a) Average trajectory and attractor length (b) Average growth and change rate

Fig. 6. Measurements in correlation to μ. Average over 1000 tests for each μ value.

organism or does the organism end with a structure that change form in a cyclic manner. Both alternatives provide interesting knowledge that would be favorable if it can be predicted already at the design point of developmental models, genome representation and/or genetic operators.

Another possibility is to investigate internal qualities of the developmental processes, i.e. growth, cell death and differentiation, and thereby define measures of different developmental phases. Two phases of interest are considered. First, a growth phase where the organism expand in size toward an ”adult” form and second, change

in the adult organism. Growth increases the number of cells ”alive” and differentiation changes the cell type. Growth is here defined (not exactly biological correct) as the transient phase of a trajectory and the chosen growth measurement is the size of the organism at the end of the transient phase, i.e. all cells of type non-void. Change rate is defined as the average number of cells that change cell type from development step to development step along the attractor. It can then be seen as a measurement of the adult life of the organism.

In order to have a complete overview of the different emergent behaviors, we use four measurements: trajectory and attractor length that may indicate information about structural and adaptive properties of the organism, growth and change rate that may give information on the activity of the developmental processes. The measurements used herein are close to complexity measures of phenotypic properties [17]. Kolmogovor inspired complexity measurements [22] are also related.

5.5 Results

In the experiments herein, the array size was set to 4x4. The size of the arrays was chosen as to be able to carry out experiments in reasonable computational time. Organisms of 4x4 cells may be considered rather small, however, the theoretical maximum attractor length is 3^{16}. As such, even at the chosen array size, the variation in trajectory and attractor length can have a huge deviation.

The average trajectory length and average attractor length were recorded and plotted as a function of the parameters, λ in Figure 4(a), M in Figure 5(a) and μ in Figure 6(a). The same was done for average growth and average change rate and results are presented in Figure 4(b), 5(b) and 6(b).

6 Conclusion

The presented experiments show that each of the used parameters have a specific ability to measure properties of the genome composition as an indication of how the resulting organism will develop. The plot in Figure 4(a), show common results with Langton’s work on λ, i.e. sudden increase in the length of trajectories, attractors and transient phase of a developing organism. However, Langton’s work was focused on potential computational properties of the system related to phase transitions. This result is encouraging as it indicates that the observed correlation between λ and the state space properties is not a special case related to the development model or a given size constraint. This is further emphasized by Langton’s work [21], where neighborhood configuration and cell types were expanded. Besides, the shown parameter correlates with [6] where different grid sizes were tested on the developmental model. The comparison of the plots in Figure 4(a), 5(a) and 6(a) show that the length of the trajectories depend strongly on the parameters value. As such, the result show that a calculation based on the genome composition can reflect a predictable developmental behavior. Such knowledge of probable path properties, e.g. length, may help evolution if there exist knowledge of what developmental path length is likely to be needed to reach a phenotype with certain structural properties. λ and M have the same power

to exploit trajectory and attractor length, whether μ is able to exploit longer paths, on average. An interpretation of such result is that μ can be used to guide towards part of the search space where genomes with long attractors are more likely to be found.

The results in Figure 4(b), 5(b) and 6(b), further emphases a relation between the measurements of genomic composition and developmental behavior. In Figure 4(b) the growth rate shows that for low values of λ the transient phase of the developmental path is rather short. Genomes with this property have a rather high probability of short developmental time with a point or short attractor. This knowledge is useful if a requirement is to develop stable organisms. Figure 5(b) shows that parameter M has the same ability as λ to represent growth rate of developing organisms but could be better suited to discover organism with higher number of structure and shape changes, especially for low M values. Figure 6(b) show that the usage of μ as measurement of genomic composition could accentuate the possibility to find organisms that develop at a higher rate with low changes in their attractor, thus being quite stable with few changes in form. Again, such knowledge could be helpful at the system design stage if information on the desired target phenotype is known.

Comparing the results in Figure 4, 5 and 6, it is possible to observe that if two or more parameters would be used together, it could be possible to compose developmental behaviors that are not achievable with a single parameter, e.g. low λ value to achieve short attractor lengths and low μ value to obtain higher growth rate. In terms of evolvability, it could be possible to add several parameters in the fitness function. However, evolving behaviors instead of looking at the effect would require a different experimental approach than the one used herein.

Moreover, when it comes to adaptivity and evolution, the results herein show that genomes with a given parameter value will most likely mutate to genomes with similar developmental behavior as long as the mutation result in an offspring with similar parameter value. Parameters could then be used to guide evolution towards favorable areas of the solution space and then remain in that area without jumping in a complete different region of the hyperspace where desired developmental behaviors are less likely to be found.

References

1. Binder, P. M.: A Phase Diagram for Elementary Cellular Automata. Complex Systems, 7, 241-247 (1993)
2. Binder, P.M.: Parametric Ordering of Complex Systems. Physical Review E, vol. 49 n. 3, pp. 2023-2025 (1994)
3. De Oliveira, G., De Oliveira, P., Omar, N.: Guidelines for Dynamics-based Parametrization of One-Dimensional Cellular Automata Rule Space. John Wiley & Sons, Inc. Vol. 6, No. 2 Complexity (2001)
4. De Oliveira, G., De Oliveira, P., Omar, N.: Definition and Application of a Five-Parameter Characterization of One-Dimensional Cellular Automata Rule Space. MIT, Artificial Life7: pp. 277-301 (2001)
5. Li, W.: Phenomenology of Nonlocal Cellular Automata. Santa Fe Institute. Journal of Statistical Physics, 68(5-6): 829-882 (1992)
6. Tufte, G., Nichele, S.: On the correlations between developmental diversity and genomic composition. GECCO 2011, ACM, pp. 1507-1514 (2011)

7. Nichele, S., Tufte, G.: Trajectories and Attractors as Specification for the Evolution of Behavior in Cellular Automata. IEEE CEC 2010 pp. 4441-4448 (2010)
8. Beer, R.D.: A dynamical systems perspective on agent-environment interaction. Artificial Intelligence, 1–2(72):173–215 (1995)
9. Bentley, P.J., Kumar, S.: Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. GECCO 1999, pages 35–43 (1999)
10. Cussat-Blanc, S., Luga, H., Duthen, Y.: From single cell to simple creature morphology and metabolism. In S. Bullock, J. Noble, R.Watson, and M. A. Bedau, editors, Artificial Life XI, pages 134–141. MIT Press, Cambridge, MA (2008)
11. Eggenberger, P.: Evolving morphologies of simulated 3d organisms based on differential gene expression. In 4th Artificial Life conference, pages 205–213. MIT press (1997)
12. Fleischer, K., Barr, A.H.: A simulation testbed for the study of multicellular development: The multiple mechanisms of morphogenesis. In 3rd Artificial Life conference, pages 389–416. Addison-Wesley (1993)
13. Forrest, S.: Emergent Computation. MIT Press (1991)
14. Gordon, T.G.W.: Exploring models of development for evolutionary circuit design. IEEE CEC 2003, pages 2050–2057. IEEE (2003)
15. Hall, B.K., Pearson, R.D., M¨uller, G.B.: Environment, development, and Evolution Toward a Synthesis. The Vienna Series in Theoretical Biology. MIT-Press (2004)
16. Kitano, H.: Building complex systems using development process: An engineering approach. In Evolvable Systems: from Biology to Hardware, ICES, LNCS, pages 218–229. Springer (1998)
17. Kowaliw, T.: Measures of complexity for artificial embryogeny. GECCO 2008. ACM (2008)
18. Kowaliw, T., Grogono, P., Kharma, N.: Environment as a spatial constraint on the growth of structural form. GECCO 2007, pages 1037–1044, New York, NY, USA (2007).
19. Kumar, S., Bentley, P.J.: Biologically inspired evolutionary development. In 5th International Conference on Evolvable Systems (ICES03), Lecture Notes in Computer Science, pages 57–68. Springer (2003)
20. Langton, C.G.: Self-reproduction in cellular automata. Physica D, 10:135–144 (1984)
21. Langton, C.G.: Computation at the edge of chaos: phase transitions and emergent computation. In S. Forrest, editor, Emergent Computation, pages 12–37. MIT Press (1991)
22. Lehre, P.K., Haddow, P.C.: Developmental mappings and phenotypic complexity. In Congress on Evolutionary Computation (CEC2003), pages 62–68. IEEE (2003)
23. Miller, J.F.: Evolving a self-repairing, self-regulating, french flag organism. GECCO 2004, Lecture Notes in Computer Science, pages 129–139. Springer (2004)
24. Miller, J.F., Banzhaf, W.: Evolving the program for a cell: from french flag to boolean circuits. In S. Kumar and P. J. Bentley, editors, On Growth, Form and Computers, pages 278–301. Elsevier Limited Oxford UK (2003)
25. Mitchell, M. Hraber, P.T., Crutchfield, J.P.: revisiting the egde of chaos: Evolving cellular automata to perform computations. Complex Systems, 7:89–130. Santa Fe Institute Working Paper 93-03-014 (1993)
26. Packard, N.H.: Dynamic Patterns in Complex Systems, chapter Adaptation Toward the Edge of Chaos, pages 293–301. World Scientific (1988)
27. Tufte, G.: Evolution, development and environment toward adaptation through phenotypic plasticity and exploitation of external information. S. Bullock, J. Noble, R. Watson, and M. A. Bedau, editors, Artificial Life XI, pages 624–631. MIT Press, Cambridge, MA (2008)
28. West-Eberhard, M.J.: Developmental Plasticity and Evolution. Oxford Univ. Press (2003)
29. Wolfram, S: Universality and complexity in CA. Physica D, 10(1-2):1–35 (1984)