Heterochrony and Artificial Embryogeny: A Method for Analyzing Artificial Embryogenies Based on Developmental Dynamics

Artur Matos*,**
Nagoya University

Reiji Suzuki**
Nagoya University

Takaya Arita**
Nagoya University

Keywords
Artificial embryogeny, heterochrony, genetic programming, neutral mutations, developmental trajectories

Abstract Artificial embryogenies are an extension to evolutionary algorithms, in which genotypes specify a process to grow phenotypes. This approach has become rather popular recently, with new kinds of embryogenies being increasingly reported in the literature. Nevertheless, it is still difficult to analyze and compare the available embryogenies, especially if they are based on very different paradigms. We propose a method to analyze embryogenies based on growth dynamics, and how evolution is able to change them (heterochrony). We define several quantitative measures that allow us to establish the variation in growth dynamics that an embryogeny can create, the degree of change in growth dynamics caused by mutations, and the degree to which an embryogeny allows mutations to change the growth of a genotype, but without changing the final phenotype reached. These measures are based on an heterochrony framework, due to Alberch, Gould, Oster, & Wake (1979 Size and shape in ontogeny and phylogeny, *Paleobiology*, *5*(3), 296–317) that is used in real biological organisms. The measures are general enough to be applied to any embryogeny, and can be easily computed from simple experiments. We further illustrate how to compute these measures by applying them to two simple embryogenies. These embryogenies exhibit rather different growth dynamics, and both allow for mutations that changed growth without affecting the final phenotype.

1 Introduction

Artificial embryogeny (AE) is a subfield of evolutionary computation (EC) concerned with the study and implementation of developmental processes, and their integration with evolutionary algorithms (EAs) [28]. In contrast to traditional EAs—where there is basically no distinction between the genotype and phenotype—AE genotypes employ generative representations, that is, the genotypes specify a process to *grow* phenotypes. This approach allows EAs to evolve phenotypes with a much higher degree of complexity than traditional representations [11]. This makes AE research an extremely important pursuit in current evolutionary computation.

* Contact author.
** Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8601, Japan. E-mail: amatos@alife.human.nagoya-u.ac.jp (A.M.); reiji@nagoya-u.jp (R.S.); arita@nagoya-u.jp (T.A.)

AE systems have been already proposed for a wide range of problems and designs. Experimental tests have shown that AEs typically outperform traditional EAs when evolving 2D patterns [3] and artificial creatures [11, 16]. These results make the field look very promising, and new embryogenies are constantly being reported in the literature. Unfortunately, this surge of activity is occurring at the expense of analysis and formalization, which is obviously becoming more necessary. Despite all of these early successes, at this time we still do not have any solid theory or model to explain what makes AEs work. Designing embryogenies is still mainly a work of intuition and of trial and error, instead of a systematic inquiry driven by theory. This is understandable given that AE is still a very young discipline, but eventually we will have to move beyond this experimentation stage, and build a good theoretical framework for the field.

Designing a general framework for AE is a difficult task, especially if we consider the huge diversity of all the existing implementations. Because the embryogenies are implemented in an artificial medium (viz., computers), AE researchers have the freedom to design them as they please. In fact we are not even limited to simulating biological development closely, and some of the proposed embryogenies employ mechanisms very different from natural development. Nevertheless, all of these embryogenies do share some common assumptions; by focusing on these assumptions alone it is possible to devise a theoretical framework that includes all the embryologies. In this article we propose a general analysis framework that does precisely this. Our framework allows us to characterize embryogenies in a quantitative way, regardless of their assumptions, and it can be easily applied by performing simple experiments. We believe that this approach is particularly important when there is a need to compare very different embryogenies.

In AE most of the existing analysis tools can be roughly classified into two groups. Each group works at a different level of organization, and they have very different objectives. In one hand, we, as AE researchers, want to understand the mechanisms that make our embryogenies work, and what makes them behave the way that they do. By mechanism here we mean the overall system used to simulate development—for instance, models of genetic regulatory networks (either discrete or continuous), and Lindenmayer systems. The tools used for this task must work at this level, and therefore are confined to the embryogeny's specific mechanism. A good example is modularity analysis: It is generally agreed that modularity is a required feature for achieving complex designs [3, 11], and some modularity measures have already been proposed [10]. Unfortunately, it is difficult to establish a measure of modularity without taking into account either the mechanism used by the embryogeny or the type of design that is being evolved. These measures may help to understand modularity in the context of a few isolated embryogenies or designs, but they do not allow us to compare degrees of modularity between embryogenies that use very different mechanisms. The framework that we will be presenting here works at an upper level of organization, and does not make any specific assumptions on the mechanisms employed by the embryogenies. Granted, it also does not allow us to explain precisely how they work, so analysis tools, such as the ones referred to above, will always be important. Our framework is meant to be complementary to these tools and not to replace them.

On the other hand, we can analyze embryogenies from an evolutionary point of view, with respect to how they perform when evolved. Studies of this sort may treat the embryogeny as a "black box," and evolve different embryogenies for a set of tasks. The results of the embryogenies can then be empirically compared, for instance, on the number of fitness evaluations required to converge, or on the quality of the generated phenotypes. Studies that employ this approach include the work of Bentley and Kumar [3], Hornby and Pollack [11], and Komosinski and Varga [18]. Bentley and Kumar [3] evolved four different embryogenies for creating 2D tessellating tiles of different sizes. In this case their major point of comparison was how well the embryogenies evolved, and how that was related to problem difficulty. They were able to show that some of the embryogenies simply could not solve the problem as the tile size increased, and how that was related to the mechanism used. Hornby and Pollack [11] and Komosinski and Varga [18] also performed empirical studies, but this time on artificial creatures. Both used embryogenies of increasing complexity on a fixed set of tasks, which included walking in both studies, and jumping in the study by Komosinski and Varga. Both

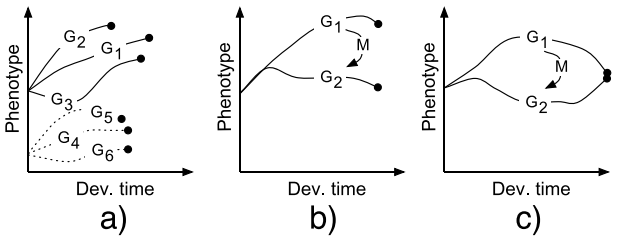

Figure 1. Artificial embryogenies viewed as developmental trajectories. (a) Two embryogenies (displayed as solid and dotted lines) generate different trajectories in the search space. (b) Variation operators, like the mutation operator (depicted with an *M*), change the trajectories, and therefore the final phenotype. (c) An example of a mutation (depicted as an arrow from G_1 to G_2) where the same phenotype is reached, but the trajectory is changed.

studies compared the embryogenies on the average fitness and convergence, and in the case of the study by Hornby and Pollack, a detailed comparison of the behavior of the mutation operator was also included. Our analytical framework is closer to these empirical studies, in that we will mainly be dealing with the phenotypes an embryogeny generates.

In our view, there are mainly two problems with the empirical studies mentioned above. The first one is that they are always performed in an evolutionary context, so they don't characterize an embryogeny per se. AEs are difficult to predict, so we can reasonably expect that some embryogenies will be better at some tasks than others. Therefore, empirical studies performed under evolution may easily dismiss an embryogeny as not good, just because a relevant task was not part of the study. In order to avoid this problem, our approach characterizes embryogenies by studying their behavior when generating or mutating phenotypes at random. From these random experiments we are able to derive a probabilistic model of how the embryogeny behaves. These models can then be further used to group related embryogenies together, and to check how they perform for given tasks.

A second, more subtle problem is that these studies do not consider the actual growth process at all. In an AE, a genotype is not mapped to one phenotype only, but to a temporal sequence of phenotypes. This temporal sequence represents the growth of the genotype, starting with its birth, until the last step in the sequence, its final stage. The previously mentioned studies are exclusively focused on these final stages, because these are the ones that are usually employed for fitness evaluation. Nevertheless, this last stage is the outcome of a complex process that usually (although not always) is dependent on the previous developmental steps.[1] We can reasonably expect that different embryogenies will exhibit different dynamics in these temporal sequences, and that some dynamics may be more adaptive than others. In a previous study [20], we have already shown that this actually happens with the cellular encoding embryogeny, so we think that the study of these dynamics should not be neglected.

In our view, a development process is better understood as a trajectory being run over the search space, where each genotype specifies a trajectory. Different embryogenies will have different biases on their translation process, so they will generate different trajectories in the search space (Figure 1a). Variation operators like mutation and crossover explore the search space by changing development trajectories, a process called *heterochrony*[2] (Figure 1b). Our analysis methodology is grounded on this concept of developmental trajectories, and we will characterize embryogenies by exploring how they generate and manipulate them. In order to do this, we have adopted an existing quantitative

1 An important exception is Hornby and Pollack's Genre embryogeny described in their empirical comparison study. Growth in this embryogeny does not work directly with the phenotypes, but with an intermediate representation; it is this representation that is changed during growth, so the phenotypes in adjacent time steps can be widely different from each other. Our framework probably is not very useful for this kind of embryogenies, but there are only a few that actually work like this.

2 As explained by Klingenberg [14], heterochrony is a rather elusive concept, and different researchers employ different definitions. Notably, heterochrony is defined differently in evolutionary and developmental biology. The definition that we are using here is closer to the evolutionary biology one, as employed by Alberch et al. [1], but not completely equivalent. We will get back to this point in Section 2.

framework from evolutionary biology, Alberch et al.'s framework [1], to our artificial embryogenies context.

Conceptually, our work is similar to a study by Stanley and Miikkulainen, where they already proposed a scale to measure heterochrony in AEs [28]. The main difference between their study and ours is that Stanley and Miikkulainen's work is only a qualitative classification, whereas we will describe ways to actually measure heterochrony quantitatively. We will explore how our framework and Stanley and Miikkulainen's work are related later in this article.

The remainder of this article is organized as follows: Section 2 gives a conceptual overview of our approach. Here we will review Stanley and Miikkulainen's work, introduce Alberch et al.'s framework, and explain the required changes to the framework for using it with AEs. Section 3 builds on Section 2, and presents the framework in a formal way. In Section 4 we will illustrate how to use the framework by applying it to two simple embryogenies. Section 5 concludes the article. In an appendix, we also include a detailed description of the two embryogenies.

2 Defining and Measuring Heterochrony

2.1 Stanley and Miikkulainen's Heterochrony Dimension

In a recent review article [28], Stanley and Miikkulainen presented a broad framework to classify artificial embryogenies. Their goal was not only to classify existing embryogenies, but also to point out further research directions that they believed were not properly explored. With this in mind, they proposed a space of five distinct categories, or *dimensions*, by which the embryogenies could be characterized. One of these dimensions was heterochrony, defined in a way conceptually similar to the definition employed in this article. Therefore it is important that we review their definition first.

According to Stanley and Miikkulainen, heterochrony is the change, over generations, of the timing and ordering of events in an embryogeny. It is well known that natural embryogenies display extreme flexibility in their growth, mostly with respect to the time and ordering of developmental events [9, 26]. For instance, there are well-known accounts of species where whole growth stages were removed by evolution, but which still have kept the adult form of the original species. The authors argue that this flexibility should be important for AEs as well, and they point out two reasons for this. The first one is that flexibility in development may lead to important phenotypic innovations: It is widely believed that some of the major innovations in evolution were caused by simple changes in the underlying growth process. The other, more subtle reason is that heterochrony allows evolution to explore neutral regions of the search space more easily. This can be better understood by going back to the concept of ontogenetic trajectories, depicted in Figure 1. In an AE, mutation allows the EA to explore the search space by changing the growth trajectories of the genotypes. Occasionally, a mutation may occur that radically changes the ontogenetic trajectory, but retains the same phenotype in the last stage of growth (Figure 1c). Usually only the last stage of growth is employed for fitness evaluation, so this new mutated genotype will have the same fitness as the previous one (in other words, it is a neutral mutation). Nevertheless, they are not the same, but as they have the same probability of being selected, the following mutations may generate two radically different phenotypes. This phenomenon, the authors argue, may make the exploration of the search space more efficient. Although the role of neutrality in artificial evolution is still greatly debated [15], there are at least a few studies showing that increased neutrality is beneficial [27, 33], so this can be an important factor in AEs.

Based on these concerns, Stanley and Miikulainen proposed a scale to measure heterochrony in AEs, and classified several of the published embryogenies on this scale. Embryogenies on the extreme left of the scale do not have any mechanisms that allow evolution to change the order or timing of events easily; heterochronic changes in these embryogenies usually require a major rewrite of the genome. Embryogenies that allow heterochronic changes to occur more easily were placed in the middle of the scale; the best embryogenies, which would allow for mutations that can change the developmental process while retaining the same final phenotype, would have been placed on the

right. Their classification of the existing embryogenies is as follows (the references have been changed to match this article):

> A few systems [2, 4, 13, 22] cannot use heterochrony because steps in their development are not parameterized or modulated in any way. Thus, changing the timing of developmental events in these systems would require altering the entire genome. In contrast, the majority of AE systems implement some kind of parameterization or signal modulation system, allowing developmental phases to taper off or initiate at different times. Bongard's [5] system is placed farthest to the right because it is the only system with a reported analysis of heterochrony. No system has implemented radical shifts in timing or the elimination of entire phases of development without disrupting the final product, as seen in natural evolution.
>
> —Stanley and Miikkulainen [28]

The reader should note, however, that this classification is meant to be only a broad, qualitative overview. Systems like the ones by Belew and Kammayer [2] or Kitano [13] can be placed, without any doubt, at the leftmost end of the scale, as they definitely do not contain any heterochronic mechanism. Most existing systems, however, do implement at least one of the mentioned mechanisms, and they were all piled together in the middle of the scale. Stanley and Miikkulainen's definition does not offer any further indication how we could grade these systems more finely, or how we could recognize a system that implements the sort of neutral mutations they describe. We agree with Stanley and Miikkulainen's definition in that these properties are important to measure, but we also believe that we need a quantitative definition in order for their scale to be really useful. With this in mind, in the next section we will introduce Alberch et al.'s framework, which is used to quantitatively measure heterochrony in natural embryogenies. We will later rephrase Stanley and Miikkulainen's concerns within this framework, and that will finally allow us to define these concepts quantitatively.

2.2 Alberch et al.'s Framework

In evolutionary biology there is an established procedure to measure and classify heterochrony in living organisms. This framework was proposed by Alberch, Gould, and colleagues in the 1970s [1]. To give an idea how this framework is applied, we will first describe a brief example drawn from the biological literature. We will base this example on data published in a study conducted by Creighton and Strauss [6], concerning heterochrony in rodents.

A common application of this framework is to classify heterochronic occurrences within a given phylogenetic hypothesis [7]. In other words, if the evolutionary relationships among a number of species are already known in advance, we can apply this framework to label the changes in the growth process that occurred among them. Let's assume that we have a phylogenetic tree as depicted in Figure 2a (this phylogenetic tree is only for illustrative purposes and doesn't represent the true relationships in the genus). The three rodent species *P. melanocarpus*, *P. maniculatus*, and *Podomys* are believed to all have a common ancestor, depicted with a black dot at the bottom of the tree. This common ancestor eventually evolved into two species: *P. melanocarpus*, and another species (represented with a white dot) that is the ancestor of the remaining two. All of the three species exhibit differences in their morphologies, for instance, in the size of their legs and feet.

Alberch et al.'s framework works exclusively with quantitative traits to define heterochrony. In this context, a trait is anything that can be measured quantitatively during development. By measuring the trait during the growth of an individual we obtain a *growth curve* for the trait. Any trait is suitable as long as it can be measured quantitatively. Some of the traits usually employed include the total body length or weight of the individual, or the lengths of individual organs. In their study Creighton and Strauss applied the framework to several different traits, but here we will be focusing on the foot length, for illustration.

Figure 2b shows growth curves of the species *P. melanocarpus* and *Podomys*, obtained from experimental data. As the reader should expect, it is not trivial to define a growth curve for an entire

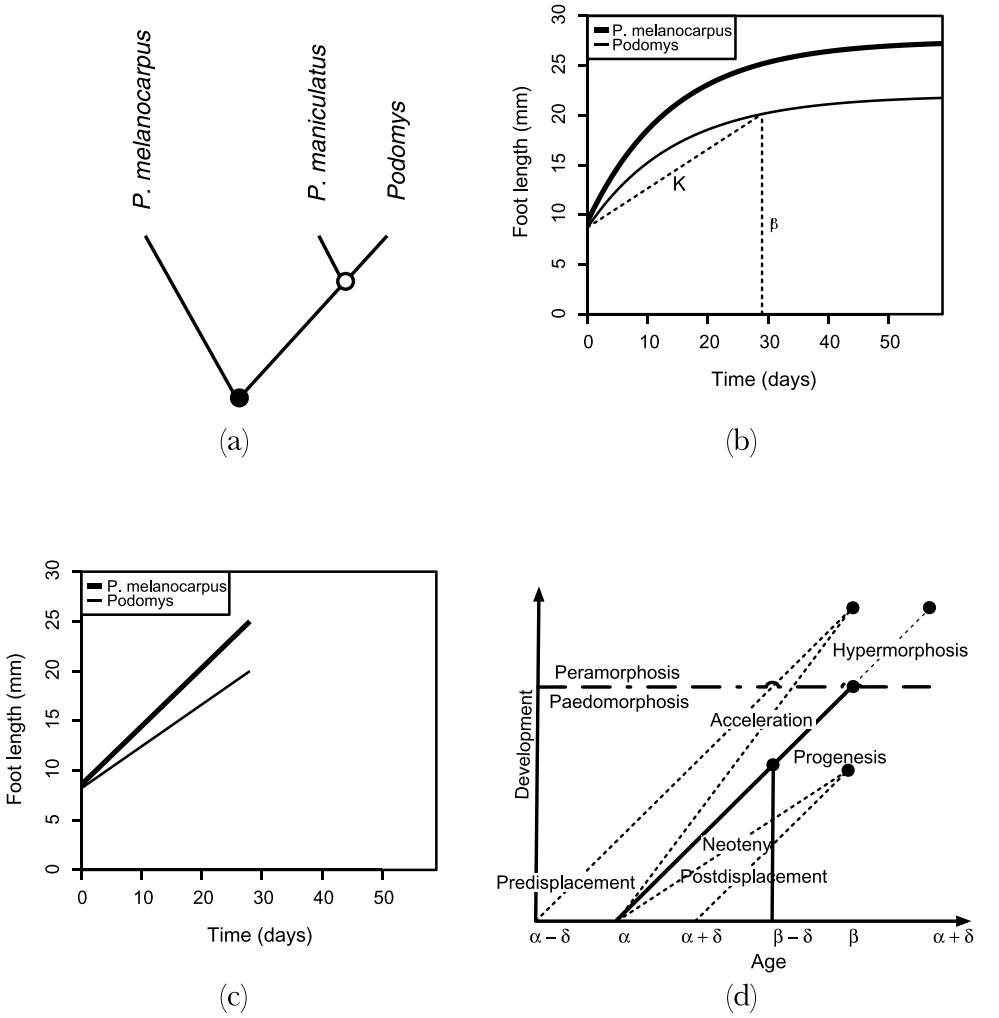

Figure 2. A simple classification example based on the work of Creighton and Strauss [6]. (a) Evolutionary relationships (cladogram) of the genus *Cricetinae*. (b) Growth curves of the species *P. melanocarpus* and *Podomys*. (c) Idealized growth curves, with the heterochrony change clearly visible. (d) The major heterochronic changes in Alberch et al.'s framework.

species. Species are made of individuals, and individuals' growth curves will differ according to their genetic contents. Also, development does not occur in isolation from the environment, and growth curves will definitely undergo a strong environmental influence. An important first step, therefore, is to try to remove all of these influences and create a typical growth curve for the species. In order to do this, the authors collected and measured the traits for several individuals, and then fitted the data to a curve, using nonlinear regression. Biological organisms tend to follow well-established growth dynamics, and several suitable mathematical models of growth exist. In this case, Creighton and Strauss fitted the data with a Von Bertalanffy curve, which tends to fit very well with biological data [30].

Heterochrony in this framework is defined by the differences found in the growth curves of the species, but they are difficult to compare directly. Therefore the framework defines three parameters that contain the most important information on the development. They are: α, the time when growth starts; β, the time when growth ends; and K, the average growth rate. In this example, α is considered to be the time of birth, and is the same in both species; β, because growth tends to decelerate drastically towards the end, was defined as the time when growth reached 90% of the final

size. *K* was computed as the slope of the line connecting the sizes at α and β, as depicted in Figure 2c. Changes in growth are much more obvious in the reduced versions of these growth curves; we can easily see that the heterochronic change occurring in these species was only a change in the rate of growth (the parameter *K*). If we assume that this specific change occurred in the *Podomys* species, then, according to the framework, this is a case of *neoteny* (slowdown of growth rate). Besides this change, the framework defines names for all individual changes in each parameter (Figure 2d). In practice, however, these changes rarely occur in isolation, so that in fact most heterochronic changes are described by more than one of the labels in Figure 2d.

2.3 Adapting the Framework

We believe that Alberch et al.'s framework, as described in the previous subsection, is particularly suitable for artificial embryogeny. First, the framework allows us to compare different systems by using their growth curves alone, without any need to refer to any underlying genetic or developmental basis. This allows us to apply it directly to AEs that are not based on natural assumptions, or even to compare embryogenies that are based on radically different mechanisms. The framework is rather easy to adopt to the AE context, but nevertheless some adaptations must be made.

Some of the most problematic parts of the framework actually become simplified when applying it to AE. EAs and AEs are highly abstracted versions of their natural counterparts, so we don't have some of the problems that plague our biologist colleagues. One obvious example is measuring growth data, which is much easier for us, the data being much more readily available. AEs do not involve dilemmas on what time frame is suitable for measuring development, or on how to deal with measurement errors [14].

However, it is important to discuss how we should extract the heterochronic parameters, that is, how can we define the parameters α, β, and *K* on AEs. As we described in the previous section, natural embryogenies tend to have well-behaved growth dynamics that can be fitted easily to mathematical growth models. Based on our previous experience [20, 21], we can state that this is not the case with AEs. Another problem is that we still don't understand the growth dynamics of AEs very well, so it is still early to propose any mathematical growth model to describe them. This doesn't mean we cannot apply this framework, because complex dynamics also happens in some natural embryogenies, for instance, during embryo development [14]. However, the method we have described in the preceding subsection is not suitable, and we will be using a much simpler method to extract the parameters. This method, based on linear regression, is robust, and is also widely used to study heterochrony in natural embryogenies [23].

The major question in applying the framework, however, is one of scope. The example that we gave uses the framework to characterize heterochrony in species, which usually have no counterpart in AE. Obviously there are EAs that employ the notion of species, but is much more natural for us to think in terms of individuals, and how the variation operators change the individuals. This is not a particular problem for the framework, as it as also been used to characterize variation in individuals of the same species [23]. The framework can be easily extended to accommodate mutation, just by treating the two individuals involved as two different species, and conducting a similar process to that in the previous example. However, it is not trivial to perform the same procedure for the crossover operator, so we will not consider it at this time. However, we do plan to extend this analysis to crossover in a later article.

Another difference from our previous example is that we will not be using the heterochronic labels in the framework, and instead we will be using directly the parameters α, β, and *K* to describe the heterochronic changes. This will allow us to rephrase Stanley and Miikkulainen's concerns in a quantitative manner, as we will be doing in the next section.

The attentive reader will note that our definition of heterochrony here is rather different from Stanley and Miikkulainen's. Their scale of heterochrony is defined exclusively by how easily an embryogeny allows heterochronic changes to occur, and this would roughly correspond to the number of heterochronic mechanisms it possesses. The definition in Alberch et al.'s framework is broader, in

that *any* mutation that creates a change in development will be considered heterochronic. We are pursuing the broadest possible definition here precisely because it doesn't make any assumptions on the mechanisms used by an embryogeny, and therefore can be used for comparing very different embryogenies. In the end, from the point of view of Alberch et al.'s framework, it doesn't really matter if the cause of change was in fact a change in the order or time of developmental events, as it will still be a heterochronic change. Heterochrony, from this point of view, is seen as an outcome of evolution, instead of a mechanism that allows evolutionary change [14, Section II]. This dichotomy in meaning does not only occur here, and it actually reflects the different meanings of heterochrony employed in evolutionary and developmental biology [25]. Nevertheless, we believe that our formalism still captures well Stanley and Miikkulainen's concerns, although in a slightly different manner.

3 The Formalism

3.1 Preliminaries

We will start our discussion with some preliminary definitions and formalizations. In this article, for all relevant purposes, an embryogeny E can be completely characterized by the items below (the notation is adapted from [29] and [31]):

- a *genotype space* Ω_E,
- a *phenotype space* $\tilde{\Omega}_E$,
- a *growth function* that maps genotypes to phenotypes, $\phi_E(g, t) : \Omega_E \times \mathcal{D}_g \rightarrow \tilde{\Omega}_E$,
- a *mutation function* $\mu_E(g) : \Omega_E \rightarrow \Omega_E$.

Ω_E is the set of all possible genotypes in the embryogeny. For instance, Ω_E could be the set of all binary vectors of length 10, or the set of all genetic programming trees with a given command set. Likewise, $\tilde{\Omega}_E$ is the set of all the possible phenotypes in a given embryogeny, for instance, all of the possible neural networks with a given depth, or all the possible 2D patterns on a 5×5 grid.

We will assume that development for any genotype g occurs in discrete steps, starting at time $t = 0$ and ending at time $t = N_g$, where N_g depends on the genotype. Therefore the *developmental time domain* $\mathcal{D}_g$ of a genotype g is a subset of the set of positive natural numbers, $\mathcal{D}_g \subset \mathbb{N}_0^+ = \{1, 2, \ldots, N_g\}$. We believe that this assumption concerning $\mathcal{D}_g$ applies to all AEs, but there are two points that must be mentioned. First, in some embryogenies, N_g is defined at the embryogeny level, that is, its value is fixed and the same for all genotypes. Our assumption covers this case as well, so this does not create any problems. Second, some embryogenies theoretically have a real, continuous $\mathcal{D}_g \subset \mathbb{R}_0^+$. This is the case when embryogenies use differential equations to model genetic regulation, and these equations must be integrated over time. In practice, however, these equations are usually integrated numerically with a fixed step size, so in the end this type of embryogeny still falls under our assumptions.

$\phi_E(g, t)$, the growth function, maps genotypes to phenotypes, including intermediate phenotypes that occur at any developmental time $t \in \mathcal{D}_g$. In almost all embryogenies, the *starting phenotype* of g, $\phi_E(g, 0)$, is fixed and is the same for all g. The *final phenotype* of g, $\phi_E(g, N_g)$, is usually the one that is employed for evaluating g. Note that we define $\tilde{\Omega}_E$ as the set of all phenotypes in an embryogeny, including intermediate and final ones.

The mutation function, $\mu_E(g)$, is a nondeterministic function that returns a mutated form of g. Based on this function, we define the *mutation set* Θ_E of E as the set of all possible mutation events occurring in E:

$$\Theta_E = \{(g_1, g_2) : g_1 \in \Omega_E, g_2 : \mu_E(g_1) = g_2\} \quad (1)$$

Formally, this is the set of all pairs in Ω_E that are linked by a single mutation operation.

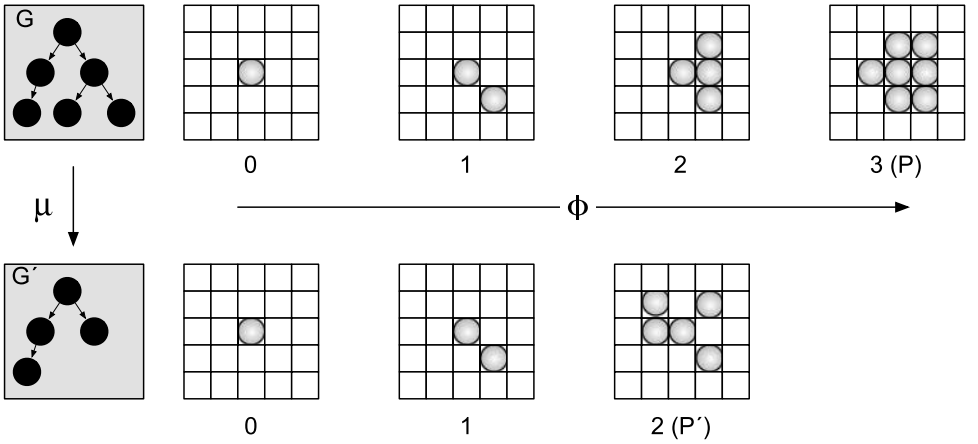

Figure 3. A sample embryogeny (which we will denote by $E1$). $E1$ is a genetic programming embryogeny, where the genotypes contain instructions for growing patterns in a 2D grid. G and G' are two genotypes in Ω_{E1}, where G' was created by mutating G (and therefore the pair (G,G') belongs to Θ_{E1}). The genotypes' N is correlated with the number of nodes in their trees; in this case, $N_G = 3$ and $N_{G'} = 2$. On the right, $P = \phi_E(G, 3)$ and $P' = \phi_E(G', 2)$ are the final phenotypes of G and G' respectively. The figure also shows all of their intermediate phenotypes.

In Figure 3, we show how these abstractions can be mapped into settings of a real embryogeny. Incidentally, this embryogeny is similar to the ones we will be using later, although any other conceivable setting could be used.

3.2 Parametrization Functions

This is mainly a formalization of the explanation of Alberch et al.'s framework given in Section 2.2. We denote the set of all traits measurable in an embryogeny as $\mathcal{T}_E$. Members of this set may include the total number of cells on a grid, the number of different colors on a grid, or the number of red cells, if the embryogeny is used to grow patterns, as in Figure 3. Or it might include the number of layers and the number of synapses, if the embryogeny is used to grow neural networks. Any conceivable trait can be used, as long as it is quantitative and changes during development. Each trait has associated with it a *trait measure function*, $M_T(p) : \tilde{\Omega}_E \rightarrow \mathbb{R}$, that computes the trait value for a given phenotype. We will assume, without loss of generality, that the trait values' domain is $\mathbb{R}$.

Assume that we pick a genotype g from Ω_E, grow it, and apply a trait measure function to each of its intermediate phenotypes—in other words, that we are computing the composite function $G_{g,T}(t) = M_T(\phi_E(g, t))$. This resulting function, $G_{g,T}(t) : \mathcal{D}_g \rightarrow \mathbb{R}$, the *growth curve* of trait T on genotype g, shows how a trait T changes its value during the growth of a given genotype. Figure 4 shows plots of some sample growth curves.

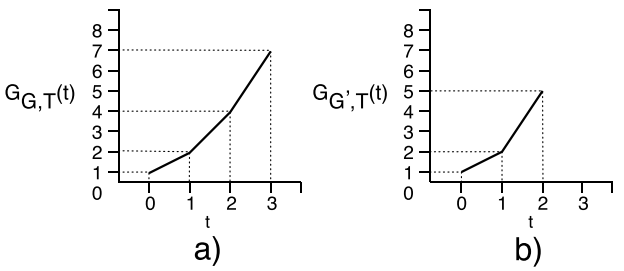

Figure 4. The growth curves of the genotypes in Figure 3: (a) G, (b) G'. The trait T used in this case is the total number of cells in the grid.

Growth curves may exhibit rather complex dynamics, and we need a way to extract relevant information from them. In the context of Alberch et al.'s framework, this is done by defining three crucial parameters of development, as explained before. In practice, there are several ways this could be done, and there is no consensus in the biological literature as to which method is the best. A common and simple approach is to use linear regression, and that is the one we will be using here. The reader is referred to Klingenberg [14] for explanations of other methods. To extract the parameters, we define the *linear parametrization function*, $L_T(g) : \Omega_E \times \mathbb{R}^3$. The resulting vector, $\mathbf{X} = \langle \alpha, \beta, K \rangle$, is computed as follows:

- α is the time $t \in \mathcal{D}_g = \{0, 1, \ldots, N_g\}$ that a change in trait value can first be observed; more formally, the first t such that $G_{g,T}(t) \neq G_{g,T}(g, t+1)$. If the trait value does not change at all, then we assume $\alpha = N_g$.
- β is the time $t \in \mathcal{D}_g = \{0, 1, \ldots, N_g\}$ that the last change in a trait value can be observed; more formally, the last t such that $G_{g,T}(t) \neq G_{g,T}(g, t+1)$. In a similar way to α, we assume $\beta = N_g$ if the trait value does not change at all.
- K, the average growth rate, is the slope of the line connecting the values at time α and β:

$$K = \frac{G_{g,T}(\beta) - G_{g,T}(\alpha)}{\beta - \alpha} \tag{2}$$

Figure 5 depicts the application of L_T to our previous examples.

We will also call the result of applying L_T the *ontogenetic parameters* of genotype g on the trait T. A pair (g_1, g_2) from the mutation set Θ_E has associated a *heterochronic change* in trait T defined as the difference in their ontogenetic parameters:

$$H_T(g_1, g_2) = L_T(g_2) - L_T(g_1) \tag{3}$$

For instance, in the example we have been following until now, the heterochronic change of the two genotypes is $H_T(G, G') = \langle 0, -1, 0.5 \rangle$. Another possibility would be to define the heterochronic change as the direct difference of the growth curves $G_{G,T}(t) - G_{G',T}(t)$, leading to a more general characterization. However, this was not defined in the original framework of Alberch et al., so we will not persue this definition here.

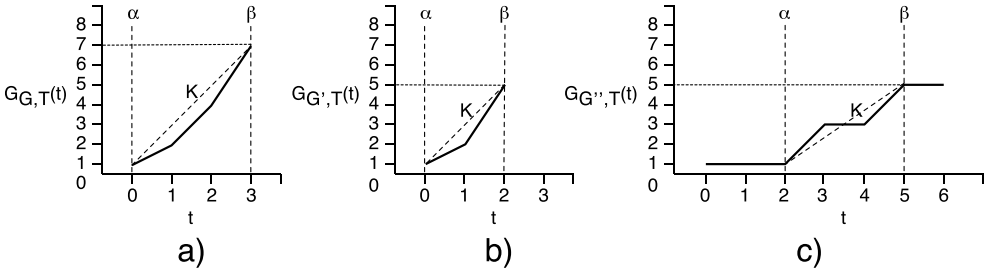

Figure 5. Computing L_T. (a) G (from Figure 4), $L_T(G) = \langle 0, 3, 2 \rangle$. (b) G' (from Figure 4), $L_T(G') = \langle 0, 2, 2.5 \rangle$. (c) A more complex example, G'', with $L_T(G'') = \langle 2, 5, 1 \rangle$.

3.3 Neutral and Heterochronic Sets

Having the framework formalized, we are now able to give more precise definitions of the concepts discussed in previous sections. We will call a mutation (g_1, g_2) *neutral* on a trait T when it does not change the trait value of the final phenotype:

$$M_T(\phi_E(g_1, N_{g_1})) = M_T(\phi_E(g_2, N_{g_2})) \tag{4}$$

This definition of neutrality is different from the notion usually implied in EC. The usual definition relates to fitness, and not to traits. In an EC context, a mutation can be considered neutral even if it changes the traits of the phenotype, as long as the fitness of the two genotypes remains the same. If we assume that the embryogeny only uses the final phenotypes for fitness evaluation, then our definition of neutrality is included in the EC's definition, but even so, they are not equivalent. As we are characterizing embryogenies in a general way (e.g., without referring to any specific fitness function), we will not use any definitions related to fitness in this article.

A mutation (g_1, g_2) is said to be *heterochronic* on a trait T when it changes the growth process in any way, that is, by creating a change in at least one of the parameters α, β, or K:

$$H_T(g_1, g_2) \neq \langle 0, 0, 0 \rangle \tag{5}$$

A point to bear in mind is that our definitions of neutral and heterochronic mutations are always based on traits computed from the phenotypes. For instance, the same mutation may be neutral for one set of traits but not another, if the mutation only changes traits in the latter set. In a similar way, the same mutation may be heterochronic for some traits but not for others. For some readers, this definition may seem strange and unnecessary. Wouldn't it be better just to avoid using traits, and define neutrality and heterochrony directly on the phenotypes? On simple phenotypes, like the 2D pattern examples we have been using here, it is indeed possible to compare the patterns directly, and in a computationally efficient way. But there are several classes of phenotypes where this would be too difficult to do or even impossible. Artificial creatures are a good example, because they usually possess a very complex graph structure with several different kinds of nodes and connections. These phenotypes cannot be compared directly, although it is possible to devise heuristic procedures that compute an approximate measure of resemblance [17]. By using a definition based on traits alone, we do not become limited to simple phenotypes.

Based on these concepts, we may define two subsets of the mutation set Θ_E of an embryogeny: The *neutral mutation set* $N_{E,T}$ contains all of the neutral mutations in Θ_E, while the *heterochronic mutation set* $H_{E,T}$ contains all of the heterochronic mutations. The relationships among the sets are depicted graphically in Figure 6. The neutral and heterochronic sets do not have to be distinct; we may further

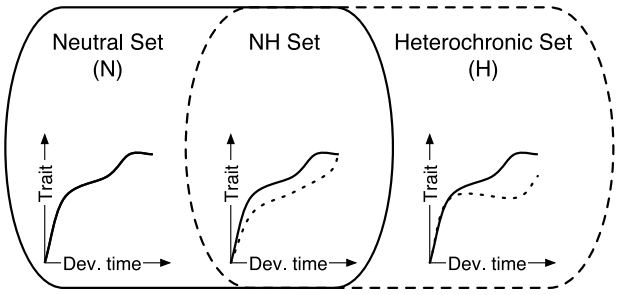

Figure 6. Relationships among the neutral, heterochronic, and NH sets of an embryogeny. The whole set, that is, the three boxes considered together, is the mutation set Θ_E. The NH set is part of both the neutral and heterochronic sets. Each box contains an example of a representative mutation on each set. The solid lines and dashed lines represent growth curves before and after mutation. For the example, in the neutral set box only the solid line is drawn, as the growth curve doesn't change with mutation.

think of an additional class of mutations that do not change the final phenotype, but still change the underlying developmental process, as explained in Section 1. These mutations, being both neutral and heterochronic, lie in a set defined as the intersection between $N_{E,T}$ and $H_{E,T}$. We will call this set the *neutral and heterochronic* (NH) *set* of a trait T, or $NH_{E,T}$ for short.

A straightforward way to characterize embryogenies based on these sets is to associate probabilities with each of them. Let's assume that we pick up a mutation event from an embryogeny at random; this mutation event could be neutral, heterochronic, or even both, and these occur with some given probabilities. We will denote the probabilities for a random mutation event to fall into one of these three categories as $P(N_{E,T})$, $P(H_{E,T})$, and $P(NH_{E,T})$ respectively:

$$P(N_{E,T}) = P((g_1, g_2) \in N_{E,T})), \qquad (g_1, g_2) \in \Theta_E \tag{6}$$

$$P(H_{E,T}) = P((g_1, g_2) \in H_{E,T})), \qquad (g_1, g_2) \in \Theta_E \tag{7}$$

$$P(NH_{E,T}) = P((g_1, g_2) \in NH_{E,T})), \qquad (g_1, g_2) \in \Theta_E \tag{8}$$

Knowing these probabilities allows us to compare different embryogenies on how their behavior affects mutation, and how much they allow the EA to explore neutral landscapes. We will explore these concepts more thoroughly when we apply them to two sample embryogenies, later in this article.

3.4 Probability Distributions

The measures shown in the previous subsection allow us to characterize the behavior of the mutation operator at a gross level, but they still cannot tell us anything about how the operator changes the growth dynamics. We will now turn to this problem.

Assume that, in a given embryogeny, we pick up a mutation event (g_1, g_2) at random from the heterochronic set $H_{E,T}$. For any given trait T, we can compute for this random mutation event its heterochronic parameters $H_T(g_1, g_2)$, which will be in the form of a vector $\mathbf{X} = \langle \alpha', \beta', K' \rangle$. As we picked up this mutation event at random, $\mathbf{X}$ is in fact a random vector. This vector has an underlying probability distribution, and we will call this distribution the *heterochronic distribution* of the trait T in the embryogeny E, or $\mathcal{H}_{E,T}$ for short. This distribution summarizes the behavior of the mutation operator under a given embryogeny. It basically states how much one single mutation may change the developmental trajectories between two genotypes.

AEs usually employ standard mutation operators, but their behavior is highly complex because any changes caused in the genotypes will ultimately be mediated by the embryogeny. Therefore different embryogenies may have radically different heterochronic distributions even if they share the same mutation operator. This is the case with the embryogenies that we will be describing later in this article, which indeed have similar representations and use the same mutation operator. Their differences are readily evident in Figure 7, where we show density plots of their heterochronic distributions.

We do not need to be limited, in this approach, to the heterochronic set alone; we can extend this concept to the set $NH_{E,T}$ as well. Let's assume that this time we pick up a mutation event (g_1, g_2) from $NH_{E,T}$, and that we compute the vector $\mathbf{X}$ following the same procedure as before. Then again $\mathbf{X}$ will be a random vector, and we will call its probability distribution the *NH distribution*, denoted $\mathcal{NH}_{E,T}$, of the embryogeny E on trait T.

This approach is similar to the one employed for the heterochronic distribution, but actually these two distributions characterize very distinct behaviors. The NH distribution characterizes how much a mutation may change development in a given embryogeny without affecting the final phenotype. This is different from the $P(NH_{E,T})$ measure proposed before, in that $P(NH_{E,T})$ specifies only how many NH mutations may occur, while the NH distribution shows how much these mutations may change the actual parameters. So they are in fact complementary tools. An example is shown in Figure 8.

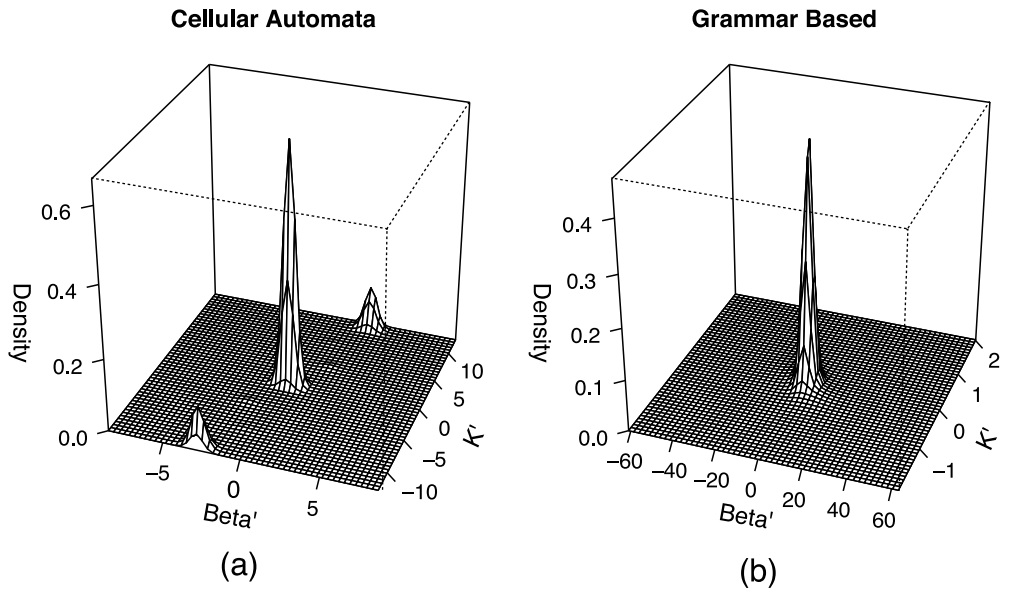

Figure 7. Sample heterochronic distributions (density plots), for the cellular automata and grammar-based embryogenies, of the number-of-cells trait (these embryogenies are described at length later in this article). Each value on the density axis corresponds to the probability that a mutation event, picked at random, will have a specified β' and K'. For simplicity, changes in α' were ignored. Plot data was obtained from experimental results and interpolated by using a kernel smoothing technique [32, 24]. (a) $\mathcal{H}_{\text{CA, number of cells}}(\text{bandwidth}(\beta') = 0.4, \text{bandwidth}(K') = 0.4)$; (b) $\mathcal{H}_{\text{grammar-based, number of cells}}(\text{bandwidth}(\beta') = 2, \text{bandwidth}(K') = 0.1)$.

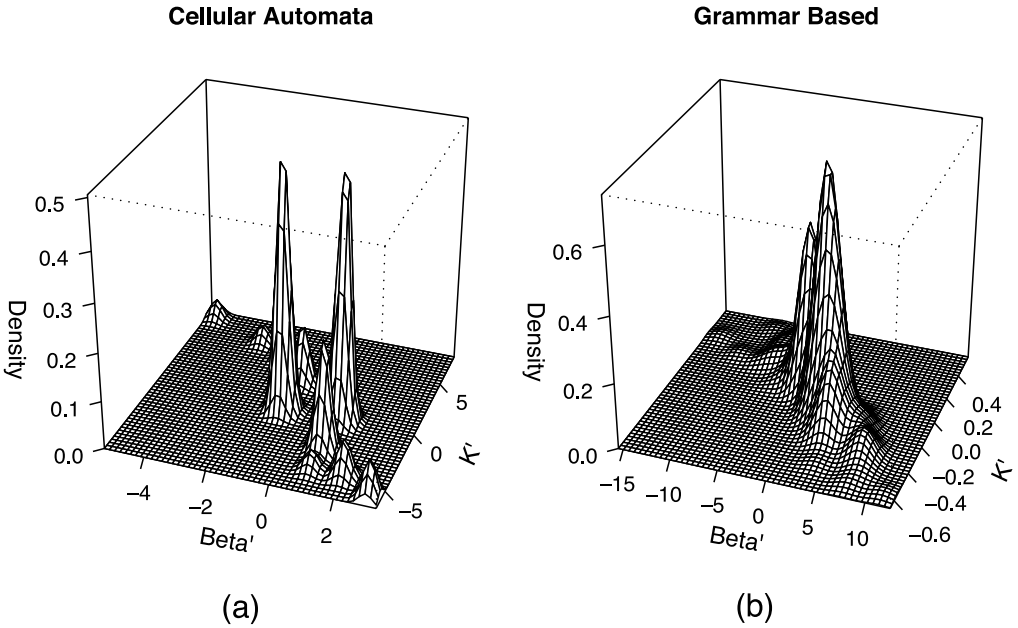

Figure 8. Sample NH distributions (density plots). The density plots were computed in a way similar to the ones displayed in Figure 7, and represent the same embryogenies. (a) $\mathcal{NH}_{\text{CA, number of cells}}(\text{bandwidth}(\beta') = 0.2, \text{bandwidth}(K') = 0.4)$; (b) $\mathcal{NH}_{\text{grammar-based, number of cells}}(\text{bandwidth}(\beta') = 1, \text{bandwidth}(K') = 0.05)$.

We can go even further and create a probability distribution based on the dynamics of the individuals themselves. Assume that, in a given embryogeny, we pick up a genotype g at random from the space of all possible genotypes Ω_E. With this g we compute its ontogenetic parameters on a given trait T, so that we end up with a vector $L_T(g) = \langle \alpha, \beta, K \rangle$. The values of $L_T(g)$ will vary according to the genotype that was picked, so, as before, $L_T(g)$ will be a random vector. We will call the distribution of this random vector the *ontogenetic distribution* of trait T in the embryogeny E, or $\mathcal{O}_{T,E}$ for short. Ontogenetic distributions are different from the previous distributions in that they characterize individuals (the kind of growth curves the embryogeny generates) instead of mutations. In Figure 9 we show the ontogenetic distributions of the two previous embryogenies.

3.5 Characterizing the Distributions

Having defined the distributions, we need a way to compare them in different embryogenies. One way to do this, which we explored in the previous section, is to compare them visually by looking at their density plots. In this section we will explain how we can compare them quantitatively by using standard statistical measures. The previous distributions are all trivariate distributions, and therefore we can analyze them by using multivariate statistics [12]. For instance, an ontogenetic distribution can be characterized by a *mean vector* $\mu(\mathcal{O}_{E,T})$:

$$\mu(\mathcal{O}_{E,T}) = \begin{bmatrix} \overline{\alpha} \\ \overline{\beta} \\ \overline{K} \end{bmatrix} \tag{9}$$

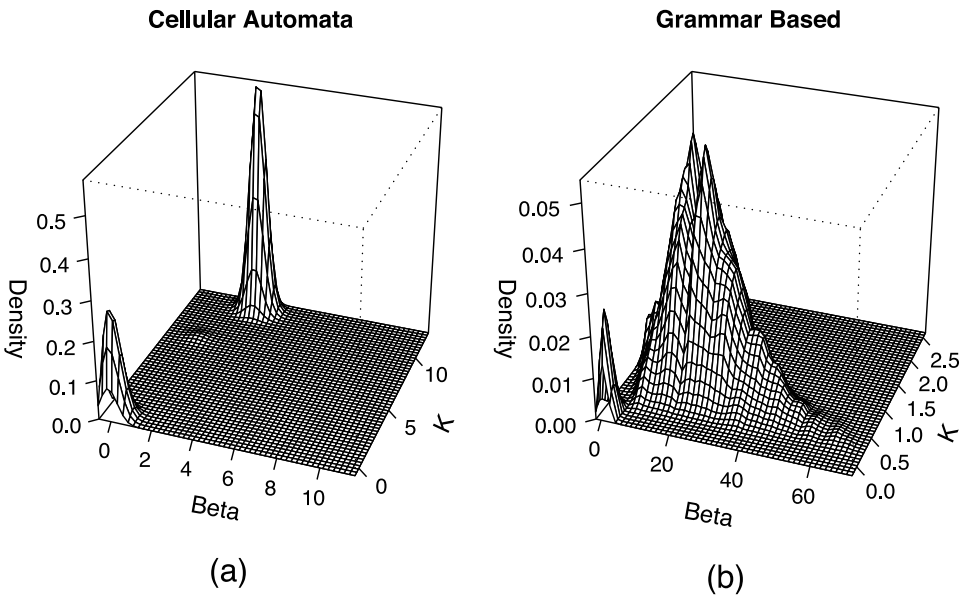

Figure 9. Sample ontogenetic distributions (density plots). The density plots were computed in a similar way to the ones displayed in Figure 7, and represent the same embryogenies. (a) $\mathcal{O}_{\text{CA, number of cells}}(\text{bandwidth}(\beta) = 0.4, \text{bandwidth}(K) = 0.4)$; (b) $\mathcal{O}_{\text{grammar-based, number of cells}}(\text{bandwidth}(\beta) = 1, \text{bandwidth}(K) = 0.1)$.

and a *variance-covariance matrix* $\Sigma(\mathcal{O}_{E,T})$:

$$\Sigma(\mathcal{O}_{E,T}) = \begin{bmatrix} \mathrm{var}(\alpha) & \mathrm{cov}(\beta,\alpha) & \mathrm{cov}(K,\alpha) \\ \mathrm{cov}(\alpha,\beta) & \mathrm{var}(\beta) & \mathrm{cov}(\beta,K) \\ \mathrm{cov}(\alpha,K) & \mathrm{cov}(\beta,K) & \mathrm{var}(K) \end{bmatrix} \tag{10}$$

where $\overline{v}$ represents the mean of the variable v, var(v) represents its variance, and cov(v_1, v_2) represent the covariance between v_1 and v_2 (for simplification, the indexes E and T were dropped from the right side of the definitions). The other distributions have similar measures, which we denote by $\mu(\mathcal{H}_{E,T})$ and $\Sigma(\mathcal{H}_{E,T})$ for the heterochronic distribution, and $\mu(\mathcal{NH}_{E,T})$ and $\Sigma(\mathcal{NH}_{E,T})$ for the NH distribution. All of these standard measures are important for characterizing embryogenies. For instance, the mean vector of an ontogenetic distribution gives us an idea of the typical ontogenetic parameters in a population, while the variance-covariance matrix of a heterochronic distribution may indicate how changes in the parameters are correlated when a mutation occurs.

Another important way to compare embryogenies is to measure the variation of their distributions. This allows us to see how much of the search space is covered by an embryogeny, and how much it allows the mutation operator to explore the search space. A useful dispersion measure for multivariate distributions is the *generalized variance* of a distribution, defined as the determinant of the variance-covariance matrix. The generalized variance is a single, summary statistic representing the "scatteredness" of a multivariate distribution, and it is akin to the variance in the univariate case. We will denote the generalized variances of the ontogenetic, heterochronic, and NH distributions by $|\Sigma(\mathcal{O}_{E,T})|$, $|\Sigma(\mathcal{H}_{E,T})|$, and $|\Sigma(\mathcal{NH}_{E,T})|$ respectively.

3.6 Obtaining the Results Experimentally

All of the measures described up until now assume that the distributions $\mathcal{O}_{E,T}$, $\mathcal{H}_{E,T}$, and $\mathcal{NH}_{E,T}$ of an embryogeny are known. In practice, most of the time we cannot compute the real distributions, because the spaces Ω_E, Θ_E, $H_{E,T}$, and $NH_{E,T}$ are extremely large, or even infinite. It is possible, however, to obtain approximate values of the measures by sampling from these spaces. We will denote the sampled spaces as $\hat{\Omega}_E$, $\hat{\Theta}_E$, $\hat{H}_{E,T}$, and $\widehat{NH}_{E,T}$. A sample genotype space $\hat{\Omega}_E$ can be obtained by generating n genotypes at random, for instance, by saving the first generation of an EA run. The mutation set $\hat{\Theta}_E$ can be obtained by generating random mutation events, that is, by generating a random genotype g, mutating it, saving the pair, and repeating the process several times. The sets $\hat{H}_{E,T}$ and $\widehat{NH}_{E,T}$ can be computed by extracting the relevant elements from $\hat{\Theta}_E$.

The distribution measures can be estimated easily from these sample spaces. For instance, let's assume that we have generated n random genotypes. If we compute $L_T(g)$ (on any trait T) for each of these genotypes, we will have n vectors with their ontogenetic parameters $X_i = \langle \alpha_i, \beta_i, K_i \rangle$, $i \in \{1, 2\ldots, n\}$. Suppose, furthermore, that we compute the mean vector $\overline{\mathbf{X}}$ and the variance-covariance matrix $\mathbf{S}$ of the set of vectors X. Then the vector

$$\hat{\mu}(\mathcal{O}_{E,T}) = \overline{\mathbf{X}} \tag{11}$$

and the matrix

$$\hat{\Sigma}(\mathcal{O}_{E,T}) = \frac{n}{n-1}\mathbf{S} \tag{12}$$

are unbiased estimators of $\mu(\mathcal{O}_{E,T})$ and $\Sigma(\mathcal{O}_{E,T})$. Due to the central limit theorem, we know that these estimators approach the values of the real distributions as n increases. Similar measures $\hat{\mu}(\mathcal{H}_{E,T})$ and $\hat{\Sigma}(\mathcal{H}_{E,T})$ can be computed for the heterochronic distribution, and $\hat{\mu}(\mathcal{NH}_{E,T})$ and $\hat{\Sigma}(\mathcal{NH}_{E,T})$ for the NH distribution.

The other measures, presented in Section 3.3, can be computed easily from the sample sets as well:

$$P(N_{E,T}) = \frac{|\hat{N}_{E,T}|}{|\hat{\Theta}_E|} \tag{13}$$

$$P(H_{E,T}) = \frac{|\hat{H}_{E,T}|}{|\hat{\Theta}_E|} \tag{14}$$

$$P(NH_{E,T}) = \frac{|\widehat{NH}_{E,T}|}{|\hat{\Theta}_E|} \tag{15}$$

where $|s|$ stands for the number of elements in the set s.

4 Applying the Framework

In this section we will apply the concepts explained in the previous section to two embryogenies. These embryogenies—which we will call the cellular automata (CA) and the grammar-based (GB) embryogenies—are two simple but representative embryogenies for evolving 2D patterns. To keep the discussion short and relevant, in this section we will only give a brief overview of how the two embryogenies work. Interested readers can find more details in the Appendix.

Both embryogenies follow the same principles presented in Figure 3. They are both based on genetic programming, and employ the same mutation operator. The embryogenies are able to grow 6×6 2D patterns where each cell can have one of four colors: gray, red, white, or blue. They also start from the same initial phenotype, in this case, an empty 2D pattern with a single gray cell in the middle. Similar to most other AEs, they allow for cell proliferation and differentiation, but not for apoptosis (cell death). In order to make the comparison fair, we kept all the settings as similar as possible.

The CA embryogeny works in a similar way to a cellular automaton. The genotype specifies a set of rules representing how a cell should set its state according to a given configuration. This configuration includes the cell's own state and the states of the eight surrounding positions. The representation used is rather flexible, and allows for both specific and general rules. For instance, rules can have don't-care symbols, indicating that they can ignore colors in given positions. It is also possible to specify rules that would be activated if there is at least one cell with a specific color, regardless of the position where it occurs. Nevertheless, the grid's boundaries are always respected, so cells at the boundaries cannot query the cells at the other extreme to set their state. The cell state can be set to any of the four colors mentioned before; if there isn't any cell yet in the given position, then a new cell will be created. However, after a cell is added to a grid, it cannot be removed again, and only its color can be changed.

The number of development steps in this embryogeny is fixed, and always set to 50. On each step, the embryogeny scans the grid from left to right and top to bottom, and attempts to apply a rule from the genotype at each position in the grid. Parts of the grid that do not include at least one cell in the current position and its eight neighboring positions are ignored. We have imposed this rule to avoid having cells appearing unconnected to other cells. For illustration purposes, an example of CA genotype is given in Figure 10.

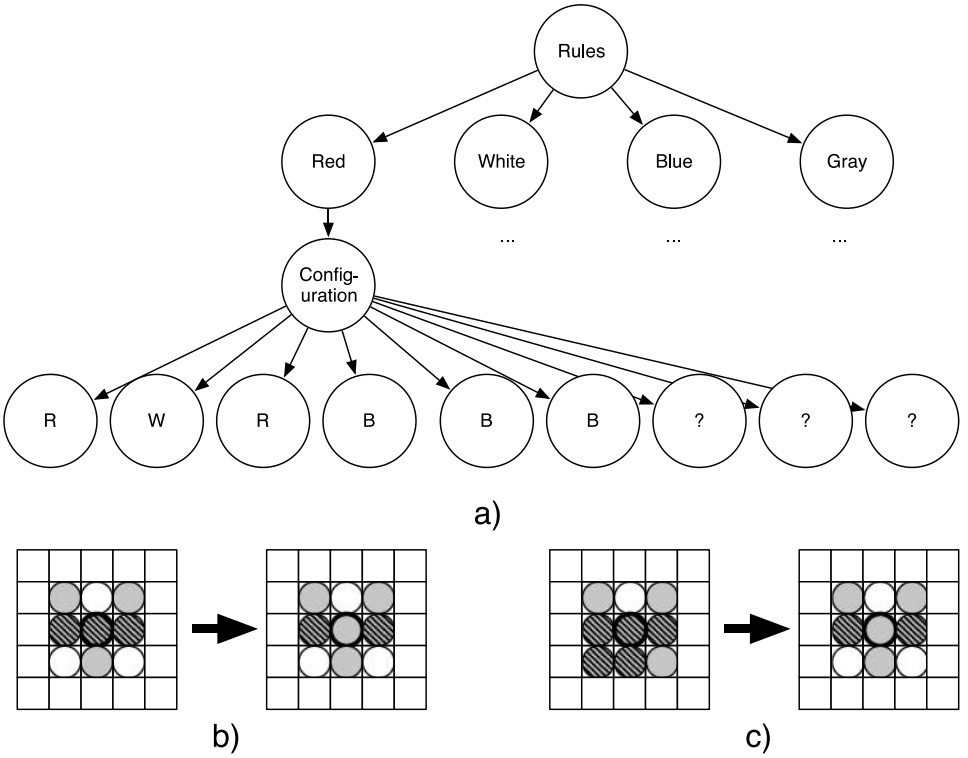

Figure 10. (a) An example genotype for the CA embryogeny. Rules are specified by the top `Rules` element, which has four children, one for each possible state. Below each outcome there is a `Configuration` node that specifies the matching part of the rule. In this example, for simplicity only the red outcome is shown. The nodes below the `Configuration` node specify states that the neighboring cells should have for this rule to be matched. The nodes specify a state for a 3 × 3 patch in the grid, starting from the top left and going to the bottom right in horizontal order. The state nodes are: R, red; W, white; B, blue; G, gray; ?, don't care. (b) and (c) show some sample matches for this rule.

The GB embryogeny has a very different developmental mechanism, which is similar to Bentley and Kumar's explicit embryogeny [3]. The genotype specifies a program, made of step-by-step instructions, to grow a tissue. Each cell in the grid has a pointer to different parts of the genetic programming tree, so each cell will execute different developmental instructions. Commands are of three types: split, query, and color commands. Split commands specify that a cell must split into two cells, and that the new cell must be created on a specified coordinate. Coordinates are always specified relative to the current cell, and they can only refer to their eight immediate neighboring positions. After a split command is executed, the old and the new cell will execute different developmental programs, specified by different children of the genetic programming node. Query commands allow a cell to query the state of the surrounding cells, again limited to its eight immediate neighbors. Finally, color commands allow the cell to change its own color. The cells can also call one automatic definition tree (ADF), which allows for some modularization of the growth process. When the cell doesn't have any more commands to execute, it terminates its development and doesn't change state again.

All the cells in the embryogeny execute their developmental instructions concurrently. In each developmental step the embryogeny executes one instruction for each cell present in the tissue, unless the cell has already finished its development. Growth finishes when all the cells have executed all of their commands, so the number of developmental steps is not fixed as in the CA embryogeny, and depends on the genotype. A sample genotype and its development is shown in Figure 11.

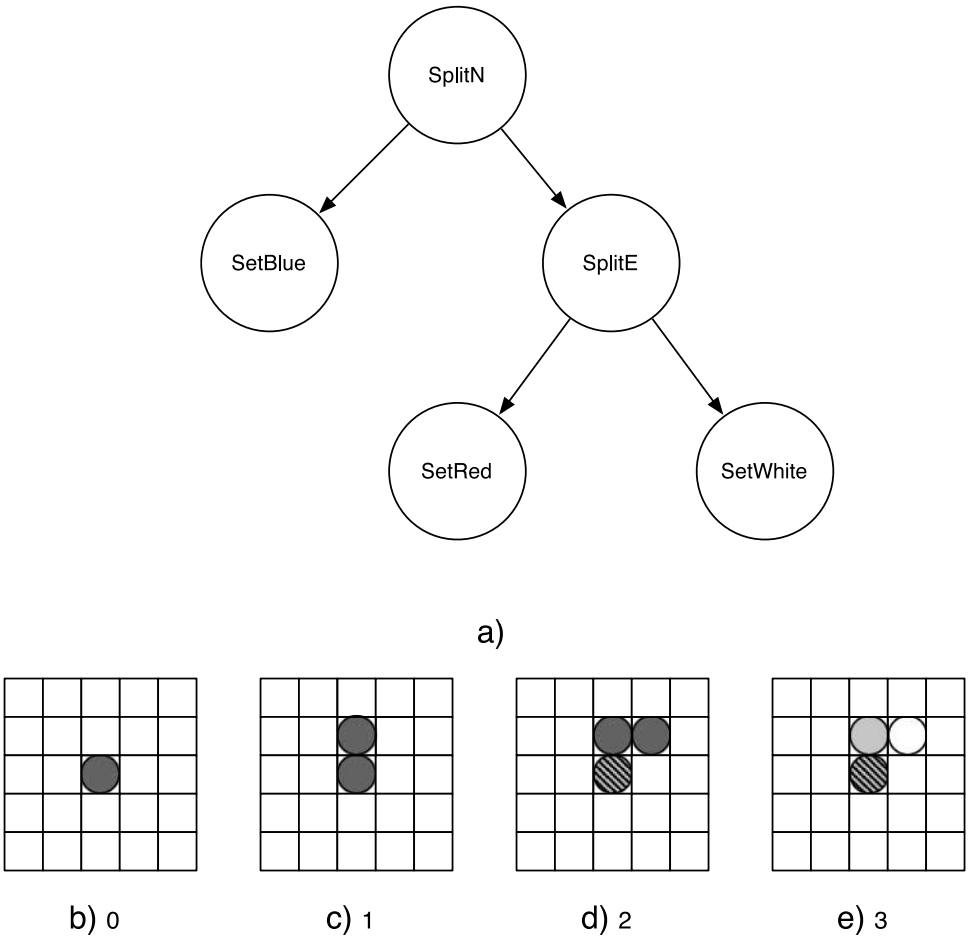

Figure 11. An example GB genotype and its development. For simplicity, neither the ADF1 tree (not used in this example) nor the terminal nodes are shown. (a) The genotype. `SplitN` commands specify that the cell should divide, with the new cell placed on the specified cardinal position. `SetN` commands specify the new state of the cell. (b) The initial phenotype. (c) After `SplitN`. (d) After (`SetBlue`, `SplitE`). (e) After (`SetRed`, `SetWhite`).

4.1 Experimental Results

We generated approximately 12,500 random genotypes and 25,000 random mutation events on each embryogeny, using the procedure explained in Section 3.6. For the sake of simplicity, in this article we will only focus on the number-of-cells trait, the same trait used before in Section 3. Readers interested in applying this framework to more complex embryogenies and traits should refer to our previous studies on this topic [20, 21].

Although the two embryogenies share lots of similar properties—they are both based on genetic programming, and both employ the same mutation operator—they have in fact very different growth dynamics. This can readily be observed in the sample growth curves displayed in Figure 12. The CA genotypes have a much higher growth rate, and reach maturation much faster than the GB genotypes. Many of the CA genotypes in our sample set grow tissues to the maximum allowable size (36 cells) in only three or four time steps. The CA embryogeny is also much more constrained in its development, with regard to the range of phenotypes it may reach. While the GB embryogeny may easily generate phenotypes with any number of cells, the CA embryogeny tends to generate tissues mainly with 1 or 36 cells, and not many outside this range (Figure 13). This is also evident when we compare the changes created by mutations in Figure 12.

The difference in dynamics becomes more evident when we consider the ontogenetic distributions of the embryogenies. We have already shown them before with density plots in Figure 9; here we will show them in a more explicit form by using a scatterplot matrix (Figure 14). The first obvious difference is in the range of the parameters α, β, and K. Because growth with the CA embryogeny is much faster—both in rate and in time—its distribution has a much smaller β range (0–6) but a much bigger K range (0–12) than the GB embryogeny ($\beta = 0$–80, $K = 0, 2$). But because the CA embryogeny is much more restrained in the dynamics it generates, the parameter values are not very spread apart, and they tend to be lumped together around a few values. This is particularly obvious with the parameter K.

How do these differences translate into quantitative results? Judging from the discussion above, we can naturally expect the CA embryogeny, being much more constrained than the GB embryogeny, to have a much smaller variance in the ontogenetic distribution. Indeed, this is what happens,

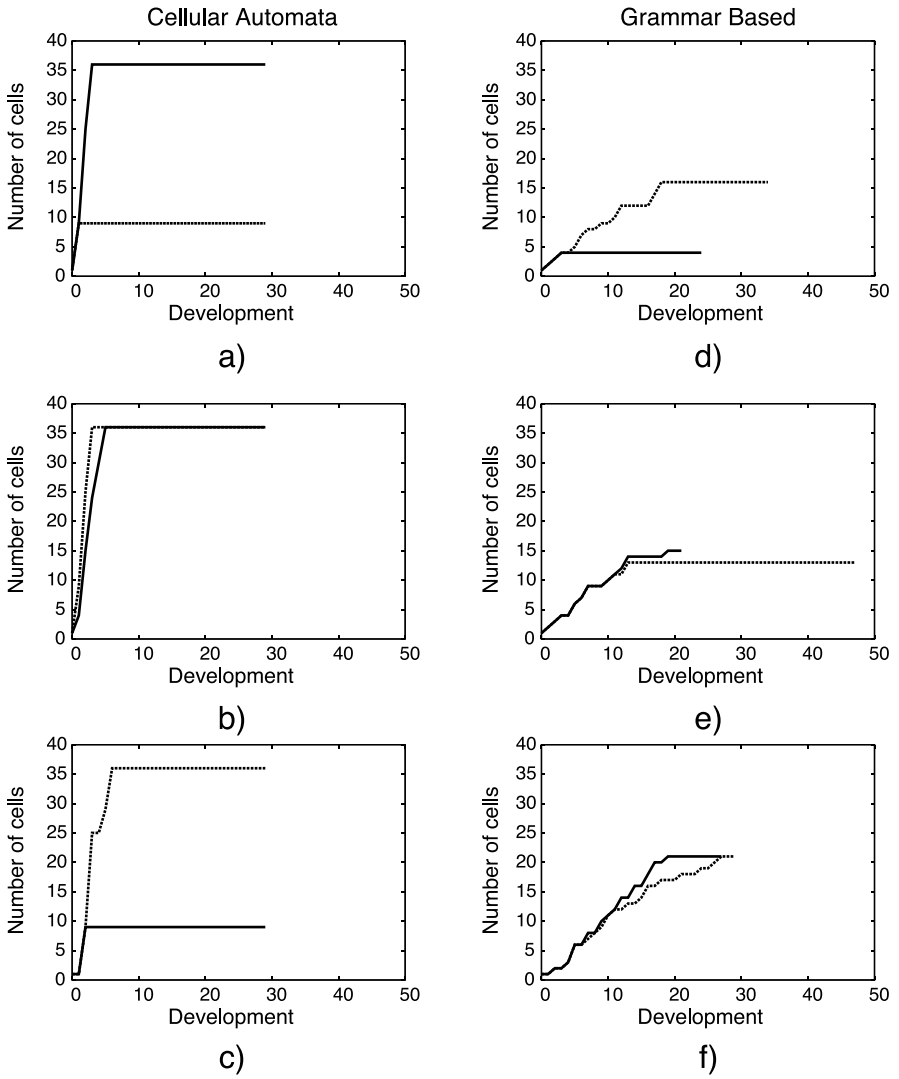

Figure 12. Sample growth curves taken from our mutation experiments. On each graph, the solid line represents the genotype generated at random, while the broken line represent the mutated genotype. (a) to (c) are from the CA embryogeny; (d) to (f) from the GB embryogeny.

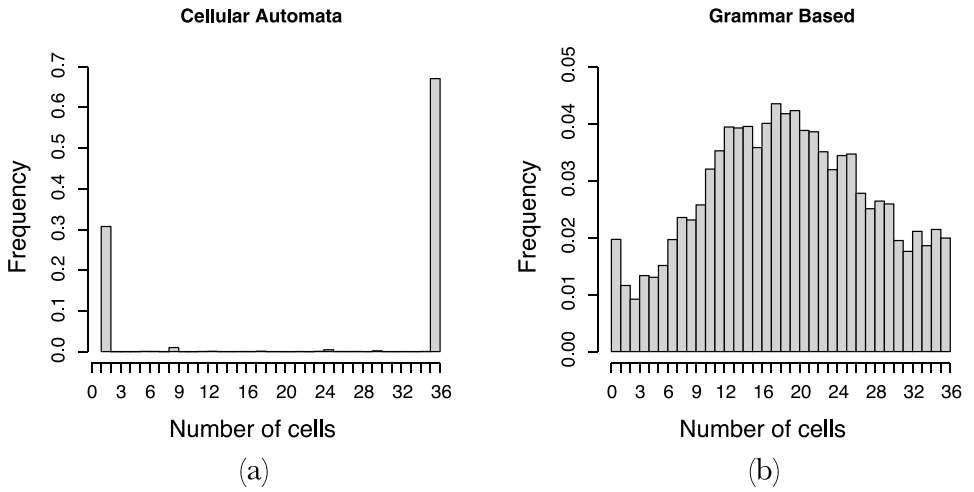

Figure 13. Histograms of the number of cells in the final phenotypes, for each embryogeny. Each bar represents the frequency of final phenotypes with the number of cells specified on the horizontal axis. Note the striking differences between the distributions.

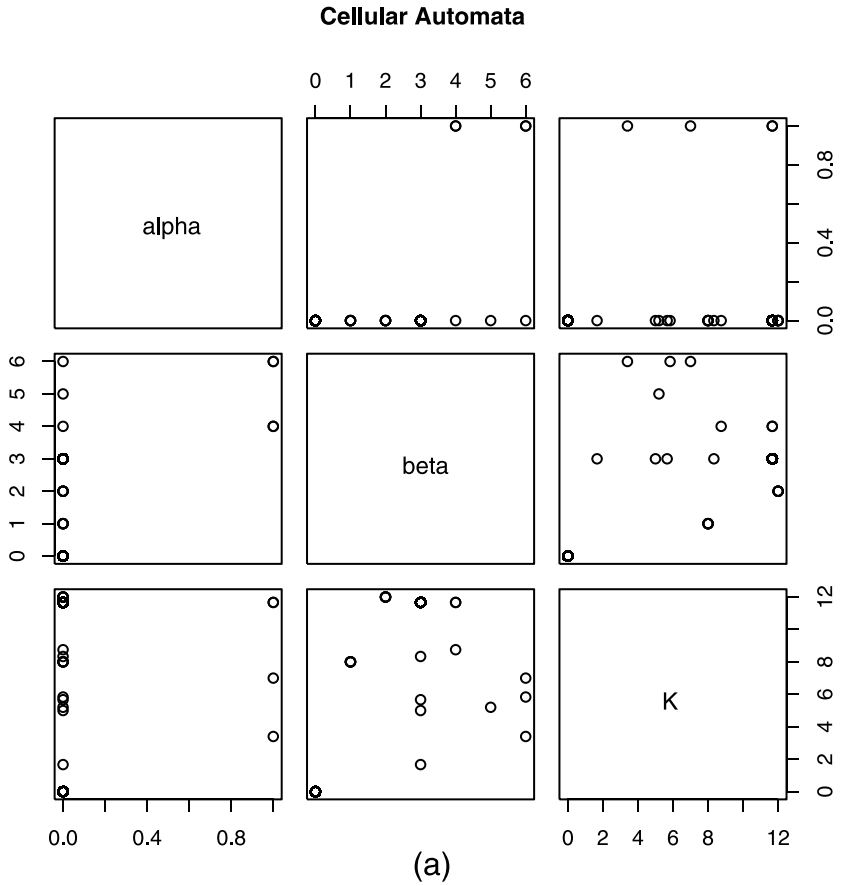

Figure 14. Ontogenetic distribution of the number of cells trait (scatterplot matrix). (a) CA embryogeny; (b) GB embryogeny.

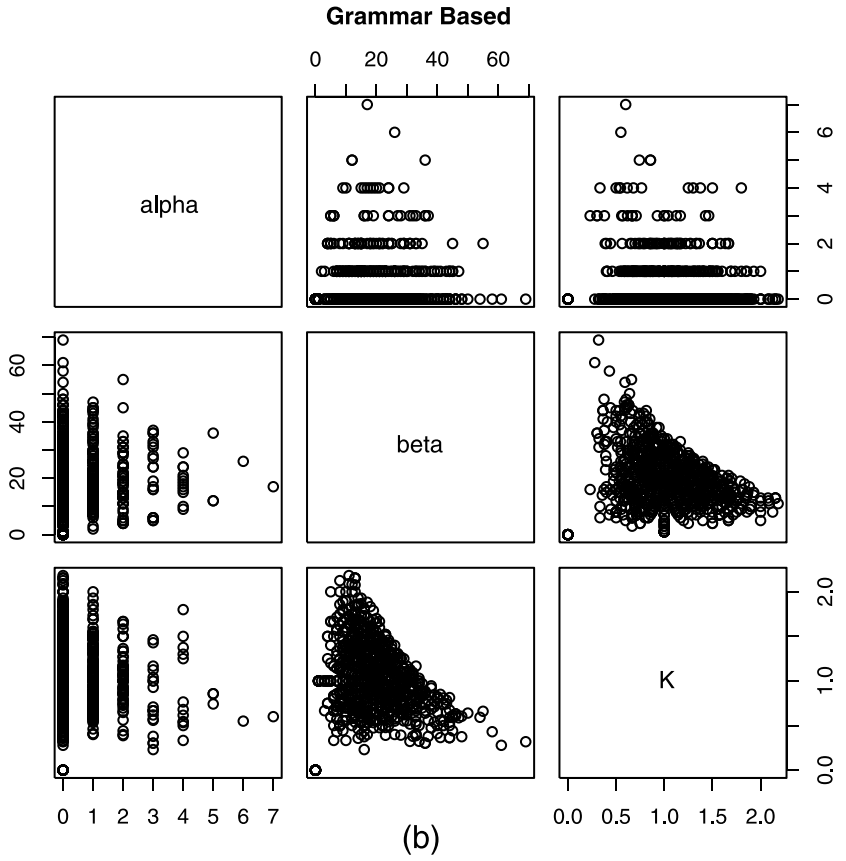

Figure 14. (*continued*)

with the GB embryogeny having a generalized variance 450 times greater than the CA embryogeny (CA = 1.82×10^{-2}, GB = $8.91 \times 10^{+0}$, Table 1). The remaining statistics are shown in Tables 1 and 2.

Results on the mutations and the heterochronic distribution are similar to those for the ontogenetic distribution (Figure 15, scatterplot matrix; Figure 7, density plot). The CA embryogeny is again more constrained than the GB embryogeny, although this time the differences are not as large as in the ontogenetic distribution. When compared, the ranges of the distribution are seen to have similar proportions (with the CA having smaller α' and β' ranges, and a much bigger K' range), but once again the CA genotypes tend to be lumped together on a few different values. The generalized variance of the heterochronic distribution, therefore, as expected, is bigger in the GB embryogeny—around 1.43 times bigger (CA = 7.6×10^{-2}, GB = 1.09×10^{-1}).

Neutral probabilities and heterochronic probabilities tend to be very similar between the embryogenies, so we will not discuss them further. The NH probabilities, however, are rather different. Out of the 25,000 mutation events, approximately 0.01% of the mutations were at the same time neutral and heterochronic in the CA embryogeny, and almost 0.07% in the GB embryogeny. This is a rather surprising result, because it shows that both embryogenies allow for mutations that are at the same time neutral and heterochronic, although only with a small probability. The shape of the NH distribution gives a better indication of how much the embryogenies allow the heterochronic parameters to change without affecting the final phenotype (Figure 16). The NH distributions overall have a smaller range of values than the parent heterochronic distribution—for instance, in the GB embryogeny, the variable β' ranges over $[-35, 35]$ in the heterochronic distribution, compared to

Table 1. Experimental results (distributions).

	Cellular automata	Grammar-based
	Ontogenetic distribution	
$\hat{\mu}(\mathcal{O}_{E,T})$	$\begin{bmatrix} 6.25 \times 10^{-3} \\ 2.07 \times 10^{+0} \\ 7.99 \times 10^{+0} \end{bmatrix}$	$\begin{bmatrix} 4.15 \times 10^{-1} \\ 1.90 \times 10^{+1} \\ 1.01 \times 10^{+0} \end{bmatrix}$
$\hat{\Sigma}(\mathcal{O}_{E,T})$	$\begin{bmatrix} 6.35 \times 10^{-3} & 1.32 \times 10^{-2} & 1.16 \times 10^{-2} \\ 1.32 \times 10^{-2} & 1.98 \times 10^{+0} & 7.34 \times 10^{+0} \\ 1.16 \times 10^{-2} & 7.34 \times 10^{+0} & 2.88 \times 10^{+1} \end{bmatrix}$	$\begin{bmatrix} 6.66 \times 10^{-1} & 4.23 \times 10^{-1} & -2.05 \times 10^{-2} \\ 4.23 \times 10^{-1} & 9.63 \times 10^{+1} & 7.28 \times 10^{-1} \\ -2.05 \times 10^{-2} & -7.28 \times 10^{-1} & 1.45 \times 10^{-1} \end{bmatrix}$
$\lvert\hat{\Sigma}(\mathcal{O}_{E,T})\rvert$	1.82×10^{-2}	$8.91 \times 10^{+0}$
	Heterochronic distribution	
$\hat{\mu}(\mathcal{H}_{E,T})$	$\begin{bmatrix} 0.00 \times 10^{+0} \\ 2.49 \times 10^{-3} \\ 1.00 \times 10^{-2} \end{bmatrix}$	$\begin{bmatrix} 3.37 \times 10^{-5} \\ 4.84 \times 10^{-2} \\ 1.02 \times 10^{-3} \end{bmatrix}$
$\hat{\Sigma}(\mathcal{H}_{E,T})$	$\begin{bmatrix} 1.02 \times 10^{-2} & 2.61 \times 10^{-2} & 3.75 \times 10^{-2} \\ 2.61 \times 10^{-2} & 2.86 \times 10^{+0} & 1.04 \times 10^{+1} \\ 3.75 \times 10^{-2} & 1.04 \times 10^{+1} & 4.08 \times 10^{+1} \end{bmatrix}$	$\begin{bmatrix} 1.10 \times 10^{-1} & 2.36 \times 10^{-1} & 9.24 \times 10^{-3} \\ 2.36 \times 10^{-1} & 2.97 \times 10^{+1} & -2.88 \times 10^{-1} \\ 9.24 \times 10^{-3} & -2.88 \times 10^{-1} & 3.79 \times 10^{-2} \end{bmatrix}$
$\lvert\hat{\Sigma}(\mathcal{H}_{E,T})\rvert$	7.6×10^{-2}	1.09×10^{-1}
	Neutral distribution	
$\hat{\mu}(\mathcal{NH}_{E,T})$	$\begin{bmatrix} 0.00 \times 10^{+0} \\ 6.02 \times 10^{-4} \\ -2.00 \times 10^{-3} \end{bmatrix}$	$\begin{bmatrix} 2.97 \times 10^{-4} \\ 8.85 \times 10^{-3} \\ -4.40 \times 10^{-4} \end{bmatrix}$
$\hat{\Sigma}(\mathcal{NH}_{E,T})$	$\begin{bmatrix} 3.37 \times 10^{-3} & 3.13 \times 10^{-3} & 5.62 \times 10^{-4} \\ 3.13 \times 10^{-3} & 1.22 \times 10^{-2} & -1.78 \times 10^{-2} \\ 5.62 \times 10^{-4} & -1.78 \times 10^{-2} & 3.84 \times 10^{-2} \end{bmatrix}$	$\begin{bmatrix} 1.25 \times 10^{-3} & 2.94 \times 10^{-4} & 5.87 \times 10^{-5} \\ 2.94 \times 10^{-4} & 1.63 \times 10^{-1} & -7.56 \times 10^{-3} \\ 5.87 \times 10^{-5} & -7.56 \times 10^{-3} & 4.33 \times 10^{-4} \end{bmatrix}$
$\lvert\hat{\Sigma}(\mathcal{NH}_{E,T})\rvert$	6.27×10^{-8}	1.60×10^{-8}

Table 2. Experimental results (probability measures).

	Cellular automata	Grammar-based
$P(H_{E,T})$	3.3×10^{-1}	3.6×10^{-1}
$P(N_{E,T})$	6.7×10^{-1}	6.3×10^{-1}
$P(NH_{E,T})$	1.8×10^{-3}	6.98×10^{-3}

[−15, 10] in the NH distribution. The NH distribution also exhibits a stronger negative correlation between the parameters β′ and K'. This is expected, because changes in one parameter must compensate changes in the other, in order for the mutation to remain neutral. For instance, if β increases, K must decrease, or otherwise the trait value will be larger in the mutated genotype. Summarizing, the GB embryogeny has a greater probability of generating NH mutations, but when they occur, the resulting changes are smaller than with the CA embryogeny. This is reflected in the generalized variance of the two embryogenies: The CA embryogeny has a generalized variance almost four times greater (CA = 6.27×10^{-8}, GB = 1.60×10^{-8}).

How are the embryogenies able to generate NH mutations? As we will see, the main mechanism behind NH mutations, in both embryogenies, is abstraction. Figures 17 and 18 show two examples of

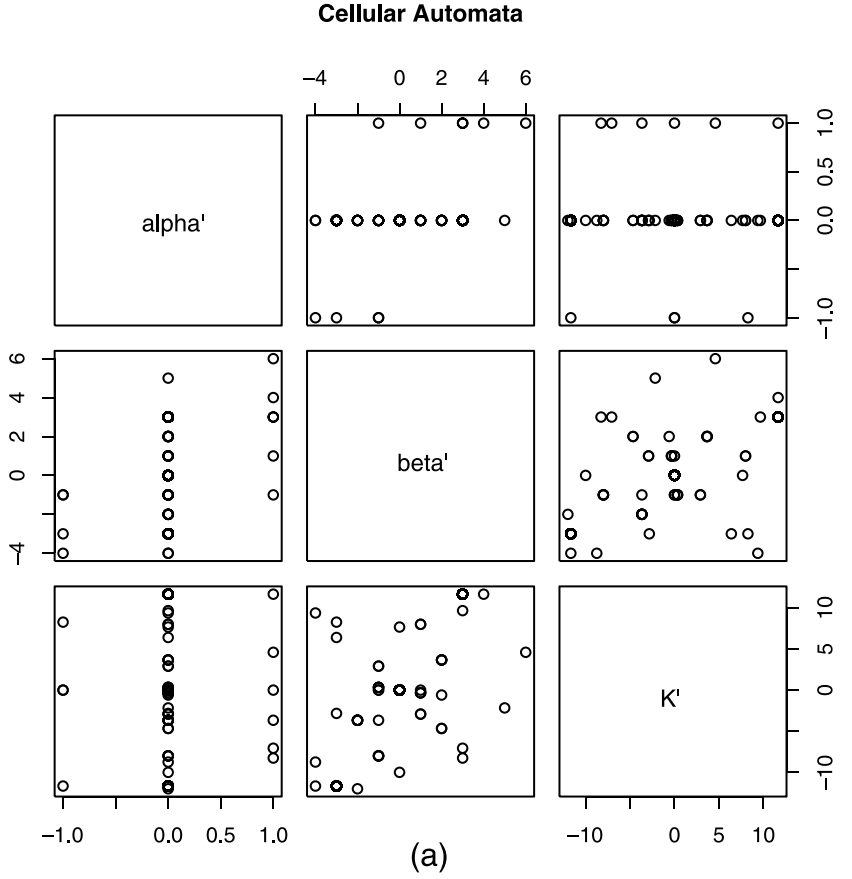

Figure 15. Heterochronic distribution of the number-of-cells trait (scatterplot matrix). (a) CA embryogeny; (b) GB embryogeny.

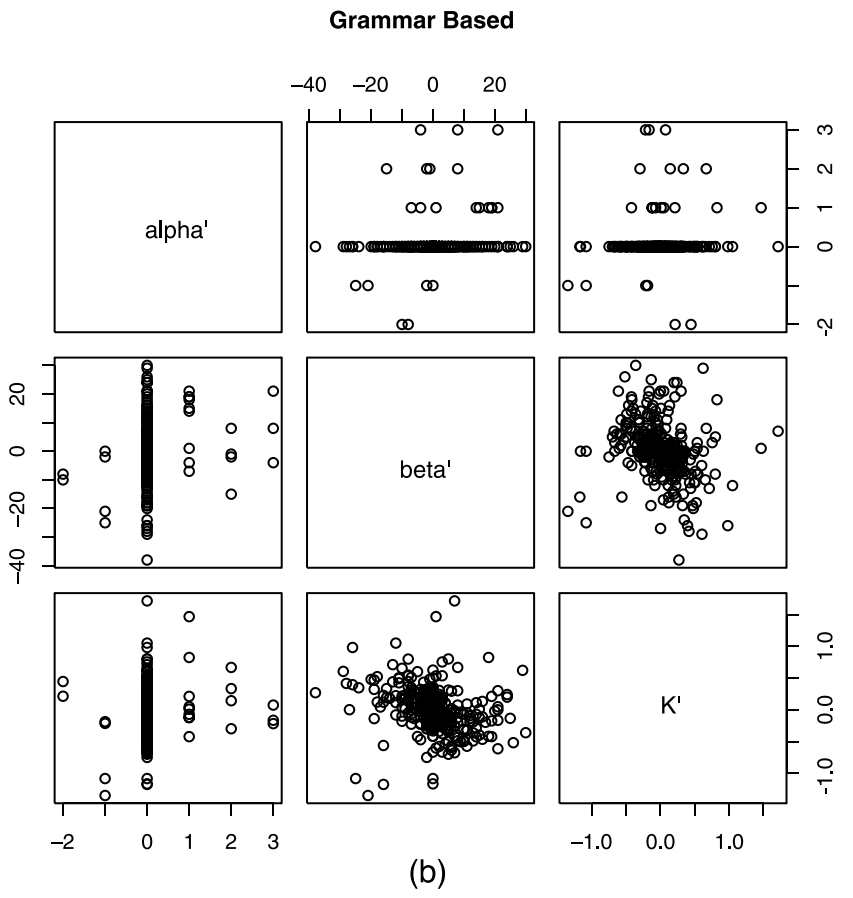

Figure 15. (*continued*)

NH mutations taken from our experiments, one for each embryogeny. In the CA example (Figure 17), the mutation causes the growth to stall in the right side of the grid. Eventually, at time $t = 4$, a rule is activated on both g_1 and g_2 that makes them reach the same state. This rule was unchanged by mutation, so it is the same for both genotypes. However, the configuration part of the rule is general enough, and matches the configurations on both genotypes at time $t = 3$. After this rule is activated, both genotypes reach the same final state, and further development will be similar from then on.

In the GB embryogeny example (Figure 18), the difference occurs at the top right of the grid, and starts at $t = 11$. It is not as pronounced as in the CA embryogeny, but the phenotypic trajectories stay apart for a much longer period. The mechanism responsible for the NH mutation is similar to that in the CA embryogeny. In this case, what is happening is that the latest developmental stages in both genotypes are directed by an ADF call. The ADF tree itself was not changed by mutation, but some of the steps before its call were altered. When the ADF tree is called, in both genotypes it rewrites most of the changes caused by the early steps. The end result is that even when these early steps are changed, the ADF call will make development converge to the same state.

Summarizing, the embryogenies show very different behavior, although they are based on very similar assumptions. Overall, the GB embryogeny is much more flexible than the CA embryogeny, not only in the range of phenotypes it may generate, but also in the dynamics of the developmental process and in the behavior of the mutation operator. This became readily evident when we applied the framework described in this article. Both embryogenies allowed to some extent for mutations that changed the growth process without changing the final phenotype, although only with a very

small probability. This was shown to be related to the ability of the embryogenies to have abstract rules and procedures.

5 Conclusion

In this article we have proposed a new approach for analyzing and comparing AEs, grounded on the concept of heterochrony. We examined different meanings of heterochrony in biology and artificial embryogeny, and argued that the most encompassing definition—the one used in evolutionary biology—is the most suitable for this field. We adapted the most popular biological framework for studying heterochrony—Alberch et al.'s heterochrony framework—to our artificial systems and proposed several ways to measure heterochrony in them. These measures are general enough to fit all AEs, and can be calculated empirically. They allow us to compare different embryogenies in a rigorous manner. To illustrate their use, we have computed the measures for two different embryogenies—an embryogeny based on cellular automata (the CA embryogeny), and another based on simple growth commands (the GB embryogeny). The CA embryogeny turned out to be much more restrained in its development than the GB one, and that was reflected in our quantitative measures.

One particular interesting result concerning our embryogenies is that both of them allow to a certain extent—albeit a small one—mutations that change the growth process without changing the final phenotype. In both embryogenies this is achieved through their abstraction mechanisms: rules

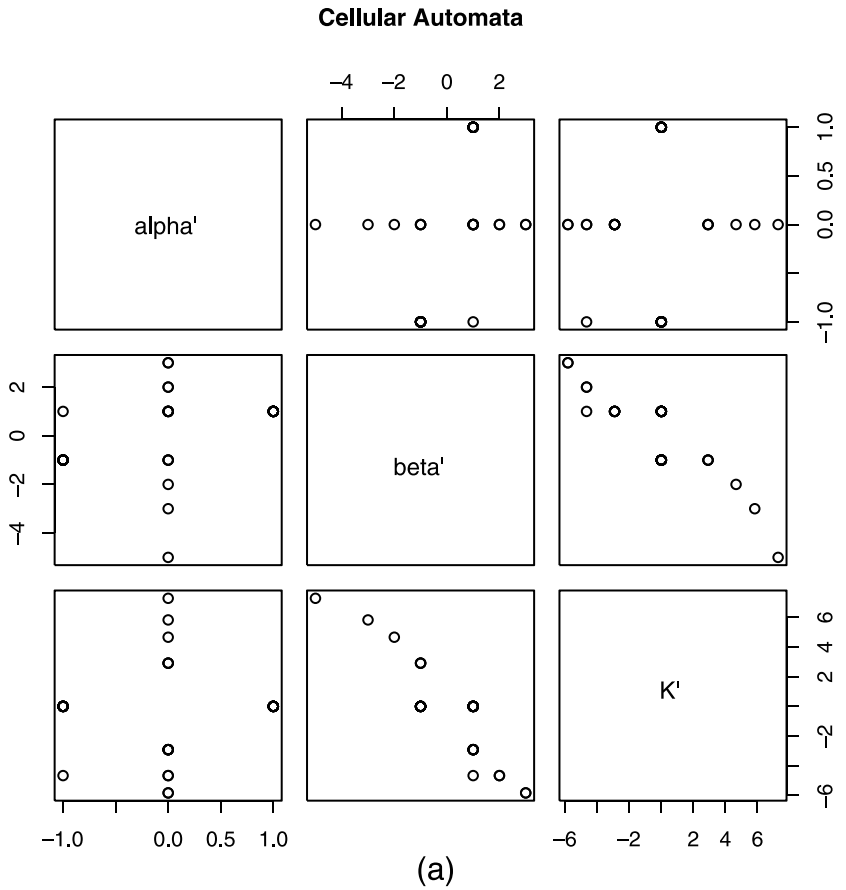

Figure 16. NH distribution of the number-of-cells trait (scatterplot matrix). (a) CA embryogeny; (b) GB embryogeny.

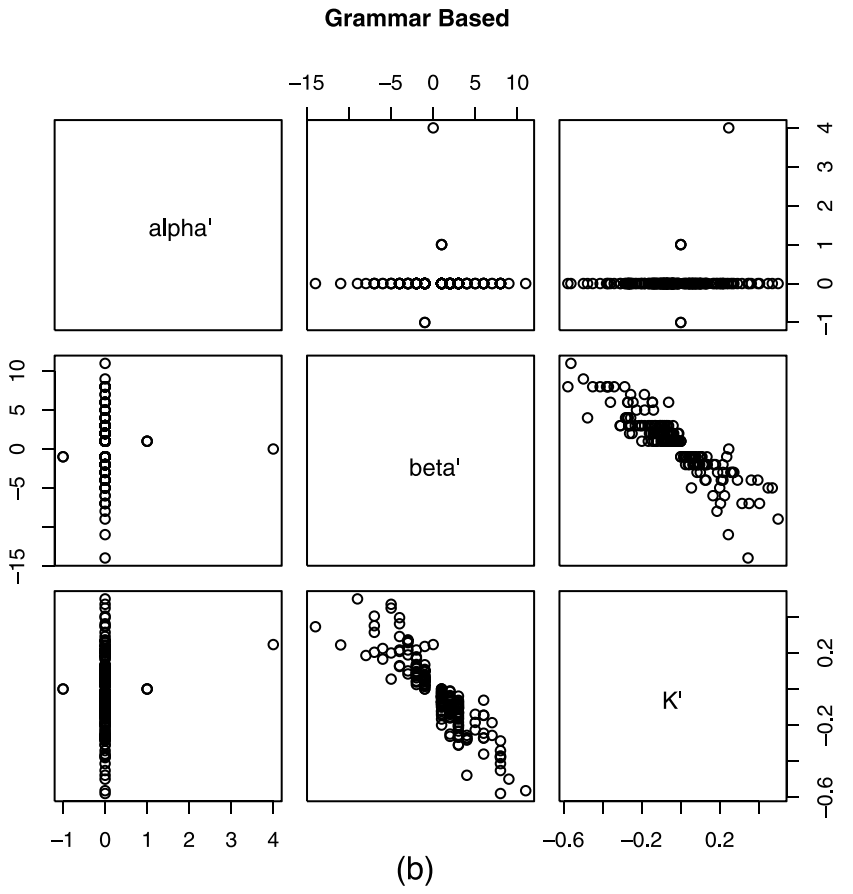

Figure 16. (*continued*)

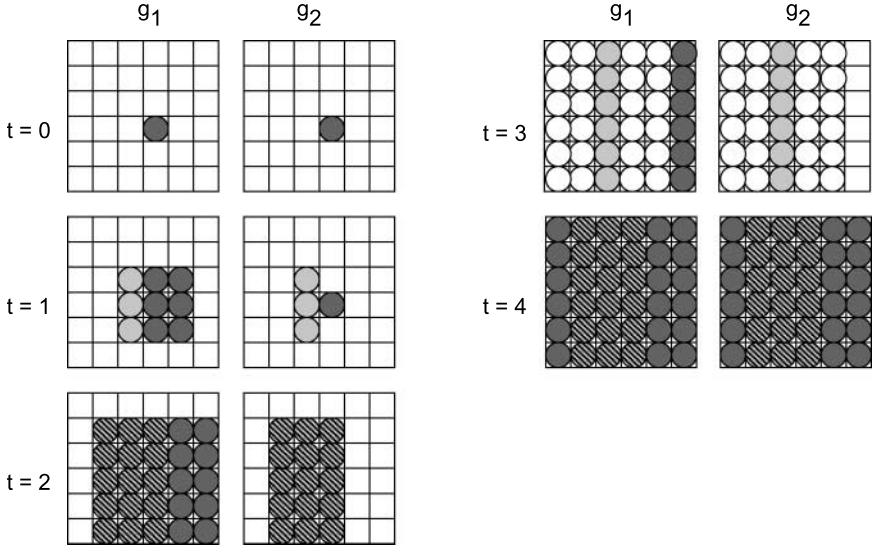

Figure 17. An example of a neutral and heterochronic mutation in the CA embryogeny. g_1 refers to the original genotype, and g_2 to the mutated one. t refers to the developmental time. Time flows from top to bottom, left to right.

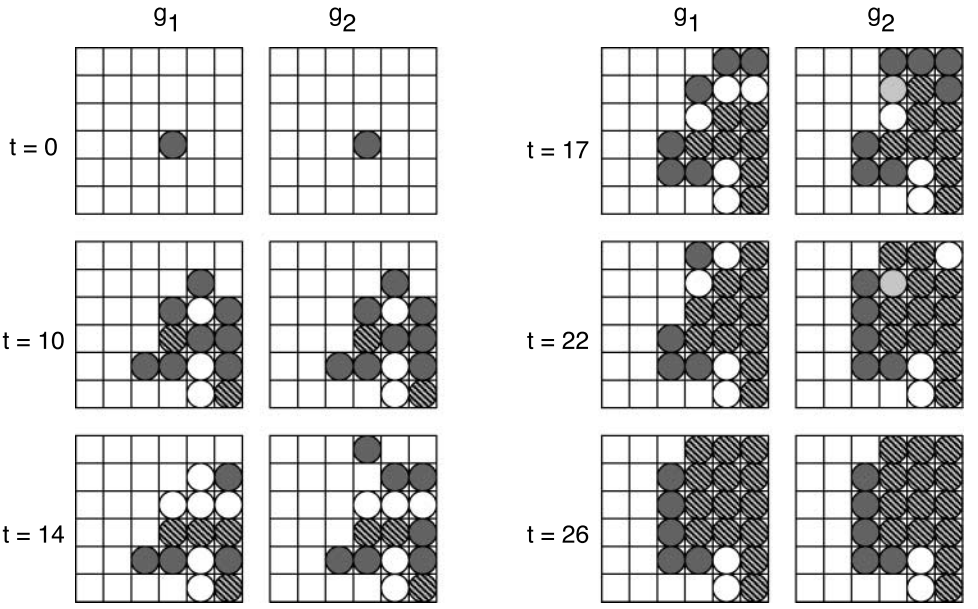

Figure 18. An example of neutral and heterochronic mutation in the GB embryogeny. The labels are similar to Figure 17.

(in the CA case), or procedures (in the GB case) that are unchanged during mutation and are general enough to bring development back to the same phenotype. This suggests two things: first, that most AEs may in fact be able to generate neutral and heterochronic mutations, that is, that this is a general property of development; second, that the extent to which this is allowed (or, using the terminology in this article, the generalized variance of the NH distribution) should be positively related to their abstraction abilities. In order to verify this, however, we will need to conduct similar analyses on a wider range of embryogenies.

We hope that the contributions in this article will allow the field of AE to achieve a more rigorous state. For this to be of any use, however, it is important that other researchers in the community readily adopt these methods, and apply them to their own embryogenies. Only with a wider range of experiments will we be able to start drawing more general conclusions. This will hopefully allow us to move over from results that only apply to specific implementations to a real, generalized understanding of how AEs work.

References

1. Alberch, P., Gould, S. J., Oster, G. F., & Wake, D. B. (1979). Size and shape in ontogeny and phylogeny. *Paleobiology*, *5*(3), 296–317.

2. Belew, R. K., & Kammeyer, T. E. (1993). Evolving aesthetic sorting networks using developmental grammars. In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.

3. Bentley, P. J., & Kumar, S. (1999). Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, & R. E. Smith (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-1999)* (pp. 35–43). San Mateo, CA: Morgan Kaufmann.

4. Boers, E. J., & Kuiper, H. (1992). *Biological metaphors and the design of modular artificial neural networks*. Master's thesis, Departments of Computer Science and Experimental and Theoretical Psychology, Leiden University, The Netherlands.

5. Bongard, J. (2002). Evolving modular genetic regulatory networks. In *Proceedings of the IEEE 2002 Congress on Evolutionary Computation* (pp. 1872–1877). Piscataway, NJ: IEEE Press.

6. Creighton, G. K., & Strauss, R. E. (1986). Comparative patterns of growth and development in cricetine rodents and the evolution of ontogeny. *Evolution*, *40*(1), 94–106.

7. Fink, W. L. (1982). The conceptual relationship between ontogeny and phylogeny. *Paleobiology*, *8*, 254–264.

8. Gruau, F. (1994). *Neural network synthesis using cellular encoding and the genetic algorithm*. Ph.D. thesis, Laboratoire de la Informatique du Parallelisme, Ecole Normale Superieure de Lyon.

9. Hall, B. K. (1992). *Evolutionary developmental biology* (2nd ed.). Dordrecht, The Netherlands: Kluwer Academic.

10. Hornby, G. S. (2005). Measuring, enabling and comparing modularity, regularity and hierarchy in evolutionary design. In H.-G. Beyer (Ed.), *GECCO '05: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. Berlin: Springer-Verlag.

11. Hornby, G. S., & Pollack, J. B. (2002). Creating high level components with a generative representation for body-brain evolution. *Artificial Life*, *8*(3), 223–246.

12. Johnson, R. A., & Wichern, D. W. (1982). *Applied multivariate statistical analysis* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

13. Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, *4*, 461–476.

14. Klingenberg, C. P. (1997). Heterochrony and allometry: The analysis of evolutionary change in ontogeny. *Biological Reviews*, *73*(1), 79–123.

15. Knowles, J. D., & Watson, R. A. (2002). On the utility of redundant encodings in mutation-based evolutionary search. In *Parallel Problem Solving from Nature—PPSN VII: 7th International Conference Proceedings* (pp. 88–98). Berlin: Springer-Verlag.

16. Komosinski, M. (2000). The world of Framsticks: Simulation, evolution, interaction. In J.-C. Heudin (Ed.), *Virtual worlds: Lecture notes in artificial intelligence 1834* (pp. 214–224). Berlin: Springer-Verlag.

17. Komosinski, M., Koczyk, G., & Kubiak, M. (2001). On estimating similarity of artificial and real organisms. *Theory in Biosciences*, *120*(3–4), 271–286.

18. Komosinski, M., & Rotaru-Varga, A. (2001). Comparison of different genotype encodings for simulated 3D agents. *Artificial Life*, *7*(4), 395–418.

19. Luke, S., Panait, L., Balan, G., Paus, S., Skolicki, Z., Popovici, E., Harrison, J., Bassett, J., Hubley, R., & Chircop, A. ECJ 16: A Java-based evolutionary computation research system. http://cs.gmu.edu/~eclab/projects/ecj/ (accessed September 2007).

20. Matos, A., Suzuki, R., & Arita, T. (2005). Evolution of development and heterochrony in artificial neural networks. In *Proceedings of the Tenth International Symposium on Artificial Life and Robotics* (pp. 635–638). Beppu, Oita, Japan: AROB Journal Office.

21. Matos, A., Suzuki, R., & Arita, T. (2006). Heterochrony and evolvability in neural network development. In *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems* (pp. 63–69). Cambridge, MA: MIT Press.

22. Nolfi, S., & Parisi, D. (1991). *Growing neural networks* (Tech. Rep. PCIA-91-15). Rome: Institute of Psychology, C.N.R.

23. Pigliucci, M. (1984). Ontogenic phenotypic plasticity during the reproductive phase in *Arabidopsis thaliana* (Brassicaceae). *American Journal of Botany*, *84*(7), 887–895.

24. R Development Core Team (2005). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing, http://www.R-project.org.

25. Raff, R. A., & Wray, G. A. (1989). Heterochrony: Developmental mechanisms and evolutionary results. *Journal of Evolutionary Biology*, *2*, 409–434.

26. Reilly, S. M., Wiley, E., & Meinhardt, D. J. (1997). An integrative approach to heterochrony: The distinction between interspecific and intraspecific phenomena. *Biological Journal of the Linnean Society*, *60*, 119–143.

27. Shipman, R., Shackleton, M., & Harvey, I. (2000). The use of neutral genotype-phenotype mappings for improved evolutionary search. *BT Technology Journal*, *18*(4), 2004.

28. Stanley, K. O., & Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, *9*(2), 93–130.

29. Toussaint, M. (2003). On the evolution of phenotypic exploration distributions. In K. A. D. Jong, R. Poli, & J. Rowe (Eds.), *Foundations of Genetic Algorithms, vol. 7 (FOGA VII)* (pp. 169–182). San Mateo, CA: Morgan Kaufmann.

30. von Bertalanffy, K. L. (1938). A quantitative theory of organic growth. *Human Biology*, *10*, 181–213.

31. Vose, M. D. (1999). *The simple genetic algorithm*. Cambridge, MA: MIT Press.

32. Wand, M., & Ripley, B. (2004). *KernSmooth: Functions for kernel smoothing for Wand and Jones* (R package version 2.22-15). http://www.maths.unsw.edu.au/~wand (accessed September 2007).

33. Yu, T., & Miller, J. F. (2006). Through the interaction of neutral and adaptive mutations, evolutionary search finds a way. *Artificial Life*, *12*, 525–551.

Appendix: The Embryogenies

In this appendix we give additional details on the embryogenies described in Section 4. This should allow interested researchers to replicate the results published in this article, if deemed necessary.

We have conducted our experiments with the Evolutionary Computation in Java package (ECJ) [19], a free EA program. For designing the embryogenies, we have used ECJ's built-in support for strongly typed genetic programming, but all other remaining settings were left to their default values. In ECJ, strongly typed genetic programming is implemented by defining a set of types, which are basically labels attached to nodes in the genetic programming tree. The types are used to indicate the return type of a given node, and also to specify the type used by its children; this information is used by the tree creation and mutation operators, to ensure that only valid trees are generated.

The types we have used in our embryogenies are as follows: `boolean` specifies a Boolean value, `state` specifies a state to be checked in the neighborhood (CA embryogeny), and `command` specifies a cell command to be executed (GB embryogeny). The list of all commands used in the CA embryogeny is shown in Table 3, and the one for the GB embryogeny in Table 4.

The CA genotypes have a rather rigid structure. The top node is always a `Rules` element with four children. Each child represents one of the four possible outcomes in order: red, white, blue, or gray (these nodes are not included in the table). Below each of the four outcomes are test subtrees, with a

Table 3. CA embryogeny commands.

Command	Return type	Children	Children's type	Description
`Configuration`	`boolean`	9	`state`	A specific configuration.
`R`	`state`	0	—	Red state.
`W`	`state`	0	—	White state.
`B`	`state`	0	—	Blue state.
`G`	`state`	0	—	Gray state.
`?`	`state`	0	—	Don't care.
`And`	`boolean`	2	`boolean`	Logical AND.
`Or`	`boolean`	2	`boolean`	Logical OR.
`Not`	`boolean`	1	`boolean`	Logical NOT.
`Any`*X*`?`	`boolean`	0	—	True if there is at least one of the specified colors in the neighborhood. *X* can have one of the following values: {`B`,`R`,`W`,`G`}.

Table 4. GB embryogeny commands.

Command	Return type	Children	Children's type	Description
Split*X*	command	2	command	Creates a new cell on the specified cardinal position. *X* can have one of the following values: {N,NE,E,SE,S,SW,W,NW}. Each cell will execute a different branch of the tree after the split.
If	command	3	boolean,command, command	Checks a condition. Executes the first code branch if the condition is true. Otherwise executes the second branch.
CX	boolean	0	—	Checks the state of a neighboring cell. *C* can have one of the following values: {R, B, W, G, Boundary}. *X* is as in the split command above.
X	command	0	—	Terminates development.
Call	command	0	—	Calls the ADF tree. (Will allow for at most three levels of recursion.)
Set*X*	command	1	command	Sets the cell's color. *X* can have one of the following values: {R, B, W, G}.
And	boolean	2	boolean	Logical AND.
Or	boolean	2	boolean	Logical OR.
Not	boolean	1	boolean	Logical NOT.

return type of boolean. During each iteration step, the embryogeny tries to match the configuration on each test subtree in order, and the first one that matches it is set to that specific outcome.

In contrast, the GB embryogeny has a looser structure. The top node of the tree is always of type command. This corresponds to the instruction that is going to be executed by the single cell in the grid when development starts. Development follows by executing the instructions in the tree, as explained before in Section 4. We believe that the descriptions in Table 4 should be self-explanatory.

The cells during development execute their instructions concurrently, and this requires some additional explanations. To simulate concurrency, we employ a scheme similar to the one used in the cellular encoding embryogeny [8]. During growth, the embryogeny maintains a list of all the active cells. On each iteration, the embryogeny iterates over all the cells in this list, and executes one single instruction (the one that the cell is currently pointing to) for each cell. Cells that finish their development are removed from this list, and new cells that are created (by split commands) are added to it. Development finishes when this list becomes empty. The cells are always iterated following the same order, with older cells being executed first, and new cells being added to the end of the list.