Emergent Diversity in an Open-Ended Evolving Virtual Community

Jose D. Fernández*,**
University of Málaga

Daniel Lobo**
University of Málaga

Gema M. Martín**
University of Málaga

René Doursat†
Ecole Polytechnique and CNRS

Francisco J. Vico**
University of Málaga

Abstract Understanding the dynamics of biodiversity has become an important line of research in theoretical ecology and, in particular, conservation biology. However, studying the evolution of ecological communities under traditional modeling approaches based on differential calculus requires species' characteristics to be predefined, which limits the generality of the results. An alternative but less standardized methodology relies on intensive computer simulation of evolving communities made of simple, explicitly described individuals. We study here the formation, evolution, and diversity dynamics of a community of virtual plants with a novel individual-centered model involving three different scales: the genetic, the developmental, and the physiological scales. It constitutes an original attempt at combining development, evolution, and population dynamics (based on multi-agent interactions) into one comprehensive, yet simple model. In this world, we observe that our simulated plants evolve increasingly elaborate canopies, which are capable of intercepting ever greater amounts of light. Generated morphologies vary from the simplest one-branch structure of promoter plants to a complex arborization of several hundred thousand branches in highly evolved variants. On the population scale, the heterogeneous spatial structuration of the plant community at each generation depends solely on the evolution of its component plants. Using this virtual data, the morphologies and the dynamics of diversity production were analyzed by various statistical methods, based on genotypic and phenotypic distance metrics. The results demonstrate that diversity can spontaneously emerge in a community of mutually interacting individuals under the influence of specific environmental conditions.

1 Introduction

1.1 The Rise of Virtual Biology

Understanding the detailed mechanisms and reconstructing the precise lineage of evolutionary biological processes constitute difficult problems for at least two reasons. First, available data (fossils,

 * Contact author.
** Research Group in Biomimetics, Department of Computer Science and Languages, University of Málaga, Severo Ochoa 4, 29590 Málaga, Spain. E-mail: josedavid@geb.uma.es (J.D.F.); dlobo@geb.uma.es (D.L.); gema@geb.uma.es (G.M.M.); fjvico@uma.es (F.J.V.)
 † Complex Systems Institute Paris Ile-de-France (ISC-PIF), CREA, Ecole Polytechnique and CNRS, 57-59, rue Lhomond, 75005 Paris, France. E-mail: rene.doursat@polytechnique.edu

DNA, extant species) is still scarce compared to the whole history of life on earth, since obviously only one instance of this history exists and the great majority of its content has been lost. Second, most significant evolutionary processes occur on the scale of millions of years, whereas modern experimental evolution in biology is limited to simple and fast-reproducing organisms that are restricted to adaptation in artificially altered environmental conditions. In fact, it can be asked whether biological experimental evolution is inherently more valuable than computer modeling and simulation, considering that the former is also significantly more costly than the latter [34]. Thus, evolution is being increasingly studied through alternative approaches involving the theoretical reconstruction and simulation of living systems at multiple scales: genetic, cellular, developmental, population, and ecosystem. In particular, computational models have the great benefit of potentially producing a large number of experiments that can condense long evolutionary periods into short computing time frames, while making vast collections of data available for the analysis and extraction of relevant properties [26, 28]. Moreover, increasing computer power and storage capacity provide the means to actually simulate explicative models of biological evolution.

Today, numerical experiments are commonplace in biological research. In the last 20 years, the field has moved from studying the evolution and population dynamics of simplified individuals [12, 38] to multi-species interactions and diversification [8, 19]. A great number of evolutionary computational models have been recently proposed, whether at the abstract level of large interacting populations [10, 48] or at the fine-grained level of multicellular growth [16]. Traditionally, models of population dynamics have been formulated as systems of differential equations [32], whether well mixed (ODEs) or spatially extended (PDEs), in which the established goal was to find analytical solutions. These approaches present a number of drawbacks, however, since their formulation and execution demand expert mathematical knowledge, yet they also need to remain simple enough to be mathematically tractable. For all these reasons, mathematical descriptions at the macroscopic level are generally unrealistic and difficult to extend with new features. On the other hand, *agent-based* models (ABMs) propose to describe communities in a bottom-up fashion through the properties of their individual members, the developmental and behavioral rules they obey, and their interactions with one another and the environment [20]. The collective behavior of the system, originating from complex multi-agent dynamics, is then obtained by computer simulation. Compared to analytical frameworks, two major properties are distinctive of the agent-based computational approach: *emergence* and *adaptation*. New properties at higher scales, which were not explicitly included in the model, appear in the system as the individuals interact. This makes it possible to describe and explain how certain properties observed in real communities relate to their lower-level features. Simultaneously, the community can also evolve and adapt to new environmental conditions, therefore broadening the scope of possible experiments and testable hypotheses.

1.2 A Generative Approach to Diversity

In this work, we focus on an especially puzzling aspect of evolution: *diversification*. We simulate a virtual community of plants to study the emergence and dynamics of genomic and phenotypic variation during evolution. To model plants, we use *rewriting systems*, a family of formal methods widely investigated in theoretical computer science and capable of encoding and generating complex structures. In this formal framework, parts of an initial object, generally strings of symbols, are iteratively replaced by other parts, generally longer string segments, following *rewrite rules*. Among the most popular and best-studied rewriting systems are *L-systems*. First proposed by Lindenmayer [27] (hence its name), L-systems are a class of formal grammars that proceed by parallel application of the rewrite rules to the current string. Parallel transformation (context-free or context-sensitive) makes L-systems especially suitable for the computational modeling of the development of unicellular colonies and multicellular organisms [21]. In particular, they have been extensively used in plant simulation [37], due to the ease of converting generated strings into 2D or 3D treelike structures by means of turtle geometry (e.g., in the Logo computer language) [29, 36]. Through the years, L-systems have found many applications in plant anatomy, physiology, and morphogenesis [2, 11, 18, 40]. Remarkably few of these efforts, however,

have examined plant generation on the *evolutionary* scale, nor do they use genetic algorithms. Yet L-systems provide a rather straightforward model suitable for evolutionary exploration: Just as in biological development, where complex organisms can be considered as emerging from the "interpretation" of one-dimensional strings (genomes), L-systems can encode complex structures through fairly simple sets of rules. Encoding large phenotypes with small genotypes via an intermediate developmental stage has been termed *indirect* or *implicit* encoding in the evolutionary computation community [41].

A first attempt at evolving L-systems was proposed by Koza [25]; it used genetic programming and represented the rewrite rules as labeled tree graphs. The goal of the search was to find specific rules that could generate a structure identical to a predefined one. Since then, other works based on evolutionary complex structures encoded by L-systems have been proposed. Jacob [21, 22] presented a variant of genetic programming to evolve context-free and context-sensitive L-systems that could generate plants similar to real ones. The fitness of individuals was based on the volume and number of blooms and leaves of the virtual plant during each iteration of the system. He also experimented with an ecosystem of different coevolving plant species [23]. Ochoa [35] designed a model for generating 2D plant morphologies from D0L systems (the simplest class of L-systems, where D stands for deterministic and 0 for context-free). It resembles the one that we propose here, although her work was more focused on exploring the model itself than on using it to investigate evolutionary dynamics. Another similar, albeit only briefly sketched, framework was proposed by Mock [31]. Ebner et al. [13, 15] presented a light-seeking 3D plant model based on L-systems. They also observed [14] the *Red Queen effect* under competitive simulations, that is, the fitness either remained constant or decreased while evolutionary progress was still going on. However, their simulation evolved to a dominant stable strategy with no further progress. Toussaint [43] defined a very similar model to study the role of neutral mutations in the evolvability of genetic representations, but it had additional layers of complexity in the genetic representation. Finally, Bornhofen and Lattaud [6, 7] evolved generic virtual plants including their full life cycle. Their simulations ended in ecosystems of tall plants with no phenotypic diversification. In [8], they concluded that the genetic search space of the D0L system was too limited, constraining evolutionary dynamics too severely and preventing it from evolving a wide variety of plant morphologies.

Here, in contrast to Bornhofen and Lattaud's conclusions, we present a morphological model based on a D0L system that shows a wide range of concurrent diversification in an open-ended evolutionary process (i.e., evolution without a definite maximum fitness, hence no optimal goal or solution to reach). Regarding the issue of genetic search space, an important aspect when designing evolutionary models of complex structures (and especially so when using an indirect encoding) concerns the proper choice of *genetic operators*. For its part, the present work largely relies on *gene duplication*. In biology, gene duplication (leading to the creation of paralogous genes) is a form of *silent* mutation in the sense that it is generally neutral with respect to selective pressure. Although in most cases copied genes could be considered "junk DNA," since they simply seem to increment redundancy, they provide in fact a fertile coding substrate for the advent of new proteins and new functions. In some models, gene duplication has been combined with traditional evolutionary operators, as in [44], which concluded that creating copies of key portions of the genome was useful because it could both retain their function and leave the copies open to incremental improvement. In the same spirit, we show here the performance of various gene duplication operators combined with usual point mutation operators (alteration, deletion, insertion), and how this can create initially non-disruptive, but later highly productive, genetic modifications.

The remainder of the article is organized as follows. Section 2 introduces our evolutionary model of virtual population, specifying mechanisms at the genetic level, the laws of interaction with the environment, and the mutation and selection rules. We then present the detailed results of a full set of numerical experiments in Section 3, followed by a discussion and our conclusions in Section 4.

2 An Evolutionary Model of Virtual Plant Population

Following standard practice in evolutionary computation studies, we present our model in four subsections: the genome and development of plant phenotypes (Section 2.1), the genetic operators

(Section 2.2), the environment and associated fitness function (Section 2.3), and the evolutionary scheme (Section 2.4). We sometimes use the term *individual* to refer to a virtual plant.

2.1 Genome and Development

As in other works in virtual botany, we choose a grammar-based indirect encoding and define plant growth as a rewriting process. More precisely, a plant develops according to a deterministic, context-free L-system, also called *bracketed D0L system*. In this type of abstract framework, gene regulation is greatly simplified, since all symbols are rewritten in parallel and the only remaining control parameter is the length of the derivation process, that is, the number of global rewriting stages, denoted n. Since our study is mainly concerned with the structural evolution of genomes, this parameter will be constant for all practical purposes and set to $n = 3$ in all individuals.

2.1.1 Genotype Representation

The D0L systems used in this article are defined as a triplet ($\{G, +, -, [,]\}$, G, $\{G \rightarrow T\}$), where there is only one nonterminal symbol, denoted G. Accordingly, there is a single production rule $G \rightarrow T$, whose right-hand side is a string T representing the *genome* of the plant, since it is the only part of the D0L system that changes from plant to plant. The genome T is composed of symbols from the alphabet $\{G, +, -, [,]\}$ and includes at least one instance of the symbol G, with the additional syntactic restriction that brackets must be balanced. Genetic expression and organism development in the proposed model consist of rewriting the axiom G by applying the rule $G \rightarrow T$ exactly $n = 3$ times, using the production rule encoded by the genome. In this way, the encoding method is rather straightforward, as first remarked in [35]: Since only one rewrite rule applies, its right-hand side codes fully, albeit indirectly, for the plant's structure.

The final string generated after three applications of the rule $G \rightarrow T$ is referred to as the *developed string*, and it is denoted $D(T)$ to highlight the deterministic nature of the system. The top row of Figure 1 illustrates this derivation process. Despite its simplicity compared to most works in L-system evolution, our encoding allows for the definition of robust genetic mutation operators (see Section 2.2).

2.1.2 Phenotype Development

The phenotype of an individual, denoted $P(D(T))$, is the complete matrix of pixels (bottom of Figure 1; see also a close-up view in Figure 4) produced from the developed string $D(T)$, also in deterministic fashion. To create the phenotype, the developed string is interpreted graphically through

$$G \quad \overset{G \rightarrow T}{\Rightarrow} \quad T = [+GG]G \quad \overset{G \rightarrow T}{\Rightarrow} \quad [+[+GG]G[+G \quad \overset{G \rightarrow T}{\Rightarrow} \quad D(T) = \quad [+[+[+GG]G[+GG]G][+GG]$$
$$G]G][+GG]G \qquad\qquad G[+[+GG]G[+GG]G][+GG]G$$
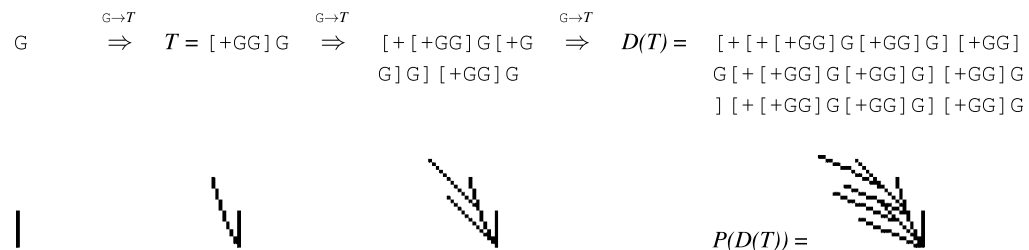$$][+[+GG]G[+GG]G][+GG]G$$

$$P(D(T)) =$$

Figure 1. Genotype-phenotype mapping. Going from a string genotype T to a graphical phenotype $P(D(T))$ via the developed string $D(T)$. Top: Genotype T, here [+GG] G, is interpreted as the right-hand side of the unique rule of a D0L system. Starting from axiom G, the rule $G \rightarrow T$ is applied three times to produce a string version of the phenotype, $D(T)$. Bottom: To obtain the final graphical phenotype $P(D(T))$, this string is then interpreted by a Logo-style turtle routine, where each symbol corresponds to a command: + and − change the turtle's direction (on the trigonometric circle, counterclockwise for positive angles), G moves forward by 10 units (drawing a branch over the pixels it covers), and brackets [and] respectively push and pop the turtle's stacked state (position *and* direction). Note the repetitive nature of $D(T)$. In this figure, a graphical equivalent is shown for each string in the rewriting process from G to $D(T)$, but only the last one corresponds to the phenotype of the plant, $P(D(T))$.

a Logo-style geometrical routine [1]. Initially, a graphic turtle pen points upwards and its memory stack is empty; then each symbol of $D(T)$ is executed as follows:

- + (−): The turtle changes its current direction by adding (subtracting) an angle of 22°.

- [: The turtle pushes (saves) onto the stack its current position and direction.

-]: The turtle pops (restores) from the stack the last saved position and direction.

- G: The turtle advances 10 units in its current direction, marking the pixels on the way in black (part of the plant) and the last pixel in red (representing a *leaf*; see Figure 4 and an explanation of its role in Section 2.3).

The value 22° of the angle parameter was chosen for its ability to generate a great variety of natural-looking plant forms (see, e.g., Figure 6). It belongs to an interval of angles that present a good compromise between too linear, broomlike shapes (at smaller angles) and too irregular, contorted shapes (at larger angles; Figure 2). Three examples of relatively simple phenotypes are displayed in Figure 4. Their genotypes are (from left to right): G[−−GG][G], G[G][−−G][++G], and G[G][−−G]++. The corresponding developed strings are not shown, as they are rather long and repetitive (which is often the case, even when the initial genotype is not).

2.1.3 Genotype-Phenotype Mapping

At this point, it is important to remark that closely similar genotypes can map to very different phenotypes (divergence) and, conversely, the exact same phenotype can be produced by very different genotypes (convergence). Divergence is illustrated in Figure 3, where genotypes G[G+[G][−G]] and G[G+[G][−−G]] differ only by one symbol, yet they produce widely dissimilar graphical forms. Convergence is due mostly (but not exclusively) to *redundancy* in genotypes and consequently in developed strings: For example, the sequence [G][G]G has the effect of drawing the same branch three times, as every closed bracket followed by an open bracket takes the graphical turtle back to its previous bifurcation point. This also means that the total number of branches of the graphical phenotype $P(D(T))$ is not equal to the number of instances of G in $D(T)$ (which is the cube of the number of instances of G in T). For example, for $T =$ G[−−GG][G] (left individual in Figure 4), the corresponding developed string $D(T)$ contains $4^3 = 64$ instances of G, whereas the corresponding phenotype $P(D(T))$ only has 33 distinct branches. We discuss again this issue of relative uncorrelatedness between genomic similarity and phenotypic similarity when defining distance measures in Section 3.2.3. The specific case of genotypic redundancy is addressed in the next section.

2.1.4 Reduction of Genotypes

The mapping from genotypes T to phenotypes $P(D(T))$ is many-to-one: there exist different genotypes that can be graphically translated into the exact same phenotype. Without loss of generality, since the mapping from T to $D(T)$ is one-to-one, let us consider only the transformation of $D(T)$ into $P(D(T))$.

   For example, the string $D(T) =$ G creates the simplest phenotype of all, consisting of only one vertical branch. This would also be the case of strings $D(T) =$ [[G]] (the additional brackets have no effect), $D(T) =$ [G][G]G (the same branch is drawn three times by the turtle), and $D(T) =$ [ ][+−G++] (no effect is produced by empty brackets, +− pairs, or any combination of + and − symbols at the end of a bracket). This raises interesting questions, in particular: Given a phenotype $P(D(T))$, what is the minimal developed string $D(T)$, and hence the minimal genotype T, that can generate it?

   While this question cannot be easily answered for every case, simple heuristics can be applied to a genotype T to remove the most blatant redundancies in order to substantially reduce the computational cost of the simulations, producing what can then be called the *reduced* genotype $R(T)$. It is important to note, however, that the reduced genotype is not part of the biological model, but only
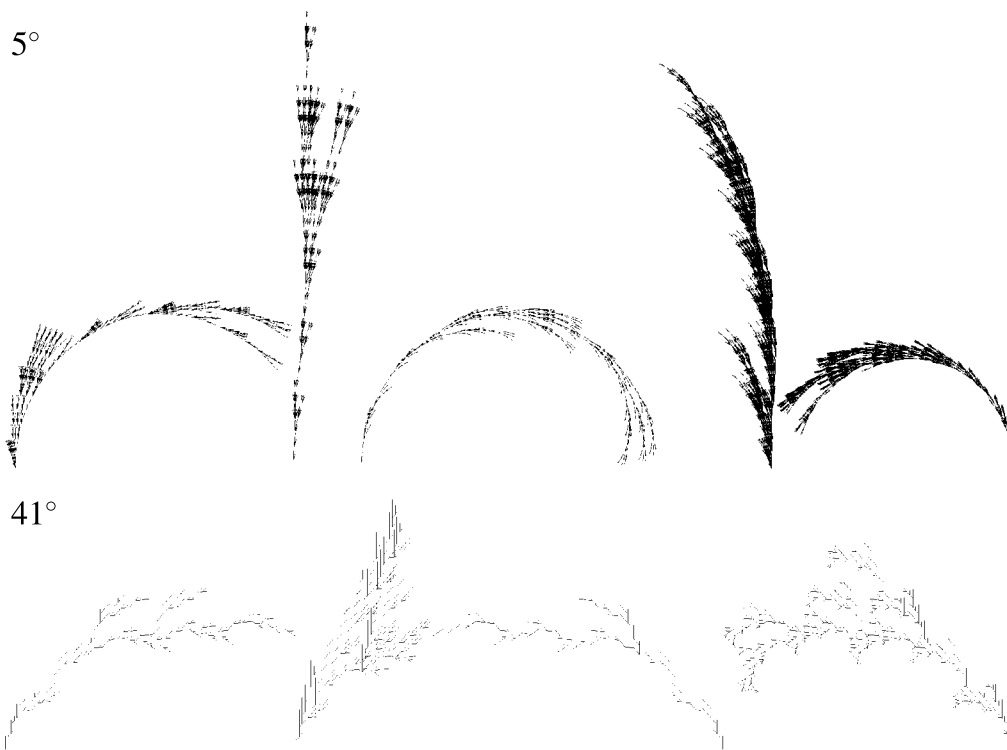
5°

41°

Figure 2. Choice of growth angle. When constructing a graphical phenotype $P(D(T))$ from a developed string structure $D(T)$, the + and − symbols are interpreted as adding and subtracting a given angle to the turtle's direction. This parameter was set to 22°; other values were tried in the evolutionary experiments but not retained. Top: Specimens obtained with an angle of 5°: Here, plants present a mostly linear, broomlike structure, which may be more or less bent into an arch. Bottom, scaled up: Specimens obtained with an angle of 41°: These plants tend to contain less material, and generally present a highly irregular, contorted structure. We opted for 22° as an interesting compromise able to generate a greater variety of natural-looking plant forms (see Figure 6).

a tool to analyze this model (see Section 3.2). The phenotype is invariant under reduction of the genotype: $P(D(T)) = P(D(R(T)))$; while T's length increases at a significantly higher rate than $R(T)$'s length, as will be later explained in Section 3.2.

The algorithmic reduction that we use here simply applies the following four transformation rules iteratively until no further change is made to the genotype:

1. Review all bracketed segments and remove any trailing + or − symbols inside them. Also remove all +− and −+ pairs wherever they appear. For example: G+−+[G+−+] becomes G+[G].

$$T_a = \texttt{G[G+[G][-G]]} \qquad T_b = \texttt{G[G+[G][--G]]}$$
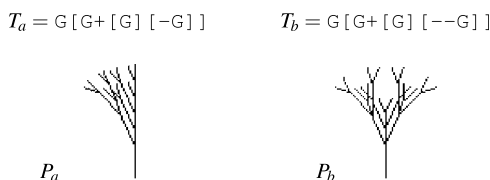
P_a P_b

Figure 3. Phenotypic divergence. Very similar genotypes can map to very dissimilar phenotypes. In this example, the genotypes T_a and T_b of two individuals (top row) differ by a single − symbol in the eighth position, yet their respective phenotypes P_a and P_b are strikingly different (bottom row).
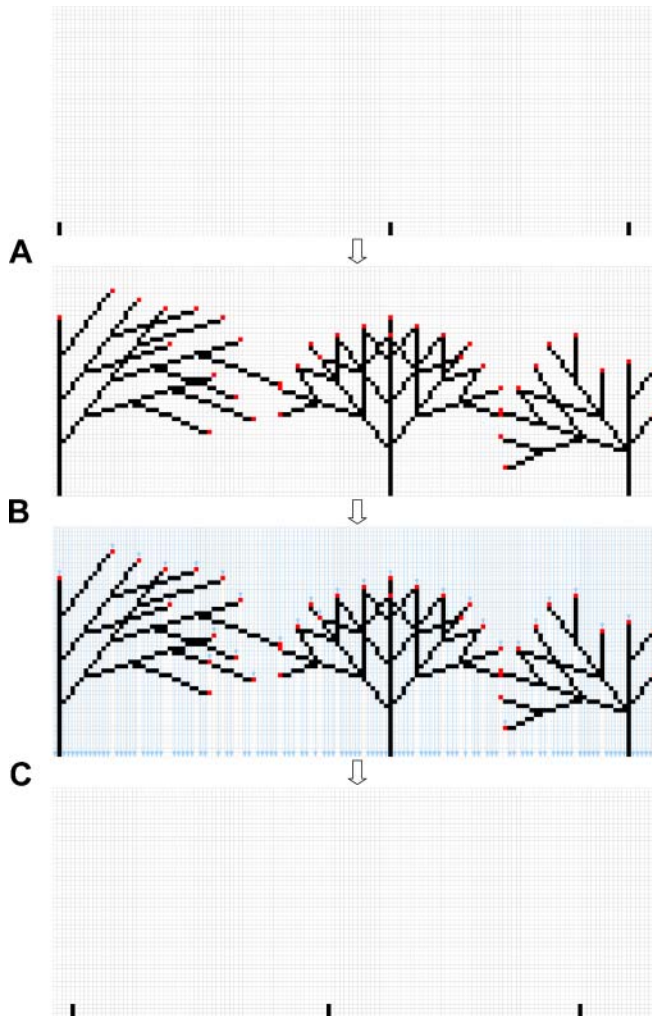
Figure 4. Simulation cycle of a plant population. (A) Plants' phenotypes grow in the environment from stolons (stem bases) spawned by the previous generation. In the electronic version, branch pixels are colored in black, leaf pixels in red. (B) In the electronic version, once plants are fully grown, they compete for ambient light in the form of vertical beams (here in blue) falling from above on the leaves. (C) The number of descendants of each plant in the next generation is a function of its fitness, calculated according to the amount of light it could capture and its number of branches. Taller plants overshadowing smaller plants are able to absorb light first, and hence have a competitive advantage. The genotypes of the simple plants displayed here are, from left to right: G[−−GG][G], G[G][−−G][++G], and G[G][−−G]++.

2. Remove empty brackets and extra layers of nested brackets, except the last pair. For example: [[[G[ ]][ ]]] becomes [G].

3. Reduce expressions of the form $[X] X$ to X and $[X] [X]$ to $[X]$, where X does not contain any brackets (to accomplish this, we use regular pattern matching for performance reasons).

4. Unnest expressions of the form $[[X_1] [X_2] \dots [X_n]]$ to produce $[X_1] [X_2] \dots [X_n]$, and expressions of the form $[[X_1] [X_2] \dots [X_n] X_{n+1}]$ to produce $[X_1] [X_2] \dots [X_n] [X_{n+1}]$, *unless* the outer pair of brackets encloses the whole genotype T. In the last case, removing it would dramatically change the phenotype, since the rule G → T is applied three times to generate $D(T)$ (whereas it would not change anything if removed from around $D(T)$).

Other, more sophisticated reduction schemes could be used (e.g., ones that would take into account semantic constraints in addition to the syntactic structure), but they would come at a higher computational cost. We opted here for simplicity and speed.

2.2 Mutation Operators

While clonal reproduction ideally creates perfect copies of successful genomes, other genetic mechanisms alter the genomic information passed on to descendants, ultimately causing evolution. We denote with M a mutation operator transforming one genotype into another: $T' = M(T)$. In the present model, mutations are constrained to preserve well-balanced brackets in the symbolic expressions. Two types of mutations are modeled here: *point mutations* and *duplications*. Point mutation operators affect the genome at the level of a single symbol or a bracketed expression in three different ways:

- M_A: *alteration*, replacing a symbol (other than brackets) by another symbol (other than brackets);

- M_D: *deletion*, removing a symbol (other than brackets) or a whole bracketed expression of the genotype (under the constraint that the resulting genotype still includes at least one G symbol);

- M_I: *insertion*, adding a symbol or an empty bracketed expression [ ] (itself susceptible to being filled later on) at a random position in the genome.

Duplication mutation operators, on the other hand, only apply to bracketed expressions and provide the genome with an additional copy of that expression in three possible ways:

- M_R: *random duplication*, inserting the copy of the bracketed expression at any position in the genome (including possibly inside itself);

- M_L: *level duplication*, inserting the copy of the bracketed expression at any position in the genome that is located at the same bracket-nesting level as the original (where the nesting level of a position can be computed as the number of open brackets minus the number of closed brackets leading to that position);

- M_T: *tandem duplication*, inserting the copy of the bracketed expression immediately after the original (note that this kind of mutation is always silent according to the genotype-to-phenotype mapping—e.g., $P(D([G])) = P(D([G][G]))$—until further mutations change either the original or the copy).

A full mutation function is calculated by composing the above six elementary operators, where each type of operator is given in turn an independent probability of occurrence of 0.05. If more than one mutation is activated, then these are applied in a sequence (i.e., each operator transforms the string just produced by the previous operator).

2.3 Environment and Fitness Function

The plants' environment is simply the discretized 2D plane containing the pixel matrix occupied by the plant phenotypes. Accordingly, a pixel of the environment can be either empty or occupied by one or several overlapping plant bits (branches or leaves). This 2D world is bounded only downwards by the horizontal ground. It extends without bounds left, right, and up. Plants are modeled solely by their aerial part, ignoring the roots, and can grow in width or height without restriction. This means that a plant can cover an unlimited number of pixels, and the length of occupied ground can increase as far as plants can reach.

Plants do not interfere with one another during growth; they develop independently as if in isolation. When immersed in the environment, the base of each plant's stem (i.e., where the turtle starts drawing) is assigned a horizontal coordinate on the ground; then the plant's branches and leaves freely expand in the open environment. As explained above, the exact phenotype $P(D(T))$ is a deterministic product of the turtle-graphic interpretation of the thrice written string $D(T)$. If any branch of the plant hits the ground (i.e., if any pixel is going to be drawn below ground level), then the plant is marked as unfit and removed from the environment and the evolving population.

The environment offers a single source of energy: light, distributed homogeneously and uniformly over space and constantly over time. Light projects from the top, vertically, one beam per pixel column, and is captured exclusively by the plants' leaves (the red terminal pixels), not by their branches. In other terms, only leaves are opaque: A unit of light hitting a leaf is fully absorbed and contributes to the energy captured by the plant. It does not propagate further down, nor is it reflected in any direction. The only two other possible events involving light are falling through a branch or vanishing into the ground. If a unit of light hits a pixel that is occupied by several overlapping leaves from different plants, then the energy contribution will be captured by only one of these, chosen at random.

Based on this virtual energy source, each individual a in the environment has a replicating potential, or *fitness*, determined in part by the total amount of light, denoted l_a, that it is able to capture. The other part contributing to the fitness, but in an inverse way, is the size of the foliage, f_a, measured as the number of branches. This number corresponds to the subset of those instances of G in $D(T)$ that caused the turtle to switch a white pixel to black (i.e., it excludes other instances of G that only caused the turtle to redraw an existing branch; see previous discussion in Section 2.1.3). In summary, the fitness F_a of each individual is based on the ratio of gained energy l_a to spent energy f_a, thus it represents the ease of building its structure (the higher F_a, the better):

$$F_a = \frac{l_a}{f_a^\alpha}$$

in which $\alpha \in [0, 1]$ is a constant exponent. The parameter α tunes the environmental cost of growing and branching, and for this reason we refer to α as the *harshness* of the environment: The higher α, the more difficult it is for the plant to increase its surface exposure to light without compromising its overall fitness. Typical values of α will be chosen in the interval [0.5, 1] (see Section 3), which corresponds to fitness values F_a roughly of the order of 1.

2.4 Evolutionary Schedule

Numerical simulations of plant populations are organized in synchronous cycles, or *generations*. At each time step, the previous generation of plants is entirely removed from the environment and a new generation is calculated by applying the following four transformations (Figure 4):

(A) *Development*: Each plant of the ith generation (offspring of another plant from the $(i - 1)$th generation) is fully developed from the ground up by calculating its phenotype $P(D(T))$ from its genotype T. Phenotypes with branches projecting below the ground are discarded from the population.

(B) *Competition*: Each plant captures the light arriving to its leaves and calculates its fitness value F. This is done by positioning the phenotype's stem in the environment at a location that was randomly chosen when the $(i - 1)$th generation was spawned. As explained in the previous section, F can be affected by the overlap with neighboring plants.

(C) *Reproduction*: Each plant then spawns a certain number of descendants (a function of its fitness), and each descendant is assigned a random position in the environment near its parent (see details below).

(D) *Mutation*: Finally, each descendant is subject to the mutation operators explained in Section 2.2, potentially modifying its genotype. After mutation, the set of all resulting genotypes make the $(i + 1)$th generation, which is ready to be processed in the same way as the previous generation, starting again from step (A).

In stage (C) the number of descendants of an individual is calculated by rounding to the nearest integer (up or down) the sum of its fitness plus a random value uniformly drawn from the interval [0, 0.6]. The rationale behind 0.6 is twofold: giving individuals with zero fitness a chance of staying in the game by maintaining one descendant, and giving individuals with fitness approximately equal to 1 a chance to have more than one descendant. Subsequently, offspring are positioned in the vicinity of the parent in a way that resembles the strategy of stolons in vegetative reproduction by clonal growth [5]: positions along the horizontal axis of the ground are uniformly drawn in a 100-pixel range centered around the stem of the parent. The width of the range influences evolutionary pressure, as it modulates the ease of colonizing empty space.

The initial population is composed of a single plant, whose genome codes for the simplest L-system able to reproduce: the single-character string G. This single-individual, single-symbol initialization scenario is at the core of the main motivation in this work, which is to study the evolution of diversity in purely emergent communities. It is thus best to start from the simplest forms of organization, both at the organism and at the population level. If we started from a population of already developed plant shapes, this would bias the results of the study.

The (A)…(D) loop is run for a preset amount of generations, 500 in this work. It can also be stopped earlier if it becomes too computationally costly due to occasional explosions in the size of T and hence $D(T)$ and $P(D(T))$.

3   Experimental Evolution of Virtual Plant Communities

We present here the results of a number of experiments in detail. Our main observation is that evolution drives a single individual containing the simplest possible genome $T = $ G toward an increasingly large population that expands along the horizontal ground in both directions and grows vertically. Our simulations of virtual plant communities illustrate how organisms proliferate, diversify, and eventually colonize their world. With each individual able to grow, absorb energy, replicate, and mutate, experiments show that the size of the population first increases, then starts to diversify. It can then take different evolutionary paths. Final distributions of phenotypes range from a spread-out collection of simple individuals to winner-take-all situations where a few individuals have developed dramatically large morphologies, extinguishing all simpler types—after which they start a Red Queen race with each other toward even greater heights and widths.

Three values of the exponent α were tried (see definition above in Section 2.3). We summarize the resulting three different worlds here, then describe their dynamics in greater detail in Section 3.1. We also present *distance* measures in Section 3.2 for a more rigorous quantitative assessment of the visible qualitative differences between these three dynamic regimes:

- α = 0.5 corresponds to a *mild* environment: The cost of growing is low, yet the population collapses rapidly because a handful of individuals manage to evolve toward very wide, tall, and branchy forms, blocking most of the light and swiftly eliminating smaller competitors.

- α = 0.75 corresponds to a *harsh* environment: The cost of growing is higher than in the previous case; thus simple individuals can proliferate while larger individuals develop, compete, and eventually become extinct, resulting in a slowly declining population size.

- α = 1 corresponds to a *very harsh* environment: In this case, the cost of growing is so high that all individuals remain simple and proliferate near the ground.

Three simulations, one for each value of α, will be analyzed through the remainder of the article from several points of view, highlighting similarities and differences between corresponding environments.

3.1   Overview of the Population Dynamics

Figures 5, 6, and 7 display a full example of simulated population in each type of environment. In the electronic version, individuals are colored randomly for visualization purposes. The first population (Figure 5) is typical of a mild environment where plants grow to a great magnitude, both in height and in breadth, and are intensely competing with neighboring individuals. The second population (Figure 6) shows a moderately harsh environment filled with a higher diversity of individuals presenting a more limited growth area and interfering less with other individuals. Simpler plants grow at the two ends of the population's domain. Finally, the last population (Figure 7) is characteristic of a very harsh environment, where plant forms remain simple and short and colonize the ground efficiently.

Generally, the type of environment is reflected in the evolution of two global variables: the *population size*, equal to the number of plants, and the global *biomass*, defined here as the total number of branches of all plants (Figure 8). In a mild environment (black line), a fast spike in the number of plants is followed by a great extinction event, as tall individuals eliminate smaller ones by overshadowing their canopies. Even in this scenario of dwindling population size, however, the few remaining plants evolve toward such gigantic forms, and do this so rapidly that the total biomass increases exponentially. By contrast, in a harsh environment (dark gray), the population size fluctuates as colonization of new
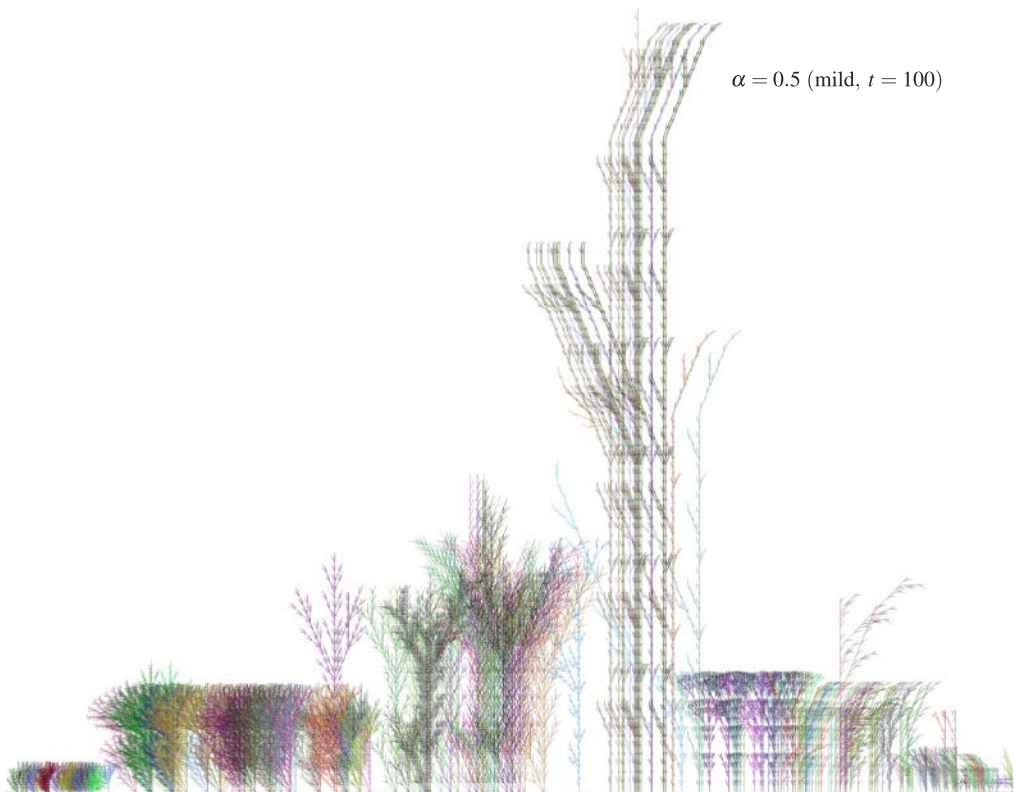

$\alpha = 0.5$ (mild, $t = 100$)

Figure 5. Typical plant world in a mild environment. Hundredth generation of one evolutionary simulation under $\alpha = 0.5$. The world covered by this plant population is 3,333 pixels wide by 2,602 pixels high. Already in the early stages, extreme competitive pressure is driving a group of plants toward a runaway Red Queen effect, in which they try to overshadow each other, while smaller individuals are at a disadvantage under the taller ones and become progressively extinct. Due to an exponential increase in CPU time and memory, simulations under these conditions have to be stopped earlier than under the other conditions, here at the 150th generation.
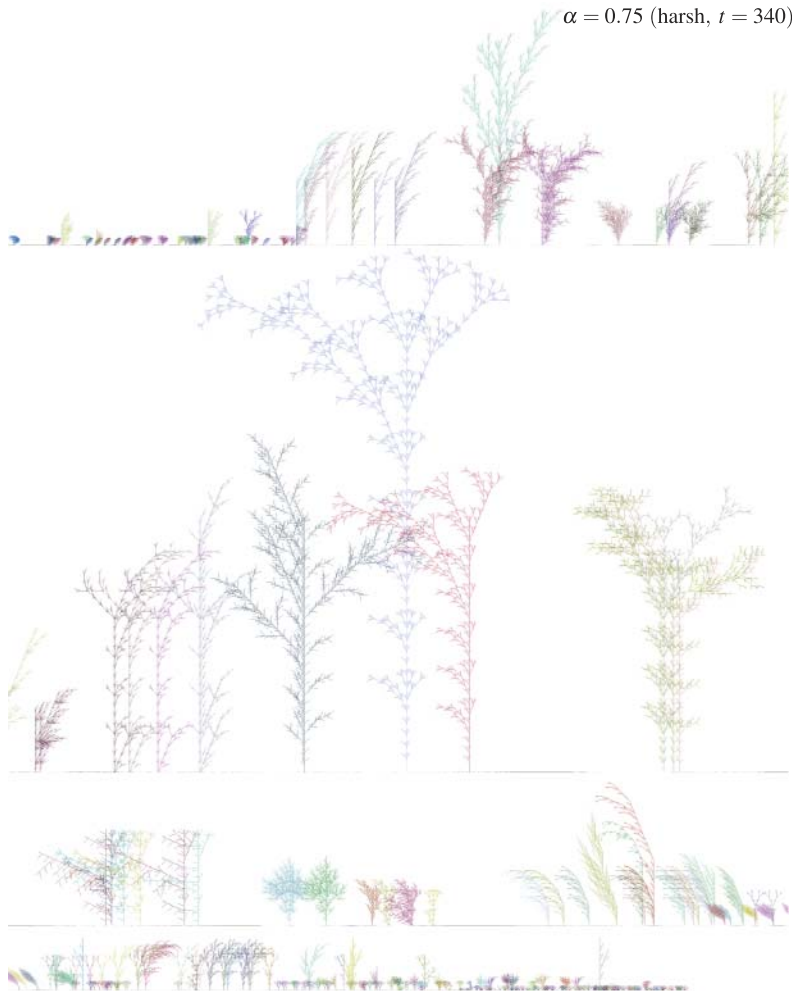
$$\alpha = 0.75 \text{ (harsh, } t = 340)$$

Figure 6. Typical plant world in a harsh environment. Three-hundred-fortieth generation of one evolutionary simulation under $\alpha = 0.75$. The world covered by this plant population is 12,627 pixels wide by 2,199 pixels high and, for clarity, has been divided into four consecutive segments, all of them at the same scale. Individuals in the central region have become larger as they compete for light. Large individuals sometimes cause the appearance of desolate areas, as they eliminate smaller neighbors and occasionally disappear themselves due to an overload of branches, thereby opening the way for simpler and faster-expanding organisms to colonize new regions.

territories is punctuated by local extinctions due to tall plants. Biomass also fluctuates, but tends to grow over larger time periods, as competition triggers the evolution of larger plants. Finally, in a very harsh environment (light gray), the population usually increases faster than in the previous case, but fluctuations remain present, since a limited degree of complexification still takes place while plants compete for light. Biomass grows very slowly, however, mostly as a consequence of the small size of the plants.

Mean fitness values ($\bar{F}$) in each kind of environment are also revealing (Figure 9). In very harsh simulations, where $\alpha = 1$, $\bar{F}$ is always less than or equal to 1 because by definition the number of light beams, l_a, captured by an individual a cannot be higher than its number of leaves and hence branches, f_a. In milder simulations, where $\alpha < 1$, some individuals can reach fitness values much larger than 1. However, the results show that the milder the environment, the lower the mean fitness level. This is due to the fact that milder environments are more crowded because plants are more prolific than in harsher environments and offspring's positions are statistically closer to their parents. Crowded conditions then create more intense competition, resulting in a significant decrease of the
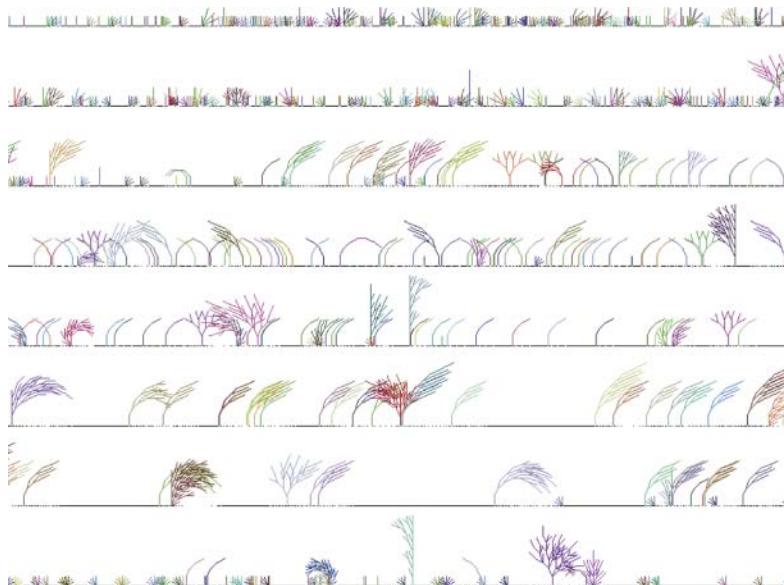
$$\alpha = 1 \text{ (very harsh, } t = 340)$$

Figure 7. Typical plant world in a very harsh environment. Three-hundred-fortieth generation of one evolutionary simulation under $\alpha = 1$. The world covered by this plant population is 7,220 pixels wide by 92 pixels high and, for clarity, has been divided into eight consecutive segments. Individuals remain simple and efficient, as biomass acquisition is severely penalized.

mean fitness. In fact, Figure 9 clearly shows that the mean fitness never rises above 1 in any environment, which indicates that high-fitness individuals constitute a minority whose sum never outweighs the remaining low-fitness population (with respect to reference level 1). Note also an interesting phenomenon of episodic *collapse* of the fitness distribution during the early generations in all environments: The mean reaches exactly 1, while the standard deviation drops to 0. These particular generations correspond to populations composed exclusively of elementary stick plants of length 1, whose fitness is exactly 1 by definition (1 unit of light divided by 1 branch, for any exponent α). At these moments, more complex individuals have all died out, albeit temporarily.

Note that the mild environmental condition $\alpha = 0.5$ is also characterized by an exploding growth in computational cost, both in memory and in time. This is why we decided to stop the simulation after 151 generations, while the time to generate the phenotypes and evaluate the fitness of the individuals was still within reasonable limits. In order to estimate the computational cost of pursuing the simulation toward later generations, an exponential curve was fitted to a 2D cloud of experimental (i, t_i) points, where i is the generation number, and t_i the CPU time required to calculate the plants up to generation i. Extrapolating from this curve, the CPU time to reach generation 200 increased 100-fold, while for generation 300 it increased 70,000-fold.

3.2 A Distance-Based Measure of Diversity

To quantitatively assess the amount of diversity of the plant population in each environment, we need a measure of distance between individuals. We propose here two definitions of distance: a genetic distance and a phenotypic distance; then we show a comparison of their abilities to differentiate among the three types of environment.

3.2.1 Definition of a Genetic Distance

We use here the *edit distance* $\mathcal{E}(T_a, T_b)$ between two strings of symbols T_a and T_b (the genomes of two individuals a and b), defined as the minimal number of insertions, deletions, and alterations of symbols

in T_a needed to transform it into T_b, or vice versa (it is indeed symmetrical, as every insertion can be reverted by a deletion and every alteration by the opposite alteration). In real-world genomics, the genetic distance between nucleotide sequences is measured in many cases by a *weighted edit distance* [4], using one of several possible weighting schemes [3] specifically fitted for each specific task. In many cases, however, genomic metrics take into account some statistical model of mutation [17]. In the
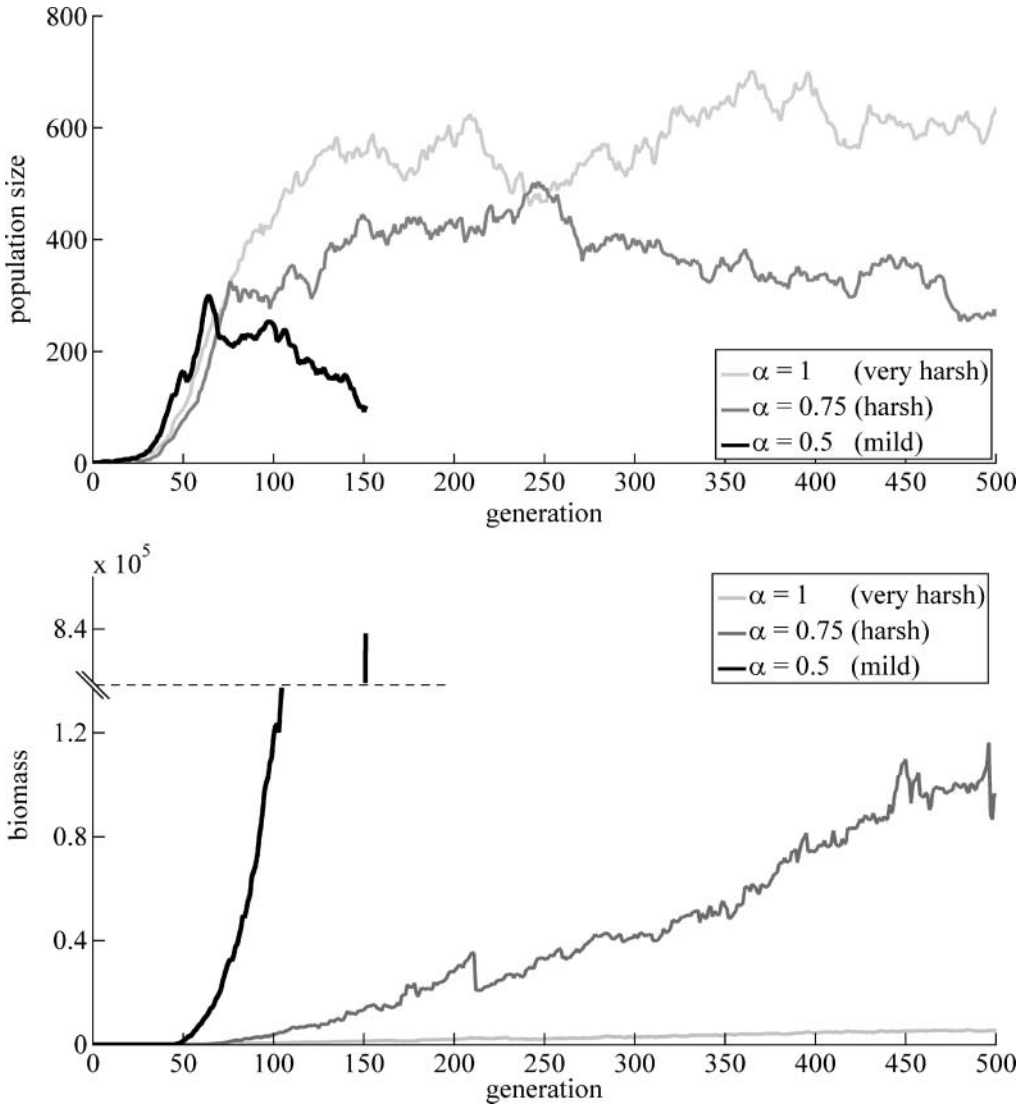

Figure 8. Evolution of population size and biomass. Top: Population size is the total number of individuals. Bottom: Biomass is the total quantity of branches across all plants. In mild simulations ($\alpha = 0.5$, black curves), population size spikes briefly before tall plants start exterminating shorter ones, as they compete to become ever taller and larger at a fast pace. Meanwhile, however, biomass continues its exponential increase (reaching about 840,000 after the 150th generation), mostly sustained by these few gigantic specimens. In harsh simulations ($\alpha = 0.75$, dark gray curves), the population size grows quickly at first, but soon starts to fluctuate, as episodic and local extinction events take place. Sudden large drops in biomass are associated with the extinction of large individuals. In very harsh simulations ($\alpha = 1$, light gray curves), the population size increases more quickly, while plant morphologies remain very simple, and consequently biomass grows very slowly compared to the other conditions. Plants are still subject to a complexification process, albeit at a smaller scale, and population size also fluctuates a little as they compete.
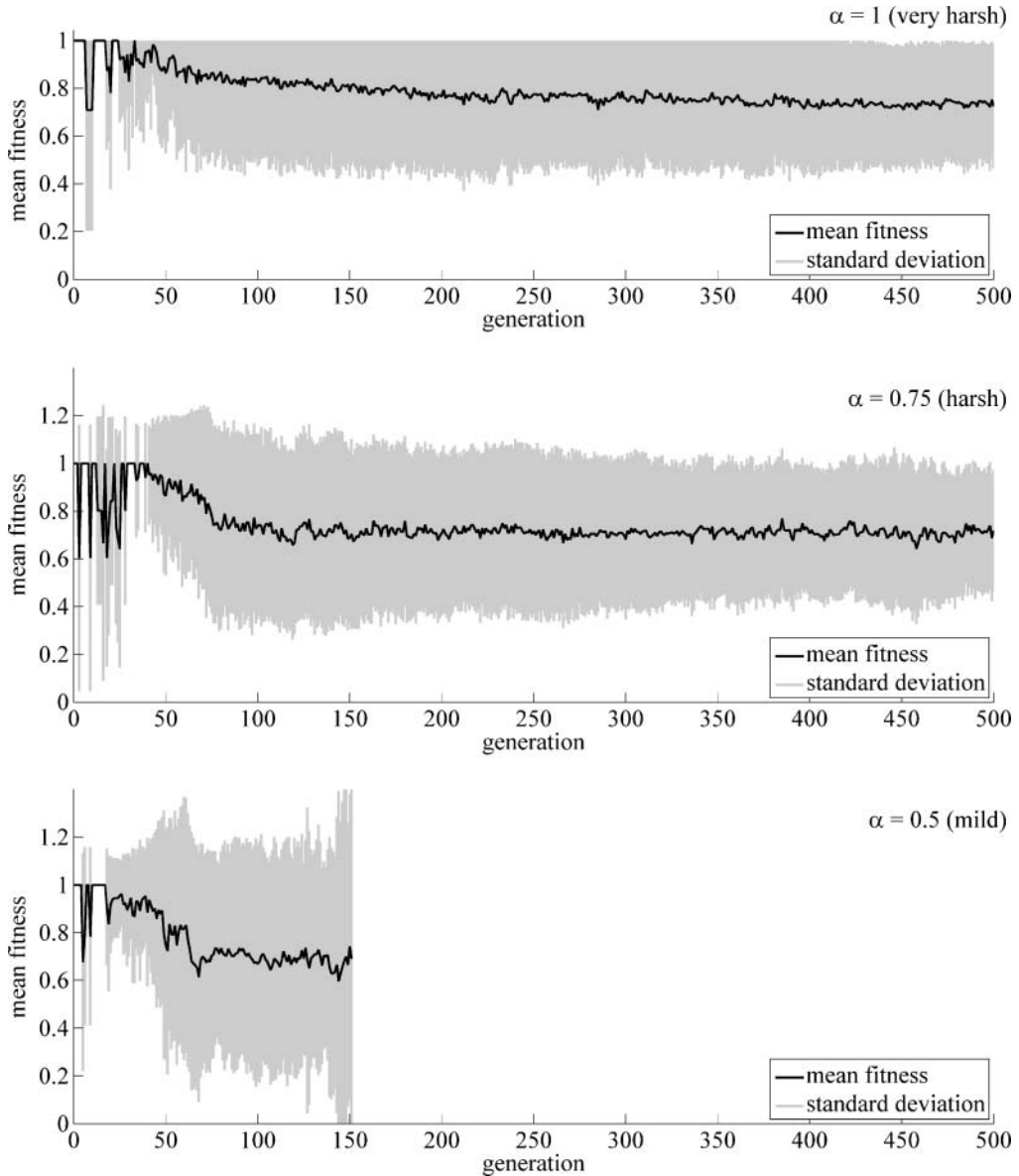
Figure 9. Evolution of mean fitness. At each time step, the mean fitness (black curves) is defined as the sum of all fitness values divided by the population size. Standard deviation bars are also shown (gray areas). Top: Very harsh condition ($\alpha = 1$). Middle: Harsh condition ($\alpha = 0.75$). Bottom: Mild condition ($\alpha = 0.5$). In all simulations, the initial value is 1, corresponding to the fitness of a single individual of genotype G (receiving one unit of light divided by one branch) in the absence of competition. Later, the mean fitness drops below 1 as plants become larger and compete for light, but the few individuals with fitness higher than 1 (when $\alpha < 1$) never outweigh the low-fitness majority. Note also episodic collapses of the fitness distribution to populations composed exclusively of one-stick plants (fitness 1, deviation 0), during the early generations. The milder the environment, the faster the mean fitness drops, and the higher the standard deviation becomes, since more plants tend to die out under the shadow of larger individuals. In the harsh and very harsh environments, the mean fitness seems to converge to slightly different values, close to 0.7.

present study, we measure genetic diversity using a raw edit distance (without weighting). For example, if $T_a = $ G+[G[−G]G]+ and $T_b = $ G[G[−−G]G]G, then $\mathscr{E}(T_a, T_b) = 3$, as we can transform T_a into T_b by removing the second symbol from T_a, adding a symbol − before the third G, and replacing the last symbol + by G.

3.2.2 Definition of a Phenotypic Distance

For a measure of the degree of dissimilarity between morphological phenotypes, we use the *Jaccard distance*, denoted $\mathcal{J}$. This distance equals the ratio of overlapping pixels to the total number of pixels occupied by both plants, which are aligned and superimposed to make their stem bases match. Thus, if P_a and P_b are the pixel sets of two aligned individuals a and b, the Jaccard distance reads

$$\mathcal{J}(P_a, P_b) = 1 - \frac{|P_a \cap P_b|}{|P_a \cup P_b|} \tag{1}$$

Figure 10 shows three distinct phenotypes P_a, P_b, and P_c, illustrating the Jaccard distance between two similar ones ($\mathcal{J}(P_a, P_b) = 0.31$) and two dissimilar ones ($\mathcal{J}(P_a, P_c) = 0.92$).

3.2.3 Comparing Genetic and Phenotypic Diversity

For a given distance $\mathcal{D}$, whether $\mathcal{E}$ or $\mathcal{J}$, we define the diversity $\delta_\mathcal{D}$ of the population as the mean over all the values of the distance matrix $[\mathcal{D}_{ab}]_{N \times N}$:

$$\delta_\mathcal{D} = \frac{1}{N^2} \sum_{a=1}^{N} \sum_{b=1}^{N} \mathcal{D}_{ab} \tag{2}$$

where N is the size of the population, and $\mathcal{D}_{ab} = \mathcal{E}(R(T_a), R(T_b))$ or $\mathcal{D}_{ab} = \mathcal{J}(P_a, P_b)$ (similar notions, such as "discrepancy," have been used elsewhere [42]). Figure 11 shows the evolution of $\delta_\mathcal{E}$ and $\delta_\mathcal{J}$ over time, using each of the two distances above. Note that the edit distance $\mathcal{E}$ was applied to reduced genotypes $R(T)$ only; interestingly, the results obtained with original genotypes T turned out to be very similar, essentially differing in scale, so we omitted them for clarity.

In milder environments, the diversity $\delta_\mathcal{J}$ based on the Jaccard distance increases faster toward its highest value 1 (in particular, the growth is considerably slower for the very harsh simulation). The evolution of diversity ($\delta_\mathcal{E}$) based on the edit distance between reduced genomes, for its part, is rather similar in the harsh and very harsh environments, while the mild conditions clearly stick out through
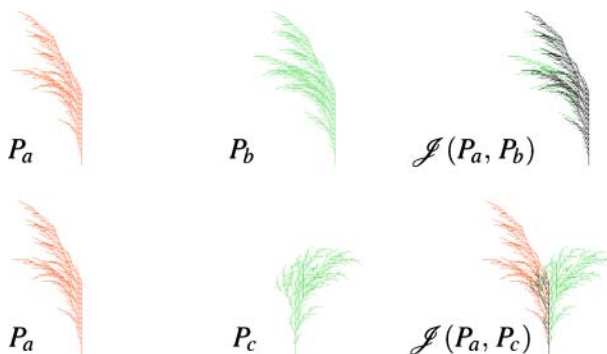

Figure 10. Illustration of the Jaccard distance. Graphical basis of the measure of similarity between individual phenotypes. Left: One same individual *a*, top and bottom. Center: Two different individuals *b* (top) and *c* (bottom). Right: Superimposition of *a* and *b* (top) and of *a* and *c* (bottom) by alignment of their stem bases. In the electronic version, black pixels represent the portion of phenotype shared by the two individuals, while green and red pixels show their symmetric difference. In the electronic version, the Jaccard distance between two phenotypes is then calculated by dividing the number of common pixels (in black) by the total number of pixels (union of the three colored areas) and subtracting the result from 1, which gives here $\mathcal{J}(P_a, P_b) = 0.31$ (top) and $\mathcal{J}(P_a, P_c) = 0.92$ (bottom).
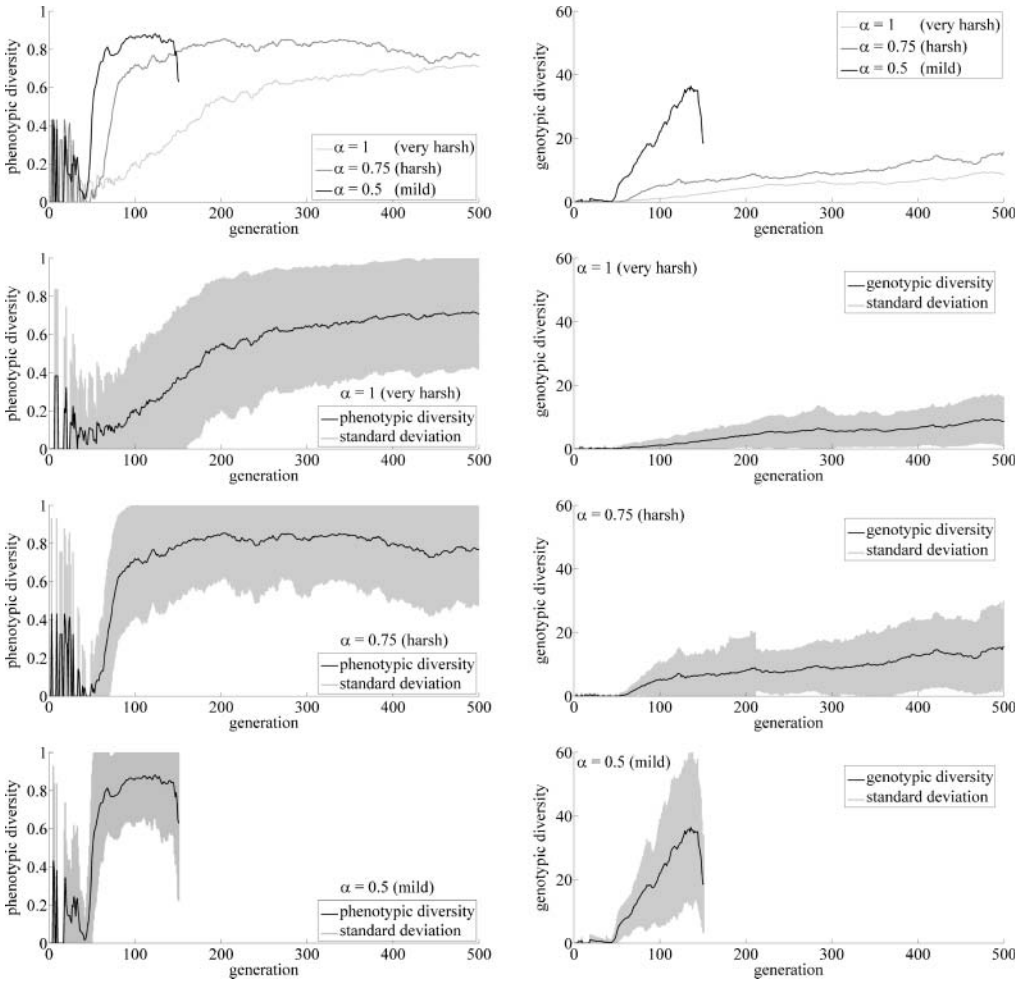
Figure 11. Evolution of diversity. Left: Phenotypic diversity $\delta_{\mathcal{J}}$ is defined as the mean of the Jaccard distance matrix $\mathcal{D}_{ab} = \mathcal{J}(P_a, P_b)$ over all pairs of individuals (a, b). Right: Genotypic diversity $\delta_{\mathcal{E}}$ is defined as the mean of the edit distance matrix $\mathcal{D}_{ab} = \mathcal{E}(R(T_a), R(T_b))$ applied to reduced genomes. Top: Comparing diversity values in the three different kinds of environments, mild ($\alpha = 0.5$, black curves), harsh ($\alpha = 0.75$, dark gray curves), and very harsh ($\alpha = 1$, light gray curves). Bottom three rows: Same diversity curves, separately for each environment, adding standard deviation bars (gray areas). The initial high-amplitude fluctuations in the phenotypic diversity are an artifact of the small population size, in which the emergence and extinction of a few mutated individuals appear large. The drop in diversity at the end of the mild simulation is due to the extinction of a large number of relatively small individuals. The sudden reduction of variance in the genotypic distance of the harsh environment, around generation 210, is due to the extinction of one plant with a large genome.

high values. This process of accelerated diversification is driven by a high growth rate in the length of the reduced genotypes.

The observed differences between the genetically based and phenotypically based notions of diversity and their evolution over generations can be ascribed to the decorrelation introduced by the indirect developmental mapping. As discussed in Section 2.1.3, a consequence of this mapping is that changes in the genotype may have wildly different consequences on the phenotype: Some mutations may cause only slight modifications in the phenotype, or none at all, while others can cause major transformations. This is famously the case of *homeobox* genes, which play a critical role in the establishment of the overall body plan of metazoan organisms [33], and whose mutation can give rise to ectopic structures, such as the development of legs in the place of antennae in *Drosophila*. Conversely,

the phenomenon of *convergent evolution* also hints at the possibility of very different genomes associated with very similar phenotypes. Thus genetic diversity and phenotypic diversity are not necessarily correlated. Figure 3 shows a case where similar genotypes give rise to clearly different phenotypes, while, on the contrary, related phenotypes encoded by very different genotypes can be seen in Figure 12.

3.3 Robustness to Mutations

Evolutionary search relies on the application of genetic operators to the individuals of a population (i.e., in this study, to their genomic strings T, *not* their developed strings $D(T)$) in order to create variability. Since reproduction is asexual here, these operators intervene only in the form of mutations. By mutating, individuals explore the fitness landscape in search of solutions adapted to the problem imposed by the environment via the fitness function. Mutation operators introduce alterations in the genome, which provoke changes in the phenotype through a developmental process. One could say that biology makes use of "indirect strategies" to encode organisms, in the sense that the genotype-to-phenotype mapping is generally highly complex. In real-world multicellular organisms, it is the result of a gigantic self-generating and self-assembling process involving between thousands and trillions of cells. In our abstract virtual model, it is produced by the linear expansion of a string that can typically contain between a dozen and a few thousand characters, giving rise to a geometric morphology covering hundreds to millions of pixels.

In this section, we use the Jaccard distance $\mathcal{J}$ (see definition above, Section 3.2.2) to study the extent of the disruption provoked by the different types of mutation operators in morphological phenotypes (only single mutations this time, not in sequence). The first five types of mutation operators described in Section 2.2 are considered: alteration (M_A), deletion (M_D), insertion (M_I), random duplication (M_R), and level duplication (M_L), leaving out tandem duplication. We use the notation $M \in \{M_A, M_D, M_I, M_R, M_L\}$ to refer to any one of these mutation operators.

Our motivation is to test the hypothesis that evolution encourages the selection of genomes that are more tolerant to mutation-induced disruption [46]. The concept of *mutational robustness* has been correlated with the evolutionary advantage of lower-than-optimal fitness plateaus under high mutation rates [47], the existence of large networks of neutral mutations [45], and the variability of the environment [30]. In our model, an evolved genome might show a more robust underlying structure
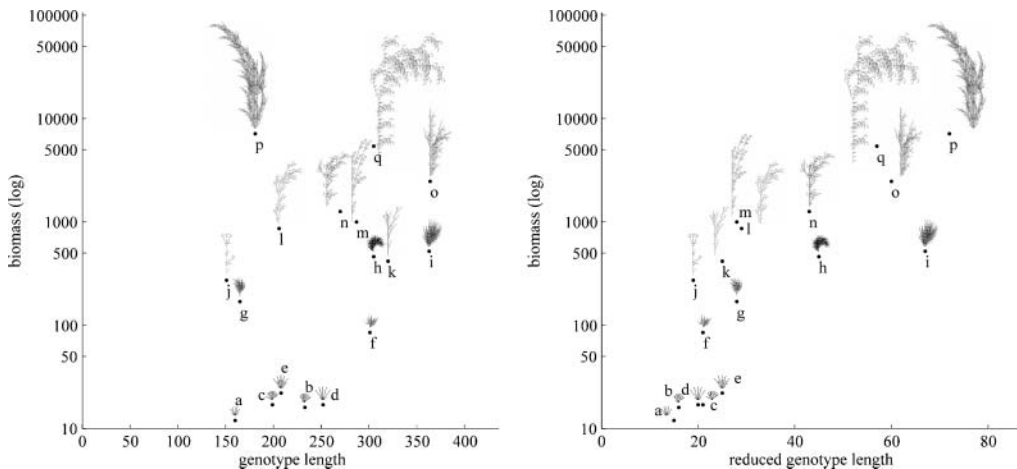

Figure 12. Biomass versus genome length. Snapshot of a few specimens from the 500th generation of a harsh-environment simulation ($\alpha = 0.75$). The biomass of a plant is defined as its total number of branches. Left: Biomass does not seem to be correlated with raw genotype length T in any meaningful way. Right: Showing the same individuals, dependence of biomass on the *reduced* genotype length $R(T)$, however, appears more correlated. Here, three trends of increasing biomass versus increasing $R(T)$ can be identified: one for individuals *a–e* (scaled 4×), another for individuals *f–i* (scaled 2×), and another for individuals *j–p* (scaled 1×). Smaller individuals are represented at a magnified scale; otherwise they would appear too small.

than a random genome, in the sense that it would be able to minimize the effects of various mutations, especially if the environmental conditions remain stable during evolution. Thus, for each mutation operator M, we are interested in measuring its effect on the phenotype of an individual a. To this aim, if b is the mutated individual such that $T_b = M(T_a)$, we define the *quantity of disruption* as the Jaccard distance between a and b, $\mathcal{J}(P_a, P_b)$.

We then calculate the mean quantities of disruption caused by various mutation operators on six different pools of genomes:

- Three pools E_m, E_h, E_{vh} of evolved genotypes, one for each kind of environment—respectively mild, harsh, and very harsh—composed of all the distinct genotypes produced by the last generation of the corresponding simulation (for E_m the 150th generation, due to the high computational cost discussed at the end of Section 3.1; for E_h and E_{vh}, the 500th generation). These pools contain 64, 245, and 593 genotypes, respectively.

- Three other pools R_m, R_h, R_{vh} of randomly generated genotypes. Each pool R_x matches the corresponding evolved pool E_x as follows: It contains the same number of genotypes, and its random strings were generated in such a way that they have the same statistical properties as the strings of E_x (average length and probability of appearance of each symbol, under the constraint of well-balanced brackets).

For each string T_a in each pool, 100 different mutated strings $T_b = M(T_a)$ were generated per mutation operator M (thus totaling 500 mutated strings per T_a), and the corresponding disruption values $\mathcal{J}(P_a, P_b)$ were evaluated. Finally, for each mutation operator M and each pool $E \in \{E_m, E_h, E_{vh}\}$, $\delta_{\mathcal{J}}^{M,E}$ was defined as the mean of all the distances $\mathcal{J}(P_a, P_b)$ between each string P_a in the pool E and the corresponding strings P_b generated through mutation operator M:

$$\delta_{\mathcal{J}}^{M,E} = \frac{1}{\|E\|} \sum_{a=1}^{\|E\|} \left(\frac{1}{100} \sum_{b=1}^{100} \mathcal{J}(P_a, P_b) \right)$$

Mutatis mutandis, we also defined the corresponding $\delta_{\mathcal{J}}^{M,R}$ for pools $R \in \{R_m, R_h, R_{vh}\}$. We refer to $\delta_{\mathcal{J}}^{M,E}$ and $\delta_{\mathcal{J}}^{M,R}$ as the *mean disruption* from the original to the mutated phenotypes for each pool and operator. Therefore, since we considered five mutation operators and six pools, a total of $5 \times 6 = 30$ mean disruption values were generated. These values can also be considered as a special type of diversity measure or "diversity jump," applied here between two different worlds (based on before-mutation–after-mutation pairs of individuals) instead of the same world (based on all possible pairs of individuals).

The final results are displayed in Figure 13. As the results for harsh and very harsh environments were nearly identical, only the results for the mild and very harsh environments are presented here (20 values). The 2D coordinates of each one of the 10 points represent the pairs of mean disruption quantities of a given mutation type over the evolved and random pools, that is, $\left(\delta_{\mathcal{J}}^{M,E}, \delta_{\mathcal{J}}^{M,R}\right)$. The main observations suggested by this chart can be summarized as follows:

- In all cases, point mutations (M_A, M_D, M_I) are far more disruptive than duplication mutations (M_R, M_L).

- In the very harsh condition, *point mutations are significantly less disruptive in evolved genomes than in random ones*, while duplication mutations are about the same (at a low level).

- In the mild condition, disruption caused by point mutations, while still lower for evolved genomes, is shifted toward significantly higher values than in the harsh condition.

In light of these results, our starting hypothesis that evolution selects for genomes that are robust against mutations is confirmed in the very harsh and harsh environments. This is to be expected,
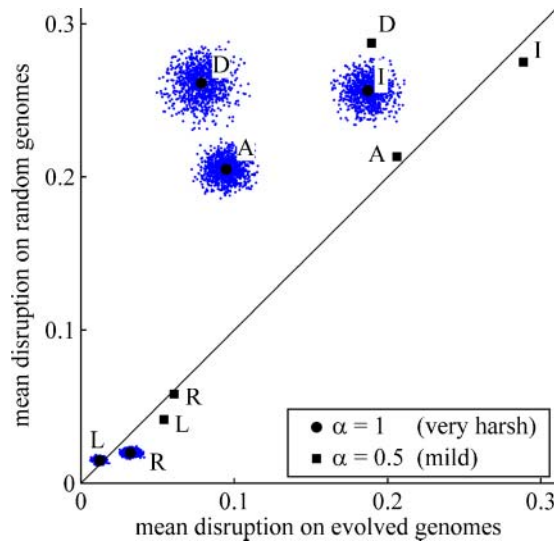
Figure 13. Robustness to mutation. Mean disruption caused by five different types of mutation operators in pools E_m, E_{vh} of evolved genomes (horizontal axis) versus corresponding pools R_m, R_{vh} of random genomes with the same statistical properties in each case (vertical axis). The disruption value for each genome in each pool is simply defined as the mean Jaccard distance between its phenotype and the phenotypes of 100 mutated versions of that genome. The coordinates of each point represent a pair of mean disruption values of a given mutation over an evolved pool and the corresponding random pool (see Section 3.3) in two different environments (squares: mild, $\alpha = 0.5$; circles: very harsh, $\alpha = 1$). It is important to note that the sizes of the pools are not identical: $|E_m| = |R_m| = 64$, while $|E_{vh}| = |R_{vh}| = 593$. In the electronic version, to show that the results are statistically robust, we have drawn 1,000 random subsamples of size 64 from the pools R_{vh} and E_{vh}, plotting the mean disruption values for each subsample as blue dots. It can be seen that the points cluster neatly around global mean values without far outliers. Mutations are labeled A for alteration, D for deletion, I for insertion, R for random duplication, and L for level duplication. Point mutations (A, D, I) are generally more disruptive than duplication mutations (R, L), and in the very harsh condition they are significantly less disruptive in evolved genomes than in random ones. Plants evolved in a very harsh environment are also more resilient to mutation-induced phenotypic changes than plants evolved in a mild environment.

since in these environments the plants are constrained to grow to small or moderate sizes, and the overall environmental conditions do not change significantly over the course of the simulation (at least on a global scale). At the same time, the resulting genotypes have been subject to a relatively high mutation rate for several hundreds of generations, and the risk of obliteration (by becoming too heavy in biomass or developing underground branches after a mutation) is always present; thus selective pressure encouraging robustness against mutations seems to be a logical effect. This would support the idea that selection in this model "sculpts" the structure of genomes in a direction that makes them more robust to the disruptive effects of mutations. For mild environments, however, the competitive pressure among individuals (in contrast to the pressure from the environment) is so high that the environmental conditions created by the neighbors change rapidly (as new individuals reach taller and taller sizes), and evolution selects for genomes that are *more* sensitive to all types of mutations, especially point mutations, in order to keep an edge over the ever larger competitors. In summary, these results reinforce the view that the environmental conditions must remain relatively constant for mutational robustness to emerge [30].

4   Discussion and Conclusions

This work combines advanced (yet minimal) models based on L-systems, genetic expression, biologically inspired mutations, and open-ended evolution in a population of interacting individuals to create a framework of evolving virtual plants. In order to characterize and understand the dynamics of diversification, data produced by the simulations was processed by simple statistical analysis based

on measures of distance. While we kept the overall model as simple as possible (through a reduced number of rules and parameters), the results show the evolution of plantlike organisms capable of increasing their degree of complexity and diversification at the same time that they are competing for common resources.

The biological value and relevance of the model is twofold. On the one hand, experiments can be fully recorded at the genomic level, since organisms are represented by one-dimensional chains of symbols, which deterministically develop into plantlike structures. Because mutations act upon genotypes, the genomic dynamics can be characterized by a statistical analysis of the pool of character strings. On the other hand, this simple genotypic model maps to a highly complex phenotypic space [9] allowing the study of rich evolutionary processes oriented toward greater organism complexity and population diversity.

We wish to stress the simplicity of our model. Other models of evolving plant communities based on L-systems are generally more complicated, making use of a great number of rules and symbols to encode the various parts of the plants. In some of these works, the conclusion was that the genetic search space of D0L systems was too limited [8]. However, this might be precisely a consequence of an overly complicated framework, since the high-dimensional parameter space that these models generate can be particularly difficult to explore and analyze. In contrast, our minimal plant system facilitates the identification of relevant biological phenomena during the simulations, in which the great volume of generated data helps, rather than hinders, the understanding of finer causality links among interacting individuals.

It is important to stress here that the present model is not based on data or parameters derived from real-world ecosystems, such as densities of species or rates of interactions between individuals. It is an abstract, artificial-life model that is not committed to any particular plant community found in nature. Yet, we believe that general parallels with biological ecology can still be drawn. More specifically, the model shows the emergence of at least two different evolutionary strategies: (1) developing a simple morphology while yielding large offspring to escape competition and colonize new areas; (2) developing a large and branchy morphology to better compete in overcrowded environments, but at the cost of reduced offspring. These two dynamics closely resemble the ecological transition from newly colonized areas to old-growth forest. In fact, the results presented here show that very harsh environments are characterized by the first strategy and mild environments by the second, reflecting their differences in developmental costs.

In summary, the biological implications of our results are the following:

1. Population diversity and individuals' complexity strongly depend on environmental conditions, confirming experimental [39] and theoretical results [24].

2. Population dynamics can be extremely varied, ranging from low-diversity populations that tend to increase in size, to very small, exclusive communities of very large individuals.

3. Environmental conditions must remain relatively constant for mutational robustness to emerge.

4. The evolution of phenotypic complexity is based upon the dynamics of genetic mutations. As these are quick to produce larger genomes, the evolutionary exploration is biased toward more complicated structures.

5. Eventually, one individual undergoes a fast excursion toward significantly more complex morphologies, extinguishing the intermediate individuals from which it evolved. These complex structures then dominate during a certain period of time, eventually themselves disappearing due to the combined effect of high internal competition and excessive enlargement.

Technically, parameter adjustment and data generation required about 700 simulated worlds, involving population sizes ranging from tens to thousands of individuals, and evolutionary times ranging

from 100 to 1,500 generations. The computational power and storage capacity required for one evolutionary experiment essentially depended on two factors: number of generations and overall biomass. With current technology, this computational effort demanded the intensive use of several tens of processing units for three months.

Greater computer power available at a lower cost provides new means to simulate complex worlds of evolving and interacting organism populations. Today, large computational resources are now accessible to conventional research groups, and more projects are undertaken in the field of artificial life, which is potentially open to any degree of model sophistication and realism. Theoretical ecology, in particular, can greatly benefit from this approach, as several open problems of population genetics can be formulated and studied in virtual communities.

Acknowledgments

References

1. Abelson, H., & diSessa, A. (1986). *Turtle geometry: The computer as a medium for exploring mathematics (artificial intelligence)*. Cambridge, MA: MIT Press.

2. Allen, M., Prusinkiewicz, P., & DeJong, T. (2005). Using L-systems for modeling source-sink interactions, architecture and physiology of growing trees: The L-PEACH model. *New Phytologist*, *166*, 869–880.

3. Apostolico, A., & Giancarlo, R. (1998). Sequence alignment in molecular biology. *Journal of Computational Biology*, *5*, 173–196.

4. Batzoglou, S. (2005). The many faces of sequence alignment. *Briefings in Bioinformatics*, *6*, 6–22.

5. Bell, A. (1991). Vegetative multiplication. In *Plant form: An illustrated guide to flowering plant morphology*. Oxford, UK: Oxford University Press.

6. Bornhofen, S., & Lattaud, C. (2006). Life history evolution of virtual plants: Trading off between growth and reproduction. In *Parallel Problem Solving from Nature IX* (pp. 808–817). Berlin: Springer.

7. Bornhofen, S., & Lattaud, C. (2008). On hopeful monsters, neutral networks and junk code in evolving L-systems. In *Proceedings of the 2008 Genetic and Evolutionary Computation Conference GECCO '08* (pp. 193–200). New York: Association for Computer Machinery.

8. Bornhofen, S., & Lattaud, C. (2009). Competition and evolution in virtual plant communities. *Natural Computing: An International Journal*, *8*, 349–385.

9. Casas, M., & Vico, F. (2009). On the performance of some bioinspired genetic operators in complex structures evolution. In *Proceedings of the 2009 Genetic and Evolutionary Computation Conference GECCO '09* (pp. 1841–1842).

10. Champagnat, N., Ferrière, R., & Méléard, S. (2006). Unifying evolutionary dynamics: From individual stochastic processes to macroscopic models. *Theoretical Population Biology*, *69*, 297–321.

11. Collado-Vides, L., Gómez-Alcaraz, G., Rivas-Lechuga, R., & Gómez-Gutierrez, V. (1997). Simulation of the clonal growth of *Bostrychia radicans* (Ceramiales, Rhodophyta) using Lindenmayer systems. *Biosystems*, *42*, 19–27.

12. Dawkins, R. (1986). *The blind watchmaker*. New York: W. W. Norton.

13. Ebner, M. (2003). Evolution and growth of virtual plants. In *Lecture Notes in Computer Science* (pp. 228–237). Berlin: Springer.

14. Ebner, M. (2006). Coevolution and the Red Queen effect shape virtual plants. *Genetic Programming and Evolvable Machines*, *7*, 103–123.

15. Ebner, M., Grigore, A., Heffner, A., & Albert, J. (2002). Coevolution produces an arms race among virtual plants. In *Lecture Notes in Computer Science* (pp. 114–170). Berlin: Springer.

16. Federici, D., & Downing, K. (2006). Evolution and development of a multicellular organism: Scalability, resilience, and neutral complexification. *Artificial Life*, *12*, 381–409.

17. Freeman, S., & Herron, J. (2003). Evolution and human health. In *Evolutionary analysis* (p. 562, box 4.1). Englewood Cliffs, NJ: Pearson Prentice Hall.

18. Gautier, H., Mech, R., Prusinkiewicz, P., & Varlet-Grancher, C. (2000). 3D architectural modelling of aerial photomorphogenesis in white clover (*Trifolium repens L.*) using L-systems. *Annals of Botany*, *85*, 359–370.

19. Gerlee, P., & Lundh, T. (2010). Productivity and diversity in a cross-feeding population of artificial organisms. *Evolution*, *64*, 2716–2730.

20. Grimm, V., & Railsback, S. F. (2005). *Individual-based modeling and ecology.* Princeton, NJ: Princeton University Press.

21. Jacob, C. (1994). Genetic L-system programming. In *Parallel Problem Solving from Nature III* (pp. 334–343). Berlin: Springer.

22. Jacob, C. (1996). Evolving evolution programs: Genetic programming and L-systems. In *Genetic Programming 1996: Proceedings of the First Annual Conference* (pp. 107–115).

23. Jacob, C. (2001). *Illustrating evolutionary computation with Mathematica.* San Mateo, CA: Morgan Kaufmann.

24. Jansen, V., & Mulder, G. (1999). Evolving biodiversity. *Ecology Letters*, *2*, 379–386.

25. Koza, J. (1993). Discovery of rewrite rules in Lindenmayer systems and state transition rules in cellular automata via genetic programming. In *Symposium on Pattern Formation SPF '93*.

26. Levin, S., Grenfell, B., Hastings, A., & Perelson, A. (1997). Mathematical and computational challenges in population biology and ecosystems science. *Science*, *275*, 334–343.

27. Lindenmayer, A. (1968). Mathematical models for cellular interaction in development: Parts I and II. *Journal of Theoretical Biology*, *18*, 280–315.

28. Lobo, D., & Vico, F. J. (2010). Evolutionary development of tensegrity structures. *Biosystems*, *101*, 167–176.

29. McCormack, J. (1993). Interactive evolution of L-system grammars for computer graphics modelling. In *Complex systems: From biology to computation.* Amsterdam: IOS Press.

30. Meyers, L. A., Ancel, F. D., & Lachmann, M. (2005). Evolution of genetic potential. *PLoS Computational Biology*, *1*, e32.

31. Mock, K. (1998). Wildwood: The evolution of L-system plants for virtual environments. In *IEEE World Congress on Computational Intelligence* (pp. 476–480).

32. Nowak, M. A. (2006). *Evolutionary dynamics: Exploring the equations of life.* Boston: Belknap Press of Harvard University Press.

33. Nusslein-Volhard, C., & Wieschaus, E. (1980). Mutations affecting segment number and polarity in *Drosophila. Nature*, *287*, 795–801.

34. Oakley, T. H. (2009). A critique of experimental phylogenetics. In T. Garland & M. R. Rose (Eds.), *Experimental evolution: Concepts, methods, and applications of selection experiments.* Los Angeles, CA: University of California Press.

35. Ochoa, G. (1998). On genetic algorithms and Lindenmayer systems. In *Parallel Problem Solving from Nature V* (pp. 335–344).

36. Prusinkiewicz, P. (1986). Graphical applications of L-systems. In *Proceedings on Graphics Interface '86* (pp. 247–253). Toronto: Canadian Information Processing Society.

37. Prusinkiewicz, P., & Lindenmayer, A. (1990). *The algorithmic beauty of plants.* Berlin: Springer-Verlag.

38. Ray, T. S. (1991). An approach to the synthesis of life. In *Artificial Life II* (pp. 371–408). Reading, MA: Addison-Wesley.

39. Rosenzweig, M. (1995). *Species diversity in space and time.* Cambridge, UK: Cambridge University Press.

40. Runqiang, B., Chen, P., Burrage, K., Hanan, J., Room, P., & Belward, J. (2002). Derivation of L-system models from measurements of biological branching structures using genetic algorithms. In *Developments in applied artificial intelligence.* Berlin: Springer.

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

42. Studer, M., Ritschard, G., Gabadinho, A., & Müller, N. S. (2010). Discrepancy analysis of complex objects using dissimilarities. In *Advances in knowledge discovery and management*. Berlin: Springer.

43. Toussaint, M. (2003). Demonstrating the evolution of complex genetic representations: An evolution of artificial plants. In E. Cantú-Paz et al. (Eds.), *Proceedings of the 2003 Genetic and Evolutionary Computation Conference GECCO '03* (p. 200). Berlin: Springer.

44. Watson, J., Hanan, J., & Wiles, J. (2008). Modeling the fitness of plant morphologies across three levels of complexity. *Biosystems*, *94*, 182–190.

45. Wilke, C. (2001). Adaptive evolution on neutral networks. *Bulletin of Mathematical Biology*, *63*, 715–730, 10.1006/bulm.2001.0244.

46. Wilke, C., & Adami, C. (2003). Evolution of mutational robustness. *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, *522*, 3–11.

47. Wilke, C., Wang, J., Ofria, C., Lenski, R., & Adami, C. (2001). Evolution of digital organisms at high mutation rates leads to survival of the flattest. *Nature*, *412*, 331–333.

48. Yamamura, N., Higashi, M., Behera, N., & Wakano, J. (2003). Evolution of mutualism through spatial effects. *Journal of Theoretical Biology*, *226*, 421–428.