NATURAL RESOURCE MODELING
Volume 15, Number 1, Spring 2002

AB INITIO MODELING OF ECOSYSTEMS WITH ARTIFICIAL LIFE

C. ADAMI
Jet Propulsion Laboratory 126-347
California Institute of Technology
Pasadena, CA 91109
E-mail: adami@jpl.nasa.gov
and
Digital Life Laboratory 136-93
California Institute of Technology
Pasadena, CA 91125

ABSTRACT. Artificial Life provides the opportunity to study the emergence and evolution of simple ecosystems in real time. We give an overview of the advantages and limitations of such an approach, as well as its relation to individual-based modeling techniques. The Digital Life system Avida is introduced and prospects for experiments with *ab initio* evolution (evolution "from scratch"), maintenance, as well as stability of ecosystems are discussed.

1. Introduction. Individual-based ecological modeling seems to be an unstoppable trend in modern ecological science, but it is not without its problems (Grimm [1999]). The attraction of this approach stems from the promise that individual-based models (IMBs) might capture emergent effects on a macroscopic level while only implementing simple interactions between agents on the microscopic level. This bottom-up approach is, in fact, a legacy of the early years of Artificial Life (see, e.g., Langton et al. [1992]) and has probably produced as much success as failure. The problems are not difficult to discern. On the microscopic level, the salient characteristics of the agents have to be decided upon, and it is rather rare that such decisions are made with strict theoretical concepts in mind. More often than not, they are subject to change and tinkering. On the macroscopic level, emergent effects are (precisely because of their nature) often subjective. The interaction between tinkering and subjective appreciation produces, again more often than not, a tendency to select effects via *parameter mutation*, i.e., a particular set of parameters may tend to be adopted as standard simply because of the subjectively interesting patterns it may produce. Thus, in the language of evolutionary biology, you often "get what you select for". This modeling disease, well-known from Artificial Life, is certainly also a problem in IMBs.

Another problem concerns the complexity of most IMBs and the ensuing difficulty in communicating salient, "universal", results. To some extent, the modeling of complex systems defeats its purpose if the complexity cannot be *reduced*. After all, "complex", when used together with "systems", is just a synonym for "not-understood".

Thus, sane IMBs should strive at reducing the complexity of a scenario, by treating cases that are as simple as possible, while retaining the essential characteristic being addressed. Furthermore, limiting cases should be included that have well-known theoretical solutions, such that baselines can be established that are beyond doubt. Because ecological modeling is supposed to replace at least some experimental work, there is no doubt that it has to be supplemented and checked against theory as much as possible (Grimm [1999]).

In this contribution I would like to propose a modeling paradigm for IMBs which avoids the pitfalls mentioned above. It is not a panacea, as its applicability to the ecological sciences is severely limited. As will become obvious, it has nothing to say about higher animals, food webs, or predator-prey interactions. However, it carries the promise to answer fundamental questions in ecology, such as concerning the emergence, maintenance, and stability of simple ecologies in simple environments. The way the model avoids the possible problems listed above is that, in the first place, the organisms in question—the microscopic agents—have not been designed. Instead, they themselves emerge via adaptation to an artificial world. Secondly, the system is simple enough that the fundamental dynamics of these organisms is well-understood theoretically (and by now experimentally), at least in the single-niche environment. Thus, there is ample room for validation and cross-checking. The organisms I am speaking about, rather than being abstract agents, are in fact self-replicating computer programs that live in the random access memory (RAM) of the computers within which the virtual world is built. They are a form of life (Harvey [1999]) whose primary purpose has been research into possible universal characteristics of evolution in simple living systems. This modeling platform (or more precisely, experimental platform, see below), called `avida`[1], has been used in the single-niche mode, without co-evolution, for the last seven years. The reasons for such caution lie precisely in validation: there is no logic in facing the complexity of multi-niche populations adapting to local conditions if the single-niche, homogeneous, situation is not fully understood. However, the time for the next step appears to have come.

2. Simulation or experiment? The main distinction between the kind of individual-based modeling described above and the approach followed in Artificial Life (in particular the type of Digital Life described below) is that the quantities which carry the system's essential properties (the"degrees of freedom") under consideration in Artificial Life, whether they be robotic, genetic, or abstract (such as cellular automata) are given a *physical*, rather than mathematical, representation. The dynamics (collective or otherwise) of these degrees of freedom is subsequently computed and *observed*. This distinction is, in most cases, not arbitrary. It relies on a fundamental discovery which lies at the heart of computational science, namely Turing's famous theorem concerning the universality of computation (Turing [1936]). Turing's theorem implies a duality between computation and physical dynamics that has wide-ranging consequences not only for computation, but indeed for physics. It states that any dynamical system with physical degrees of freedom that is "complicated enough"[2] is capable of universal computation, which means that a computer constructed using these degrees of freedom can calculate "any"[3] function. The theorem also implies that all universal computers are strictly equivalent, in the sense that any such computer can *simulate* any other. An under-appreciated consequence of the theorem is that it works "both ways": If "complex enough" dynamics imply universal computation, then it is clear that any universal computer (such as, e.g., any von Neumann-architecture computer) has to have "interesting" physical dynamics at its core. With "interesting" I mean that the degrees of freedom and their dynamics could be used, or shaped, to do almost anything.

Let me give an example. With a computer, there are two ways to investigate wave propagation in media. The usual one is to write down the relevant differential equations and solve them numerically. In this case, the wave is represented mathematically (say $\sin(x)$), and we would not be tempted to assert that there is an actual wave anywhere inside of the computer. This is, of course, the central idea in simulation. Here is another way to solve this problem with a computer. Suppose that, instead of programming the differential equations, I instead program memory locations within RAM to interact in a specific manner. In particular, I program adjacent locations (which we can view as potential wells carrying charges) to interact like springs with a particular mass, spring constant, and a given amount of friction. It would then be possible to observe an excitation propagate through the RAM of the computer precisely like the sine wave in the simulation, only that this wave would be *real*. Granted, this would be a perverse way of solving wave propagation with a computer, but the example serves to illustrate that universal computers do indeed rely on physical dynamics at their core, and that these physical degrees of freedom (charges in potential wells) are as real as billiard balls or mechanical cogs or biochemical molecules, all of which have been used to construct universal computers.

When a computer is used in such a "perverse" manner, it is clear that one is performing experiments, not simulations. Under which circumstances does this approach present an advantage over simulation? Precisely in the case where the construction of the physical dynamics would be far too complicated in any other medium than the computer, and when the architecture of the computer is best suited for the problem at hand. Because the physical degrees of freedom in a computer are primarily used to store, transmit, and manipulate information, physical dynamics of information-bearing degrees of freedom are most easily implemented. This encompasses the dynamics of simple living systems, in particular those whose essence is informational (i.e., self-replicating molecules). A typical example would be a computer virus. Computer viruses physically populate a computer's memory and physically replicate in it. As it is clear that the viruses are not simulated, and as it is clear from the duality mentioned above that there is no reason to draw a fundamental distinction between one sort of information-bearing degree of freedom (e.g., biomolecules) or charges in potential wells, self-replicating programs within a computer's memory have to be considered on an equal footing as, say, *Mycoplasma mycoides* thriving in our nasal passages. Turing's insight, thus, demolishes the barrier between "real" and "artificial" life.

3. Digital life. The field of Digital Life was, in fact, directly inspired by computer viruses. In 1989, Steen Rasmussen at Los Alamos National Laboratory created a "reserve" for computer viruses inside of a computer (Rasmussen et al. [1999]), by creating virtual CPUs that executed programs written in Redcode, a type of assembly language used in a computer game called "CoreWars," where the objective is to write code that invades and takes over another computer's memory. It turned out that the best strategy to win this game was to write self-replicating programs, and these were used by Rasmussen to inoculate that special space inside of the computer. As a consequence, the "world" in which these programs live is virtual, i.e., simulated, while the actual programs (because they are physical) are real. This can be roughly compared to what happens in *in vitro* experiments with *E. coli* bacteria. The environment, (namely the Petri dish with its nutrients) that these bacteria live in is entirely artificial, and very controlled. The bacteria on the other hand are real. While leading to important insights, Rasmussen's experiments in evolution were not successful because his choice of world led to very fragile programs: while trying to copy themselves they inevitably overwrote adjacent programs, such that populations died very quickly. This shortcoming was lifted by Tom Ray with the now famous tierra software (Ray [1992]), in which the first interesting dynamics of "alien life" could be observed. Ray's tierra inspired the creation of the avida system at Caltech in 1993 (Adami and Brown [1994], Adami [1998]); avida is the most widely used Digital Life platform today. Because of the flexibility of avida as an experimental platform, it has been used to address a wide variety of problems in the evolution and dynamics of simple living systems (Chu and Adami [1997], Adami et al. [1998], Adami et al. [1999], Ofria and Adami [1999], Ofria et al. [1999], Lenski et al. [1999], Wagenaar and Adami [2000]).

From its initial design, avida emphasized simplicity and accountability. As an experimental platform, it would have to be possible to focus on particular aspects of living systems while "turning off" those that could detract from the question being asked. Thus, for example, sexual crossover between organisms was not to be implemented before the dynamics of asexual reproduction was fully understood and mapped out (a stage yet to be reached). World geometry was chosen either as a two-dimensional grid that wraps on itself (a flat torus) in order to avoid boundary effects, or else a well-stirred reactor without any geometry. In both cases, the world is isotropic and homogeneous, i.e., it is a single-niche environment (see Figure 1).

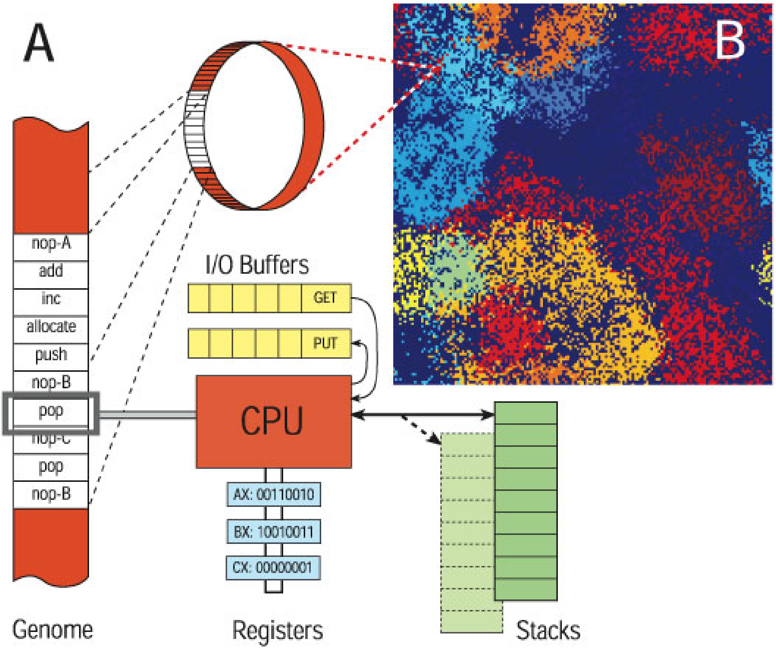

FIGURE 1. The virtual world of avidians. (**A**) Simple virtual CPU that executes avidian code. The CPU operates on three registers and two stacks. Input and output from and to the environment is achieved via dedicated I/O buffers. (**B**) Physical arrangement of programs on a 2D lattice in the `Avida` world. Different colors indicate different program genotypes.

Avidians are thus computer programs that self-replicate within a virtual world created for them inside of a computer. Unlike ordinary computer viruses, however, their replication mechanism is imperfect, leading to mutations which allow them to adapt to their world and grow in complexity. Their genetic code consists of instructions taken from a set designed to be simple and universal (in the sense that, again, almost any program can be written in that language). Naturally, many different such sets can be designed and indeed, because different instruction sets can be viewed as different types of *chemistry*, this opens up the opportunity to investigate how the dynamics of simple living systems depends on such a choice. In biochemistry, the equivalent would be the freedom to investigate biochemistries based on widely different sets of amino acids. In `avida`, we usually use a standard instruction set of 28 different instructions which are superficially similar to Intel i860 assembly instructions, liberally supplemented with instructions that allow self-replication. Examples of such instructions are `copy`, which copies an instruction from one memory location to another (and which has a probability to fail set by the mutation rate), `allocate`, which allocates memory space before replication can begin, and `divide`, which separates a mother and daughter program and places the daughter, either near the progenitor on the grid or anywhere, depending on the choice of geometry. Other instructions manipulate the virtual CPU's registers and stacks, and perform logical, bitwise, operations on random numbers that are abundant in the organism's environment. Different instruction sets (artificial chemistries) can affect the dynamics of the population markedly, in particular with respect to evolvability and robustness (Ofria et al. [2002]).

The heart of the avidian system is the energy metabolism of the organisms. The primary resource, without which no program can survive, is CPU time. It plays the role that the carbon source plays for bacteria in a Petri dish, except that it is not substitutable. CPU time is distributed in "time slices" to each organism in the population. The relative amount received by each organism depends on a number of factors. A default amount is distributed according to genome length, in order to make the replication process genome-length independent (the generation time is proportional to length). On top of that, CPU "bonus" time is given out for those programs that have developed computational genes. Such genes are stretches of program which read numbers from the environment, perform computations on them, and write them back out. Rewarded computations are, currently, logical operations on binary numbers, with up to three inputs (this is a total of 78 different operations). Since the only instruction available for such computations is a logical NAND[4], complex computations require the evolution of significantly complex code. It is worth noting that most evolutionary experiments are started with a simple ancestral genotype that is only 20 instructions long and whose only function is self-replication, yet organisms with sequence lengths of several hundred instructions performing a good fraction of all possible computations readily emerge in these experiments. Because it is these computations which provide the organism with the "energy" (in the form of CPU time) it needs to replicate, we can think of this computational code as the genes that code for the organism's metabolism. To this extent, we are able to observe the emergence of metabolic genes in self-replicating organisms, and thus the evolution of complexity (Adami et al. [1999]).

From a biological point of view, avidian populations are extremely simple, by design. In particular, it is possible in this scenario to make a clean distinction between the population and the environment that the population is adapting to, due to the fact that there is virtually no co-evolution in this system. On the other hand, many of the most interesting questions in evolutionary biology, and in ecology in particular, have to do with co-evolution of organisms and adaptation to local conditions, for example to depletable and substitutable local resources. In the following, I shall outline the type of changes that are required in the avidian system to investigate such questions, and discuss possible experiments in which the emergence, maintenance, and stability of simple ecosystems can be studied.

4. Resource competition and ecosystems in avida. As described above, most of the adaptive activity in `avida` is geared towards the evolution of "metabolic", i.e., computational, genes. Typical examples of computations ("tasks") being rewarded and performed would (for two inputs i and j) be[5] (i AND j), (i XOR j), (NOT i OR j), etc. In a sense, these genes can be viewed as the analog of exothermal catalytic reactions that are being carried out by the organisms, as they allow a more "efficient" exploitation of the primary resource, CPU time. The product of the reaction is the computational result, while the substrate it uses are the numbers available in the environment. In the standard avidian world, however, these numbers are inexhaustible and isotropically distributed. To move towards exhaustible and local resources, rather than limiting the numbers you can read, or where you can read them, each task instead is associated with an abstract resource, whose presence is required in order for the organisms to reap the reward of the computation. Let us imagine, for the moment, a simple world in which there are only three different possible tasks, say the ones introduced above. Then, we associate resource "A" with AND, "B" with NOR[6], and "C" with XOR. We can now load up the world with these resources, and we can limit them. For example, we can stipulate that every time an organism performs an AND, a certain amount of resource A is depleted, and similarly for the other resources. There are a number of obvious but important consequences of such a scenario. First of all, it is clear that computational genes can *only* evolve in regions where the corresponding resource is present. Thus, local differences in resource abundances will lead to different genes evolving in different areas, and a multi-niche population must form. Second, depletion of resources forces new selective pressures on the population, and strategies must be evolved to avoid starvation. Of course, for this to be nontrivial we must assume that there is a certain rate of substrate renewal, and the most economic way of implementing this is via a typical flow reactor.

At the same time, it appears reasonable to propose that performance of a computation in the presence of the enabling resource might transform the resource rather than use it up. In such a model for example, we could have

$$A \xrightarrow{\texttt{AND}} B,$$

$$B \xrightarrow{\texttt{NOR}} C,$$

$$C \xrightarrow{\texttt{XOR}} A.$$

Of course, in a system such as this, any type of resource chemistry can be implemented or tested. Still, extremely simple chemistries such as the one above already lead to a number of interesting questions and suggest experiments whose outcome is not immediately obvious.

For example, let us imagine we start an experiment in which three resources A, B, and C are distributed spatially such as in Figure 2. If resources are continuously renewed, but not transformed, specialists will evolve in the three different habitats, and no species can invade the other. If resources become scarce and if they are, through usage, transformed according to the rules above, we can expect two different scenarios. Either an organism will develop that carries all three genes (for AND, NOR, and XOR) making it effectively an autotroph, or we will witness colonies of heterotrophs forming at the boundary where the three resources A, B, and C meet in Figure 2. Such colonies would represent simple ecosystems whose members rely on each other's metabolic products.

After such colonies are formed (and surely we can imagine breeding much more complex ecosystems, built on more complex artificial resource chemistries) it becomes possible to study their stability. For

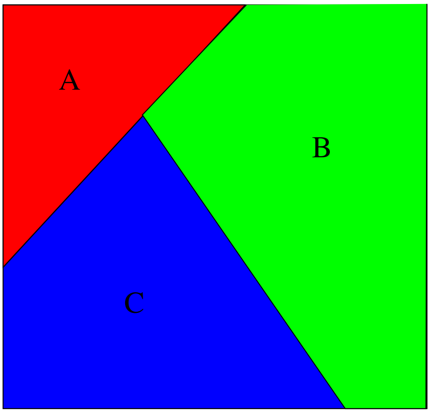

FIGURE 2. Spatial distribution of transformable resources for an experiment testing allopatric speciation, and the evolution of heterotrophy.

example, such ecosystems can be tested at different levels of starvation. Also, because it is possible to manipulate each species, and each resource as well as its strength (in units "bonus CPU time") separately, it ought to be possible to remove one species at a time in order to observe either a self-healing or a collapse of the ecosystem. Such a strategy might also serve to determine whether there are "keystone" species present in the ecosystem, whose removal is tantamount to ecosystem collapse.

Another question that might be addressed experimentally with such a system concerns the well-known "plankton paradox" of ecology, in which n species appear to be able to coexist on $k < n$ resources, violating the competitive exclusion principle (Hutchinson [1961]). A recent explanation of the paradox (Huisman and Weissing [1999]) invokes nonequilibrium dynamics of at least three species, which can coexist on less than three resources because three-species competition can generate sustained oscillations. In such a scenario, species displace each other in a cyclical fashion, dominating via usage of one resource while becoming limited by another. While such a scenario may indeed explain the $n > k$ puzzle, it is at this point entirely theoretical. Within Digital Life, experiments can be carried out to observe if, and under which circumstances, such oscillations do indeed arise, and whether they contribute to increasing the diversity of the population.

5. Conclusions. Individual-based modeling of ecosystems represents a powerful new tool for the study of complex ecosystems, but it generates its own sets of problems, both in the areas of validation and communication. At least for simple ecosystems composed of simple organisms, Artificial Life can be useful in addressing some issues fundamental to the ecological sciences, pertaining to the emergence, evolution, maintenance, and stability of such simple ecosystems. The simplicity of the system, while maintaining a complex fitness landscape, allows the design of experiments (and their controls) with conclusive results. In such a system, parameters can be set at will, and measurements can be recorded noise-free that perhaps are impossible to obtain in natural systems. At the same time, these results have to be reproducible and fit within a theoretical construct which allows an interpretation of the data. Experiments, according to the standard philosophy of science, are the potential "falsifiers" of theory. Without a framework to compare to, any amount of collected data is worthless.

While a system such as `avida` has nothing to say about realistic complex ecosystems of several trophic levels, such as are often the object of IMBs, it is conceivable that the type of experimental approach followed there, guided by theoretical insight and a tendency towards parsimony, may profit IBM practitioners too.

Acknowledgments. I would like to thank Steve Railsback for inviting me to speak at the 85th Annual Meeting of the Ecological Society of America (where part of this work was presented), Roland H. Lamberson for the opportunity to contribute to this volume, as well as an anonymous referee for valuable comments. This work was supported in part by the National Science Foundation under contract No. DEB-9981397. A portion of the research described in this paper was carried out at the Jet Propulsion Laboratory, under a contract with the National Aeronautics and Space Administration.

ENDNOTES

[1] `Avida` is free software and can be downloaded from the Digital Life Laboratory's site at `http://dllab.caltech.edu`

[2] The minimum complexity requirement is that a universal Turing machine can be constructed from these degrees of freedom.

3. In reality “almost any” function. Technically, at most partially recursive functions can be calculated by universal computers, a restriction due to the “Halting Problem”.

4. NAND is the logical operation which is the negation of the AND operation performed on two binary inputs. It can be used as a primitive to construct all possible logical operations.

5. Besides the computational primitive NAND mentioned above, there are several other distinct logical operations on two input bits, such as AND, OR, and XOR. The latter, “exclusive OR”, returns “true” (or the bit value “1”) only if one and only one of the two input bits are “true” (or “1”). XOR is an example of a difficult logical operation to perform armed with only NAND, and a skilled programmer needs at least 19 lines of avidian code to construct it. Nature, i.e., evolution, can do somewhat better.

6. NOR stands for NOT (i OR j), and is another independent logical operation on two input bits.

REFERENCES

C. Adami [1998], *Introduction to Artificial Life*, Springer, New York.

C. Adami and C.T. Brown [1994], *Artificial Life IV*, Proc. 4th Internat. Workshop on the Synthesis and Simulation of Living Systems (R.A. Brooks and P. Maes, eds.), MIT Press, Cambridge, MA, p. 377.

C. Adami, C. Ofria and T.C. Collier [1999], *Evolution of Biological Complexity*, Proc. Natl. Acad. Sci. USA **97**, 4463–4468.

C. Adami, R. Yirdaw and R. Seki [1998], *Critical Exponent of Species-Size Distribution in Evolution*, Proc. of Artificial Life VI (Los Angeles, June 27–29, 1998) (C. Adami, R. Belew, H. Kitano, and C. Taylor, eds.), MIT Press, Cambridge, MA, 221–227.

J. Chu and C. Adami [1997], *Propagation of Information in Populations of Self-Replicating Code*, Proc. of Artificial Life V (Nara, Japan, May 16–18, 1996) (C.G. Langton and T. Shimohara, eds.), MIT Press, Cambridge, MA, 462–469.

V. Grimm [1999], *Ten Years of Individual-Based Modeling in Ecology: What Have We Learned and What Could We Learn in the Future?* Ecol. Model. **115**, 129–148.

I. Harvey [1999], *Creatures from Another World*, Nature **400**, 618–619.

J. Huisman and F.J. Weissing [1999], *Biodiversity of Plankton by Species Oscillation and Chaos*, Nature **402**, 407–410.

G.E. Hutchinson [1961], *The Paradox of the Plankton*, Amer. Natur. **95**, 137–145.

C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, eds. [1992], *Artificial Life II*, Proc. 2nd Internat. Workshop on the Synthesis and Simulation of Living Systems, Addison Wesley, Redwood City.

R.E. Lenski, C. Ofria, T.C. Collier and C. Adami [1999], *Genome Complexity, Robustness and Genetic Interactions in Digital Organisms*, Nature **400**, 661–664.

C. Ofria and C. Adami [1999], *Evolution of Genetic Organization in Digital Organisms*, Proc. of DIMACS Workshop on Evolution as Computation (Princeton University, January 11–12, 1999) (L. Landweber and E. Winfree, eds.), Springer-Verlag, New York, p. 167.

C. Ofria, C. Adami and T.C. Collier [2002], *Design of Evolvable Computer Languages*, IEEE Trans. Evol. Comput. **6**.

C. Ofria, C. Adami, T.C. Collier and G. Hsu [1999], *Evolution of Differentiated Expression Patterns in Digital Organisms*, Lecture Notes in Artif. Intell. **1674**, 129–138.

S. Rasmussen, C. Knudsen, R. Feldberg and M. Hindsholm [1999], *The Core-World: Emergence and Evolution of Cooperative Structures in a Computational Chemistry*, Physica **D 42**, 111–134.

T.S. Ray [1992], *Artificial Life II*, Proc. 2nd Internat. Workshop on the Synthesis and Simulation of Living Systems (C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, eds.), Addison Wesley, Redwood City, p. 371.

A. Turing [1936], *On Computable Numbers with an Application to the* Entscheidungsproblem, Proc. London Math. Soc. Ser. 2 **43**, p. 544; **42**, p. 230.

D. Wagenaar and C. Adami [2000], *Influence of Chance, History and Adaptation on Evolution in* Digitalia, Proc. of Artificial Life VII (M.A. Bedau, J.S. McCaskill, N.H. Packard and S. Rasmussen, eds.), MIT Press, Cambridge, MA, 216–220.