

# The Evolutionary Dynamics of Digital and Nucleotide Codes: A Mutation Protection Perspective

William DeJong\*<sup>1</sup> and Hans Degens<sup>2</sup>

<sup>1</sup>INI-Research, innovation and change inquiry, P.O. Box 1073, 2600 BB Delft, The Netherlands

<sup>2</sup>Institute for Biomedical Research, into Human Movement and Health, Manchester Metropolitan University, John Dalton Building, Oxford Road, Manchester M1 5GD, UK

**Abstract:** Both digital codes in computers and nucleotide codes in cells are protected against mutations. Here we explore how mutation protection affects the random change and selection of digital and nucleotide codes. We illustrate our findings with a computer simulation of the evolution of a population of self replicating digital amoebae. We show that evolutionary programming of digital codes is a valid model for the evolution of nucleotide codes by random change within the boundaries of mutation protection, not for evolution by unbounded random change. Our mutation protection perspective enhances the understanding of the evolutionary dynamics of digital and nucleotide codes and its limitations, and reveals a paradox between the necessity of dysfunctioning mutation protection for evolution and its disadvantage for survival. Our mutation protection perspective suggests new directions for research into mutational robustness.

**Keywords:** Evolutionary change, evolutionary dynamics, evolutionary programming, mutation protection, mutational robustness.

## INTRODUCTION

Digital codes in the form of electronic, magnetic or optical strings of 1's and 0's are as omnipresent in information technology as nucleotide codes in the form of DNA strings of A's, C's, G's and T's are in living nature. Although both digital and nucleotide codes can be adapted to changing circumstances by random processes, they are protected against errors or mutations in an operating environment as computers and cells, respectively. We contend that evolutionary theory largely ignores mutation protection and the consequences of this protection for possible limitations of the evolutionary dynamics of digital and nucleotide codes. In this article, we describe these limitations in relation to the evolutionary programming literature and illustrate them with a computer simulation of the evolution of a population of digital amoebae. Our mutation protection perspective articulates the potential of evolutionary programming to simulate evolution, and enhances the understanding of the evolutionary dynamics of digital and nucleotide codes and its limitations. In addition, it articulates the ambiguity between 'mutational robustness' – that is the persistence of an organismal trait under genetic perturbations [1] – and evolvability [2, 3], and suggests new directions for research into mutational robustness.

## MUTATION PROTECTION OF DIGITAL AND NUCLEOTIDE CODES

In digital codes, the bits are usually packed into sets of eight bits called bytes. Normally, seven bits are used to record information and one bit – called the 'parity bit' – is

used for mutation protection, usually denoted as error protection [4]. The seven-bit string 1000001, for example, codes the letter A and 0110011 codes the number 3. If the number of 1's in the seven-bit string is even, the parity bit is given the value 1, else the value 0. If one of the seven information bits changes, for instance, by radiation, heat, or mechanical influences, they no longer correspond with the parity bit value; this is detected, an error message is generated, and the program stops, is aborted, or a switch is made towards a back-up. In heavy duty environments, often multiple back-ups are used to reach mutational robustness of the digital code [5].

In nucleotide codes, natural decay is also antagonized. Every human cell loses daily about 5000 adenine or guanine bases by depurination and about 100 cytosine bases by deamination [6]. Fortunately, the nucleotide code is continuously checked for mutations using the redundancy of information present in pairs of nucleotide strings, pairs of chromatids, and pairs of chromosomes, and repaired by a large repertoire of DNA repair enzymes [7, 8]. This redundancy based mutation protection shows a great resemblance with the redundancy based mutation protection of digital codes and seems of major importance for the mutational robustness of nucleotide codes [9, 10]. Because hereditary diseases and cancer are caused by irreparable mutations [11, 12], stringent safety protocols in industry and society are present to limit the mutative effects of radiation or chemicals [13]. As mutational robustness seems to antagonize evolution, much research into mutational robustness is aimed at resolving this ambiguity [14, 15].

## ADAPTATION OF DIGITAL AND NUCLEOTIDE CODES BY RANDOM PROCESSES

Digital codes are usually adapted deliberately by changing their control parameters or by (de)activating program

\*Address correspondence to this author at the INI-Research, innovation and change inquiry, P.O. Box 1073, 2600 BB Delft, The Netherlands; Tel: +31152850958; Fax: +31152850959; E-mail: dejong@ini-research.nl

modules present within the code, often by clicking on a set of menus. Adaptation can also be achieved by random processes using a computerized search strategy consisting of repeated cycles of random variation of control parameters or (de)activation of program modules followed by selection of a combination of advantageous parameters or modules, until an optimal combination of parameters or program modules is found. In this technique – called evolutionary programming or artificial evolution [16-18] – the search program that operationalizes the process of random variation, recombination, and selection, stays unchanged and limits the random change of the digital code 1) to the random variation of its control parameters within predefined boundaries and 2) to the random switching on or off of predefined program modules. Any other random change of the digital code – for instance the random change of individual bits or the expansion of the code by copying a random string of bits and inserting it elsewhere in the code – is prohibited by the mutation protection at the bit-level and by spelling and syntax error protection present at the higher levels of the code. Consequently, bounded random change of the digital code is present. When reviewing the literature on evolutionary programming, only bounded random change of digital codes can be found, even in the sophisticated AVIDA environment where a fixed set of predefined low-level computer instructions are combined at random resulting in independent programs that compete with one another for run time [19]. When a string of, for example, 80 predefined computer instructions is taken to move a computer processor from a predefined initial state to a predefined end state, random recombination of these instructions and giving competitive advantage to strings of instructions that consume little processor time can produce alternative routes to the end state that take about 30 instructions only [20]. During the optimizing process the predefined set of processor instructions stays unchanged and their recombination takes place within the boundaries of the mutation protection at the bit-level and the spelling and syntax error protection at the higher levels of the underlying digital codes. This limitation is also present in computer simulations of the construction of random nucleotide sequences by bioengineering and testing their effects on a biological system [17]. In this case a random sequence of predefined operators is constructed and fed into a computer processor that understands these operators. The random sequence of operators is constructed within the boundaries of mutation protection at the bit-level and the higher levels of the digital code. Therefore, also the computer simulation of this bioengineering technique provides no valid model for unbounded random change of a nucleotide code at the nucleotide-level.

Nucleotide codes can also be adapted by random processes. In contrast to digital codes, where program modules can only be switched on or off, program modules in nucleotide codes – genes – can have many gradations between being silent to being fully expressed and can cover a broad spectrum of possible effects. This provides nucleotide codes with a massive potential to adapt to changing circumstances; see for example the regulation of the lac operon in *E. coli* that tunes the production of three Lac proteins to the need to metabolize lactose [21]. Additional evolutionary potential is provided by the random recombination of gene variations – alleles – from the gene pool of a population by cross-over during the production of gametes and the generation and

selection of advantageous allele combinations. If, for example, the habitat of a population of Darwin finches changes and almost solely hard seeds are available, finches with a combination of alleles that produce a broad beak will survive, while during periods that small insects prevail finches with a combination of alleles that produce a sharp beak will become prevalent in the population [22]. In artificial breeding programs, the recombination and selection of alleles can produce a wide variety of dogs, pigeons, tulips etc. in a short time [23]. The recombination and selection act at the gene-level, function within the boundaries of mutation protection, and do not produce new alleles. Moreover, they do not expand the size of the nucleotide code but provide a final line of defense against propagation of gene code expanding mutations to the next generation by comparing alleles of the father with that of the mother; if they are not of exactly the same length, the cross-over fails and the production of gametes is aborted [24, 25].

### COMPUTER SIMULATION OF RANDOM CHANGE OF NUCLEOTIDE CODES

To illustrate the difference between random change of digital and nucleotide codes within the boundaries of mutation protection and unbounded random change, we present a computer simulation of the evolutionary dynamics of a population of digital amoebae. Each so called Damoeb consists of a small (3.3 Kbytes) C++ program that imports two digits from an input file, processes them into another digit, and exports it to an output file. The procession of the input depends on the value of a control parameter in the program code of the Damoeb, which can have the value 1, 2, 3 or 4, regulating the activation of the operator for summation, subtraction, division, or multiplication, respectively. A replication and random variation (RRV) program is used to make a copy of a Damoeb and to assign at random and with differing probabilities a value 1, 2, 3 or 4 to the control parameter of the copy Damoeb, resulting into an  $\alpha$ -type Damoeb, a  $\beta$ -type Damoeb, a  $\gamma$ -type Damoeb, or a  $\delta$ -type Damoeb, respectively; the copy Damoeb receives the control parameter value of the original Damoeb with a 94% chance, or one of the three alternative values of the control parameter with a 2% chance each. The RRV program simulates the exchange of alleles that are present in the gene pool of an amoeba population [26]; it can also be viewed as simulating gene regulation and the inheritance of gene expression to posterity [27, 28]. During a replication time interval  $\tau$ , a Damoeb enters the RRV-program once, and after an existence of  $5 \tau$  a Damoeb is deleted. The simulation is started with one  $\alpha$ -, one  $\beta$ -, one  $\gamma$ - and one  $\delta$ -type Damoeb. They are fed with the digit pair (20, 5) and allowed to replicate freely until the population consists of about 1000 Damoebes equally distributed over each type. Subsequently, selection rule S1 is imposed on the population, which allows only Damoebes that produce an output digit between 0 and 20 to replicate. Hereafter, the share of  $\beta$ - and  $\gamma$ -type Damoebes in the population grows strongly at the expense of the  $\alpha$ - and  $\delta$ -types, which produce an output digit of 25, respectively 100. The  $\alpha$ - and  $\delta$ -types, however, do not become extinct because the RRV-program allows them to arise sporadically from the replication of  $\beta$ - and  $\gamma$ -type Damoebes. After about 6 replication cycles of random variation and selection, the distribution of Damoeb-types reaches a new dynamic equilibrium. Next,

selection rule S1 is replaced by rule S2, allowing only Damoebes to reproduce that have an output greater than 50. Now, the population moves towards a distribution with mainly  $\delta$ -type Damoebes and a very small share of  $\alpha$ -,  $\beta$ -, and  $\gamma$ -type Damoebes. When S2 is replaced by selection rule S3, demanding the output to be between 0 and 10 or between 20 and 50, the  $\alpha$ - and  $\gamma$ -type Damoebes start to dominate the population. The population of Damoebes (Fig. 1) shows the same evolutionary dynamics as observed in for instance populations of bacteria or finches [22].

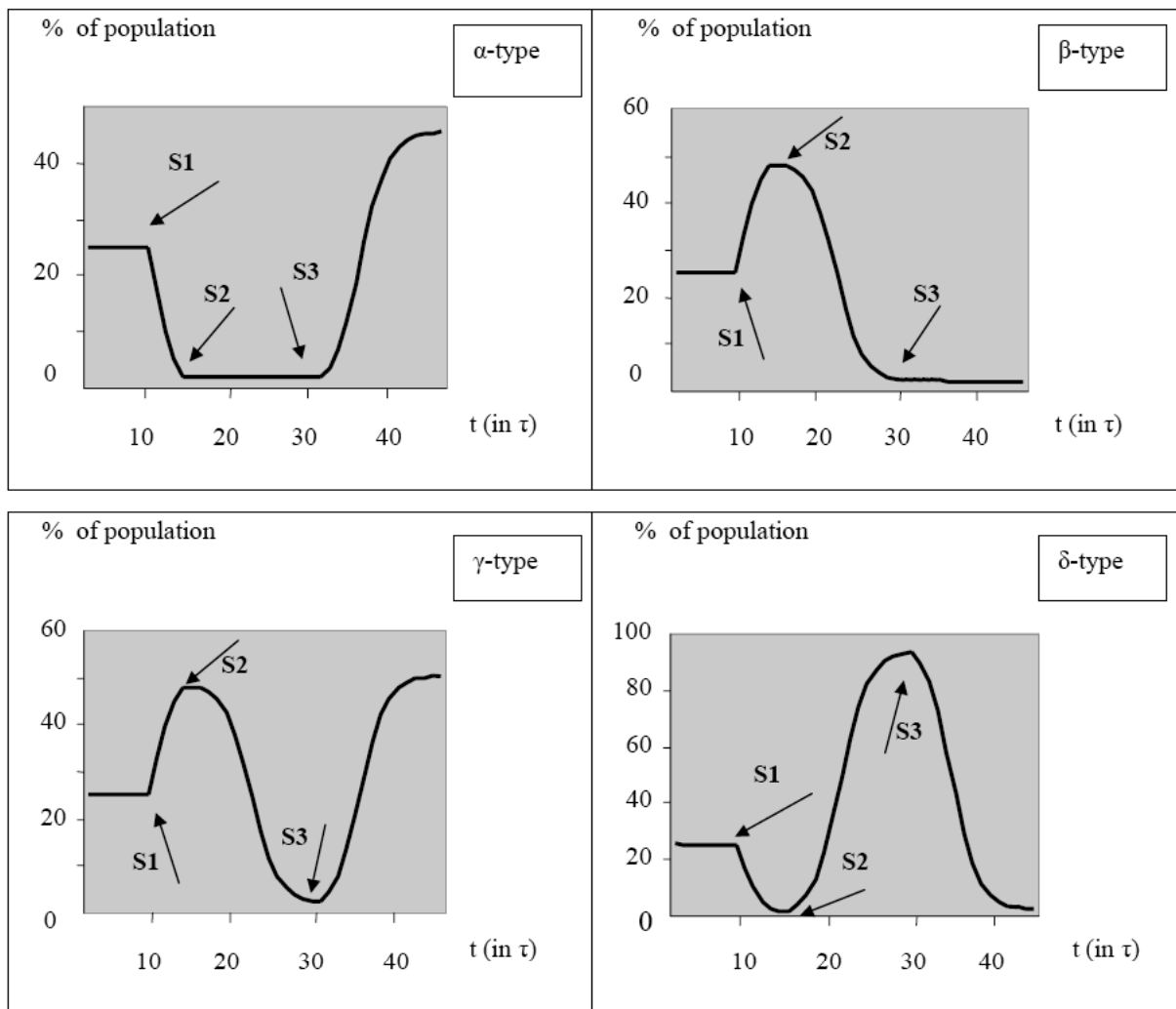
Random change of digital codes within the boundaries of the mutation protection present at the bit-level and the higher levels of the code can thus accurately simulate the evolutionary dynamics of populations of organisms. Random change of digital codes beyond the boundaries of the standard mutation protection, on the other hand, is impossible. This can be illustrated by expanding the RVV module with a submodule that randomly changes bits of the digital code of a Damoeb and inserts copies of random parts elsewhere in the code. When the expanded RVV module is taken into operation and a Damoeb is subjected to it, every mutated Damoeb generates error messages at the bit-level, or spelling and syntax

errors at the higher levels of the program code. Random change of digital codes beyond the boundaries of mutation protection thus requires the mutation protection to be switched off. In digital codes this is impossible because mutation protection is an intrinsic part of digital codes and the program languages in use. Therefore, unbounded random change of nucleotide codes – for instance by random change of individual nucleotides or insertion of a copy of a random part of the nucleotide code elsewhere in the code – cannot be simulated by unbounded random change of digital codes.

Our simulations show a population of Damoebes adapting to changing environmental pressures while the intrinsic mutation protection of the Damoebes is fully functioning. Evolvability and mutational robustness thus do not exclude one another, as other computer simulations confirm [15].

**EVOLUTIONARY DYNAMICS: LIMITATIONS AND MUTATION PROTECTION PARADOX**

Our mutation protection perspective advances the understanding of the evolutionary dynamics of both digital and nucleotide codes. It reveals that random change of digital codes is limited to the variation, recombination, and selec-



**Fig. (1).** Evolutionary dynamics of a Damoeb population as a response to selection rules S1, S2 and S3 imposed respectively at  $9 \tau$ ,  $15 \tau$ , and  $29 \tau$ , where  $\tau$  is the replication time of a Damoeb. For explanation of the selection rules see text.

tion of predefined parameters, operators, or program modules, as a consequence of the normal mutation protection of digital codes that is present at the bit-level and the higher levels of a digital code. The technique of evolutionary programming of digital codes by random processes operates within the boundaries of mutation protection and provides an excellent model for the random change of nucleotide codes by the mechanisms of gene-regulation and recombination of alleles from the gene pool of a population and selection of advantageous combinations. These mechanisms for random change of nucleotide codes operate within the boundaries of mutation protection present at the nucleotide-level and the higher levels of the code, and do neither produce new alleles nor expand the length of the nucleotide code.

Unbounded random change of digital codes – for instance by random change of bits or by copying a random string of bits and inserting it elsewhere in the code – is impossible because the normal mutation protection is an intrinsic part of digital codes and the program languages in use, and cannot be switched off. Therefore, evolutionary programming of digital codes is only a valid model for the evolution of nucleotide codes by random change within the boundaries of mutation protection, not for the evolution of nucleotide codes by unbounded random change.

Unbounded random change of nucleotide codes through the accumulation of irreparable, advantageous, code-expanding, inheritable mutations at the level of individual nucleotides, as proposed by evolutionary theory, requires the mutation protection at the level of the individual nucleotides and at the higher levels of the code to be switched off or at least to dysfunction. Dysfunctioning mutation protection, however, is the origin of cancer and hereditary diseases, which reduce the capacity to live and to reproduce. Our mutation protection perspective of the evolutionary dynamics of digital and nucleotide codes thus reveals the presence of a paradox in evolutionary theory between the necessity and the disadvantage of dysfunctioning mutation protection. This mutation protection paradox, which is closely related with the paradox between evolvability and mutational robustness, needs further investigation. The research may focus on the relationships between the redundancy-based mutation repair systems, the buffering and inheritance of irreparable mutations, the evolvability of mutation repair, and the capacity to live of organisms. The research into mutational robustness may profit from the insights acquired in the research of cancer and hereditary diseases [12]. When investigating mutational robustness a clear distinction between random code change within the boundaries of mutation protection and beyond these boundaries appears essential.

## REFERENCES

[1] Félix, M.A.; Wagner, A. Robustness and evolution: concepts, insights and challenges from a developmental model system. *Heredity*, **2008**, *100*, 132-140.

- [2] Wagner, A. *Robustness and Evolvability in Living Systems*. Princeton University Press: Princeton NJ, **2005**.
- [3] de Visser, J.A.; Hermisson, J.; Wagner, G.P.; Ancel Meyers, L.; Bagheri-Chaichian, H.; Blanchard, J.L.; Chao, L.; Cheverud, J.M.; Elena, S.F.; Fontana, W.; Gibson, G.; Hansen, T.F.; Krakauer, D.; Lewontin, R.C.; Ofria, C.; Rice, S.H.; von Dassow, G.; Wagner, A.; Whitlock, M.C. Perspective: Evolution and detection of genetic robustness. *Evolution*, **2003**, *57*(9), 1959-72.
- [4] Linux Information Project. *ASCII: A Brief Introduction*. [http://www.linfo.org/ascii.html] accessed January **2011**.
- [5] Singh, A.D.; Murugesan, S. Fault-Tolerant Systems. *IEEE Computer*, **1990**, *7*, 15-18.
- [6] Alberts, B.; Johnson, A.; Lewis, J.; Raff, M.; Roberts, K.; Walter, P. *Molecular Biology of the Cell*, 4<sup>th</sup> ed. Garland Science: New York, **2002**.
- [7] Friedberg, I.C.; Walker, G.C.; Siede, W. *DNA Repair and Mutagenesis*, American Society of Microbiology Press: Washington DC, **1995**.
- [8] Wood, R.D.; Mitchell, M.; Lindahl, T. Human DNA repair genes. *J. Mutation Res.*, **2005**, *577*(1-2), 275-283.
- [9] Pearson, H. Surviving a knockout blow. *Nature*, **2002**, *415*, 8-9.
- [10] Lolle, S.J.; Victor, J.L.; Young, J.M.; Pruitt, R.E. Genome-wide non-mendelian inheritance of extra-genome information in Arabidopsis. *Nature*, **2005**, *434*, 505-509.
- [11] Beral, V.; Reeves, G.; Shigamatsu, I.; Theissen, J.W. Childhood thyroid cancer in Belarus. *Nature*, **1992**, *359*, 680-681.
- [12] COSMIC, Catalogue of Somatic Mutations in Cancer, [http://www.sanger.ac.uk/genetics/CGP/cosmic/] accessed January **2011**.
- [13] American Institute for Cancer Research. *Food, Nutrition, and the Prevention of Cancer: A Global Perspective*. Author. Washington DC, **1997**.
- [14] Wagner, A. Robustness and evolvability: A paradox resolved. *Proc. Roy. Soc. London Series*, **2008**, *B 275*, 91-100.
- [15] Draghi, J.A.; Parsons, T.L.; Wagner, G.P.; Plotkin, J.B. Mutational robustness can facilitate adaptation. *Nature*, **2010**, *463*, 353-355.
- [16] Fogel, G.B.; Corne, D.W. *Evolutionary Computation in Bioinformatics*, Elsevier Science: San Francisco CA, **2003**.
- [17] Foster, J.A. Evolutionary computation. *Nat. Genet.*, **2001**, *2*, 428-436.
- [18] Koza, J.R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press: Cambridge MA, **1992**.
- [19] Adami, C.; Brown, C.T. Evolutionary Learning in the 2D Artificial Life Systems Avida. In: *Proc. Artificial Life IV*; Brooks, R.; Maes, P. Eds.; MIT Press: Cambridge MA, **1994**, pp. 377-381.
- [20] Yedid, G.; Bell, G. Macroevolution simulated with autonomously replicating computer programs. *Nature*, **2002**, *420*, 810-812.
- [21] Jacob, F.; Monod, J. Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, **1961**, *3*, 318-356.
- [22] Gibbs, H.L.; Grant, P.R. Oscillating selection on Darwin's finches. *Nature*, **1987**, *327*, 511-513.
- [23] Cadieu, E. Coat variation in the domestic dog is governed by variants in three genes. *Science*, **2009**, *2*, 150-153.
- [24] Kimball, J.W. *Biology*, 6<sup>th</sup> ed., Wm. C. Brown Publishers: Dubuque, **1994**.
- [25] Kimball, J.W. The biology pages, crossing over and genetic recombination in meiosis. [http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/C/CrossingOver.html] accessed January **2011**.
- [26] Awadalla, P. The evolutionary genomics of pathogen recombination. *Nat. Genet.*, **2003**, *4*, 50-59.
- [27] Pembrey, M. E. Time to take epigenetic inheritance seriously. *Eur. J. Hum. Genet.*, **2002**, *10*, 669-671.
- [28] Perdew, G.H.; Vanden Heuvel, J.P.; Peters, J.M. *Regulation of Gene Expression*. Humana Press: Totowa NJ, **2006**.

Received: October 10, 2010

Revised: February 13, 2011

Accepted: February 15, 2011

© DeJong and Degens; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.