A cellular automaton model for the study of DNA sequence evolution

G.Ch. Sirakoulis[a], I. Karafyllidis[b,*], Ch. Mizas[a], V. Mardiris[a], A. Thanailakis[b], Ph. Tsalides[b]

[a] *Ulysses Ltd., Pipinou 4, Kavala 65201, Greece*
[b] *Department of Electrical and Computer Engineering, Democritus University of Thrace, Laboratory of Electrical and Electronic Materials Technology, Xanthi GR-671 00, Greece*

Abstract

Cellular automata are introduced as a model for DNA structure, function and evolution. DNA is modeled as a one-dimensional cellular automaton with four states per cell. These states are the four DNA bases A, C, T and G. The four states are represented by numbers of the quaternary number system. Linear evolution rules, represented by square matrices, are considered. Based on this model a simulator of DNA evolution is developed and simulation results are presented. This simulator has a user-friendly input interface and can be used for the study of DNA evolution.
© 2003 Elsevier Ltd. All rights reserved.

1. Introduction

Biologists and Computer Scientists and Engineers have recently put combined efforts in the interdisciplinary task to understand the information storage and processing in DNA [1,2], giving thus rise to Bioinformatics. Bioinformatics may be defined as a discipline that generates computer tools, databases, hardware, algorithms and methods to support genomic and post-genomic research. It comprises the study of DNA structure, function and evolution, gene and protein expression, protein production, structure and function, genetic regulatory systems and clinical applications [3,4].

Methods successfully used in Computer Science and Engineering have recently been used in modeling and simulation of the DNA structure, function and evolution [3,4]. Because of the vast amount

of information stored into the DNA structure new models, algorithms and dedicated processors are expected to be developed soon.

The aim of this work is to introduce cellular automata (CAs) as a model for DNA structure, function and evolution. A simulator with a user-friendly input interface was developed. This simulator can be used for the study of DNA evolution. CAs, originally developed by John von Neumann as models of self-reproducing systems [5], have been extensively used to model and simulate physical systems and processes [6–9]. Furthermore CAs have been successfully used in modeling and simulations of Environmental and Biological systems, such as forest fire spreading [10], oil slick movement and spreading [11], greenhouse effect on insect and microorganism geographical distribution and population dynamics [12], effects of population movement and vaccination on epidemic propagation [13], tumor invasion and growth [14,15], and dynamics of the evolution of HIV infection [16]. CAs have also been used as high performance simulators of the immune system [17,18].

This paper is organized as follows: All the necessary background in CAs is given in Section 2. In Section 3, DNA is modeled as a one-dimensional CA, and in Section 4 this model is used to simulate DNA evolution. In Section 5, the graphical user interface is described and some examples of its use are presented. The conclusions of this work are presented in Section 6.

2. Cellular automata

CAs were originally introduced by von Neumann [5] and Ulam [19] as a possible idealization of biological systems, with the particular purpose of modeling biological self-reproduction. Since then CAs have been reinvented several times under various names such as "cellular spaces," "tessellation automata," "cellular structures," "cellular spaces" and "iterative arrays" [8]. During the last two decades CAs have been extensively used as mathematical idealizations of physical systems in which space and time are discrete, interactions are local and physical quantities take on a finite set of discrete values. DNA will be modeled in this paper as a one-dimensional CA and, therefore, only one-dimensional CAs will be presented in this section.

A one-dimensional CA consists of a regular uniform lattice, which may be infinite in size and expands in a one-dimensional space. Each site of this lattice is called *cell*. At each cell a variable takes values from a discrete set. The value of this variable is the *state* of the cell. Fig. 1(a) shows a one-dimensional CA. The CA lattice consists of identical cells, $\ldots, i-3, i-2, i-1, i, i+1, i+2, i+3, \ldots$, and the corresponding states of these cells are $C_{i-3}, C_{i-2}, C_{i-1}, C_i, C_{i+1}, C_{i+2}$ and C_{i+3}.

The state of the ith cell takes values from a predefined discrete set:

$$C_i \in \{c_1, c_2, c_3, \ldots, c_n\}, \tag{1}$$

where $c_1, c_2, c_3, \ldots, c_n$ are the elements of the set. This set may be a set of integers, a set of real numbers, a set of atoms, a set of molecules, or even a set of properties. If the set contains only the two binary numbers, i.e. $C_i \in \{0, 1\}$, the CA is called *elementary*.

The CA is a dynamic system, which *evolves* in time. The CA evolves in discrete time steps and its evolution is manifested by the change of its cell states with time. The state of each cell is affected by the states of its neighboring cells. All the cells that affect the change of the state of the ith cell are the *neighborhood* of this cell. The neighborhood is defined as follows:

$$N(i,r) = \{C_{i-r}, \ldots, C_{i-3}, C_{i-2}, C_{i-1}, C_i, C_{i+1}, C_{i+2}, C_{i+3}, \ldots, C_{i+r}\}, \ \ r = 0, 1, 2, 3, \ldots, m, \tag{2}$$
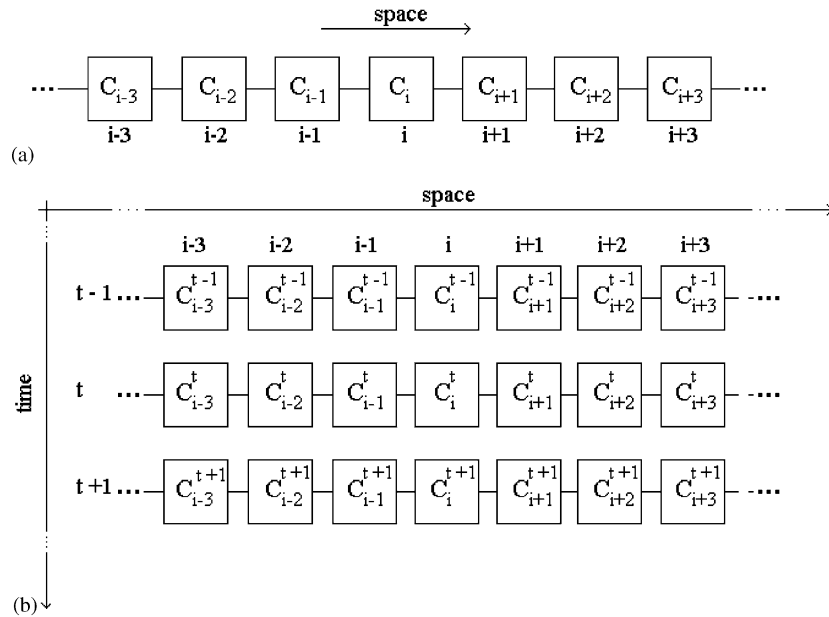
Fig. 1. (a) A one-dimensional CA, (b) the evolution of a one-dimensional CA.

where r is the size of the neighborhood. If $r = 1$, which is the most usual case then the neighborhood of the ith cell consists of the same cell and its left and right immediate neighbors:

$$N(i, 1) = \{C_{i-1}, C_i, C_{i+1}\}. \tag{3}$$

The state of the ith cell at time step $t + 1$ is affected by the states of its neighbors at the previous time step t, i.e. the state of the ith cell at a time step is a function of the states of its neighbors at the previous time step:

$$C_i^{t+1} = F(C_{i-r}^t, \ldots, C_{i-3}^t, C_{i-2}^t, C_{i-1}^t, C_i^t, C_{i+1}^t, C_{i+2}^t, C_{i+3}^t, \ldots, C_{i+r}^t). \tag{4}$$

This function is the CA *evolution rule*. The upper index in the state symbol denotes the time step. C_i^{t+1} is the state of the ith cell at time step $t + 1$. If $r = 1$, Eq. (4) becomes

$$C_i^{t+1} = F(C_{i-1}^t, C_i^t, C_{i+1}^t). \tag{5}$$

Fig. 1(b) shows the evolution of a one-dimensional CA. The horizontal axis is *space* and the vertical axis is *time*. Each row represents the CA at each time step and each column represents the state of the same cell at various time steps.

3. Cellular automaton model of DNA

A schematic DNA structure is shown in Fig. 2(a). DNA can be modeled as a one-dimensional CA. In this model, the phosphate chain corresponds to the CA lattice and the deoxyribose sugars to the CA cells. At each sugar molecule one of the four bases A, C, T and G may bind. These four
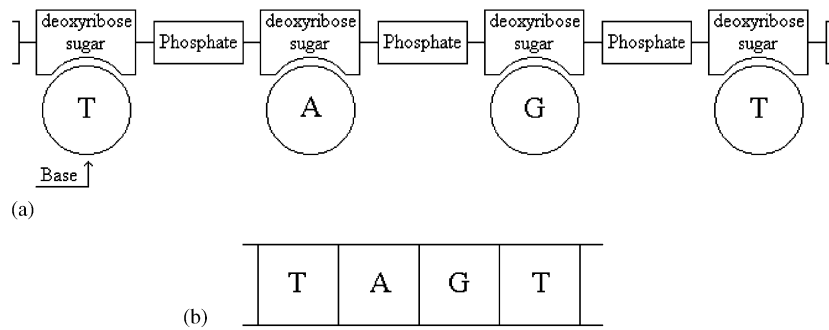
Fig. 2. (a) A schematic DNA structure, (b) the CA that models the DNA structure.

bases correspond to the four possible states of the CA cell. Fig. 2(b) shows the CA that models the DNA structure shown in Fig. 2(a).

The state of the ith cell of this CA takes values from the discrete set that comprises the four bases:

$$C_i \in \{A, C, T, G\}. \tag{6}$$

In non-sexual reproduction, the DNA molecule is passed from an individual to its offspring, whereas in sexual reproduction, the DNA of the offspring consists of parts of the parental DNA. We define as an *evolution event* a change in state, which may occur in one or more CA cells. Therefore, mutation is an evolution event and it corresponds to cell state changes. In the case of non-sexual reproduction, if a DNA strand is passed unaltered from one generation to the other, then no state change occurs, and the CA does not evolve. The CA evolves if a change in one of its cells occurs, either during the reproduction process or during the life of the individual carrying the DNA.

The time step in CA evolution is the time interval between two CA cell changes and, therefore, the time flow is not uniform. Consider for example a non-sexual reproducing species that the mean life of its individuals is 1 year. Suppose that a CA cell state change (DNA mutation) occurs now, the next one occurs in 10 years, the next one in 3 years and the next one in 6 months. Then the first time step represents 10 years of real time, the second 3 years and the third 6 months. But, in the CA model all time steps are equivalent, i.e. the difference in real time between the first, the second and the third time step does not become evident. A result of modeling DNA as a CA is that the DNA strand and the individuals passing it from one generation to the other may exist in different time scales and, therefore, the DNA evolution is time-like separated from the life of the individuals that carry it.

The main question that rises when one tries to model DNA is whether mutations are completely random or not. As explained above, DNA mutations are represented by CA cell state changes. If mutations are completely random, then CAs, which are deterministic computational models, cannot model DNA evolution. In this case probabilistic methods, such as Markov chains may be appropriate. Although the answer to this question is not known, there are some indications that mutations and, therefore, DNA evolution may not be completely random [20,21]. One of the indications is that life on earth is about 10^{17} s old, whereas the DNA of complex mammals comprises about 10^9 bases. The evolution of such complex living beings in this relatively very short time period is an indication that evolution is not completely random, but may be determined by some evolution rules [21].

We will proceed to the model construction by assuming that mutations, i.e. CA cell changes are not completely random, but depend on the states of some of the cells that are located near by. Neighbor-dependent mutation has been studied using Markov chains and revealed biases in mutation rates that depend on the neighboring bases [22]. Suppose that a state change at the ith cell occurs, and a time step is taken. In the model, presented here it is supposed that the state of this cell has changed as a result of the effect of the states of its neighbors. The new state of the ith cell at this time step (which is generally the $t + 1$ step) is given by

$$C_i^{t+1} = \hat{M}(C_{i-r}^t, \ldots, C_{i-3}^t, C_{i-2}^t C_{i-1}^t, C_i^t, C_{i+1}^t, C_{i+2}^t, C_{i+3}^t, \ldots, C_{i+r}^t). \tag{7}$$

Eq. (7) is a more general expression of the evolution rule given in Eq. (4), where the function F has been replaced by an operator, $\hat{M}$, which is a more general mathematical abstraction. An operator may be a mathematical function, a logic function, a matrix, etc. The operator operates on the state of the neighborhood of the ith cell at time step t and produces the state of this cell at time step $t + 1$.

In Eq. (7) cell states are one of the four bases A, C, T and G. Operators act on numbers and symbols that represent numbers. Therefore, the four bases must be represented by numbers. Since there are only four bases, the most appropriate way of representing them by numbers is to correspond each one of them to a respective number of the quaternary number system, which contains only four numbers, i.e. 0, 1, 2 and 3. We represent the bases with numbers as follows:

$$A \to 0, C \to 1, T \to 2, G \to 3. \tag{8}$$

A vast number of evolution rules can be applied to the CA that models DNA. Furthermore, evolution rules that include base insertion *and/or* base deletion may be used. Usually, when a new CA is proposed, the linear evolution rules are the first ones to be studied. The study of linear rules reveals the dynamics of the CA evolution and provides a very good insight to the structures created by evolution. The use of linear rules is further justified by the fact that a linear algebra has already been successfully used to the analysis of mutation rates [23].

In the case of linear evolution rules the operator $\hat{M}$ of Eq. (7) is a matrix, M, and the evolution rule takes the form

$$
\begin{bmatrix} \vdots \\ C_{i-2}^{t+1} \\ C_{i-1}^{t+1} \\ C_i^{t+1} \\ C_{i+1}^{t+1} \\ C_{i+2}^{t+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & M_{i-2,j-2} & M_{i-2,j-1} & M_{i-2,j} & M_{i-2,j+1} & M_{i-2,j+2} & \cdots \\ \cdots & M_{i-1,j-2} & M_{i-1,j-1} & M_{i-1,j} & M_{i-1,j+1} & M_{i-1,j+2} & \cdots \\ \cdots & M_{i,j-2} & M_{i,j-1} & M_{i,j} & M_{i,j+1} & M_{i,j+2} & \cdots \\ \cdots & M_{i+1,j-2} & M_{i+1,j-1} & M_{i+1,j} & M_{i+1,j+1} & M_{i+1,j+2} & \cdots \\ \cdots & M_{i+2,j-2} & M_{i+2,j-1} & M_{i+2,j} & M_{i+2,j+1} & M_{i+2,j+2} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ C_{i-2}^t \\ C_{i-1}^t \\ C_i^t \\ C_{i+1}^t \\ C_{i+2}^t \\ \vdots \end{bmatrix}. \tag{9}
$$

The column matrix at the right-hand side of Eq. (9) is formed by the states of all CA cells at time step t. This matrix is multiplied by the matrix M, which represents the evolution rule. The

matrix elements $M_{i,j}$ may take only two values, namely 0 and 1. The column matrix at the left-hand side of Eq. (9) is the result of the matrix multiplication and it contains the states of all CA cells at time step $t + 1$. All the additions are modulo 4 additions. In the case of a CA with n cells (DNA strand with n bases) the column matrices have n rows and the matrix M is a square matrix with n columns and n rows. Each square matrix M represents a CA rule. Consider, for example, a very small DNA strand which at present time t has seven bases: { G, C, T, G, A, G, T }. This strand is represented by the following numbers: $\{3, 1, 2, 3, 0, 3, 2\}$. Suppose that this DNA strand evolves according to the following evolution rule:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

The CA state at the next time step is calculated using Eq. (9) as follows:

$$\begin{bmatrix} 3 \\ 1 \\ 2 \\ 2 \\ 0 \\ 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 2 \\ 3 \\ 0 \\ 3 \\ 2 \end{bmatrix}. \tag{11}$$

It is reminded that the additions are modulo 4. The CA state at time $t + 1$ is $\{3, 1, 2, 2, 0, 3, 2\}$ and the DNA strand at this time is $\{G, C, T, T, A, G, T\}$. The fourth base has changed from G to T. Consider another evolution rule applied to the same DNA strand, given by the matrix:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12}$$

In this case the CA state at time $t+1$ is {3, 1, 3, 3, 2, 3, 2} and the DNA strand at this time is {G, C, G, G, T, G, T}. The third base changed from T to G and the fifth from A to T. As mentioned before each (different) matrix M corresponds to a different evolution rule.

In elementary CAs, given an evolution pattern the evolution rule that generated it can be determined [24,25]. It is very probable for such a method to exist for CAs that model DNA. In this case if the evolution of the DNA strand at various time steps is given, it will be possible to determine the evolution rule (or rules) that generated the evolution. After that, since the evolution rule and the DNA strand at present time are known, it may be possible to predict the next evolution event (or events) and, therefore, the DNA strand at the next time step.

4. Simulation of DNA sequence evolution using the proposed model

The model developed in the previous section will now be used to simulate the evolution of DNA sequences. Most of the studies on mathematical models of DNA are limited to nearest-neighbor interaction [26]. Because of that, we have chosen to use in our simulations an evolution rule that incorporates only nearest-neighbor interaction, and it is given by the following matrix:

$$
M = \begin{bmatrix}
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & 1 & 1 & 0 & 0 & 0 & \cdots \\
\cdots & 1 & 1 & 1 & 0 & 0 & \cdots \\
\cdots & 0 & 1 & 1 & 1 & 0 & \cdots \\
\cdots & 0 & 0 & 1 & 1 & 1 & \cdots \\
\cdots & 0 & 0 & 0 & 1 & 1 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix} .
\tag{13}
$$

All the elements in a matrix row are zero, except the three neighboring elements that are equal to one. If this matrix is multiplied by the column matrix formed by the states of all CA cells, at time step t, the state of the ith element at time step $t+1$ will be the modulo 4 addition of its own state and the states of its left and right neighbors (cells $i-1$ and $i+1$, respectively), at time step t.

Fig. 3(a) shows the simulated evolution of a DNA sequence. The simulation starts with a random sequence of a DNA strand with 30 bases and produces the strands for 30 successive time steps. Base A is shown in white, C in light gray, T in dark gray and G in black. Fig. 3(b) shows the number of cells with the same DNA base at various time steps.

Fig. 4(a) shows the simulated evolution of a periodic DNA sequence of 30 bases for 30 time steps. The initial sequence comprises repetitions of the triplet CCT. Fig. 4(b) shows the number of cells with the same DNA base at various time steps.

Fig. 5(a) shows the simulated evolution of a random DNA sequence of 30 bases for 30 time steps. In this case, only a part of the sequence changes, whereas the rest remains unaltered. Fig. 5(b) shows the number of cells with the same DNA base at various time steps.

These simulations show that the evolution data visualization is straightforward, and the evolution patterns can be easily studied and interpreted. The simulator presented in this section is available from http://www.ulyssestech.com.
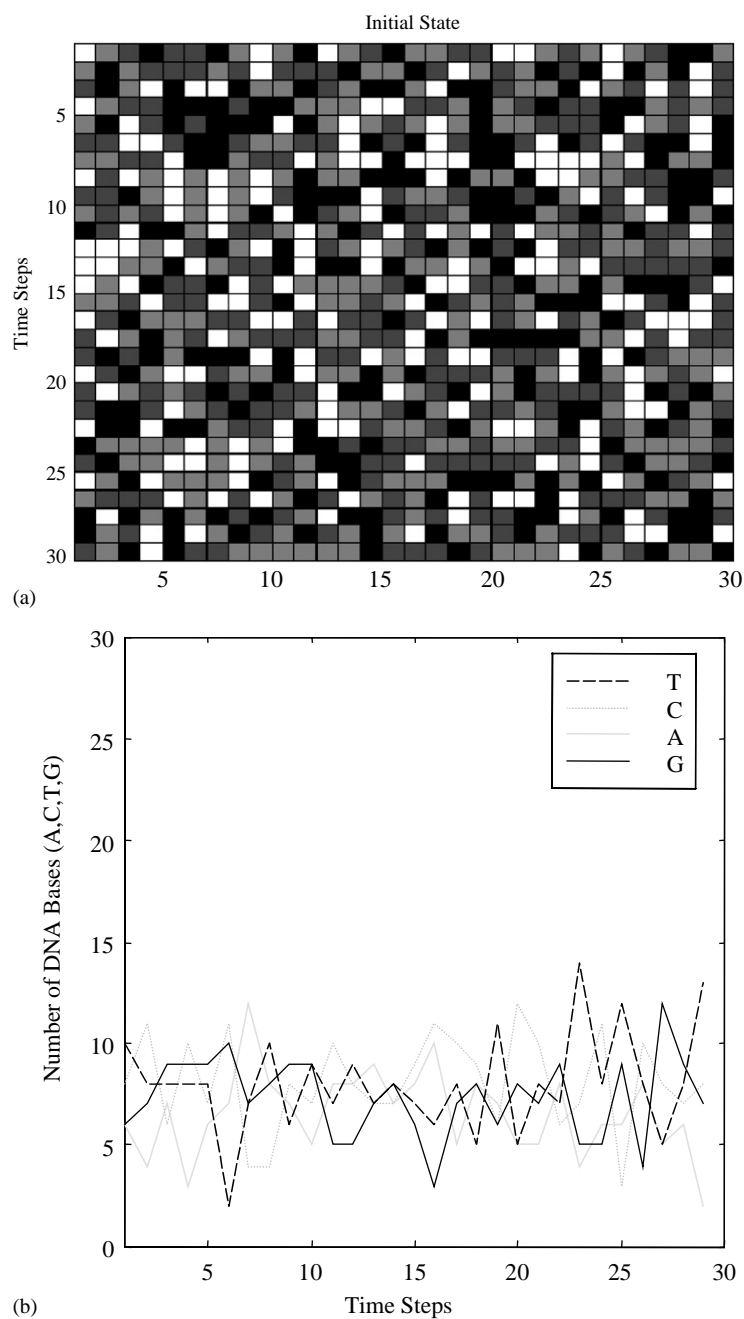
Fig. 3. (a) Simulated evolution of a random DNA sequence, (b) the number of cells with the same DNA base at various time steps. (A: white, C: dark gray, T: light gray and G: black).
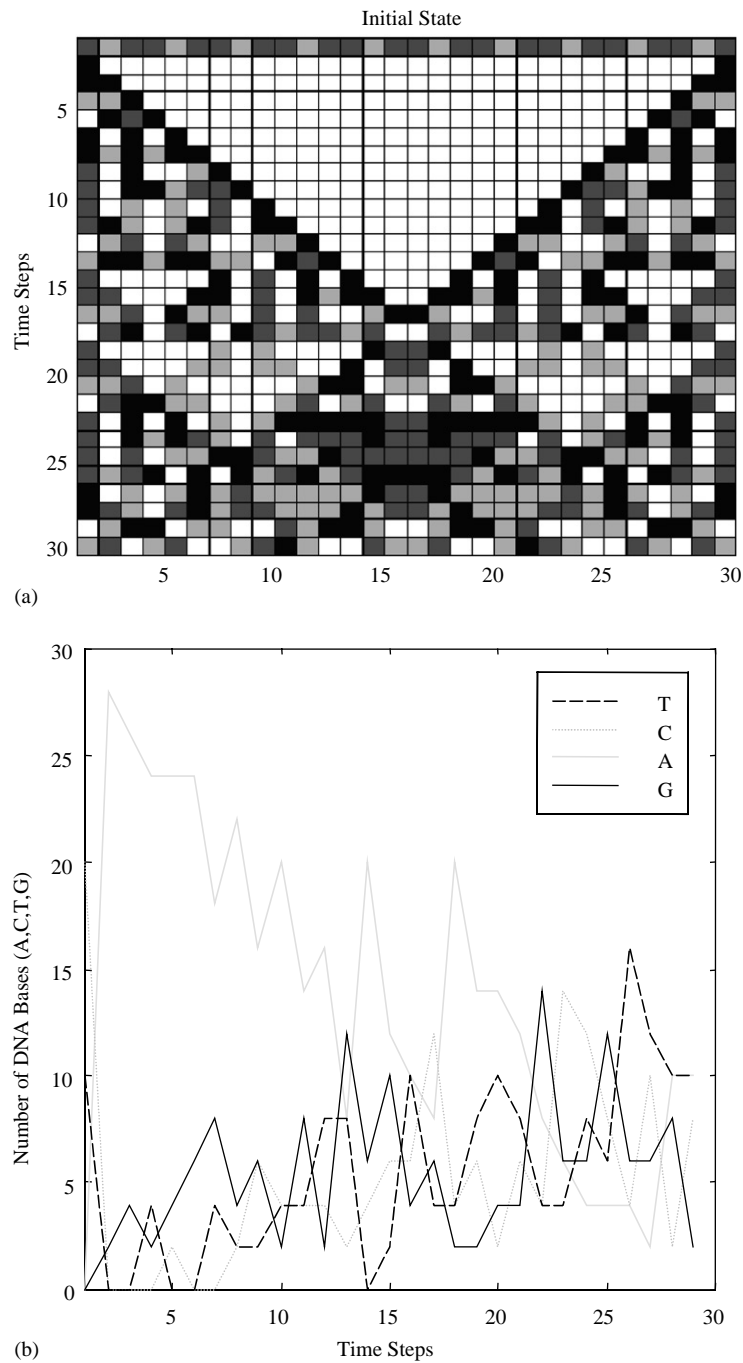
Initial State

(a)

(b)

Fig. 4. (a) Simulated evolution of a periodic DNA sequence, (b) the number of cells with the same DNA base at various time steps. (A: white, C: dark gray, T: light gray and G: black).
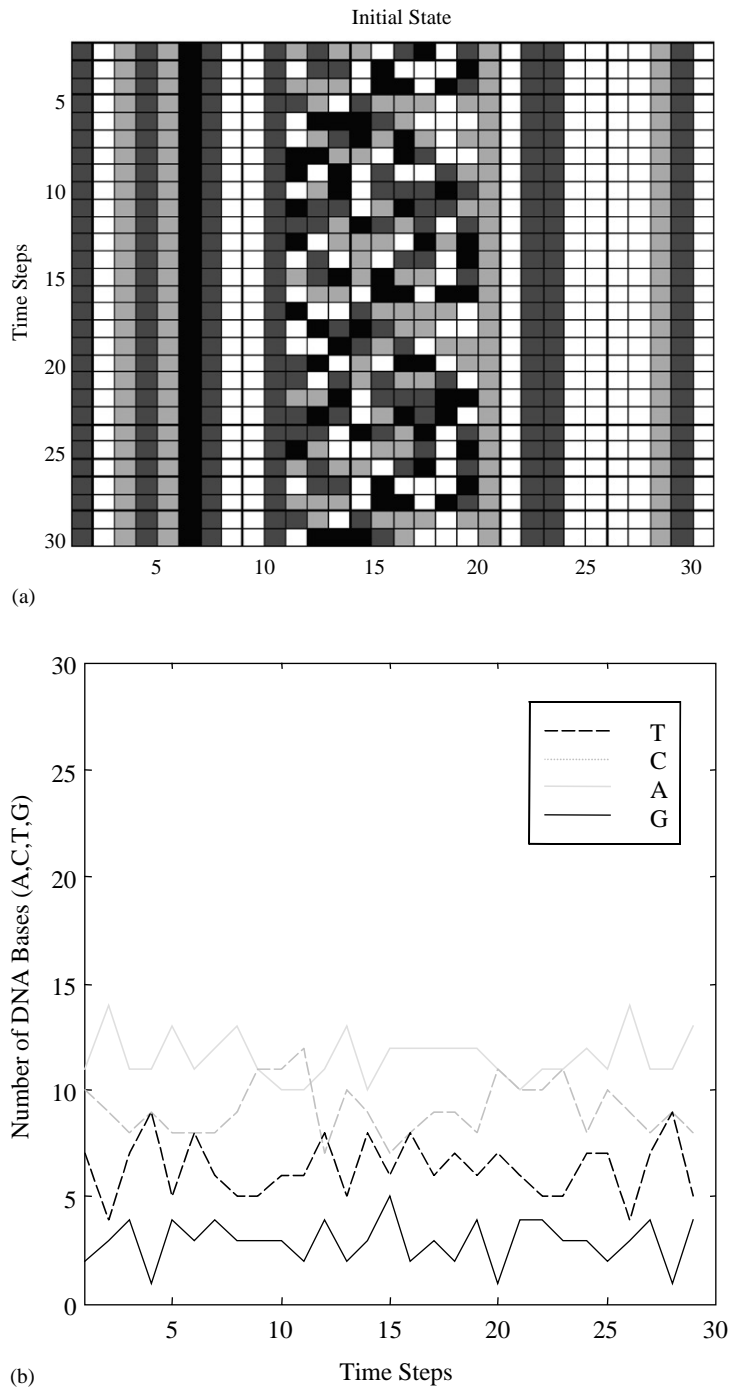
Fig. 5. (a) Simulated evolution of a random DNA sequence. In this case only a part of the sequence changes, (b) the number of cells with the same DNA base at various time steps. (A: white, C: dark gray, T: light gray and G: black).

5. The graphical user interface of the simulator

The simulator can be used as a computer tool for the study of DNA evolution. No previous knowledge of CAs or computer programming is necessary to use the simulator, because of the user-friendly graphical user interface that was developed.

The graphical user interface is shown in Fig. 6. In the field "Initial DNA sequence" the user inserts the sequence which will be used as initial. By clicking on the radio button beside the "Default DNA sequence 1", a previously defined DNA sequence is used as initial. There are three predefined initial sequences that are used in order to familiarize the user with the interface. By clicking on the radio button beside the "Random DNA sequence", a randomly generated DNA sequence is used as initial. A random number generator that has been incorporated into the simulator generates the sequence. Therefore, a different sequence is generated each time this radio button is selected. The user can enter his/hers own DNA sequences by clicking on the radio button beside the "Manual DNA sequence". After that he/she can enter the sequence into the blank field on the right side of "Manual DNA sequence". The user enters a sequence of the capital letters A, C, G and T.

The number of evolution time steps is entered in the field "Set Maximum time of DNA evolution". The user can use the default number of time steps, which is 30, or enter another number by clicking on the radio button "Set at." After that the number is entered into the blank field on the right.
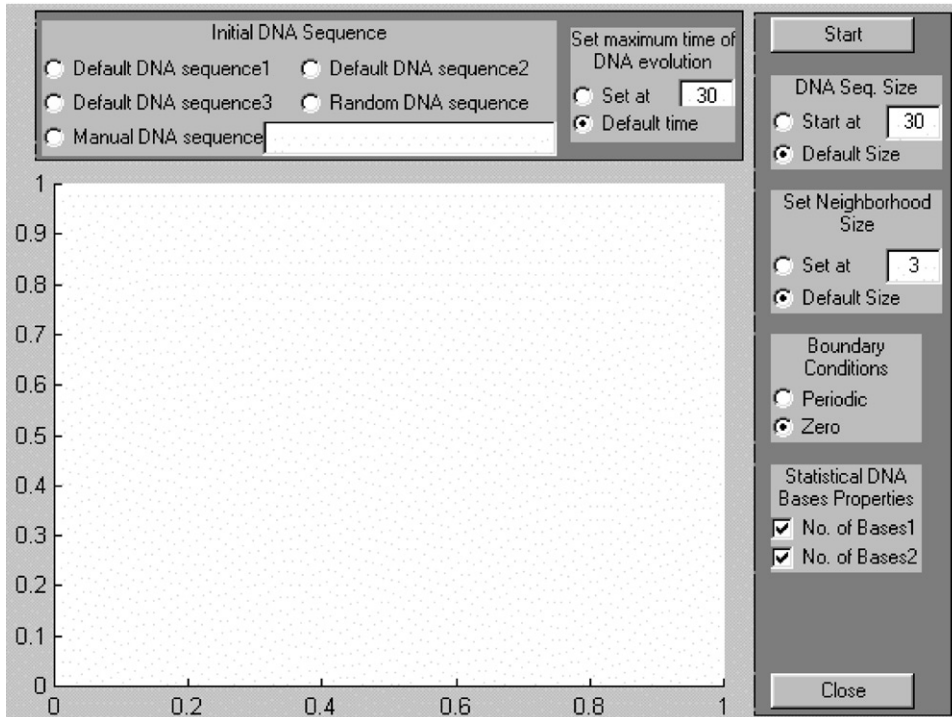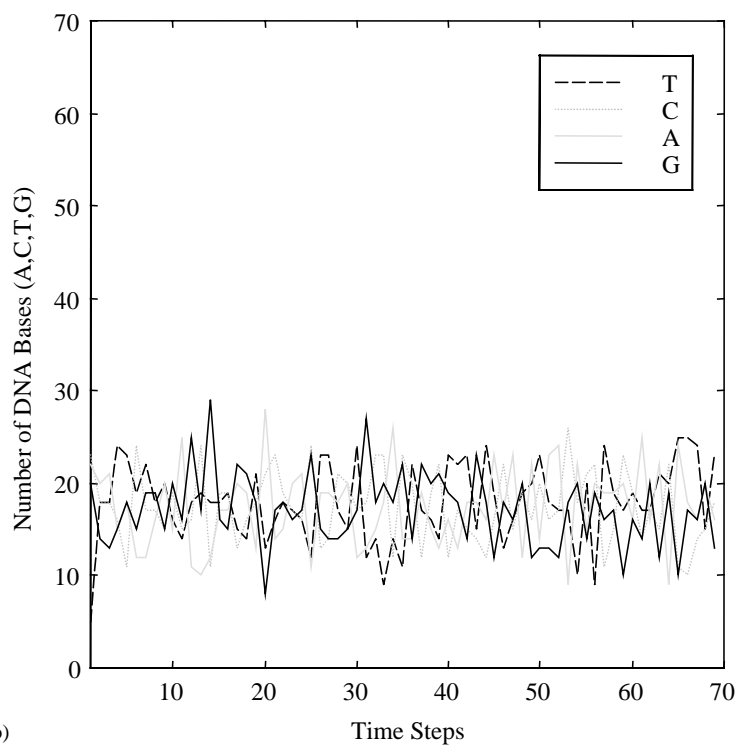

Fig. 6. The graphical user interface of the simulator.

(a)

(b)

Fig. 7. Results produced by the simulator, (a) the evolution pattern, (b) number of bases at various evolution steps.

The size (number of bases) of the DNA sequence is entered in the field "DNA Seq. Size". The user can use the default size, which is 30, or enter size by clicking on the radio button "Set at". After that the number is entered into the blank field on the right.

The size of the CA cell neighborhood is entered in the field "Set Neighborhood Size". The user can use the default size, which is 3, or enter size by clicking on the radio button "Set at". After that the number is entered into the blank field on the right.

The user can set periodic or zero boundary conditions by clicking the corresponding radio button in the field "Boundary Conditions". Some statistical properties, such as the number of bases at each evolution step, are displayed if the user clicks on the check buttons "No. of Bases" in the "Statistical DNA Bases Properties" field.

After setting all the simulation parameters, the simulator is activated by clicking the button "Start" on the top-right of the interface.

Fig. 7(a) shows results produced by the simulator. The initial sequence was random, the evolution time steps were set to 70. The size of the DNA sequence and the size of the neighborhood were set to 70 and 31, respectively. Zero boundary conditions were imposed.

In the blank frame of the interface of Fig. 6, the evolution pattern is now displayed. The *x-axis* is the number of cells and the *y-axis* the number of evolution steps. Base A is shown in white, C in light gray, T in dark gray and G in black.

Fig. 7(b) shows the number of bases at various evolution steps and has been displayed because the "No. of Bases" check buttons were selected.

6. Conclusions

CAs have been introduced as a model for DNA structure, function and evolution and a simulator with a user-friendly input interface was developed. This simulator can be used for the study of DNA evolution. DNA was modeled as a one-dimensional cellular automaton. In this model, the phosphate chain corresponds to the CA lattice and the deoxyriboze sugars to the CA cells. There are four possible states per cell. These states are the four DNA bases A, C, T and G. These four states are represented by numbers of the quaternary number system. Linear evolution rules, represented by square matrices, were considered. Based on this model a simulator of DNA evolution was developed and simulation results have been presented. CAs appear to be a promising model for DNA, because the DNA structure, function and evolution can be simulated using several mathematical tools (such as linear algebra and operators), introduced through the use of CAs. Furthermore, it is very likely that a methodology will be developed for determining the evolution rules generating given evolution patterns.

References

[1] S.K. Moore, Understanding the human genome, Inst. Electr. Electron. Eng. (IEEE) Spectrum 37 (2000) 33.
[2] S.K. Moore, Making chips to probe genes, Inst. Electr. Electron. Eng. (IEEE) Spectrum 38 (2001) 54.
[3] A.D. Baxevanis, B.F., Ouellette, Bioinformatics, A Practical Guide to the Analysis of Genes and Proteins, Wiley-Interscience, New York, 1998.

[4] R. Durbin, S. Eddy, A. Krogh, G. Mitchison, Biological Sequence Analysis, Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, Cambridge, 2000.

[5] J. von Neumann, Theory of Self-reproducing Automata, University of Illinois Press, Urbana, 1966.

[6] S. Wolfram, Cellular Automata and Complexity, Addison-Wesley, New York, 1994.

[7] P. Manneville, N. Boccara, G.Y. Vichniac, R. Bidaux, Cellular Automata and Modeling of Complex Physical Systems, Springer, Berlin, 1990.

[8] B. Chopard, M. Droz, Cellular Automata Modeling of Physical Systems, Cambridge University Press, Cambridge, MA, 1998.

[9] R.J. Gaylord, K. Nishidate, Modeling Nature, Cellular Automata Simulations with Mathematica, Telos Springer, Berlin, 1996.

[10] I. Karafyllidis, A. Thanailakis, A model for predicting forest fire spreading using Cellular Automata, Ecol. Modeling 99 (1997) 87.

[11] I. Karafyllidis, A model for the prediction of oil slick movement and spreading using Cellular Automata, Environ. Int. 23 (1997) 839.

[12] I. Karafyllidis, A model for the influence of the greenhouse effect on insect and microorganism geographical distribution and population dynamics, BioSystems 45 (1998) 1.

[13] G.Ch. Sirakoulis, I. Karafyllidis, A. Thanailakis, A Cellular Automaton model for the effects of population movement and vaccination on epidemic propagation, Ecol. Modeling 133 (2000) 209.

[14] A.R. Kansal, S. Torquato, G.R. Harsh IV, E.A. Chiocca, T.S. Deisboeck, Simulated brain tumor growth dynamics using a three-dimensional cellular automaton, J. Theoret. Biol. 203 (2000) 367.

[15] A.A., Patel, E.T., Gawlinski, S.K. Lemieux, R.A. Gatenby, A cellular automaton model of early tumor growth and invasion: the effects of native tissue vascularity and increased anaerobic tumor metabolism, J. Theoret. Biol. 213 (2001) 315.

[16] R.M.Z. dos Santos, On the dynamics of the evolution of HIV infection, http://arXiv.org/pdf/cond-mat/0008081 (2000).

[17] A.T. Bernardes, R.M.Z. dos Santos, Immune network at the edge of chaos, J. Theoret. Biol. 186 (1997) 173.

[18] M. Bernaschi, F. Castiglione, S. Succi, A high performance simulator of the immune system, Future Gen. Comput. Systems 15 (1999) 333.

[19] S. Ulam, Some ideas and prospects in biomathematics, Ann. Rev. Bio. 12 (1974) 255.

[20] J. McFadden, J. Al-Khalili, A quantum mechanical model of adaptive mutation, BioSystems 50 (1999) 203.

[21] H.P. Schwefel, Deep insight from simple models of evolution, BioSystems 64 (2002) 189.

[22] P.F. Arndt, C.B. Burge, T. Hwa, DNA sequence evolution with neighbor-dependent mutation, http://arXiv.org/pdf/physics/0112029 (2001).

[23] M.E. Jones, S.M. Thomas, K. Clarke, The application of a linear algebra to the analysis of mutation rates, J. Theoret. Biol. 199 (1999) 11.

[24] A. Adamatzky, Identification of Cellular Automata, Taylor & Francis, London, 1994.

[25] B.H. Voorhees, Computational Analysis of One-dimensional Cellular Automata, World Scientific, Singapore, 1996.

[26] J.F.R. Archilla, P.L. Christiansen, Yu.B. Gaididei, Interplay of nonlinearity and geometry in a DNA-related, Klein-Gordon model with long-range dipole-dipole interaction, Phys. Rev. E (Stat. Nonlinear, Soft Matter Phys.) 65 (2002) 016609/1.

Georgios Ch. Sirakoulis was born in Athens, Greece in 1973. He received the Diploma and Ph.D. degrees in Electrical and Computer Engineering from the Democritus University of Thrace, Greece, in 1996 and 2001, respectively. For his Diploma Thesis, he received a prize of distinction from the Technical Chamber of Greece (TEE). His current research emphasis is on cellular automata theory and applications, bioinformatics, modeling and simulation of microelectronic devices and processes, TCAD, VLSI design applications using VHDL, VLSI physical design, and genetic algorithms. He is with Ulysses Ltd. from 2001. The R&D Department of Ulysses Ltd. is focused on providing integrated customized software solutions for the emerging demands of the Life Sciences. He is a member of the IEEE and of the TEE.

Ioannis Karafyllidis was born in Thessaloniki, Greece in 1959. He received the Diploma in Engineering and Ph.D. degrees in Electrical Engineering from the Aristotle University of Thessaloniki, Greece, in 1983 and 1991, respectively. In 1992, he joined the Department of Electrical and Computer Engineering, Democritus University of Thrace, Greece, as a faculty

member, where he is currently an Associate Professor. He has published more than 70 papers in refereed journals and conferences and acts as a referee for several international journals. His current research emphasis is on nanoelectronics, nanotechnology, quantum computing, and cellular automata theory and applications. He is a fellow of the Institute of Nanotechnology, a member of the IEEE and a member of the Technical Chamber of Greece (TEE).

Charilaos Mizas was born in Thessaloniki, Greece in 1971. He received the Diploma degree in Electrical and Computer Engineering from the Democritus University of Thrace, Greece, in 1996. His current research emphasis is on bioinformatics, low power design, parallel architectures, computer arithmetic, and VLSI signal processing. In 2000, he received an award from Intel and IBM for the Low Power Design project LPGD ESPRIT IV #25256. He is with Ulysses Ltd. from 2001. The R& D Department of Ulysses Ltd. is focused on providing integrated customized software solutions for the emerging demands of the Life Sciences. He is a member of the TEE.

Vassilios Mardiris was born in Kavala, Greece in 1969. He received the Diploma degree in Electrical and Computer Engineering from the Democritus University of Thrace, Greece, in 1996. His current research emphasis is on cellular automata theory and applications, bioinformatics, neural networks and applications, parallel architectures and computer arithmetic. In 2000, he received an award from Intel and IBM for the Low Power Design project LPGD ESPRIT IV #25256. He is with Ulysses Ltd. from 2001. The R& D Department of Ulysses Ltd. is focused on providing integrated customized software solutions for the emerging demands of the Life Sciences. He is a member of the TEE.

Adonios Thanailakis was born in Greece in 1940. He received B.Sc. degrees in Physics and Electrical Engineering from the University of Thessaloniki, Greece, in 1964 and 1968, respectively, and the M.Sc. and Ph.D. degrees in electrical engineering and electronics from UMIST, Manchester, England, UK, in 1968 and 1971, respectively.
He has been a Professor of Microelectronics in the Department of Electrical and Computer Engineering, School of Engineering, Democritus University of Thrace, Xanthi, Greece, since 1977, and he was a visiting Professor in the Departments of Electrical and Computer Engineering, University of Manitoba, Canada, and George Mason University, VA, USA, during the Academic years 1984–1985 and 1996–1997, respectively. He has been active in electronic device research since 1968. His current research activities include amorphous materials and devices, photovoltaic conversion of solar energy, applications of group theory in physical and computational systems, and VLSI systems design. He served as Rector (President) of the Democritus University of Thrace (DUTH), as Dean of the School of Engineering of DUTH, and as Chairman of the Electrical and Computer Engineering Department, School of Engineering of DUTH. He has published a great number of scientific and technical papers, as well as five textbooks on materials, devices, electronic noise, photovoltaic conversion of solar energy, and VLSI systems design. Prof. Thanailakis is a Member of the Institution of Electrical Engineers (IEE), the Institute of Physics (IOP), the European Physical Society (EPS), the Technical Chamber of Greece (TEE), and the Institute of Heliotechnics of Greece (IHT).

Philippos Tsalides was born in Mirina Limnou, Greece, on October 14, 1953. He received the Diploma degree from the University of Padova, Italy, in 1979 and the Ph.D. degree from DUTH, Greece, in 1985. He is Professor of applied electronics in the Department of Electrical & Computer Engineering, DUTH, Greece. His research interests include VLSI Architectures, VLSI Systems, BIST Techniques and Application of Cellular Automata to Image Processing, as well as in Computational Systems. He has published a number of papers and a textbook on VLSI Systems (Basic Principles of Design and Fabrication). He is a Fellow member of the IEE and a member of TEE.