Information Sciences 547 (2021) 828–840

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# DNA sequence reconstruction based on innovated hybridization technique of probabilistic cellular automata and particle swarm optimization

Wesam M. Elsayed [a], Mohammed Elmogy [b,*], B.S. El-Desouky [c]

[a] Mathematics Dept., Faculty of Science, Mansoura University, Mansoura, Egypt
[b] Information Technology Dept., Faculty of Computers and Information, Mansoura University, Mansoura, Egypt
[c] Mathematics Dept., Faculty of Science, Mansoura University, Mansoura, Egypt

ARTICLE INFO

*Article history:*
Received 24 May 2019
Received in revised form 24 August 2020
Accepted 27 August 2020
Available online 2 September 2020

*Keywords:*
DNA sequence reconstruction
Computational biology
Mutation rates
Probabilistic cellular automata (PCA)
Particle swarm optimization (PSO)

ABSTRACT

DNA sequence reconstruction is a challenging research problem in the computational biology field. The evolution of the DNA is too complex to be characterized by a few parameters. Therefore, there is a need for a modeling approach for analyzing DNA patterns. In this paper, we proposed a novel framework for DNA pattern analysis. The proposed framework consists of two main stages. The first stage is for analyzing the DNA sequences evolution, whereas the other stage is for the reconstruction process. We utilized cellular automata (CA) rules for analyzing and predicting the DNA sequence. Then, a modified procedure for the reconstruction process is introduced, which is based on the Probabilistic Cellular Automata (PCA) integrated with Particle Swarm Optimization (PSO) algorithm. This integration makes the proposed framework more efficient and achieves optimum transition rules. Our innovated model leans on the hypothesis that mutations are probabilistic events. As a result, their evolution can be simulated as a PCA model. The main objective of this paper is to analyze various DNA sequences to predict the changes that occur in DNA during evolution (mutations). We used a similarity score as a fitness measure to detect symmetry relations, which is appropriate for numerous extremely long sequences. Results are given for the CpG-methylation-deamination processes, which are regions of DNA where a guanine nucleotide follows a cytosine nucleotide in the linear sequence of bases. The DNA evolution is handled as the evolved colored paradigms. Therefore, incorporating probabilistic components help to produce a tool capable of foretelling the likelihood of specific mutations. Besides, it shows their capabilities in dealing with complex relations.

## 1. Introduction

Mathematical approaches and algorithms can model several biological problems. Therefore, it seems to be very beneficial for both mathematics and biologics to combine the results of their disciplines. One of the most critical problems is the analysis of deoxyribonucleic acid (DNA), which is described by a sequence of bases. A DNA sequence holds four nucleic acid bases, which are adenine (A), cytosine (C), guanine (G), and thymine (T). A and T are a complement to each other. Similarly,

* Corresponding author.
*E-mail addresses:* melmogy@mans.edu.eg (M. Elmogy), b_desouky@mans.edu.eg (B.S. El-Desouky).

https://doi.org/10.1016/j.ins.2020.08.102
0020-0255/© 2020 Elsevier Inc. All rights reserved.

C and G are also a complement to each other. DNA is a paired strand molecule where the two strands are linked to each other. The linkage is done over a hydrogen bond among a pyrimidine on one strand and a purine on the other or vice versa. A strand of DNA is formed by a sequence of these four bases, which is transcribed to yield another analogous sequence in the DNA replication procedure [26].

As DNA stores genetic information, it is typical for phylogeny because the DNA of a complicated organism has some similarities with the DNA of a simpler one. Modeling DNA does not follow a specific procedure. In some modeling techniques, it demands to incorporate concepts from multidisciplinary fields, such as chemistry, physics, thermodynamics, and computer science.

This study will concentrate on models that are established on the notion of probabilistic cellular automata (PCA). On the one hand, we used our suggested probabilistic rules to study DNA evolution. A DNA strand can be viewed as a row of cells where every cell holds one of the four bases (A, C, G, or T). This sequence is transcribed to yield another analogous sequence in the DNA replication procedure [6]. The challenge in modeling DNA using Cellular Automata (CA) is the representation of the problem in a way that maps it in a real scenario that follows CA rules. We used CpG methylation followed by deamination and mutation of CG/CG to TG/CA as an example for computing mutation rates. The obtained rules by CA during DNA modeling can give a beneficial vision about the neighboring base pairs' effects on the DNA sequences evolution [26].

On the other hand, as CA showed to be a promising tool in modeling DNA, we use these probabilistic rules again to reconstruct DNA sequence. For the extraction of optimum transition rules, we proposed a hybridized mechanism of both PCA and Particle Swarm Optimization (PSO). This integration makes the proposed framework more efficient and achieves optimum transition rules. Therefore, this study aims to detect the neighborhood rules, which consider the effect of mutations that occurred throughout the sequences' evolution. Besides, this study attempts to get rid of uncertainties in the intermediate sequences. It is fulfilled by introducing stochastic elements to our proposed technique, which leads to a more qualified simulation.

Also, the proposed model can be beneficial for understanding bacterial resistance to antibiotics, which represents a significant threat to public health. With thousands of known unique resistance genes, only the DNA analysis can give detailed genetic information. This retrieved information is required for an accurate evaluation of the existing resistance mechanism. Bacterial genomes are dynamic. They are exposed to diverse genetic events, including duplications, mutations, transpositions, inversions, recombination, insertion, and deletions. As the proposed model is a probabilistic evolution model, it will be convenient for the analysis of the DNA sequences of functional genes during phylogenesis.

The remainder of this article is organized as follows. Section 2 discusses some basic concepts. The design of DNA sequence modeling as a PCA procedure and a brief introduction to the PSO algorithm are presented. Section 3 discusses the literature review. Section 4 describes the proposed probabilistic model for studying the mutation of DNA sequences. Section 5 illustrates in detail how we merge the PCA and PSO algorithms and shows the innovative algorithm for evolution rules. Section 6 elucidates the simulation and experimental results. Section 7 investigates the conclusion of the study, along with further future expectations of this work.

## 2. Basic concepts

In this section, some basic concepts will be introduced. First, we will discuss DNA modeling based on PCA. Then, the PSO technique will be elucidated.

### *2.1. DNA modeling regarding PCA*

In this section, the DNA sequence modeling is tackled as a stochastic procedure. This assumption is obtained from the observations, which clarified the randomly happen mutations. For example, Arndt et al. [2] deduced a neighbor-dependent impact in the mutagenesis process. CA are discrete-state systems comprising of a countable lattice of analogous cells that communicate with their neighbors [1,33,9]. These networks can take any number of dimensions, starting from a one-dimensional sequence of cells. The state of a CA is entirely detected by the values of the variables at each cell. Here, a one-dimensional CA is used for modeling DNA sequence, which can be expressed as a line of adjacent bases [41,35,25,15]. Its mathematical formulation can be defined as a 4-uplet, as presented in Eq. (1).

$$Z = (Q_d, S, N, f) \tag{1}$$

where $Z$ is a CA system, $Q$ is cell space, and $d$ denotes the dimension of this space. $S$ describes the states of all cells. $N$ denotes a set of all cells within the neighborhood. Finally, $f$ is the evolution rule where it detects how the state of the cell can alter.

In the beginning, the PCA is simulated from any preliminary configuration [36,38]. Then, a sequence of configurations is generated. Each configuration is attained from the previous state through an asynchronous update of all sites.

$$S_i \in \{S_1, S_2, S_3, \dots, S_n\} \tag{2}$$

$$P_{t+1}(S) = \sum_{\acute{S}} W(S|\acute{S}) P_t(\acute{S}) \tag{3}$$

where $W(S|\acute{S})$ is the probability of transition from state $\acute{S}$ to state $S$ with the two properties, which are listed in Eqs. (4) and (5).

$$W(S|\acute{S}) \geqslant 0 \tag{4}$$

$$\sum_{S}(S|\acute{S}) = 1 \tag{5}$$

In the beginning, the PCA is simulated from any preliminary configuration [36,38]. Then, a sequence of configurations is generated. Each configuration is attained from the previous state through an asynchronous update of all sites.

### 2.2. Particle swarm optimization (PSO)

The PSO algorithm is a population-based stochastic optimization mechanism, which is proposed by [8,31,37,14]. The PSO algorithm composes a set of feasible solutions that evolve to achieve an adequate solution for a problem. Its main target is to achieve a global optimum of a real-valued function outlined in a search space that is named fitness function.

PSO is initiated with a set of arbitrary particles. Then, it looks for the optimum solution by updating the succeeding generations. At every iteration, each particle is updated by two subsequent "**best**" values. The premier one is the best solution that is obtained until now. This value is called $P_i$ (or $P - best$). The other best value that is pursued by the PSO algorithm is the best value, which is attained so far by any particle. This value is a global best, and it is symbolized by $P_g$ or $(g - best)$. After detecting these two values, the particle updates its speed according to Eq. (6) and its location by Eq. (7).

$$V_i^{t+1} = V_i^t + C_1 r_1 (P_i^t - X_i^t) + C_2 r_2 (P_g^t - X_i^t) \tag{6}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{7}$$

where $V_i$ is the speed of each particle. $X_i$ is the current location of each particle. $C_1$and $C_2$are acceleration constants. $r_1$and $r_2$are arbitrary numbers in the range [0, 1]. $P_i$ is the best position of each particle. $P_g$ is the best position of the swarm. The original PSO has been modified by Shi and Eberhart [31]. They introduced an inertia weight $(\omega)$ to balance exploitation and exploration by modifying Eq. (6) with Eq. (8).

$$V_i^{t+1} = \omega V_i^t + C_1 r_1 (P_i^t - X_i^t) + C_2 r_2 (P_g^t - X_i^t) \tag{8}$$

The PSO algorithm is highly common because of its simple implementation and capability of fast convergence to a rationally good solution. On the other hand, it has some limitations, which can be summarized in the following points:

- Premature convergence.
- Tending to get stuck in local optima.
- Low solution precision.

To overcome the above limitations, this study proposed an innovative hybridized model of PCA and PSO. First, PSO is considered as a simple and easy to execute technique. Then, the simplicity and performance of the PSO algorithm imply that it is inexpensive in terms of memory requirements [30]. Therefore, these reasons led us to make this hybridization between PCA and PSO for better and fast optimization tool.

## 3. Literature review

In this section, some prior studies are discussed, which have considered the evolution of DNA sequences by taking mutations into account. Afterward, the study investigates the efforts that represent the DNA sequence reconstruction process with different techniques.

### 3.1. DNA sequence evolution

The DNA sequence evolution has been discussed by many studies [21,34,35,29]. These studies analyzed the impact of neighboring bases on occurring a mutation. For example, Bulmer [3] found that there is an apparent growth in the transitions frequency from C and G bases. Also, he concluded that there are a few impacts of neighbor bases on the frequencies of transitions from T and A bases. Finally, he determined the transition frequency from these bases. The transition frequency is decreased by having C on the left (or G on the right). Besides, it is incremented by having A on the left (or T on the right).

Arndt et al. [2] introduced a model for the DNA sequence evolution that regard biases in mutation rates. These mutation rates relied on the knowledge of the neighboring bases. They improved an evolution analysis model by assuming non-linear dynamics techniques. They concluded that phylogenetic analysis could be broadened to involve neighbor-dependent impacts. All the previous attempts showed that neighbor bases have some impact on the mutation process. However, none of these studies investigated the effects of the neighboring base through each step of the evolution process. CA model can employ this neighbor reliance during the DNA sequences evolution.

### 3.2. DNA sequence reconstruction

Nowadays, reconstructing evolutionary history is still considered as one of the leading research issues. With the rise of molecular sequencing technologies, advanced computational approaches have been proposed to reconstruct phylogenies [27,28,5]. The programs described in [10] were designed to help in the assembly of long DNA sequences from the much shorter ones obtained as primary data. They detected the overlapping state between fragments and how to be oriented.

Peltola et al. [24] described the first program to control the DNA sequence reconstruction process. Their proposed technique is implemented in three stages. The premier stage calculated overlaps among fragment pairs. They showed these fragments as edges in a directed graph, which has a vertex for each fragment. Second, their procedure elected overlaps from the graph. Finally, the third phase integrated these overlaps into a synchronous alignment of the fragments from which a sequence was extracted.

As our model is probabilistic, we formed a combination of PSO and PCA to extract CA rules properly. Many studies used this combination successfully. Fengxia and Gang [7] used CA integrated with PSO for simulation, which improved the ability of premature convergence. Experiments showed that their algorithm had powerful global searchability. Besides, it could effectively improve the capability of premature convergence. Their method made some improvements but did not completely solve the premature convergence issue.

Pagel [23] used maximum likelihood models to deduce ancestral character states for discrete binary characters that have only two states. However, the generalization to more than two states demands no new concepts. He utilized a Markov model of binary character evolution on phylogenies to reconstruct ancestral states.

As indicated above, there are some limitations in the current related work. However, these algorithms contained the premier explanation of the sequence reconstruction problem with an error. They did not solve or approximate this problem. Nearly, most of the existing research considered the sequence reconstruction problem from the perspective of computational learning theory [8,16,40,34,20]. Also, these methods considered the mutations as deterministic events in the DNA reconstruction process. However, it was commonly appropriate that mutations occur randomly [2,12,3,21,39].

To overcome the limitations mentioned above, we combine PSO and PCA to develop a technique for DNA reconstruction problem. We proposed a new PCA-PSO algorithm, which integrates the cellular space, cells, and neighbors of CA with the PSO configuration. The PSO algorithm is applied to discover the optimal and convenient transition rules of CA for the reconstruction process.

## 4. The proposed DNA evolution model

In this section, the proposed DNA evolution model will be discussed in detail. First, let's postulate that A, C, G, and T bases occur at different frequencies as calculated in [2]. Neighbor-dependent mutation rates are calculated according to both these frequencies and our new suggested rules [6]. Then, the six parameter rate matrix model command of the general time-reversible (GTR) is utilized to specify the calculated rates of each type of nucleotide change. We inserted a random sequence with 30 bases as a first initial taxon.

The evolution of the DNA sequence can be represented as a colored graphical representation for the DNA bases. We get the graphical output by assigning a color to each base. Red color is used for A, green color is assigned for C, yellow color is utilized for G, and blue color is used for T. To test the suggested rules, we used for the processes $ApC \rightarrow ApA$ or $GpT \rightarrow TpT$. Besides, we allow only a single transversion rate (i.e., $p = 1$). With these processes, the maximum-likelihood solution for the mutation rates became more credible, with $Q_{AG} = Q_{TC} = 3.10p, Q_{CT} = Q_{GA} = 3.78p, R_{CGCA} = R_{CGTG} = 43.02p$, and $R_{ACAA} = R_{GTTT} = 4.35p$. From these values, we form 3 matrices as listed in Eqs. (9)–(11).

$$Q = \begin{bmatrix} 0 & 0 & 3.78 & 0 \\ 0 & 0 & 0 & 3.1 \\ 3.1 & 0 & 0 & 0 \\ 0 & 3.78 & 0 & 0 \end{bmatrix} \tag{9}$$

$$R_l = \begin{bmatrix} 4.35 & 4.35 & 0 & 0 \\ 43.02 & 0 & 43.02 & 0 \\ 0 & 0 & 0 & 4.35 \\ 0 & 0 & 43.02 & 4.35 \end{bmatrix} \tag{10}$$

$$R_r = \begin{bmatrix} 4.35 & 43.02 & 0 & 0 \\ 4.35 & 0 & 43.02 & 0 \\ 0 & 43.02 & 0 & 43.02 \\ 0 & 0 & 4.35 & 4.35 \end{bmatrix} \tag{11}$$

where $Q$represents the rates of single-nucleotide occurrence. $R_l$ is the rates of the left pair neighboring nucleotides. $R_r$ is the rates of the right pair neighboring nucleotides. Then, these matrices are substituted in the mutation probability $(Q + R).\Delta t$. The time increment $(\Delta t)$ must be selected such that all non-diagonal transition probabilities in Eqs. (9)–(11) are small $(\ll 1)$. So, we get the mutation matrices as shown in Fig. 1.

The DNA mutation plays a significant role in the DNA sequence evolution [13]. Neighbor-dependent mutations influence the evolution of DNA sequences. We used the previously calculated mutation rates, as listed in Fig. 1, where we utilized the calculated frequencies by [2]. Therefore, we got the mutation rates that produced our evolution. Fig. 2 presents the simulation results of a DNA sequence evolution. The simulation begins with an arbitrary DNA sequence that contains 30 bases. It generates the sequences for 30 consecutive time steps, where A is shown in red, C in green, T in blue, and G in yellow.

## 5. The Proposed PSO-PCA reconstruction technique

In this section, we introduce an innovative mechanism for DNA sequence reconstruction based on a hybrid PSO-PCA technique. First, the search space is partitioned into cells by applying CA. At any time, a set of particles looks for a local optimum. Then, the best solution is found in their neighborhood cells.

In the PSO algorithm, each particle updates its immediate location together with all particle states. In PCA, every cell is closely incorporated with its neighbors and their positions governed by a specific rule. Its immediate state is updated with neighbor's states. In our proposed PSO-PCA algorithm, the state of the cell is evaluated by its state besides its neighbor's states. The last speed updates each cell's state. Then, to approach the optimal state, the current cell's state is updated once more regarding both immediate and optimal states. PSO-PCA modeling is as follow:

- **Cells**: Each particle is interpreted as a single cell.
- **Cellular space**: The group of all cells in the space (1-dimensional CA is considered in this scenario).
- **State space**: The cell's state is mentioned as the location of every particle. The state of $i^{th}$ cell is calculated by Eq. (12).

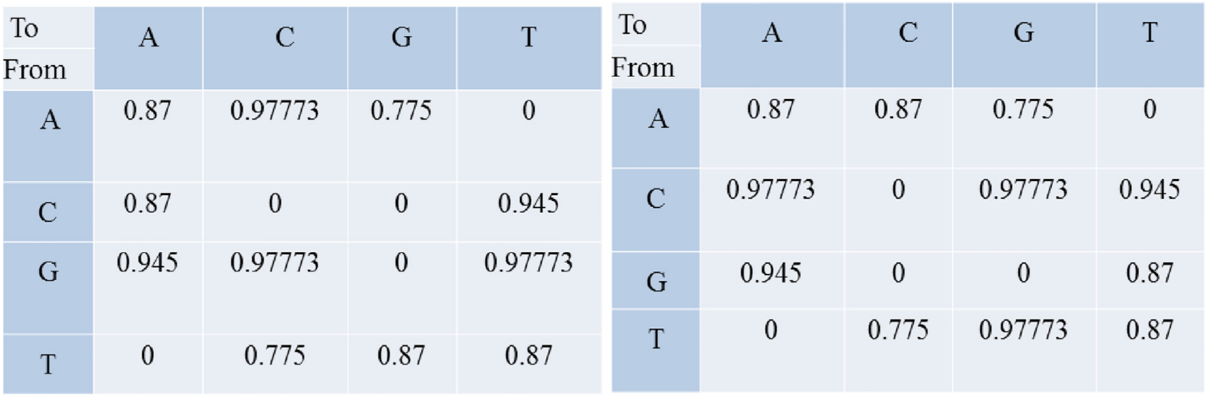
$$S_i^t = X_i^t \tag{12}$$

- **Neighborhood**: The whole cells that may influence the variation of the state of the $i^{th}$ cell are the neighbor cells. The neighborhood is presented by Eq. (13).

$$N(i, r) = S_{i-r}, \ldots, S_{i-2}, S_{i-1}, S_i, S_{i+1}, S_{i+2}, \ldots, S_{i+r}, r = 0, 1, 2, \ldots, m \tag{13}$$

where $r$ is the size of the neighborhood. Here, we use $r = 1$, then the neighbors of the $i^{th}$ cell are comprised of the cell itself and its right and left direct neighbors.

$$N(i, 1) = S_{i-1}, S_i, S_{i+1} \tag{14}$$



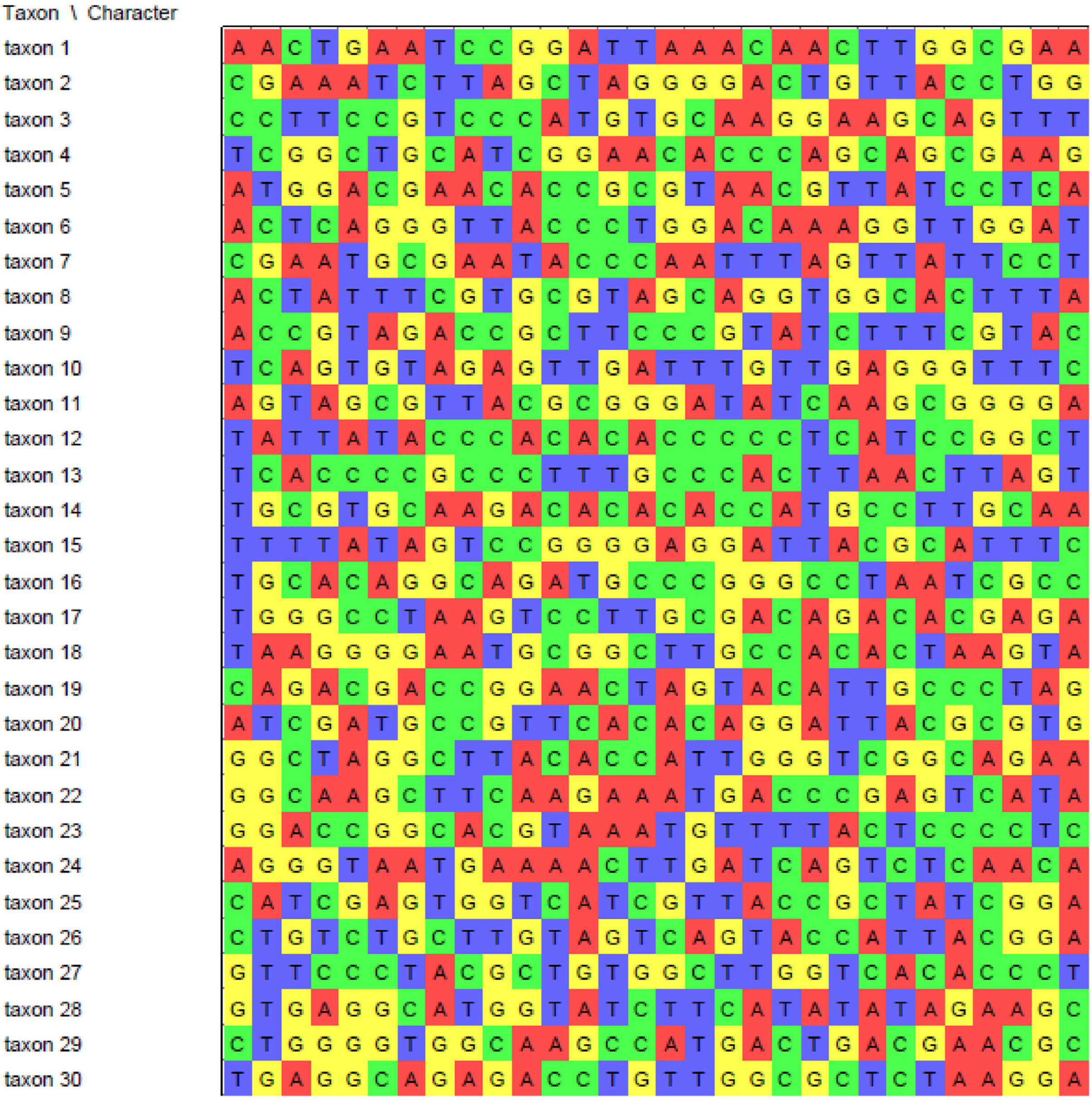
| To<br>From | A | C | G | T |
|---|---|---|---|---|
| A | 0.87 | 0.97773 | 0.775 | 0 |
| C | 0.87 | 0 | 0 | 0.945 |
| G | 0.945 | 0.97773 | 0 | 0.97773 |
| T | 0 | 0.775 | 0.87 | 0.87 |

a) Left Mutation Matrix

| To<br>From | A | C | G | T |
|---|---|---|---|---|
| A | 0.87 | 0.87 | 0.775 | 0 |
| C | 0.97773 | 0 | 0.97773 | 0.945 |
| G | 0.945 | 0 | 0 | 0.87 |
| T | 0 | 0.775 | 0.97773 | 0.87 |

b) Right Mutation Matrix

**Fig. 1.** The mutation matrices: If a site is located at the cell's position, then its mutation probability is given by (a). Otherwise, its mutation probability is given by (b).



| Taxon \ Character | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| taxon 1 | A | A | C | T | G | A | A | T | C | C | G | G | A | T | T | A | A | A | C | A | A | C | T | T | G | G | C | G | A | A |
| taxon 2 | C | G | A | A | A | T | C | T | T | A | G | C | T | A | G | G | G | G | A | C | T | G | T | T | A | C | C | T | G | G |
| taxon 3 | C | C | T | T | C | C | G | T | C | C | C | A | T | G | T | G | C | A | A | G | G | A | A | G | C | A | G | T | T | T |
| taxon 4 | T | C | G | G | C | T | G | C | A | T | C | G | G | A | A | C | A | C | C | C | A | G | C | A | G | C | G | A | A | G |
| taxon 5 | A | T | G | G | A | C | G | A | A | C | A | C | C | G | C | G | T | A | A | C | G | T | T | A | T | C | C | T | C | A |
| taxon 6 | A | C | T | C | A | G | G | G | T | T | A | C | C | C | T | G | G | A | C | A | A | A | G | G | T | T | G | G | A | T |
| taxon 7 | C | G | A | A | T | G | C | G | A | A | T | A | C | C | C | A | A | T | T | T | A | G | T | T | A | T | T | C | C | T |
| taxon 8 | A | C | T | A | T | T | T | C | G | T | G | C | G | T | A | G | C | A | G | G | T | G | G | C | A | C | T | T | T | A |
| taxon 9 | A | C | C | G | T | A | G | A | C | C | G | C | T | T | C | C | C | G | T | A | T | C | T | T | T | C | G | T | A | C |
| taxon 10 | T | C | A | G | T | G | T | A | G | A | G | T | T | G | A | T | T | T | G | T | T | G | A | G | G | G | T | T | T | C |
| taxon 11 | A | G | T | A | G | C | G | T | T | A | C | G | C | G | G | G | A | T | A | T | C | A | A | G | C | G | G | G | G | A |
| taxon 12 | T | A | T | T | A | T | A | C | C | C | A | C | A | C | A | C | C | C | C | C | T | C | A | T | C | C | G | G | C | T |
| taxon 13 | T | C | A | C | C | C | C | G | C | C | C | T | T | T | G | C | C | C | A | C | T | T | A | A | C | T | T | A | G | T |
| taxon 14 | T | G | C | G | T | G | C | A | A | G | A | C | A | C | A | C | A | C | C | A | T | G | C | C | T | T | G | C | A | A |
| taxon 15 | T | T | T | T | A | T | A | G | T | C | C | G | G | G | G | A | G | G | A | T | T | A | C | G | C | A | T | T | T | C |
| taxon 16 | T | G | C | A | C | A | G | G | C | A | G | A | T | G | C | C | C | G | G | G | C | C | T | A | A | T | C | G | C | C |
| taxon 17 | T | G | G | G | C | C | T | A | A | G | T | C | C | T | T | G | C | G | A | C | A | G | A | C | A | C | G | A | G | A |
| taxon 18 | T | A | A | G | G | G | G | A | A | T | G | C | G | G | C | T | T | G | C | C | A | C | A | C | T | A | A | G | T | A |
| taxon 19 | C | A | G | A | C | G | A | C | C | G | G | A | A | C | T | A | G | T | A | C | A | T | T | G | C | C | C | T | A | G |
| taxon 20 | A | T | C | G | A | T | G | C | C | G | T | T | C | A | C | A | C | A | G | G | A | T | T | A | C | G | C | G | T | G |
| taxon 21 | G | G | C | T | A | G | G | C | T | T | A | C | A | C | C | A | T | T | G | G | G | T | C | G | G | C | A | G | A | A |
| taxon 22 | G | G | C | A | A | G | C | T | T | C | A | A | G | A | A | A | T | G | A | C | C | C | G | A | G | T | C | A | T | A |
| taxon 23 | G | G | A | C | C | G | G | C | A | C | G | T | A | A | A | T | G | T | T | T | T | A | C | T | C | C | C | C | T | C |
| taxon 24 | A | G | G | G | T | A | A | T | G | A | A | A | A | C | T | T | G | A | T | C | A | G | T | C | T | C | A | A | C | A |
| taxon 25 | C | A | T | C | G | A | G | T | G | G | T | C | A | T | C | G | T | T | A | C | C | G | C | T | A | T | C | G | G | A |
| taxon 26 | C | T | G | T | C | T | G | C | T | T | G | T | A | G | T | C | A | G | T | A | C | C | A | T | T | A | C | G | G | A |
| taxon 27 | G | T | T | C | C | C | T | A | C | G | C | T | G | T | G | G | C | T | T | G | G | T | C | A | C | A | C | C | C | T |
| taxon 28 | G | T | G | A | G | G | C | A | T | G | G | T | A | T | C | T | T | C | A | T | A | T | A | T | A | G | A | A | G | C |
| taxon 29 | C | T | G | G | G | G | T | G | G | C | A | A | G | C | C | A | T | G | A | C | T | G | A | C | G | A | A | C | G | C |
| taxon 30 | T | G | A | G | G | C | A | G | A | G | A | C | C | T | G | T | T | G | G | C | G | C | T | C | T | A | A | G | G | A |

**Fig. 2.** The simulated evolution of an arbitrary DNA sequence (A: red, C: green, T: blue, and G: yellow). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- **Update rules**: Every cell updates its state with its immediate location, velocity, individual optimal value, and its neighbors' optimal values. The state of the $i^{th}$ cell at a time step is a function of the states of its neighbors at the prior time step.

  $$S_i(t+1) = f(S_{i-1}(t), S_i(t), S_{i+1}(t)) \tag{15}$$

- **Discrete time step**: It represents the number of iterations in the PSO.

DNA may be modeled as a 1-dimensional CA. The four DNA bases stand for the probable states of a CA cell. The state of the $i^{th}$ cell of this CA gets values from the discrete set that incorporates the four bases.

$$S_i \in A, C, T, G \tag{16}$$

**Table 1**
The left-hand sides of 64 transitions of CA with a neighbor size equals to one.

| State No. | Possible states | State No. | Possible states | State No. | Possible states | State No. | Possible states |
|---|---|---|---|---|---|---|---|
| 0 | 000 | 16 | 001 | 32 | 002 | 48 | 003 |
| 1 | 100 | 17 | 101 | 33 | 102 | 49 | 103 |
| 2 | 200 | 18 | 301 | 34 | 302 | 50 | 303 |
| 3 | 300 | 19 | 201 | 35 | 202 | 51 | 203 |
| 4 | 010 | 20 | 011 | 36 | 012 | 52 | 013 |
| 5 | 110 | 21 | 111 | 37 | 112 | 53 | 113 |
| 6 | 310 | 22 | 311 | 38 | 312 | 54 | 313 |
| 7 | 210 | 23 | 211 | 39 | 212 | 55 | 213 |
| 8 | 130 | 24 | 031 | 40 | 032 | 56 | 033 |
| 9 | 330 | 25 | 131 | 41 | 132 | 57 | 133 |
| 10 | 230 | 26 | 331 | 42 | 332 | 58 | 233 |
| 11 | 020 | 27 | 231 | 43 | 232 | 59 | 023 |
| 12 | 120 | 28 | 021 | 44 | 022 | 60 | 123 |
| 13 | 320 | 29 | 121 | 45 | 122 | 61 | 323 |
| 14 | 220 | 30 | 321 | 46 | 322 | 62 | 223 |
| 15 | 030 | 31 | 221 | 47 | 222 | 63 | 333 |

The bases are coded with numbers as follows: $A \rightarrow 0, C \rightarrow 1, T \rightarrow 2, G \rightarrow 3$. Here, we take into account just the rules with a neighborhood-sized by 1. The transitions of base-pairs through evolution are clarified in Table 1. The right-hand side of each transition can be one of the four base-pairs.

Here, CAs have four states for each cell. Hence, the number of all probable rules is $4^{4^3}$. The whole rule space must be investigated. It detects the assumed CA rules that control the evolution of the DNA sequence. Here, PSO is applied in order to investigate the enormous CA rule space.

Evolution is visualized with the aid of a phylogenetic tree that represents a group of organisms that are connected [22,32,4,11,18,27]. A phylogenetic tree is a tree demonstrating the evolutionary mutual relations among diverse species or other organisms that are accepted to have a mutual ancestor. Several species, organisms, or genomic sequences are represented on the leaves of the tree. Our work seeks to detect the rules for neighbor-based mutations that may have been resulted in the sequence evolutions. Here, we used linear rules, which represent a promising tool for analyzing mutation rates [31]. The linear evolution rules status takes a matrix format as listed in Eq. (17). Consequently, the velocity is updated using Eq. (18).

$$\begin{bmatrix} \vdots \\ S_{i-1}^{t+1} \\ S_i^{t+1} \\ S_{i+1}^{t+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & M_{i-1,j-1} & M_{i-1,j} & M_{i-1,j+1} & \cdots \\ \cdots & M_{i,j-1} & M_{i,j} & M_{i,j+1} & \cdots \\ \cdots & M_{i+1,j-1} & M_{i+1,j} & M_{i+1,j+1} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ S_{i-1}^{t} \\ S_i^{t} \\ S_{i+1}^{t} \\ \vdots \end{bmatrix} + \begin{bmatrix} \vdots \\ V_{i-1}^{t+1} \\ V_i^{t+1} \\ V_{i+1}^{t+1} \\ \vdots \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} \vdots \\ V_{i-1}^{t+1} \\ V_i^{t+1} \\ V_{i+1}^{t+1} \\ \vdots \end{bmatrix} = \omega \begin{bmatrix} \vdots \\ V_{i-1}^{t} \\ V_i^{t} \\ V_{i+1}^{t} \\ \vdots \end{bmatrix} + C_1 r_1 \left( \begin{bmatrix} \vdots \\ P-best_{i-1}^{t} \\ P-best_i^{t} \\ P-best_{i+1}^{t} \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ S_{i-1}^{t} \\ S_i^{t} \\ S_{i+1}^{t} \\ \vdots \end{bmatrix} \right) + C_2 r_2 \left( \begin{bmatrix} \vdots \\ g-best_{i-1}^{t} \\ g-best_i^{t} \\ g-best_{i+1}^{t} \\ \vdots \end{bmatrix} - \begin{bmatrix} \vdots \\ S_{i-1}^{t} \\ S_i^{t} \\ S_{i+1}^{t} \\ \vdots \end{bmatrix} \right) \tag{18}$$

The states of all CA cells at time step $t$ are represented by the column array at the right-hand side of Eq. (17). This array is multiplied by the evolution rule array $(M)$. The array components $M_{i,j}$may hold only two values (0 and 1). The output array at the left-hand side of Eq. (17) contains the states of all CA cells at time step $t + 1$.

### 5.1. Fitness function evaluation

In this stage, the fitness function should be evaluated after representing each cell. The optimal solution is obtained based on the resulting fitness value. In the DNA sequence reconstruction, the optimum solution represents the highest matching score of the bases among the sequences at the previous steps. First, we have to compute the evolution of DNA sequences to extract the proper rules for the reconstruction process, as shown in Fig. 2. The matching score is evaluated by enumerating the equivalent nucleotide of sequences. The matching score for a pair of sequences is computed by using Eq. (19).

$$Score_{S_{i,j}\&\hat{S_{i,j}}} = \begin{cases} 0 & if\, bases\, do\, not\, match \\ Score_{S_{i,j}\&\hat{S_{i,j}}} + 1 & otherwise \end{cases} \tag{19}$$

where, $Score_{S_{i,j}\&\hat{S_{i,j}}}$ is a matching score of two consecutive sequences. $i$and $j$are the indices of the cell. After calculating the score of each two consecutive sequences, the total score is evaluated by Eq. (20).

$$max f(x) = \sum_i \sum_j Score_{S_{i,j}\&\hat{S_{i,j}}} \tag{20}$$

where, $f(x)$ indicates the fitness value for individual $i$ and $j$ of the PSO. max indicates that our target is to get the maximum value of $f(x)$. The optimal solution is the one that has the largest value of $f(x)$. The fitness function is determined by summing all scores (Eq. (19)). The main goal is to find out symmetry relations among various sequences. Algorithm 1 lists the proposed PSO-PCA algorithm for the election of CA evolution rules.

**Algorithm 1** The proposed PSO-PCA rule extraction algorithm.

```
1: Initialize swarm-size, Dsearch space dimension, Nnumber of neighbors and iterMmax iteration times.
2: for i = 1 to swarm-size do
3:    Particle[i].position = random(0,1,2,3)
4:    Particle[i].velocity = 0
5:    Particle[i].fitness = 0
6: fori = 1 to swarm-size do
7:    P-best[i]=particle[i]
8: G-best = particle[1]
9: for i = 1 to swarm-sizedo
10:    P-best[i]=fitness($X_i$)
11: while criterion is not fulfilled do
12:    for$p = 1$to swarm-size do
13:       foreach evolution step do
14:          Compute next rule
15:       Update particle's velocity $V_i(t+1)$
16:       Update cell's state $S_i(t+1)$
17:       Estimate fitness
18:    for$i = 1$to swarm-size do
19:       if particle[i].fitness $\geqslant$P-best[i].fitness then
20:          P-best[i]=particle[i]
21:       if $G - best[i].fitness > particle[i].fitness$ then
22:          G-best[i]=particle[i]
23:    for$i = 1$to swarm-size do
24:       Update PSO parameters: position and velocity
25: Return to step 2
```

## 6. Simulation and results

This section deals with the experimental setup and outcomes that are achieved after the simulation. The proposed system is implemented by using the Matlab 2019b [19] Program. Besides, we utilized the Mesquite program [17], which is an analysis tool for evolutionary biology. It offers a simple and powerful tool, which motivated us to use it for DNA simulation. For hardware specifications, we implemented the proposed system on a computer with an Intel Core i7 processor (8th generation, 1.8 GHz) with 16 GB Ram. The DNA data used to support the findings of this study are available at [ https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi?mode=Info&id=48640]. Our proposed PSO-PCA algorithm is started by inserting the following parameters to obtain the desired probabilistic rules:

- The native sequence of the DNA.
- The sequences of the DNA that belong to in-between evolution steps.
- The eventual sequences of the DNA.
- The maximum iteration numbers.

We used phylogenetic trees for the reconstruction process. Besides, they are used for representing the species samples that are used for simulation. For each branch, we applied a set of CA rules for altering a predecessor sequence to the offspring sequence. First, the proposed technique compares the current sequence, with the produced sequence at the next time step using a similarity score. We defined this score as the proportion of the number of matching base pairs in two consecutive sequences. Then, the proposed algorithm randomly chooses a CA rule at each time step and applies it to the immediate sequence. Then, we observe the progression of similarity score, meanwhile evolution. Non-uniform probabilistic rules are used for simulation based on the fact that not all the base-pairs will be altered at every time step. In our evolution, we attempt to dynamically generate a CA rule utilizing a sequence achieved within the evolution and the consecutive sequence. Fig. 3 clarifies the dynamic structure of a rule.

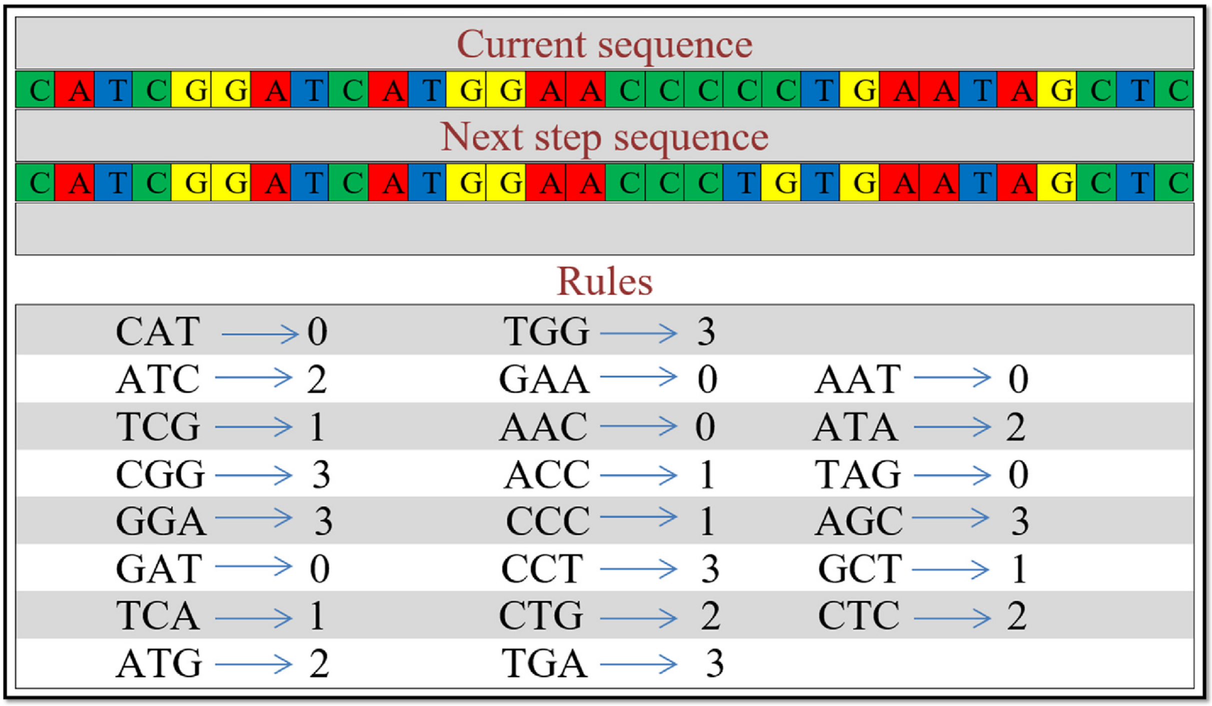
We aim to generate a CA rule by using a sequence resulting from the evolution process along with its next step sequence. The transitions on the left-hand side are established by applying a rule on the immediate sequence. The next step sequence configures the right-hand direction of the transitions. For instance, the left-hand side of a transition is outlined by the premier three base-pairs of the immediate sequence, labeled CAT, as shown in Fig. 3. As for the right-hand side of the transition, it is outlined by the corresponding base-pair in the consecutive sequence, 0.

The major aim of this study is to detect the most probable rules for mutations, which take into consideration the impact of the neighbor cells. Table 2 demonstrates some of the resulted rules that were stratified to some of the phylogenetic tree branches obtained from random DNA sequences. We used sequences of size equals 100 bases, as shown in Fig. 4. The achieved rules by using the proposed technique are used instantly for foretelling of next step sequences and the construction of the phylogenetic tree itself. In Fig. 4, the DNA evolution outcomes after 100 generations in the DNA sequences, which are generated by the implementation of the parallel CA rule. The resulted CA rule is demonstrated in Fig. 5.

The PSO-PCA is proved to be a successful procedure for reconstructing the evolution paradigm of the given DNA sequences. We developed a proper algorithm that efficiently elicits the CA rule, which controls the evolution of the sequence. During these random experiments, the innovated procedure dictated the possible rules that produced the given evolution paradigm. The algorithm showed to be a successful simulation tool. As a result, we demonstrated that having a series of DNA sequences that represent a set of evolution steps, this procedure can be utilized for generating the probabilistic rules of this evolution pattern. These rules are properly capable of reconstructing DNA sequences. Our attempt to incorporate probabilistic components produces a system capable of predicting the likelihood of particular mutations. Also, our technique properly showed to be a promising tool for simulating the evolution of large sequences, as in Fig. 6:

First, the data clarified in Fig. 6 is tested with the help of the Mesquite simulation software. Then, the mutation rates are applied besides using the proposed innovative PSO-PCA algorithm. Finally, the obtained results can be shown in Fig. 7.

The protein, DNA, or RNA sequence can be used to classify the sequence into sets of analogous bases that share characteristics in terms of their function. It can be quite beneficial in recognizing the functions of new sequences. Also, it can be beneficial in phylogenetic prediction. Studying DNA evolution and the effect of mutation will help recognize bacterial species as well as potential antibiotic resistance mechanisms. It will lead to a chance to employ DNA sequence information to guide medication.



Current sequence

C A T C G G A T C A T G G A A C C C C C T G A A T A G C T C

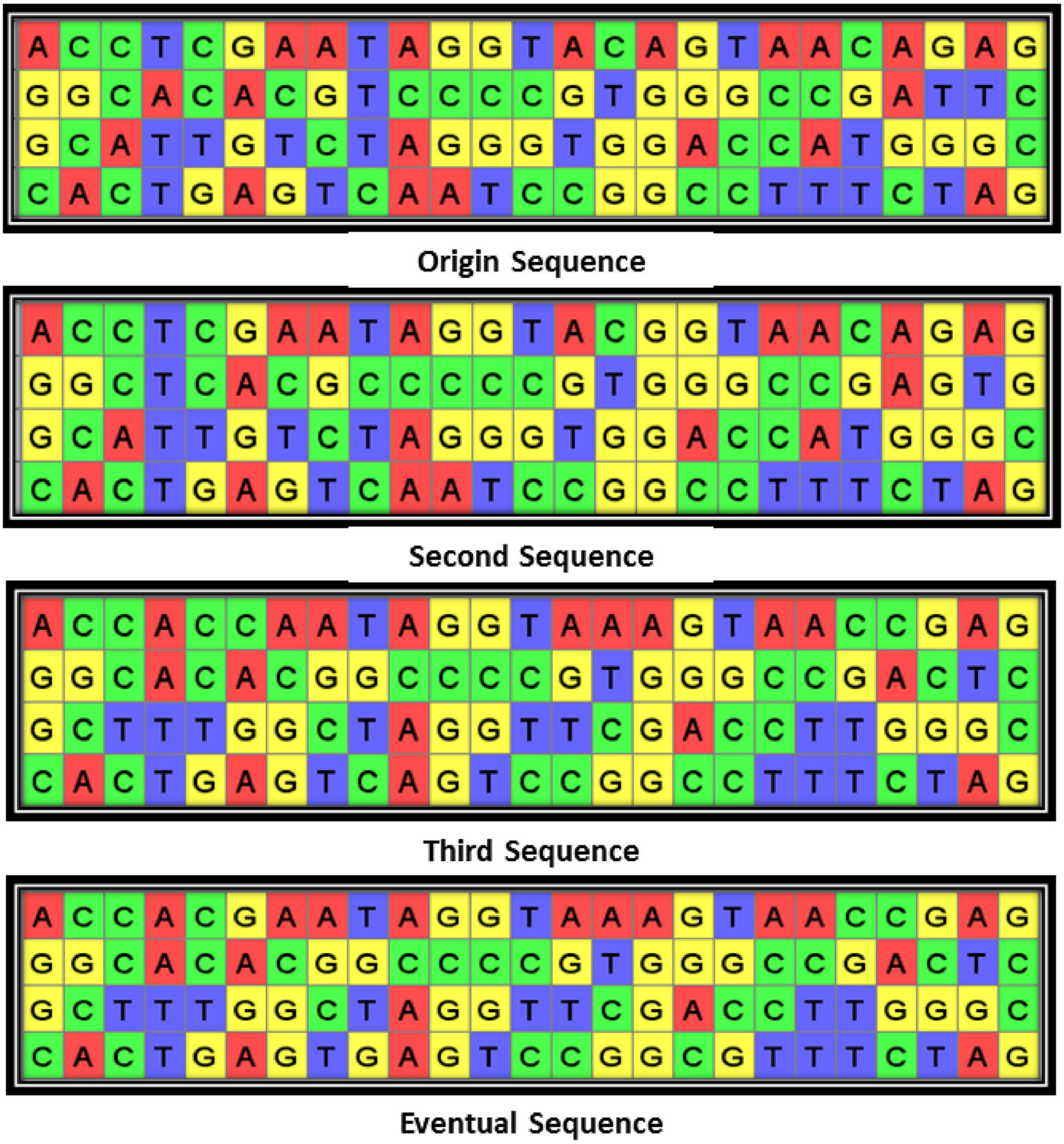
Next step sequence

C A T C G G A T C A T G G A A C C C T G T G A A T A G C T C

Rules

| | | |
|---|---|---|
| CAT → 0 | TGG → 3 | |
| ATC → 2 | GAA → 0 | AAT → 0 |
| TCG → 1 | AAC → 0 | ATA → 2 |
| CGG → 3 | ACC → 1 | TAG → 0 |
| GGA → 3 | CCC → 1 | AGC → 3 |
| GAT → 0 | CCT → 3 | GCT → 1 |
| TCA → 1 | CTG → 2 | CTC → 2 |
| ATG → 2 | TGA → 3 | |

**Fig. 3.** An example of the formation of CA rules.



A C C T C G A A T A G G T A C A G T A A C A G A G
G G C A C A C G T C C C C G T G G G C C G A T T C
G C A T T G T C T A G G G T G G A C C A T G G G C
C A C T G A G T C A A T C C G G C C T T T C T A G

Origin Sequence

A C C T C G A A T A G G T A C G G T A A C A G A G
G G C T C A C G C C C C C G T G G G C C G A G T G
G C A T T G T C T A G G G T G G A C C A T G G G C
C A C T G A G T C A A T C C G G C C T T T C T A G

Second Sequence

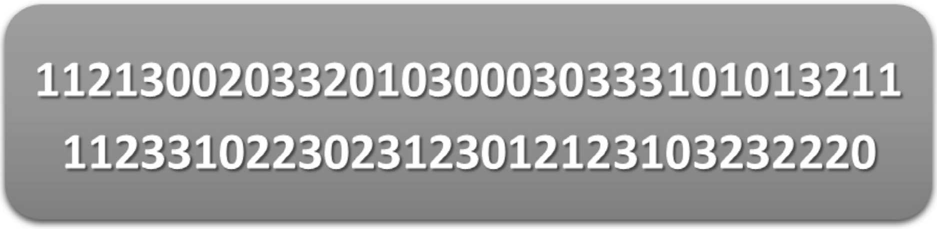
A C C A C C A A T A G G T A A A G T A A C C G A G
G G C A C A C G G C C C C G T G G G C C G A C T C
G C T T T G G C T A G G T T C G A C C T T G G G C
C A C T G A G T C A G T C C G G C C T T T C T A G

Third Sequence

A C C A C G A A T A G G T A A A G T A A C C G A G
G G C A C A C G G C C C C G T G G G C C G A C T C
G C T T T G G C T A G G T T C G A C C T T G G G C
C A C T G A G T G A G T C C G G C G T T T C T A G

Eventual Sequence

**Fig. 4.** The DNA sequences corresponding to the origin, eventual, and two in-between evolution steps.



11213002033201030003033310101321 1
11233102230231230121231032322220

**Fig. 5.** The used CA rule to reproduce the evolution pattern in Fig. 4.

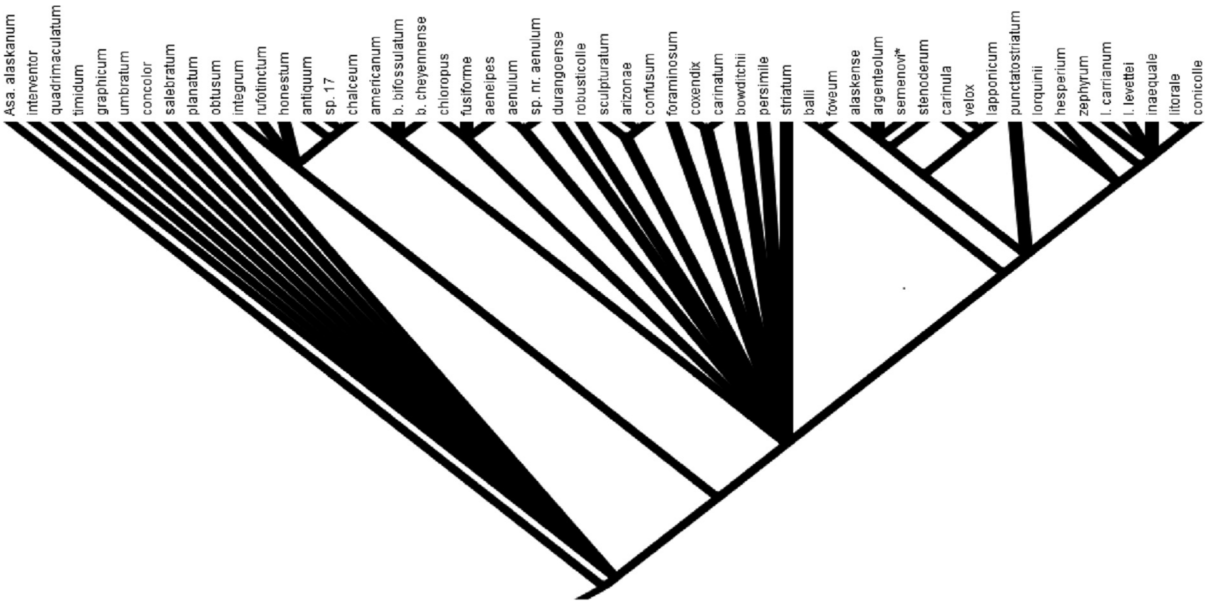


**Fig. 6.** An example of the phylogenetic tree with 53 samples.

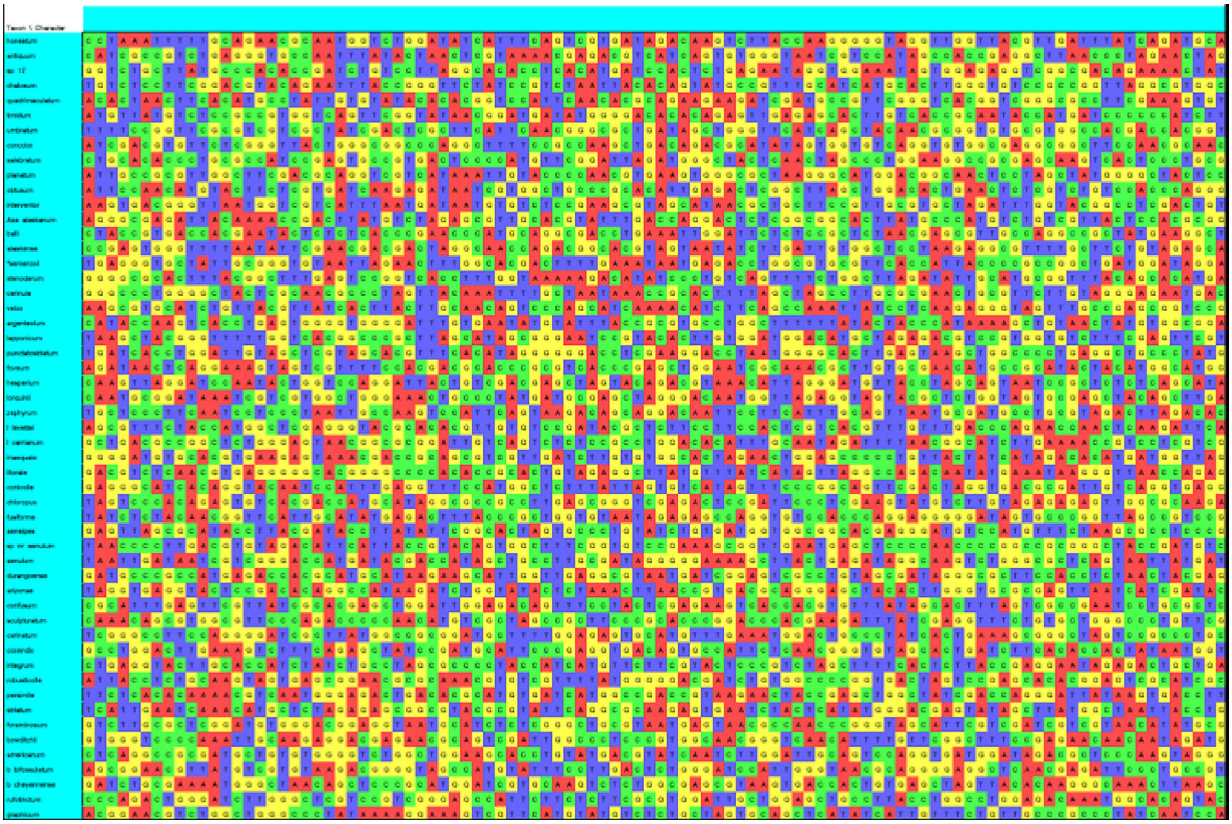

Taxon \ Character

**Fig. 7.** The simulation results of the evolving DNA characters after using our rates and the probabilistic rules.

Finally, we presented a comparative survey on DNA sequence using our model and deterministic methods based on genetic algorithm. Our proposed technique is capable of the following:

- It can be used for analyzing the evolution of many various species. It helps in many practical applications, including drug detection, population surveillance, and management.
- Dealing with these models, such as PCA, enables us to discover the influence of some neighbors base-pairs evolution.
- We were able to resolve uncertainties (i.e., detecting anonymous base-pairs in intermediate sequences and the number of time steps for evolution).
- The proposed model of evolution is a probabilistic model. Therefore, it will be convenient for the DNA sequence analysis of functional genes during phylogenesis.

On the other hand, other methods that consider only the deterministic direction focus on the neighbor-dependent mechanics of DNA sequence alteration without considering the processes of natural selection. Therefore, the deterministic methods are not suitable for the analysis of DNA sequences.

## 7. Discussion and conclusion

This paper introduced a novel technique based on PCA to investigate the rules for the neighbor bases regarding mutation effects. CA proved to be a robust system for analyzing DNA mutations. CA rules are generated by simulating DNA mutations. They can give us beneficial insights about the influence of neighboring base pairs on the evolution of DNA sequences. The proposed tool for simulation is based on the usage of PSO, as it extracts the PCA evolution rules very efficiently. There are paramount concerns that neighboring base-pairs influence mutations of DNA base-pairs. We tried to reveal this correlation by modeling DNA as a CA model where the CA rules govern the DNA mutations. Due to the enormous rule space comprised in our simulation, we adopt the PSO algorithm for extracting these rules efficiently. Then, we applied the resulted rules for predictions of sequences in phylogenetic trees. Simulating DNA as a PCA model facilitates viewing, analyzing, and comparing various sequences.

Also, our method is suitable for the comparison of many long sequences. Establishing powerful and reasonable hybridization strategies is required for creating beneficial and practical models for predicting most of the future changes that occur during the evolution of DNA sequences. It will highly increase information on which mutations are most popular in certain bacteria. Besides, PCA can reveal patterns in enormous amounts of gene expression data, and discover groups of disease-related genes to be able to detect medicine.

The main contributions of this paper can be summarized in the following points:

- A methodology is developed to locate the impact of neighboring DNA base-pairs on the mutation of a base-pair.
- The model presented here is based on the assumption that mutations are probabilistic events, and that their evolution can be modeled using PCA.
- A hybridized technique is developed to discover the optimal and proper transition rules of CA for the reconstruction task. This integration increases the performance of the algorithm.
- A modified method is proposed for the reconstruction of DNA sequences based on PCA integrated with the PSO algorithm.

In our future work, we will popularize this study to handle distinct rules for diverse structures of the neighborhood regarding mutation effects. Particularly, diverse neighborhood sizes with larger sizes will be discussed. Also, we can simulate the evolution and reconstruction of DNA sequences on small-world networks. We might be able to use this probabilistic model to foretell possible mutations of viruses and other pathogens. For example, we may hopefully be able to explain the CORONA virus.

## CRediT authorship contribution statement

**Wesam M. Elsayed:** Data curation, Writing - original draft, Formal analysis, Methodology, Conceptualization, Software. **Mohammed Elmogy:** Data curation, Writing - original draft, Formal analysis, Methodology, Conceptualization, Project administration, Validation, Writing - review & editing. **B.S. El-Desouky:** Project administration, Validation, Writing - review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

We thank Prof. Dr. E. Ahmed, Mathematics Dept., Faculty of Science, Mansoura, EGYPT, for his guidance and comments.

## References

[1] Andrew Adamatzky, Identification of Cellular Automata, Springer, New York, New York, NY, 2009, pp. 4739–4751.

[2] Peter F. Arndt, Christopher B. Burge, Terence Hwa, Dna sequence evolution with neighbor-dependent mutation, J. Comput. Biol. 10 (3–4) (2003) 313–322.

[3] Michael Bulmer, Neighboring base effects on substitution rates in pseudogenes, Mol. Biol. Evol. 3 (4) (1986) 322–329.

[4] Frédéric Delsuc, Henner Brinkmann, Hervé Philippe, Phylogenomics and the reconstruction of the tree of life, Nat. Rev. Genet. 6 (5) (2005) 361.

[5] Safia Djemame, Mohamed Batouche, Combining cellular automata and particle swarm optimization for edge detection, Int. J. Comput. Appl. 57 (14) (2012).

[6] Wesam M. Elsayed, Mohammed Elmogy, B. El-Desouky, Evolutionary behavior of dna sequences analysis using non-uniform probabilistic cellular automata model, Ciencia e Tecnica Vitivinicola 32 (2017) 137–148.

[7] Yu Fengxia, Li Gang, The simulation and improvement of particle swarm optimization based on cellular automata, Proc. Eng. 29 (2012) 1113–1118.

[8] Gary B. Fogel, Kumar Chellapilla, David B. Fogel, Reconstruction of dna sequence information from a simulated dna chip using evolutionary programming, in: International Conference on Evolutionary Programming, Springer, 1998, pp. 427–436.

[9] Niloy Ganguly, Biplab K. Sikdar, Andreas Deutsch, Georey Canright, P. Pal Chaudhuri, A survey on cellular automata, centre for high performance computing, dresden university of technology. Report, Technical Report 9, 2003.

[10] T.R. Gingeras, J.P. Milazzo, D. Sciaky, R.J. Roberts, Computer programs for the assembly of dna sequences, Nucleic Acids Res. 7 (2) (1979) 529–543.

[11] Nick Goldman, Maximum likelihood inference of phylogenetic trees, with special reference to a poisson process model of dna substitution and to parsimony analyses, Syst. Zool. 39 (4) (1990) 345–361.

[12] S.T. Hess, J.D. Blake, R.D. Blake, Wide variations in neighbor-dependent substitution rates, J. Mol. Biol. 236 (4) (1994) 1022–1033.

[13] M.E. Jones, S.M. Thomas, K. Clarke, The application of a linear algebra to the analysis of mutation rates, J. Theor. Biol. 199 (1) (1999) 11.

[14] James Kennedy, Russell C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on, vol. 5, IEEE, 1997, pp. 4104–4108.

[15] Kim Laurio, Fredrik Linker, Ajit Narayanan, Regular biosequence pattern matching with cellular automata, Inf. Sci. 146 (1) (2002) 89–101.

[16] Ming Li, Towards a dna sequencing theory (learning a string), in: Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on, IEEE, 1990, pp. 125–134.

[17] W.P. Maddison, D.R. Maddison, Mesquite: a modular system for evolutionary analysis, version 3.61. The MathWorks Inc.http://www.mesquiteproject.org, 2019.

[18] Vladimir Makarenkov, T-rex: reconstructing and visualizing phylogenetic trees and reticulation networks, Bioinformatics 17 (7) (2001) 664–668.

[19] MATLAB. version 9.6.0 (R2019a). The MathWorks Inc., Natick, Massachusetts, 2019.

[20] Ch Mizas, G.Ch. Sirakoulis, V. Mardiris, Ioannis Karafyllidis, N. Glykos, R. Sandaltzopoulos, Reconstruction of dna sequences using genetic algorithms and cellular automata: towards mutation prediction?, Biosystems 92 (1) (2008) 61–68

[21] Brian R. Morton, Irie V. Bi, Michael D. McMullen, Brandon S. Gaut, Variation in mutation dynamics across the maize genome as a function of regional and flanking base composition, Genetics 172 (1) (2006) 569–577.

[22] Gary J. Olsen, Hideo Matsuda, Ray Hagstrom, Ross Overbeek, fastdnaml: a tool for construction of phylogenetic trees of dna sequences using maximum likelihood, Bioinformatics 10 (1) (1994) 41–48.

[23] Mark Pagel, The maximum likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies, Syst. Biol. 48 (3) (1999) 612–622.

[24] Hannu Peltola, Hans Söderlund, Esko Ukkonen, Seqaid: a dna sequence assembling program based on a mathematical model, Nucl. Acids Res. (1984).

[25] David Posada, Keith A. Crandall, Evaluation of methods for detecting recombination from dna sequences: computer simulations, Proc. Natl. Acad. Sci. 98(24) (2001) 13757–13762.

[26] Aharon Razin, Arthur D. Riggs, Dna methylation and gene function, Science 210 (4470) (1980) 604–610.

[27] Naruya Saitou, Masatoshi Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Mol. Biol. Evol. 4 (4) (1987) 406–425.

[28] Naruya Saitou, Masatoshi Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees, Mol. Biol. Evol. 4 (4) (1987) 406–425.

[29] Hans-Paul Schwefel, Deep insight from simple models of evolution, BioSystems 64 (1–3) (2002) 189–198.

[30] Yang Shi, Hongcheng Liu, Liang Gao, Guohui Zhang, Cellular particle swarm optimization, Inf. Sci. 181(20) (2011) 4460–4493. Special Issue on Interpretable Fuzzy Systems.

[31] Yuhui Shi, Russell Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, IEEE, 1987, pp. 69–73.

[32] Adam Siepel, David Haussler, Phylogenetic estimation of context-dependent substitution rates by maximum likelihood, Mol. Biol. Evol. 21 (3) (2004) 468–488.

[33] Moshe Sipper, Evolving Uniform and Non-uniform Cellular Automata Networks, World Scientific, 1997, pp. 243–285.

[34] G. Ch Sirakoulis, I. Karafyllidis, R. Sandaltzopoulos, Ph. Tsalides, A. Thanailakis, An algorithm for the study of dna sequence evolution based on the genetic code, BioSystems 77 (1–3) (2004) 11–23.

[35] G.Ch. Ch Sirakoulis, Ioannis Karafyllidis, Ch. Mizas, V. Mardiris, Adonios Thanailakis, Ph. Tsalides, A cellular automaton model for the study of dna sequence evolution, Comput. Biol. Med. 33 (5) (2003) 439–453.

[36] Mariëlle Stoelinga, An introduction to probabilistic automata, Bull. EATCS 78 (176–198) (2002) 2.

[37] Ravi Shankar Verma, Vikas Singh, Sanjay Kumar, Dna sequence assembly using particle swarm optimization, Int. J. Comput. Appl. 28(10) (2011).

[38] Xuan Xiao, Shi-Huang Shao, Kuo-Chen Chou, A probability cellular automaton model for hepatitis b viral infections, Biochem. Biophys. Res. Commun. 342 (2) (2006) 605–610.

[39] Yingxu Yang, S.A. Billings, Neighborhood detection and rule selection from cellular automata patterns, IEEE Trans. Syst. Man Cybern.-Part A 30 (6) (2000) 840–847.

[40] Ji-Hong Zhang, Ling-Yun Wu, Xiang-Sun Zhang, Reconstruction of dna sequencing by hybridization, Bioinformatics 19 (1) (2003) 14–21.

[41] Shihua Zhou, Bin Wang, Xuedong Zheng, Changjun Zhou, Study and application of DNA cellular automata self-assembly, Springer, 2014, pp. 654–658.