Modeling and Simulation of Genome Evolution Using Linear Boolean Functions Associated with One Dimensional Cellular Automata

By Prashanthi Govindarajan, Sathya Govindarajan & Ethirajan Govindarajan

Abstract: Structural and functional behavior of genomes could be studied using one dimensional binary-valued three neighborhood cellular automata updating rules. These updating rules are linear Boolean functions, and they are applied to the adjoint sequences of adenine, (A), Thymine (T), Guanine (G) and Cytosine (C) corresponding to the characteristic sequence of a genome. This paper proposes the use of linear Boolean functions, and demonstrates the textural or fractal behavior of genome evolution in terms of nucleotide adjoints.

Keywords: linear boolean functions, cellular automata, genome evolution.

GJCST-G Classification: J.3
Modeling and Simulation of Genome Evolution Using Linear Boolean Functions Associated with One Dimensional Cellular Automata

Prashanthi Govindarajan *, Sathya Govindarajan* & Ethirajan Govindarajan*

Abstract- Structural and functional behavior of genomes could be studied using one dimensional binary-valued three neighborhood cellular automata updating rules. These updating rules are linear Boolean functions, and they are applied to the adjoint sequences of adenine (A), Thymine (T), Guanine (G) and Cytosine (C) corresponding to the characteristic sequence of a genome. This paper proposes the use of linear Boolean functions, and demonstrates the textural or fractal behavior of genome evolution in terms of nucleotide adjoints.

Keywords: linear boolean functions, cellular automata, genome evolution.

I. Introduction

The four nucleotides A, T, G, and C get connected by phosphodiester bonds to form strands. Strand formation depends on innumerable factors related to inter and intra cellular parameters and functions. One cannot precisely say that a particular strand gets formed using such and such rules. The infinite possibilities of strand formation cannot be determined experimentally or in the framework of classical genetics. One can alternatively formulate a notion of “Language of Genomes” wherein one can finitely specify infinite strands, Fig. 1 shows a finitely generated quaternary tree structure of strand formation of nucleic acids.

![Fig. 1: Quaternary tree structure for strand formation](image)

To be precise, Fig. 1 shows three levels of nucleotides. One can generate 64 strands of length 3. As the length increases, the number of strands increases as per the formula $4n$, where $n$ is the length of the strand. Strands of length three are called triplet codons or 3-tuple codons. Similarly, one can think of $n$-tuple codons where $n$ is any number.

Parallel Prediction of Nucleotides using One Dimensional Cellular Automata

A genome sequence is a chain of four nucleotides A, T, G and C. The numerical representation of a genome sequence is a sequence of four numbers 1, 2, 3 and 4. Linear prediction of a strand could be carried out using linear prediction algorithms from a sub sequence of length 8. Alternatively, one can evolve generations of genome sequences from a given full-length genome sequence using one-dimensional cellular automata rules. Section 2 describes the notions of adjoints of nucleotides corresponding to a genome sequence. Section 3 describes the notions of cellular automata and linear Boolean functions. Section 4 provides the results of applying linear Boolean functions on adjoint strings of nucleotides. Section 5 demonstrates the results of combining evolution patterns of adjoint sequences dyadically. Section 6 presents various observations made from the study and proposes future perspectives of cellular automata-based genome analytics.

II. Adjoints of Nucleotides

Adjoint of a particular nucleotide in a genome sequence is the binary sequence obtained by substituting the particular nucleotides in the genome sequence by 1’s and the others by 0’s. For example, let us consider a sample sequence G, A, A, T, G, A, T, T, A, C, C, A, A, G, G, C of length 16. Now the adjoint of adenine (A) is the binary string $A(n) = 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0$. The adjoint of thymine (T) is the binary string $T(n) = 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0$. The adjoint of guanine (G) is the binary string $G(n) = 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1$. The adjoint of cytosine (C) is binary string $C(n) = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0$. The first segment of 40 nucleotides of a genome sequence of Brucella Suis 1330 is considered here for a case study. The actual length of the genome sequence of Brucella Suis 1330 is 5806. The sample sequence is given below.

`GAATGATTCAAGGCGAAGCTCAAGCTCTCCTTCCTTGG`

Author a: Chief Executive Officer, PRC Global Technologies Inc. Ontario, Canada. e-mail: sathyag413@gmail.com

Author b: Director, PRC Global Technologies Inc. Ontario, Canada. e-mail: prashanthi89@gmail.com
Now the adjoints of this sample sequence of length 40 are given below.
A(n) = 01100100101100001100001100000000000000000
T(n) = 00010011000000000000000100010101001100
G(n) = 10010000000001100000001000000000000011
C(n) = 0000000001100011000101001011011000

A cellular automaton is an idealized parallel processing system consisting of an array of numbers (1-D, 2-D and more) realized using updating rules based on certain neighborhood. For example, a one-dimensional cellular automaton would consist of a finite length array as shown below.

### III. Cellular Automata and Linear Boolean Functions

A cellular automaton is an idealized parallel processing system consisting of an array of numbers (1-D, 2-D and more) realized using updating rules based on certain neighborhood. For example, a one-dimensional cellular automaton would consist of a finite length array as shown below.

Consider an ith cell in the array. This cell has a neighbor i-1 on its left and another i+1 on its right. All three put together is called a three-neighborhood. One can assign a site (cell) variable $\xi_i$, to the three-neighborhood cells. At a particular instant of time, these variables take on numerical values, say either a 0 or 1. In such a case, the variables are denoted as $\xi_{i-1}$, $\xi_i$, and $\xi_{i+1}$. The value of the ith cell at the next instant of time is evaluated using an updating rule that involves the present values of the ith, (i-1)th and (i+1)th cells. This updating rule is essentially a linear Boolean function of three variables. One can construct 256 linear Boolean functions as updating rules of one-dimensional three-neighborhood binary-valued cellular automata. Each rule defines an automaton by itself. So, one-dimensional binary-valued three-neighborhood cellular automata (123CA) rules could be used to model adjoints of a genome sequence. The first thirty linear Boolean functions of cellular automata 123CA are listed below with their decimal equivalents.

#### Linear Boolean Function

<table>
<thead>
<tr>
<th>Expression</th>
<th>Decimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>0</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\xi</em>{i+1}$</td>
<td>1</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>2</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\xi_{i+1}$</td>
<td>3</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>4</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\xi</em>{i+1}$</td>
<td>5</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}<em>{i+1}\bar{\xi}</em>{i+1} + \xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>6</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>7</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>8</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>9</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\xi_{i+1}$</td>
<td>10</td>
</tr>
</tbody>
</table>

IV. Cellular Automata Evolutions of Genome Adjoints

The genome sequence of Brucella Suis 1330 is considered here for a case study. Due to space limitations, a part of the genome sequence and its adjoints are shown below. As defined already, adjoint of genome sequence concerning a particular nucleotide is the binary string obtained by marking a ‘1’ in the place of that particular nucleotide and by marking a ‘0’ in the places of other nucleotides. A segment consisting of 60 nucleotides of Brucella Suis 1330 is shown below.

<table>
<thead>
<tr>
<th>Nucleotide</th>
<th>Adjoint of Nucleotide</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>A</td>
</tr>
<tr>
<td>G</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>G</td>
</tr>
</tbody>
</table>

The adjoints of the genome sequence segment are given below.

<table>
<thead>
<tr>
<th>Adjoint</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>0</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\xi</em>{i+1}$</td>
<td>1</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>2</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\xi_{i+1}$</td>
<td>3</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>4</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\xi</em>{i+1}$</td>
<td>5</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}<em>{i+1}\bar{\xi}</em>{i+1} + \xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>6</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>7</td>
</tr>
<tr>
<td>$\xi_{i-1}\bar{\xi}<em>i\bar{\xi}</em>{i+1}$</td>
<td>8</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\bar{\xi}_{i+1}$</td>
<td>9</td>
</tr>
<tr>
<td>$\xi_{i-1}\xi_i\xi_{i+1}$</td>
<td>10</td>
</tr>
</tbody>
</table>
Cellular automata evolutions of adjoints of a genome are carried out using 256 rules of 123CA. As an example, rule number 137 of 123CA, that is, $(\xi_{i-1} \xi_i \xi_{i+1}) + (\xi_i \xi_{i+1})$ is applied to adjoints of Brucella Suis 1330 genome and results shown below in Fig. 2.

![Evolution of A(n)](image1)

![Evolution of T(n)](image2)

![Evolution of G(n)](image3)

![Evolution of C(n)](image4)

Fig. 2: Evolution of adjoints using rule 137 of 123CA

The size of the images shown in Fig. 2 is 500x500, though the actual size is 5806x500. The first 500 columns of the actual images are clipped and presented here for visual clarity. From Fig. 2, it is clear that the evolution pattern of each adjoint is different. One can observe that there are certain fractal patterns in the evolutions and such fractals are distributed in the images very differently. For instance, the zoomed in versions of the evolution patterns of A(n), T(n), G(n) and C(n) using rule 137 are shown in Figs. 3, 4, 5 and 6 respectively.

![Zoomed in version of evolution pattern of A(n)](image5)

![Zoomed in version of evolution pattern of T(n)](image6)

![Zoomed in version of evolution pattern of G(n)](image7)

![Zoomed in version of evolution pattern of C(n)](image8)
It is a matter of an empirical study to examine the conjoint behavior of various evolution patterns of adjoints and it could be carried out by combining evolution patterns of adjoints dyadically. The various dyadic operations are (i) Boolean addition, (ii) Boolean subtraction, (iii) Boolean multiplication, (iv) Boolean division, (v) Dyadic relation of maximum and (vi) Dyadic relation of minimum. Out of these six different dyadic operations and relations, the Boolean operation of binary addition is considered here for the intended study.

**Fig. 5:** Zoomed in version of evolution pattern of G(n)

**Fig. 6:** Zoomed in version of evolution pattern of C(n)

**V. Dyadic Operations between Cellular Automata Evolutions of Genome Adjoints**

The zoomed in versions of the Boolean additions of evolution patterns of A(n), T(n), G(n) and C(n) using rule 137 are shown in Figs. 7, 8, 9 and 10 respectively.

**Fig. 7:** Boolean addition of evolution patterns of adjoints

**Fig. 8:** Zoomed in version of addition of patterns of A and T

**Fig. 9:** Zoomed in version of addition of patterns of A and G

**Fig. 10:** Zoomed in version of addition of patterns of A and C
VI. Observations and Conclusions

From the above empirical study, it is observed that cellular automata modeling and simulation of evolutions of adjoints of a given genome sequence and the inter-pattern operations and relations exhibit distinct patterns of fractals and fractal distributions. The novel technique and results presented in this paper are outcome of prolonged research carried out in the mathematical modeling of genomes and their evolutions. It is evident that one can as well look into the possibilities of genome editing using such cellular automata tools.

ACKNOWLEDGEMENT

The authors express their profound gratitude to the management of Pentagram Research Centre Private Limited, Hyderabad for their boundless support in carrying out research in their premises. The authors further put on record the invaluable guidance of Prof. Dr E G Rajan, President of Pentagram Research and data scientists Mr. Rahul Sharma, Mr. Srikanth Maddikunta and Mr. Shubham Karande.

REFERENCES

6. Robin McKie (24 February 2013). "Scientists attacked over claim that ‘junk DNA’ is vital to life". The Observer.

© 2020 Global Journals


35. Dan Graur, The Origin of Junk DNA: A Historical Whodunit


39. Another source is genome duplication followed by a loss of function due to redundancy.


48. Grünewald, Thomas G P; Bernard, Virginie; Gilardi-Hebenstreit, Pascale; Raynal, Virginie; Surdez, Didier; Aynaud, Marie-Ming; Mirabeau, Olivier; Cidre-Aranaz, Florencia; Tirode, Franck. "Chimeric EWSR1-FLI1 regulates the Ewing sarcoma susceptibility gene EGR2 via a GAGA microsatellite". Nature Genetics 47 (9): 1073–1078. doi:10.1038/ng.3363. PMC 4591073. PMID 26214589.


55. Callaway, Ewen (March 2010). "Junk DNA gets credit for making us who we are". New Scientist.

56. "Plagiarized Errors and Molecular Genetics", talkorigins, by Edward E. Max, M.D., Ph.D.

