A Novel Approach to Genome Editing using Cellular Automata Evolutions of Adjoints Sequences

By Prashanthi Govindarajan, Sathya Govindarajan & Ethirajan Govindarajan

Abstract- This paper proposes a novel method for genome editing using cellular automata evolutions of adjoints of Adenine, Thymine, Guanine, and Cytosine. The adjoints of the given a genome sequence are the characteristic binary string sequences. For example, the adjoint of Adenine of a given genome sequence is a binary string consisting of 0’s and 1’s where 1’s corresponds to the presence of Adenine in the genome sequence. So, one can have four adjoint sequences of Adenine, Thymine, Guanine, and Cytosine corresponding to a given genome sequence. One-dimensional three neighborhood binary value cellular automata rules can be applied to an adjoint sequence and the desired number of evolutions could be obtained. This rule is defined by a linear Boolean function and one can have 256 such linear Boolean functions. Genome editing is carried out by superimposing the evolved adjoint sequence on the original genome sequence or on its successive evolutions. In this manner, one can have four ways of genome editing using four adjoint sequences and evolutions.

Keywords: genome editing, cellular automata, evolutions of adjoints, linear boolean functions.

GJCST-G Classification: J.3
A Novel Approach to Genome Editing using Cellular Automata Evolutions of Adjoints Sequences

Prashanthi Govindarajan*, Sathya Govindarajan* & Ethirajan Govindarajan*

Abstract- This paper proposes a novel method for genome editing using cellular automata evolutions of adjoints of Adenine, Thymine, Guanine, and Cytosine. The adjoints of the given a genome sequence are the characteristic binary string sequences. For example, the adjoint of Adenine of a given genome sequence is a binary string consisting of 0’s and 1’s where 1’s corresponds to the presence of Adenine in the genome sequence. So, one can have four adjoint sequences of Adenine, Thymine, Guanine, and Cytosine corresponding to a given genome sequence. One-dimensional three neighborhood binary value cellular automata rules can be applied to an adjoint sequence and the desired number of evolutions could be obtained. This rule is defined by a linear Boolean function and one can have 256 such linear Boolean functions. Genome editing is carried out by superimposing the evolved adjoint sequence on the original genome sequence or on its successive evolutions. In this manner, one can have four ways of genome editing using four adjoint sequences and evolutions.

Keywords: genome editing, cellular automata, evolutions of adjoints, linear boolean functions.

1. INTRODUCTION

Genome editing is essentially the process of introducing required changes in a given DNA. A protein or enzyme cuts certain portions of a given DNA and substitutes the target sequences of nucleotides by specific chains of nucleotides. For ages, people have been working on the notion of Genome editing. However, only recently, the spurt of activities has been reported in the literature about genome editing. Before getting into the details of the genome editing, it would be apt to reason out why genome editing is viewed as a significant activity. Genome editing is carried out as a health initiative to heal even the so-called incurable diseases associated with genetic problems. Clinically there are many methods of editing genomes among which CRISPR technique is recognized as a reliable technique ratified by the U.S. Food and Drug Administration, pushing in a new era of cancer treatment. CRISPR based therapy is designed to treat blood and bone marrow cancer, which usually affects children and young adults. This therapy is known as CAR-T therapy, and it has shown remarkable results in patients. By eliminating those genes which cause disease, physicians can treat various illnesses ranging from heart disease to Alzheimer’s.

In addition to curing diseases, gene therapy (gene editing) could be used to stop inherited disease in its tracks to save endangered species, and more so, to resurrect extinct species. All such gene therapy techniques are clinical laboratory-based. Alternatively, one can try out the possibilities of developing some of them for genome editing using computational tools and concepts. Genome is a string of nucleotides, and its characteristic sequence is a sequence of A, T, G, and C. Gene editing, in the computer science point of view, is a pattern searching and substituting process, meaning, genome editing is essentially a string processing operation. A genome is a subset of a free monoid A* of an alphabet A which consists of the primitive symbols A, T, G, and C. So, gene editing is viewed as a map ϕ that connects A* to A*. It is in this context, this paper introduces a novel concept of genome editing cellular automata evolutions of adjoint strings of a genome. Section 2 of this paper describes the fundamental notions of adjoints of a genome and their evolution using one-dimensional cellular automata rules defined by linear Boolean functions. Section 3 presents the technique of editing genome sequences using the cellular automata generations of adjoint sequences. Section 4 illustrates the concept with the help of a case study.

II. ADJOINTS OF A GENOME SEQUENCE AND THEIR EVOLUTIONS USING ONE-DIMENSIONAL THREE NEIGHBORHOOD CELLULAR AUTOMATA

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 of BrucellaSuis 1330 for a case study. The actual length of the genome sequence of BrucellaSuis 1330 is 5806. 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>i-1</th>
<th>i</th>
<th>i+1</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>

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 $ξ_{i-1}$, $ξ_i$, and $ξ_{i+1}$ to the three neighborhood cells. At a particular instant of time, these variables take on numerical values, say either a 0 or a 1. In such a case, the variables are denoted as $ξ_i$, $ξ_{i-1}$, and $ξ_{i+1}$. 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 basically 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 twenty linear Boolean functions of cellular automata 123CA are listed below with their decimal equivalents.

<table>
<thead>
<tr>
<th>Linear Boolean Function</th>
<th>Decimal Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0$</td>
<td>$0$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$1$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$2$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$3$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$4$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$5$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$6$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$7$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$8$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$9$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$10$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$11$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$12$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$13$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$14$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$15$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$16$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$17$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$18$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$19$</td>
</tr>
<tr>
<td>$ξ_{i-1}ξ_iξ_{i+1}$</td>
<td>$20$</td>
</tr>
</tbody>
</table>

For the case study rule number 90 is applied to the adjoints of BrucellaSuis 1330 genome sequence and 500 evolutions generated. Rule 90 is shown below.

$ξ_{i-1}ξ_iξ_{i+1} + ξ_{i-1}ξ_iξ_{i+1}$

Since the image of the 500 evolutions of BrucellaSuis 1330 is quite large, a small portion of the images are presented in this paper. Fig. 1 shows evolutions of the adjoints of A(n) and T(n).

![Evolution of Adjoint A(n) using rule 90 of 123CA](image1)

![Evolution of Adjoint T(n) using rule 90 of 123CA](image2)

**Fig. 1:** Evolutions of A(n) and T(n)

![Evolution of Adjoint G(n) using rule 90 of 123CA](image3)

![Evolution of Adjoint C(n) using rule 90 of 123CA](image4)

**Fig. 2:** Evolutions of G (n) and C (n)

### III. NUCLEOTIDE ADJOUTNS BASED GENOME EDITING

As outlined earlier, a genome sequence is a subset of a free monoid $A^*$ of an alphabet $A$ which consists of the primitive symbols A, T, G and C and gene editing is a map $ϕ$ that connects $A^*$ to $A^*$. The map $ϕ$ is a rule that transforms a sequence into another desired sequence. One can use four types of symbol to symbol substitution formulas given below for genome editing.

<table>
<thead>
<tr>
<th>Symbol to be substituted</th>
<th>Substitution symbol</th>
<th>Formula Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>A $\lor$ T $\lor$ G $\lor$ C</td>
<td>$A$</td>
<td>1</td>
</tr>
<tr>
<td>A $\lor$ T $\lor$ G $\lor$ C</td>
<td>$T$</td>
<td>2</td>
</tr>
<tr>
<td>A $\lor$ T $\lor$ G $\lor$ C</td>
<td>$G$</td>
<td>3</td>
</tr>
<tr>
<td>A $\lor$ T $\lor$ G $\lor$ C</td>
<td>$C$</td>
<td>4</td>
</tr>
</tbody>
</table>

Where the symbol $\lor$ denotes the relation of logical OR. Application of formula #1 to any genome sequence is called A-latch. Similarly one can think of T-latch, G-latch and C-latch. A nucleotide latch would give rise to conversion of any genome sequence into a sequence consisting of that particular nucleotide in one step. This is called ‘Nucleotide Saturation’. Thus one can have A-saturation, T-saturation, G-saturation, and C-saturation. To be precise, A-latching of any genome sequence transforms it into A-saturated sequence and it
holds for other three types of latching also. The question that arises here is that whether it is possible to edit a genome sequence using nucleotide latching techniques at preferred locations in the sequence. One such possibility is latching of a genome sequence using cellular automata evolutions of adjoints. Once a genome sequence is latched, the resulting edited sequence is called its elementary transformation. Two types of latching are discussed here (i) latching of elementary transformations of a genome sequence with various cellular automata evolutions of adjoints and (ii) latching of original genome sequence with various cellular automata evolutions of adjoints. Fig. 3 portraits the first approach of latching of elementary transformations of a genome sequence with various cellular automata evolutions of adjoints.

Figs. 5 to 8 portrait the first approach of latching of elementary transformations of a genome sequence with various cellular automata evolutions of adjoints.

Fig. 5 shows the result of A-latching the genome sequence, that is, Adenine based genome editing. Since the image is quite large, a small portion of it is shown here. It was observed that the A-saturation of the genome sequence occurred while editing the previous elementary transformation of the genome sequence using the 32nd evolution of the A(n).

Elementary transformations of the genome sequence are edited using the rule number 90 based cellular automaton evolution of A(n). A-saturation occurred when the 32nd evolution is used for editing.

Fig. 6 shows the result of T-latching the genome sequence, that is, Thymine based genome editing. Since the image is quite large, a small portion of it is shown here. It was observed that the T-saturation of the genome sequence occurred while editing the previous elementary transformation of the genome sequence using the 40th evolution of the T(n).

Fig. 7 shows the result of G-latching the genome sequence, that is, Guanine based genome editing. Since the image is quite large, a small portion of it is shown here. It was observed that the G-saturation of the genome sequence occurred while editing the previous elementary transformation of the genome sequence using the 15th evolution of the G(n).
Elementary transformations of the genome sequence are edited using the rule number 90 based cellular automaton evolution of GT(n). G-saturation occurred when the 15th evolution is used for editing.

**Fig. 7:** Guanine based genome editing (G-saturation at 15th evolution)

Fig. 8 shows the result of C-latching the genome sequence, that is, Cytosine based genome editing. Since the image is quite large, a small portion of it is shown here. It was observed that the C-saturation of the genome sequence occurred while editing the previous elementary transformation of the genome sequence using the 21st evolution of the C(n).

**Fig. 8:** Cytosine based genome editing (C-saturation at 21st evolution)

**Approach #2**

Editing of a genome sequence with various cellular automata evolutions of adjoints

Figs. 9 to 14 portrait the second approach of latching of a given genome sequence with rule number 90 based cellular automaton evolutions of adjoints. Figs. 9 shows Adenine based genome editing on original sequence both in Text Form and in Image Form. The Image Form is obtained using a color coding scheme which paints a red color for Adenine, green color for Thymine, blue color for Guanine and yellow color for Cytosine. Fig. 10 shows the predominant horizontal lines and vertical lines of the image of the edited genome separately. An image processing tool of line detector is used for this purpose.

**Fig. 9:** Edited genome using evolutions of A(n)

**Fig. 10:** Horizontal and Vertical lines of A-latch detected

Figs. 11 shows Thymine based genome editing on original sequence both in Text Form and in Image Form. Fig. 12 shows the predominant horizontal lines and vertical lines of the image of the edited genome separately.

**Fig. 11:** Edited genome using evolutions of T(n)

**Fig. 12:** Horizontal and Vertical lines of T-latch detected
Figs. 13 shows Guanine based genome editing on original sequence both in Text Form and in Image Form. Fig. 14 shows the predominant horizontal lines and vertical lines of the image of the edited genome separately.

![Guanine based genome editing on original sequence (Text Form)](image1)

![Guanine based genome editing on original sequence (Image Form)](image2)

**Fig. 13:** Edited genome using evolutions of G(n)

![Horizontal lines Detected](image3)

![Vertical lines Detected](image4)

**Fig. 14:** Horizontal and Vertical lines of G-latch detected

Figs. 15 shows Cytosine based genome editing on original sequence both in Text Form and in Image Form. Fig. 16 shows the predominant horizontal lines and vertical lines of the image of the edited genome separately.

![Cytosine based genome editing on original sequence](image5)

![Cytosine based genome editing on original sequence (Image Form)](image6)

**Fig. 15:** Edited genome using evolutions of C(n)

![Horizontal lines Detected](image7)

![Vertical lines Detected](image8)

**Fig. 16:** Horizontal and Vertical lines of C-latch detected

---

**Dyadic Operations among edited genomes**

Two relational operations ‘max’ and ‘min’ are carried out on the A-latch, T-latch, G-latch and C-latch edited genomes. Adenine is represented by the number 1, Thymine by 2, Guanine by 3 and Cytosine by 4. Now, the relation \( \max(x, y) \) of two nucleotides is evaluated as the maximum of the numerical values of the nucleotides under comparison. Similarly, the relation \( \min(x, y) \) of two nucleotides is evaluated as the minimum of the numerical values of the nucleotides under comparison.

Fig. 17 shows the result of comparing A-latch and T-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

![AT Max](image9)

![Horizontal line](image10)

![Vertical line](image11)

**Fig. 17:** Max of A and T latches and its horizontal and vertical lines

Fig. 18 shows the result of comparing A-latch and C-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

![AG Max](image12)

![Horizontal line](image13)

![Vertical line](image14)

**Fig. 18:** Max of A and G latches and its horizontal and vertical lines

Fig. 19 shows the result of comparing A-latch and C-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

![AG Max](image15)

![Horizontal line](image16)

![Vertical line](image17)
Fig. 19: Max of A and C latches and its horizontal and vertical lines

Fig. 20 shows the result of comparing T-latch and G-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

Fig. 20: Max of T and G latches and its horizontal and vertical lines

Fig. 21 shows the result of comparing T-latch and C-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

Fig. 21: Max of T and C latches and its horizontal and vertical lines

Fig. 22 shows the result of comparing G-latch and C-latch of BrucellaSuis 1330 genome with the max operator along with its horizontal and vertical lines.

Fig. 22: Max of G and C latches and its horizontal and vertical lines

Fig. 23 shows the result of comparing A-latch and T-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines.

Fig. 23: Min of A and T latches and its horizontal and vertical lines

Fig. 24 shows the result of comparing A-latch and G-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines.

Fig. 24: Min of A and G latches and its horizontal and vertical lines

Fig. 25 shows the result of comparing A-latch and C-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines.

Fig. 25: Min of A and C latches and its horizontal and vertical lines

Fig. 26 shows the result of comparing T-latch and G-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines.

Fig. 26: Min of T and G latches and its horizontal and vertical lines
Fig. 27 shows the result of comparing T-latch and C-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines. Fig. 28 shows the result of comparing G-latch and C-latch of BrucellaSuis 1330 genome with the min operator along with its horizontal and vertical lines.

**Fig. 27:** Min of T and C latches and its horizontal and vertical lines

**Fig. 28:** Min of G and C latches and its horizontal and vertical lines

V. RESULTS AND DISCUSSIONS

The following observations are made from the case study.

1. Editing a genome is a map or a function that transforms a genome into a desired genome sequence.
2. Two approaches could be undertaken for genome editing (i) Editing of elementary transformations of a genome sequence with various cellular automata evolutions of adjoints and(ii) Editing of the given genome sequence with various cellular automata evolutions of adjoints.
3. Using approach 1, one would end up with saturation in short steps. Using approach 2, one would be able to generate edited versions that exhibit periodicities and generic evolutions.
5. The result of each operation exhibits unique behavior in periodicity and generic form.
6. Interpretation of these properties could be made in a better way only by an expert in genetic engineering.

ACKNOWLEDGEMENTS

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.
10. Dan Graur, Yichen Zheng, Nicholas Price, Ricardo B. R. Azevedo1, Rebecca A. Zufall and EranElhaik (2013). "On the immortality of television sets: "function" in the human genome according to the
41. (No citation provided)

40. (No citation provided)

39. (No citation provided)


27. Comings (1972), on the other hand, gave what must be considered the first explicit discussion of the nature of "junk DNA," and was the first to apply the term to all noncoding DNA.

26. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

25. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

24. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

23. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

22. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

21. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

20. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

19. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

18. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

17. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

16. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

15. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

14. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

13. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

12. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

11. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

10. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

9. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

8. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

7. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

6. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

5. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

4. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

3. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

2. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.

1. "junk" DNA has been defined as DNA that is not required for normal cellular function. However, the term "junk DNA" is not universally accepted, and alternative terms such as "non-coding DNA" are often used instead.
