Biologically compatible algorithm for Data storage in DNA

[1]Rasika Naik, [2]Rahul Mhatre, [3]Sanchay Srivastava, [4]Alok Dadlani, [5]Suraj Malpani

[1]Department of Electronics and Telecommunication,
[1]University of Mumbai, India

***Abstract*—A number of methods have been proposed over the last decade for embedding information within deoxyribonucleic acid (DNA). However, mostly none of them comply with bio-logical restrictions. Not adhering to these restrictions can potentially be detrimental to the organism hosting the artificial information-carrying DNA. We propose an algorithm of embedding information in non-coding DNA (ncDNA) without altering the organism's genetic behavior. The proposed scheme also leads to high volume data density and depends on adoption of sequence transformation algorithms**

***Index Terms*— DNA, Data storage, Compression**

I. Introduction

This rapid generation of Big Data requires more storage place and these storage requirements are increasing day by day or rather hour by hour! The conventional storage systems comprise of Digital Memory which is a promising technology. But as the data quantities accrue and become astronomical this technology will require more hardware. Production costs will increase and even the waste generation would take place at a magnified rate.

To cater to this need of future, artificial DNA data embedding technology is under development. This technology developed for storing this information is evolving at a reasonable pace. These devices provide limited storage and will prove insufficient in near future. The potential of deoxyribonucleic acid (DNA) for use as a storage medium of digital data was realized just over a decade ago. To date several data embedding algorithms have been proposed. However, as we will see later, none of them fully comply with some recently highlighted biological restrictions. Not adhering to these restrictions can potentially be detrimental to the organism hosting the artificial information-carrying DNA. We propose an algorithm of embedding information in non-coding DNA (ncDNA) without altering the organism's genetic behavior. The proposed scheme also leads to high volume data density and depends on adoption of sequence transformation algorithms.

II. Prior work

The DNA data embedding field was born a little over a decade ago with the seminal paper by Clelland et al. [1], in which the authors proposed and implemented a data embedding scheme. Alphanumeric data was embedded using a trivial assignment of base groupings to characters. The synthesized DNA in this case was embedded in vitro, but not sub-cloned into an organism's genome. The work of Clelland et al. was built upon by Wong et al. [2], in which they performed in vivo embedding of data in bacterial ncDNA regions. Similar to Clelland et al's encoding scheme, a base to alphanumeric translation table was used. Two bacteria were selected for embedding, E. coli and D. radiodurans. The latter has the ability to survive in harsh environments such as those containing high levels of ionizing radiation, implying that the encoded message would also be resilient under such conditions.

The first paper to discuss error correction for information encoded in DNA was by Smith et al [3]. Since any information embedded in DNA is replicated from generation to generation, any difference between encoded information may be resolved by examining copies obtained from different organisms. Also, there exists genetic machinery in the cell which maintains DNA, providing limited error correction. Despite such inherent error correction abilities, the use of error correction methods at the encoding stage is required to reliably retrieve information after many generations of a host organism.

Arita and Ohashi [4] developed an embedding algorithm which operates in pcDNA regions. The algorithm encodes binary data and was successfully tested in vivo. The main pitfall of this method is that it requires that the original DNA sequence be available at the decoder end in order to decode the embedded message.

One paper of significance was written by Heider and Barnekow [5], in which they proposed two versions of a data embedding algorithm, entitled "DNA-Crypt". The ncDNA version of the DNA-Crypt algorithm is a trivial mapping of bits to bases. The authors also proposed a pcDNA version of their algorithm, and went on to test their proposal in vivo [6]. It was suggested that Hamming code be used in conjunction with DNA-Crypt to increase robustness under mutations, although note that error correction can actually be applied on any DNA data embedding method.

The use of repetition coding as an explicit DNA data embedding method was first proposed by Yachie et al [7]. The premise behind their algorithm is that errors may be corrected by embedding redundant copies of information throughout an organism's genome. The authors performed in vivo embedding of binary data in multiple ncDNA regions. Also included was an in silico analysis of their method, showing the data recovery rate for a varying mutation rate. This work was expanded upon by Haughton and Balado [8].

The first paper to discuss performance analysis of data embedding algorithms and propose performance bounds was by Balado [9]. The achievable rate for both ncDNA and pcDNA under substitution mutations when codons are uniformly distributed was presented. Further bounds were proposed by Balado and Haughton in [10].

III. BIOLOGICAL CONSTRAINTS

Encoding information in sexually reproducing organism is very difficult due to the effects of genetic crossover. Thus, our aim is to produce more biologically compatible DNA for the host organism so that:

i. Our host organism doesn’t get harmed and,
ii. Our information passes through the next generations without any errors.

There exist two distinct regions within the genomes of living organisms: protein-coding (pcDNA) regions and non-protein coding (ncDNA) regions. In the past, ncDNA was thought to have no function, however recent research suggests that up to 80% of ncDNA may be responsible for regulatory functions. In the remaining 20% of ncDNA, it is safe to assume that DNA can be freely overwritten. On the other hand pcDNA regions are responsible for the encoding of proteins, which are the basic building blocks of life. It is possible to modify pcDNA regions to encode information; however the constraints which an algorithm must operate under are more restrictive.

It is essential that any data embedding process does not harm the functionality of the host organism. In order to develop reliable data embedding algorithms the constraints which enable robust encoding must be clear. A modified ncDNA region (in order to embed information) should not be mistaken as a pcDNA region by the genetic machinery. This implies that start codons should not appear in the modifications. This method does not completely guarantee that start codons will not be created; instead, it is designed such that the probability of start codons appearing is low. Moreover, this low likelihood only applies to all the six possible reading frames of DNA. In any case it might still happen that a modified region which originally did not contain start codons may acquire them due to mutations accumulated over a number of generations. This is clearly a potentially unavoidable scenario for any method.

The encoding process starts by identifying whether the input data is of long sequence or short. Shorter messages have more probability of repetition and thus can be further compressed using probabilistic compression techniques. For small sequence data the first step is to perform Burrow wheeler transform and move to front transform on the original text. This generates better context information and obtain high compression ratio. Adaptive Huffman encoding method further compresses the original text message to a much smaller size.

Second phase of the encoding technique is to convert the encrypted binary strand into nucleotide sequence. Although many other mapping functions can be used in literature almost none of the techniques abide by the several biological constraints on a DNA molecule. Our mapping technique is unique and abides by constraints making it viable to be safely inserted into a biologically entity.

Our aim is to optimally embed information within ncDNA while observing the no start codons constraint. Firstly, observe that as |X| = 4 it is possible to encode information by trivially assigning a two bit sequence to each base. This is the foundation of the ncDNA embed-ding algorithm DNA-Crypt by Heider and Barnekow [5], among others. However such a static mapping of bits to DNA symbols does not take into account the no start codons constraint discussed in the previous section. Using such a mapping it is possible that some particular messages will produce start codons in the information-carrying strand. One might think that simply avoiding messages which translate into start codons would bypass this problem. However, this is far from being a solution because there are three possible reading frames where the genetic machinery might find a start codon, plus three additional reading frames in the antiparallel complementary strand.

In order to address this issue we use a variable symbol mapping that we describe next. For generality it is assumed that the host DNA belongs to a eukaryotic organism, for which the start codons are “ATG”, “CTG” and “TTG”, with the complementary codons on the opposite strand being “CAT”,“CAG” and “CAA”. Taking the first two bases of these triplets, the following set of special duplets is defined:

D = {AT, CT, TT, CA}

These duplets indicate that the next encoded symbol in a DNA sequence is a special case since a start codon may be produced if the wrong symbol is encoded. Such a situation is avoided by constantly examining the trailing dinucleotide sequence, d =[yi−2 , yi−1], where i represents the position of encoding within the information-carrying DNA sequence y. If the concatenation of the previous two bases d with the current base yi has the potential to create a start codon (that is, if d ∈ D), then the algorithm restricts the choice of yi to a subset of bases Sd such that no start codon can be produced. Otherwise yi can be freely chosen from X. In order to reflect these conditions, a graduated mapping from the subset Sd to message bits is used to encode the symbol yi.

A schematic of the algorithm is shown in Fig 1. The encoded DNA sequence y is constructed by reading the binary message m and at each point examining the previously encoded dinucleotide d. A lookup of table is performed using d and the next bit(s) to be encoded m, from the message vector m. The base y ∈ Sd is selected for encoding using m ∈ Md. This mapping is performed by locating m in the set Md and choosing the base y from Sd at the corresponding position. Given the dinucleotide sequence d the next message base to be encoded is one belonging to the set Sd. Each bit message found in Md corresponds to a base in Sd.

D	AT	CT	T T	C	$X^2 \setminus D$
|S**d**|	3	3	3	3	4
S**d**	A	A	A	G	A
	T	T	T	T	T
	C	C	C	C	C
					G
	↓ Encode			Decode ↑	
Md	0	0	0	0	00
	10	10	10	10	01
	11	11	11	11	10
					11

Fig 1: Look up table for data mapping algorithm.

The entire encoding technique is summarized in Fig 2.

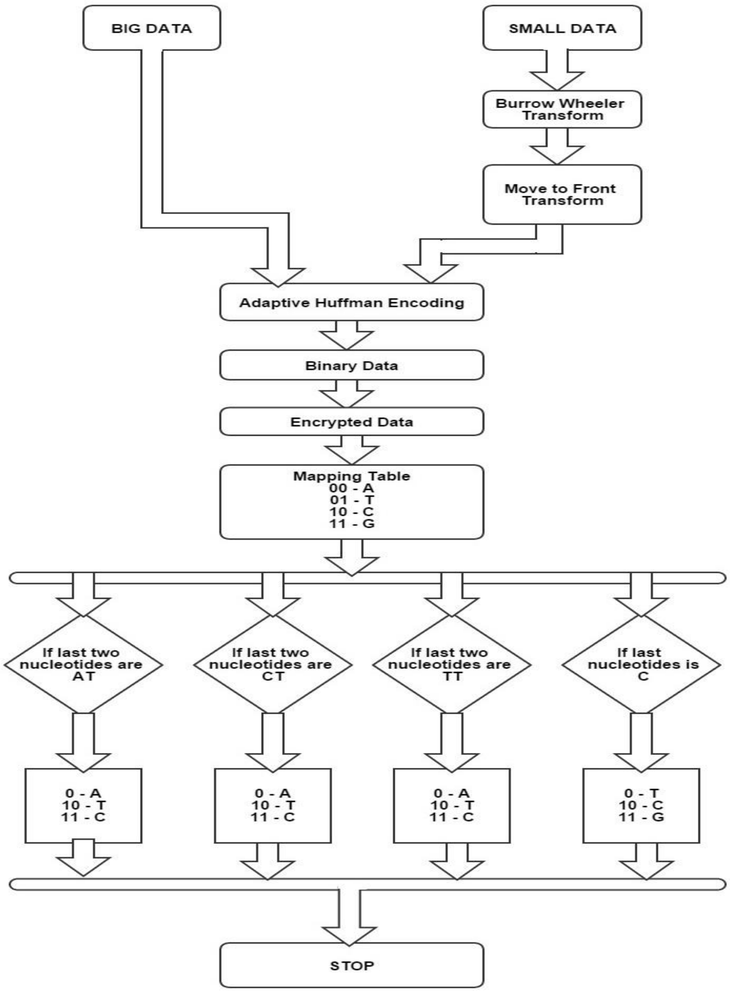


Fig 2: Data to DNA mapping algorithm

The decoding of message can be performed by reversing the encoding scheme. This nucleotide sequence can be artificially synthesized and inserted into the host to maintain the attributes of hereditary media and durable data storage for intensive period of time. We have not proceeded in implementing the biological protocols to insert the sequence in genome of bacteria.

IV. Advantages over earlier method

Biological constraints included a couple of constraints which need to be tackled to consider a method error free. These constraints tackling capacity defines the superiority of one algorithm over other. Firstly, the information-carrying DNA sequence should not hinder the host organisms' development (that is, it should be as biocompatible as possible). Secondly, the embedded data should be retrievable as close as possible to a theoretical threshold (Shannon's capacity), determined by the number of generations a message has been transmitted along and the mutation rate between generations. Finally, the algorithms should make economical use of DNA in terms of data storage, that is, maximize the embeddable payload for a given sequence length.

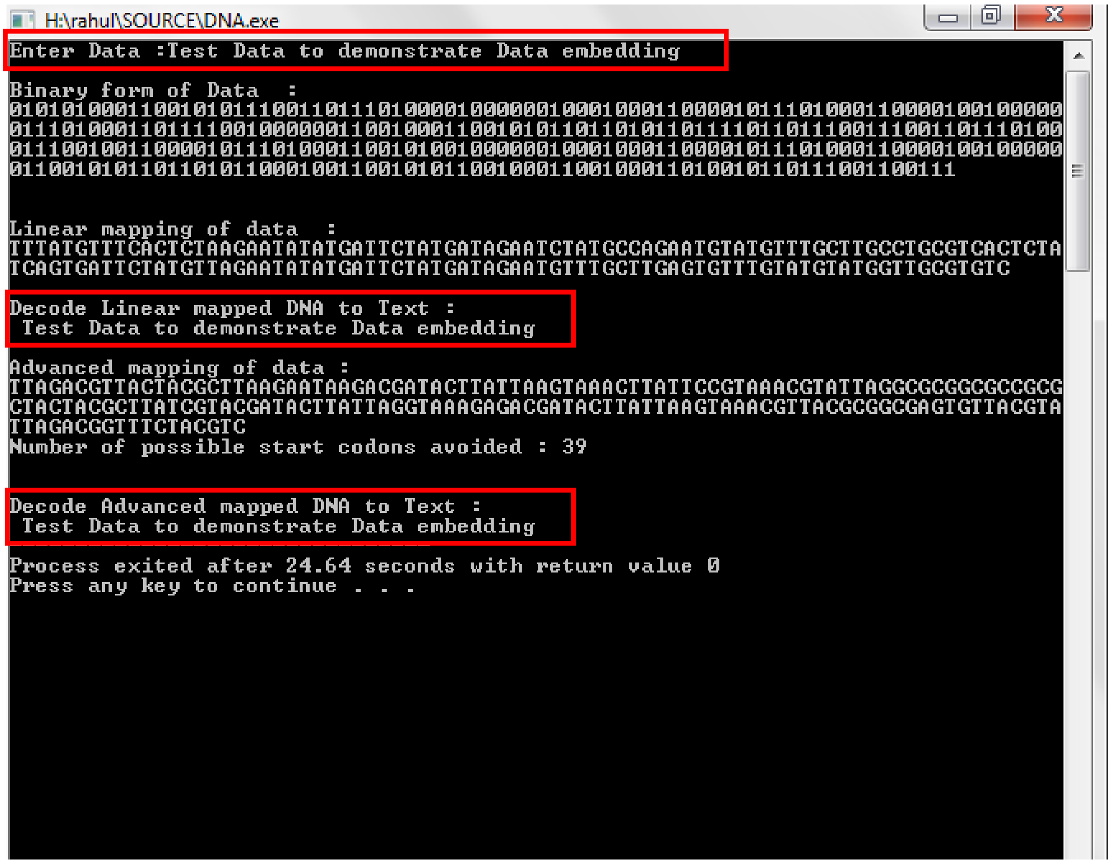
Our proposed work is stated as better by considering the face that it does satisfy these biological constraints. Advanced Mapping Method adheres to biological constraints mentioned above. The results showing the same can be witnessed in the Simulation Results section below. According to the algorithm the 'start codon' must be avoided in order to make the algorithm bio-compatible. As expected the advanced mapping successfully eliminated any start codon thus making it viable for implantation into any biological entity without any detrimental effects.

V. Simulation results

With present emphasis on development of algorithm and learning necessary tools for this project we have successfully implemented the algorithm in C++ code compatible with hardware. Two algorithms were implemented.

1. Direct mapping on binary data bits to DNA nucleotides.
2. Advanced mapping adhering to biological constraints.

Both mapping and decoding for both methods was implemented successfully and the output of the code is seen in Fig 3.



```
Enter Data :Test Data to demonstrate Data embedding

Binary form of Data  :
01010100011001010111001101110100001000000100010001100001011101000110000100100000
01110100011011110010000001100100011001010110110101101111011011100111001101110100
01110010011000010111010001100101001000000100010001100001011101000110000100100000
011001010110110101100010011001010110010001100100011010010110111001100111


Linear mapping of data  :
TTTATGTTTCACTCTAAGAATATATGATTCTATGATAGAATCTATGCCAGAATGTATGTTTGCTTGCCTGCGTCACTCTA
TCAGTGATTCTATGTTAGAATATATGATTCTATGATAGAATGTTTGCTTGAGTGTTTGTATGTATGGTTGCGTGTC

Decode Linear mapped DNA to Text :
 Test Data to demonstrate Data embedding

Advanced mapping of data :
TTAGACGTTACTACGCTTAAGAATAAGACGATACTTATTAAGTAAACTTATTCCGTAAACGTATTAGGCGCGGCGCCGCG
CTACTACGCTTATCGTACGATACTTATTAGGTAAAGAGACGATACTTATTAAGTAAACGTTACGCGGCGAGTGTTACGTA
TTAGACGGTTTCTACGTC
Number of possible start codons avoided : 39

Decode Advanced mapped DNA to Text :
 Test Data to demonstrate Data embedding

Process exited after 24.64 seconds with return value 0
Press any key to continue . . .
```

Fig 3: Output of algorithm

VI. Future Scope

Further features for making the DNA molecule easier to read are into discussion. Smaller DNA strands are easier to synthesize and retrieve. Hence data compression techniques like adaptive Huffman coding are being integrated to reduce the data to be mapped. Addition of encryption will make the data more secure and easier to store. Since 'time taken to encode and decode' is an important matter to be considered, implementation of fixed length DNA strands with each strand containing the data to be stored and also the address of the strand, is under consideration. This way we can fragment the data and retrieve only those parts which are needed. This makes the process faster and economical.

The later part is of developing a soft core processor for this dedicated DNA embedding. Open source soft core processors like MICROBLAZE, LEON III, etc. can be used to implement the same. Implementing the algorithms on these dedicated processors rather than on C++ which use advanced multipurpose computer processors will make the process faster.

VII. CONCLUSION

Thus, we have proposed an algorithm which will be biologically compatible for storing data in DNA. To test the reliability of this algorithm it was implemented using C++ and results were studied. This was done to propose that this method has the capability to be practically realizable. Also to show that it can be realized in a better way than the present ones.

With the growing rate of data produced at a time in this world increasing day by day and limited storage capacity involved in present day electronic methods, this method will act as a pioneer in big data storage. Our method helps this technology to become faster and robust. Thus the above mentioned algorithm and implementation promise us better storage opportunities in future.

VIII. REFERENCES

[1] Clelland CT, Risca V, Bancroft C: Hiding messages in DNA microdots. Nature 1999, 399(6736):533–534.

[2] Wong PC, Wong K, Foote H: Organic data memory using the DNA, approach. Comms ACM 2003, 46:95–98.

[3] Smith GC, Fiddes CC, Hawkins J P Cox, J P: Some possible codes for encrypting data in DNA. Biotech Lett 2003, 25(14):1125–1130.

[4] Arita M, Ohashi Y: Secret signatures inside genomic DNA. Biotechnol Prog 2004, 20(5):1605–1607.

[5] Heider D, Barnekow A: DNA-based Watermarks using the DNA-Crypt Algorithm. BMC Bioinformatics 2007, 8(176)

[6] Heider D, Barnekow A: DNA watermarks: A proof of concept. BMC Mol Biol 2008, 9(40).

[7] Yachie N, Sekiyama K, Sugahara J, Ohashi Y, Tomita M: Alignment-based approach for durable data storage into living organisms. Biotechnol Prog 2007, 23(2):501–505

[8] Haughton D, Balado F: Repetition coding as an effective error correction code for information encoded in DNA. Bioinformatic Bioeng, IEEE Int Symp 2011, 0:253–260

[9] Balado F: On the Shannon capacity of DNA data embedding. In IEEE International Conference on Acoustics, Speech and Signal (ICASSP). Dallas, USA; 2010:1766–1769

[10] Balado F, Haughton D: Gene tagging and the data hiding rate. In 23nd IET Irish Signals and Systems Conference. Ireland: Maynooth; 2012.