HIGH DENSITY DATA STORAGE IN DNA USING AN EFFICIENT MESSAGE ENCODING SCHEME

Rahul Vishwakarma[1] and Newsha Amiri[2]

[1]Tata Consultancy Services, India
derahul@ieee.org
[2]Bangalore University, India

ABSTRACT

This paper suggests a message encoding scheme for small text files in nucleotide strands for ultra high data density storage in DNA. The proposed scheme leads to high volume data density and depends on adoption of sequence transformation algorithms. Compression of small text files must fulfill special requirement since they have small context. The use of transformation algorithm generates better context information for compression with Huffman encoding. We tested the suggested scheme on collection of small text size files. The testing result showed the proposed scheme reduced the number of nucleotides for representing text message over existing method and realization of high data density storage in DNA.

KEYWORDS

Encoding, Nucleotides, Compression, Text

1. INTRODUCTION

This DNA consists of double stranded polymers of four different nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). The primary role of DNA is long-term storage of genetic information. This feature of DNA is analogous to a digital data sequence where two binary bits 0 and 1 are used to store the digital data. This analogous nature of DNA nucleotide with Binary Bits can be exploited to use artificial nucleotide data memory [1] [2]. For example, small text message can be encoded into synthetic nucleotide sequence and can be inserted into genome of living organisms for long term data storage. Further, to enhance the data density for encoded message, original text message can be compressed prior to encoding.

Currently, there exist many losses-less compression algorithms for large text les. All of them need sufficient context information for compression, but context information in small le (50 kB to 100 kB) is difficult to obtain. In small les, context information is sufficient context information only when we process them by characters. Character based compression is most suitable for small les up to 100 kB. Thus we need a good compression algorithm [3], which requires only small context or we need an algorithm that transforms data into another form. An alternative approach is to use Burrow Wheeler transform followed by Move to Front transform. The Huffman encoding is used to convert the original file into compressed one.

The paper suggests a compression scheme for small text message with an introduction of mapping table to encode the data into nucleotide sequence to increase the data density. The organization of paper will be as follows: Section 2 presents a method for data preparation using transforms and compression scheme. Section 3 describes the mapping function for encoding the message into nucleotide sequence. Section 4 describes the method for message encoding and retrieval. Section 5 shows the performance result, and Section 6 contains the conclusion of this paper.

DOI : 10.5121/ijitcs.2012.2204

2. PRIOR WORKS

There has been much advancement in the use of DNA as a data storage device. One of the most critical steps in the realization of biological data storage is the conversion of digital data to nucleotide sequence. Below are few mentioned works which tried to encode the information to be stored in biological sequence.

Battail proposed the idea of using hereditary media as a media for information transmission in communication process.[4] Shuhong Jiao devised a code for DNA based cryptography and stegano-cryptography and implemented in artificial component of DNA.[5] Nozomu Yachie used keyboard scan codes for converting the information to be encoded into hexadecimal value and finally binary values. The last step was to translate the bit data sequence into four multiple oligonucleotide sequence. This was mapped with the nucleotide base pairs. [6] Chinese University of Hong Kong used Quaternary number system to transform the information for mapping it to nucleotides. First they obtained ASCII vale of the information and used the mapping table 0=A, 1=T, 2=C and 3=G for the formation of nucleotide strand. In this method of encoding nucleotides the number of binary bits used for representing the digital information was same as the nucleotide strand. [7]

3. DATA PREPARATION

3.1. Context Information Generation

Currently there are many compression methods that require good context in order to achieve a good compression ratio. One of them is Burrow Wheeler transform [8] [9]. BWT can achieve good compression ratio provided that there is a sufficient context which is formed by frequent occurrence of symbols with same or similar prefixes. Maximizing the context leads to better compression ratio. The Burrow Wheeler algorithm is based on the frequent occurrence of symbol pairs in similar context and it uses this feature for obtaining long strings of the same characters. These strings can be transformed to another form with move to front (MTF) transformation.

3.2. Compression of Text File

We used statistical compression method [10] to compress the data obtained after transformation. The chosen statistical compression scheme was Huffman encoding [11]. Input consists of alphabet A and set W represented in equation (1) and (2) respectively. Output is a set of binary sequence in equation (3), which must satisfy the goal (4) for all the codes with the given condition [12].

$$A = \{a_1, a_2, \ldots, a_n\} \quad (1)$$

$$W = \{w_1, w_2, \ldots, w_n\} \quad (2)$$

$$w_i = \text{weight}(a_i),\ 1 < i < n$$

$$C(A,W) = \{c_1, c_2, \ldots, c_n\} \quad (3)$$

4. MAPPING FUNCTION

4.1. Mapping Table

Mapping table consists of binary bits and nucleotides. Binary value is represented as 0 and 1. Nucleotides are represented as A, C, G and T. Four binary bits are represented by two nucleotide base pairs resulting in sixteen such combinations as shown in Mapping Table [13]. The reason for choosing four bits for two nucleotides is that the output of Huffman encoding

here is Hexadecimal value (radix =16). So we need sixteen such combinations to represent this in binary and then nucleotides.

Mapping Table

Binary - nts.	Binary - nts.	Binary - nts.	Binary - nts.
0000 - AA	0100 - AC	1000 - AG	1100 - AT
0001 - CA	0101 - CC	1001 - CG	1101 - CT
0010 - GA	0110 - GC	1010 - GG	1110 - GT
0011 - TA	0111 - TC	1011 - TG	1111 - TT

4.2. Encryption

The encoded message must be encrypted in order to maintain its security [14]. For this purpose One time pad encryption is used. The requirement for one time pad is that the number of bits of random key must be the same length as of the message to be encoded. The encryption is processed character by character. The secrecy property of the encrypted message depends upon the generated random pad and the decryption of the message is impossible without knowing the true random key and makes it mathematically unbreakable [15].

5. MESSAGE ENCODING AND RETRIEVAL

We have implemented the data encoding in nucleotides by integrating the trans-formation algorithm with statistical compression scheme. Here we have demonstrated our encoding and message retrieval scheme on small text: OPERATION BARBAROSSA.

The first step was to perform Burrow wheeler transform and move to front transform on the original text. This was done to generate better context information and obtain high compression ratio.

The security of the encoded message was maintained by encryption method. The encryption method used was One Time Pad where a randomly generated binary strand was XORed with the binary strand obtained from Huffman en-coding. We used a random function generator for generating the random binary sequence. Only in the last step, Huffman encoding method was introduced which compressed the original text message to a much smaller size. The next step toward message encoding was to use mapping table. The generated binary strand obtained after Huffman encoding was mapped to nucleotides according to the mapping table.

Second phase of our work was to convert the encrypted binary strand into nucleotide sequence. Although many other mapping functions can be used, but for our convenience we used two nucleotides to represent four binary bits, as hexadecimal (radix =16) value is being converted to four bit binary representation and thus leading to formulation of original text message in form of nucleotide sequence.

The decoding of message can be performed by reversing the encoding scheme. This is explained in Figure 1. 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 [1]. We have not proceeded in implementing the biological protocols to insert the sequence in genome of bacteria.

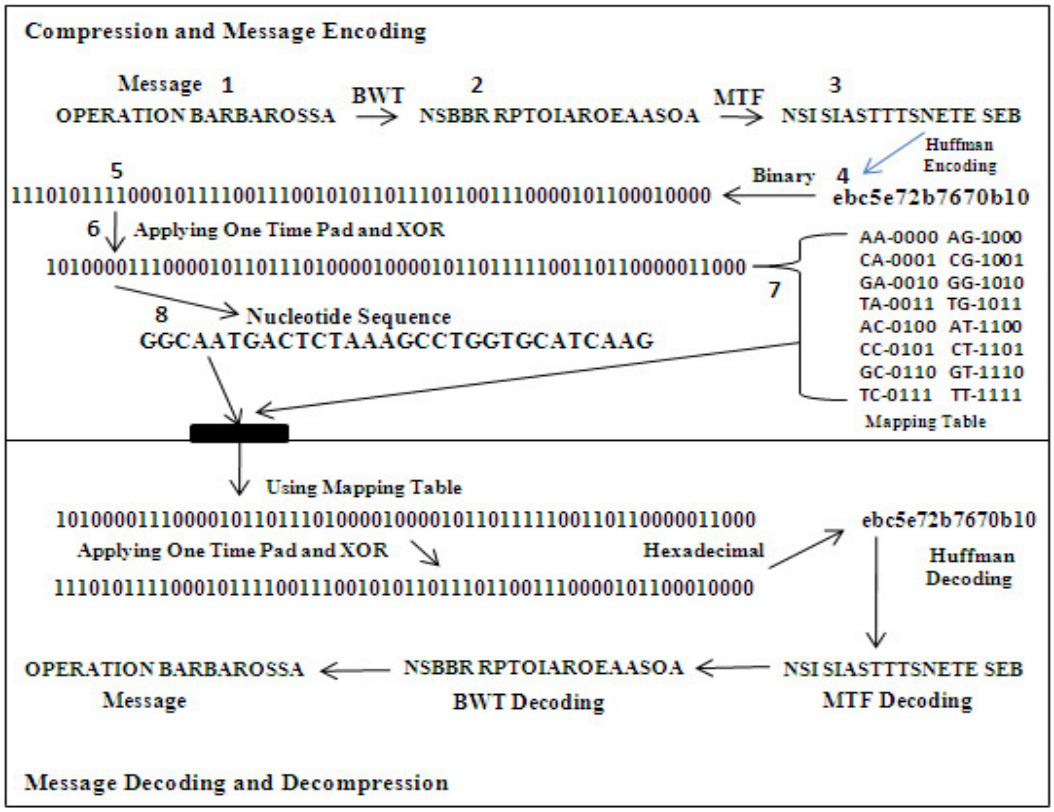

Fig. 1. Message encoding and decoding scheme in nucleotides.

6. PERFORMANCE

The suggested method was compared with two different methods on a set of les. The les that we used for testing is available at http://testdata.idlecool.net. This is a collection of single line texts. The original form was used for compression and encoding scheme.

The first method was comparison of nt-1 and nt-3, where nt-1 represents the number of nucleotides used to represent the original text message after converting the text message to binary and mapping it with nucleotides, and nt-3 represents the number of nucleotides obtained after encoding the nucleotide sequence after performing transformation and then applying compression algorithm on the same text message. In this algorithm, Burrow-Wheeler-Transformation and Move to Front transform was applied to the text message.

The comparison in the second method was used for demonstrating the importance of context information generation using transformation algorithm. Here we compared nt-2 and nt-3, where nt-2 represents the number of nucleotides obtained after encoding then nucleotides without applying transformation algorithm and nt-3 with transformation algorithm on the same text le. The compression efficiency was tested with many tests over several les. The size of text les chosen varied from 140 Bits to 700 Bits.

Experimental result showed that the maximal compression efficiency was achieved by applying transformation in the first step. This transformation generates better context information needed to compress text les of very small size (1000 Bytes). Result for encoding nucleotides without performing transform is depicted in the Figure 3. This shows that transformation prior to compression reduces the number of nucleotides to represent the same text message. The mean compression factor for the eight tested le was 2.076. As can be seen form results above, our

algorithm for small message is better when incorporated with transformation algorithm. Even though, the difference in compression factor was 0.294. This is a step toward reducing the number of nucleotides for message strand and consecutively reducing the cost factor in artificially synthesizing the nucleotide strand for encoding message.

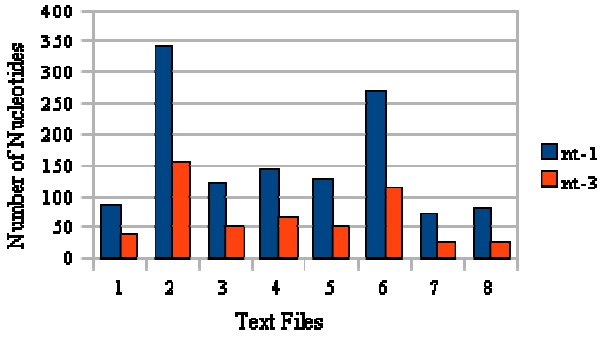

Fig. 2. Comparison between nt-1 and nt-3.

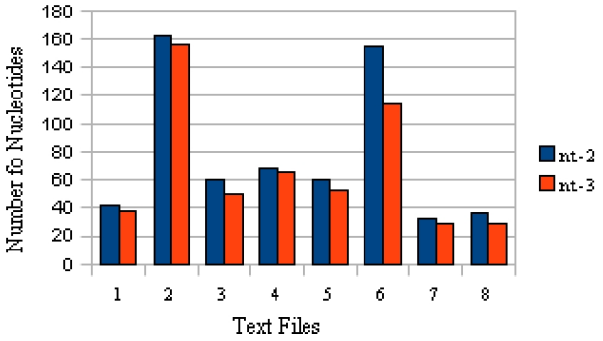

Fig. 3. Comparison between nt-2 and nt-3

7. CONCLUSION

This paper describes a data encoding method to achieve high volume data density by reducing the number of nucleotides. The primary focus of this study was to encode data for les of very small size.

Data encoding method was performed into two steps. The first step was to compress the original text message. This was achieved using transformation and compression algorithm. Second step was introduction of mapping table, which finally maps the binary strand to nucleotide sequence.

The contributions of this study are summarized as follows. First, while majority of previous experiments just mapped the binary strand to nucleotide sequence, this study reduced the number of nucleotides to represent the same in-formation, finally reducing the cost factor for artificially synthesizing nucleotide strand in laboratory. Second, this study uses transformation on original text message prior to compression to achieve better context information, resulting in a compression factor of 2.37.

This scheme may be useful for many applications which need to store small message for long period of time, e.g. military implications [16], signatures of living modified organism (LMOs) and as valuable heritable media. Future work may focus on modification of transformation algorithm and designing other mapping function for encoding nucleotide sequence.

REFERENCES

[1] Cox, J.P.: Long-term data storage in DNA. Trends Biotechnol. 19, 247–250 (2001).

[2] Bancroft, C., Bowler, T., Bloom, B., Clelland, C.T.: Long-term storage of information inDNA. Science 293, 1763–1765 (2001)

[3] Ziviani, N., Moura, E., Navarro, G., Baeza-Yates, R.: Compression: a key for nextgeneration text reyrival systems. IEEE Computer 33, 37–44 (2000)

[4] Battail, G.: Heredity as an Encoded Communication Process. IEEE Transactions on Information Theory 56(2), 678–687 (2010)

[5] Jiao, S., Goutte, R.: Code for encryption hiding data into genomic DNA of living organisms. In: Signal Processing ICSP 2008. pp. 2166–2169 (2008)

[6] Yachie, N., Sekiyma, K., Sugahara, J., Ohashi, Y., Tomita, M.: Alignement- Based Approach for Durable Data Storage into Living Organ-isms. Biotechnol. Prog. 23, 501–505 (2007)

[7] Chinese University of Hong Kong, http://www.cuhk.edu.hk/cpr/pressrelease/101124e.htm

[8] Lansky, J., Chernik, K., Vlickova, Z.: Comparison of Text Models for BWT. In: Data Compression Conference, DCC 2007, p. 389 (March 2007)

[9] Burrows, M., Wheeler, D.J.: A block-sorting lossless data compression algorithm.Technical Reprot, Digital System Research Center Research Reoprt 124 (1994)

[10] Pandya, M.K.: Compression: efficiency of varied compression techniques. Technical Report, University of Brunnel, UK (2000)

[11] Huffman, D.A.: A method for the construction of minimum-redundancy codes. Proceedings of IRE 40(9), 1098–1101 (1952)

[12] Nelson, M., Gailly, J.L.: The Data Compression Book. M and T Books (1995)

[13] . Ercegovac, M.D., Lang, T., Moreno, J.: Introduction to Digital Systems. John Wiley and Sons Inc., Chichester (1999)

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

[15] Shannon, C.: Communication Theory of Secrecy Systems. Bell System Tech-nical Journal 28(4), 656715 (1949)

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