Enhanced Single Bit DNA Squeezer (ESBDNAS)

Dr. S. Panneer Arokiaraj[1], Alam Jahaan[2]

[1]Associate Professor, Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India

[2]Research Scholar, Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India

ABSTRACT

DNA Sequences contain concatenation of the four nucleotides or bases namely, Adenine, Guanine, Thymine and Cytosine. These nucleotides form a double stranded helix with each base of one strand joined to its complement on the other strand by using hydrogen bonds through base pairing rules. DNA compression may be carried out by reducing redundancy and exploiting the properties of DNA sequences. This paper proposes ENHANCED SINGLE BIT DNA SQUEEZER (ESBDNAS) an enhanced bit based lossless compression algorithm to compress a DNA sequence which implements two stages and is similar to SBDNAS (Single Bit DNA Squeezer) Algorithm. The proposed method converts the bases to bits 0 or 1 using the BDNAS algorithm and exploits the properties of DNA sequences that are inherent in it by substituting exact repeats, palindromes and their respective reverses for selected sub-sequences. This method achieves a compression ratio that is better than the existing lossless bit based DNA sequence compression algorithms.

Keywords : DNA Sequence, Lossless Compression, Bit Based Method, DNA Compression, Bit DNA Squeezer, SBDNAS, Bit- Based DNA Compression

I. INTRODUCTION

Compression aims at reduction of file size for efficient storage, spontaneous transmission and cost effectiveness. Compression also deals with Dynamic exchange of data, Prompt Searches, Easy Retrieval, and Speedy Transfer of Data.

DNA Molecule

DNA is the molecule present in the cells of living organisms which stores genetic instructions used in the various process of living organisms such as growth and development, functioning and reproduction and it forms the basis for life [1]. DNA comprises of a phosphate group, a 5-carbon sugar (Deoxyribose) and a nitrogenous base which is any one of {A, C, T, G} that is Adenine, Cytosine, Thymine and Guanine respectively as shown in Figure1. Many such molecules combine to form a single strand and each base is connected to its complement on the next strand with hydrogen bonds [2]. Twisting two such strands gives rise to a double stranded helix as in Figure 2.

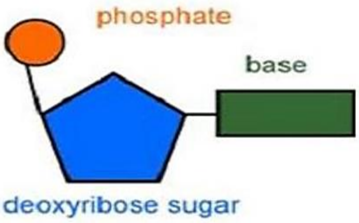

Figure 1 : DNA Molecule

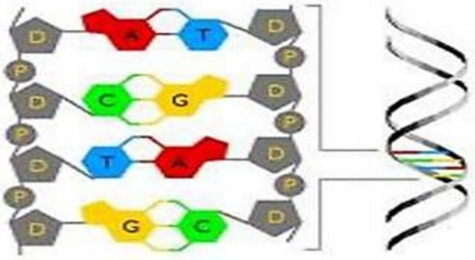

Figure 2 : Helical Structure

CSEIT183760 | Received : 10 Oct 2018 | Accepted : 24 Oct 2018 | September-October - 2018 [3 (7) : 345-351]

Properties of a DNA sequence

On observation of sub-sequences in any DNA sequence, certain properties are noticed which form the criteria for various DNA compression algorithms.

- Repeated substrings in DNA sequences.
- Repeated complements of substrings
- Repeated palindromes.
- Repeated reverse complements [3]

Repeated substrings are redundant fragments of a selected sequence that occurs in the same order. Repeated palindrome is the fragment in the reverse order like any text complement that is spelt the same from backwards, for example the word 'dad'. In repeated complements and reverse complements, the base pairing rule is used and each nucleotide is replaced by its complement, that is A is replaced by T and C is replaced by G or vice versa.

DNA Compression

The essentiality for DNA compression arises in order to overcome storage constraint and accomplish high speed transmission by reducing the size of sequence to reduced size. Generally, any sequence may be compressed by identifying redundancy and the vital information it contains. Lossless compression methods need to be developed to compress DNA sequences so as to preserve the originality of the sequence. [4]

Performance Metrics

There are a number of Quality measures used to determine the appropriate compression algorithm, here only Compression ratio is used since the example sequence considered to implement the ESBDNAS algorithm is very small compared to any original DNA sequence.

Compression ratio: The ratio between the sizes of compressed to original sequence [5].

$$\textbf{Compression Ratio} = \frac{\textbf{Compressed file size}}{\textbf{Original File Size}}$$

While analysing an algorithm, time complexity and space complexity [6] are generally considered. In the proposed algorithm, space complexity has been computed in terms of compressed size whereas, time complexity depends on Compression and Decompression time. Since only a miniscule DNA sequence has been used to implement the algorithm also due to the use of higher end hardware and software, Compression and Decompression time is negligible hence has not been considered, whereas compressed size has been computed and in turn compression ratios have been calculated for different case scenarios such as:

- Best Case: Best case efficiency of ESBDNAS algorithm has maximum replacements of vector repeats or palindromes or their respective complements.
- Average Case: The Average case efficiency defines the compression ratio of a random input which is different from the worst case and the best-case efficiency.
- Worst case: In the worst case there are no or very few replacements made. In this algorithm, the worst-case compression ratio is the highest.

This paper is organized as follows. Section II reviews various existing two bit based DNA compression algorithms. Section III describes the proposed method and Section IV analyses the results obtained. Finally followed by conclusion in Section V.

II. TWO BIT BASED DNA COMPRESSION ALGORITHMS

General 2 bit based methods comprises of a pre-processing stage, in which the bases are assigned 2 bits each such as, A=00, T=01, G=10, C=11 and then the coding stage follows which may be new techniques or enhanced existing techniques [7] .

One bit based technique Bit DNA Squeezer (*BDNAS*) [8] converts each base to either 0 or 1 and Single Bit DNA Squeezer (*SBDNAS*) [9] further compresses the reduced DNA sequence using (*VCT)* technique. The improvement of the single bit based technique over a 2 bit based technique is that each base is assigned a single bit, hence the size of the base reduces from 8 bits (1 byte) to 1 bit, whereas, in 2 bit based techniques, a base of 1 byte (8 bits) is reduced to 2 bits. The following two-bit based algorithms contain two phases with each phase being similar in the pre-processing phase but different in the coding phase:

Rajeswari et al, proposed the GENBIT Compress tool [10] for compressing DNA sequences based on a novel concept of assigning binary bits. Here, a fragment with length n of a DNA sequence is input, and is divided into n/4 fragments. They got a compression ratio for best case of 1.125 bpb, 1.727 bpb for average case and 2.238 bpb for worst case.

HUFFBIT Compress was proposed by Rajeswari et al., [11] for DNA sequences. In this algorithm a bit pre-processing stage takes place before the encoding. It achieves the compression ratio using the concept of Extended Binary Tree. Compression Ratio was calculated using a 1000 base sequence in order to compute best, average and worst case. An average of 1.006 bits per base for best case, 1.611 bpb for average case and 2.109 bpb for worst case was achieved.

Rajeswari et al., later on introduced the DNABIT Compress tool [12] which assigns binary bits in the bit pre-processing stage to exact and reverse repeat fragments of DNA sequences. The average compression ratio is below 1.58 bits per base

GenCodex inroduced by Satyanvesh et al., [13] is a two phased algorithm it uses graphical processing units and multi-cores. A code byte is used for every eight bytes of the compressed data which is set to 1 if there is a repetition.

DNACRAMP by Prasad et al., [14] performs the encoding and decoding process for both repetitive and non-repetitive sequences with the help of a two-stage index bounded linear array data structure using basic procedural language. The algorithm DNACRAMP obtains better compression ratios with an average of 1.143 bpb.

Prasad introduced PGBC "Partitioned Group Binary Compression" [15] suitable for non-repetitive DNA sequences of Genomes. Here encoding process starts after bit pre-processing where each six part is taken as a partition.

Functionality Of BDNAS & SBDNAS Algorithms:

- **BDNAS (Bit DNA Squeezer)** Algorithm: An improvement over general two bit based Algorithms. It concentrates on reducing the sequence size considerably by assigning a single bit (0, 1) per base.

The compression process involves two steps:

Step1: The frequency of each nucleotide is calculated.

Step2: Depending on the order of the frequency of each nucleotide in the dataset. The nucleotide with highest frequency is assigned the value 0. The nucleotide with second highest frequency is assigned the value 1. The nucleotide with third frequency is searched and its position is recorded in a position map then it is assigned the value 0. Finally, the last nucleotide's position is recorded in the position map then it is assigned the value 1.

The Decompression process uses the position map and the compressed sequence as input, it converts all the 0's and 1's in the compressed sequence according to the positions recorded in the position map to its corresponding third and fourth nucleotides in the compressed sequence respectively. Finally, the remaining 0's are converted to the first nucleotide and 1's to the second nucleotide respectively.

- **SBDNAS (Single Bit DNA Squeezer)** Algorithm: An optimal algorithm that enhances the BDNAS technique by using it as the pre-processing Phase and then compresses the sequence further using the Vector Coding Technique (VCT). It searches for a chosen vector and replaces only the exact repeats with a unique symbol.

The decompression process is achieved by executing both the Phases in reverse order. Decoding is performed for the coding Phase first then the decompression for pre-processing Phase (BDNAS) is performed.

III. PROPOSED ALGORITHM

An Improved algorithm Enhanced Single Bit DNA Squeezer (ESBDNAS) is proposed by enhancing the SBDNAS technique by modifying its Phase II, it's coding phase. The phase I or the pre- processing phase implements the BDNAS algorithm to assign a single bit for each base and Phase II implements the Extended Vector coding Technique (EVCT) which searches for a chosen vector and encodes not only the exact repeats, but encodes the complements, reverse and reverse complements of the vector too.

- **Methodology:**

The compression process involves two Phases: Phase I for pre- processing or converting bases to bits and Phase II for Enhanced Vector Coding Technique.

Phase I: Pre-processing Phase where **BDNAS** algorithm is implemented. In this Phase each nucleotide is assigned one bit either 0 or 1 according to the decreasing order of its frequency. An example sequence of 200 bases is used as input for which the compressed sequence along with the position map is the output.

Phase II: Coding Phase implements the Enhanced Vector Coding Technique (**EVCT**). The output sequence from the previous phase is considered as the input sequence for Phase II. A row vector is selected and is searched in the input sequence for exact repeats, palindromes, complements and reverse complements. Then for each outcome unique characters are assigned. This process of searching and assigning a unique character to different outcomes is repeated for the entire sequence.

- **ESBDNAS Encoding Algorithm:**

Input: Input example sequence Containing A, T, G, C
Output: Encoded Sequence & Position Map
Procedure Encode:
Begin
Step I: Begin compression for PhaseI (Pre-Processing Phase) using BDNAS algorithm
Step II: Calculate the frequency of each nucleotide in given input sequence.

1. Depending on the frequency assign 0 to all the highest frequency and 1 to all the second highest frequency.
2. Again search for nucleotide with third frequency and note down its position in the position map then replace it with 0.
3. Similarly replace the final nucleotide with 1 after noting down its position in the position map.
4. Compressed Sequence and Position Map are created as output of the source sequence.

Step III: Consider compressed sequence and start with Coding Phase using Enhanced Vector Coding Technique.

1. Select a single row vector with more than 8 bits.
2. Search for properties of DNA sequences such as occurrence of the exact vector, its reverse, its complement and reverse complement.
3. Replace each occurrence with a unique character ‘a’, ’b’, ’c’, ’d’ respectively based on each property for the entire sequence.
4. The output is the completely compressed sequence obtained.

End

- **ESBDNAS DECODING ALGORITHM**

Input: Input sequence, Position Map
Output: Original Sequence
Procedure Decode:
Begin

1. Decompression of Phase II (Coding Phase) is carried out first
2. Search for the unique character ‘a’
3. Replace it with the Vector for the entire sequence
4. Repeat step2 and step3 for characters ’b’, ‘c’, ‘d’
5. Decompression of Phase I is carried out by reading the Position Map
6. For each entry equivalent to 0 in Position Map, go to corresponding position in compressed sequence and change it to 3rd Nucleotide.
7. Similarly change the entry for 1 in Position Map to 4th nucleotide at the same position in compressed sequence.
8. Replace all the remaining 0’s to 1st nucleotide and 1’s in compressed sequence to 2nd nucleotide.
9. The original sequence is obtained.

End

Illustration of ESBDNAS

Consider an example DNA sequence containing 200 bases which is a concatenation of the four nucleotides A, T, C, G.

```
CTGTATTGGAACATGCCCCTACCTAAAGCCAGC
CCCGGTCGCCCTAAAAGGCATTTATTGGAATTA
AAAGCCAGGGACACTACCTTTCATTTTAGGGAC
CTTTTATTTAAATTTTATGGAGTTTAACGATAAA
TCCCCTTTAAATTACACTAAAGGAGAGACAGCC
AAGGGAAATGAACTAAACGGTAAATAAACGGA
GA
```

Phase I: BDNAS Algorithm is applied to the DNA sequence above

Step 1: Total Nucleotides (bases) = **200**

Frequency of A (f_A)= 70; Frequency of T (f_T) = 51;
Frequency of G (f_C) =41; Frequency of C (f_G) = 38
Such that, Đ = $\Sigma(f_A + f_T + f_C + f_G)$ = **200**.
Since $f_A > f_T > f_C > f_G$

Step 2 : Assigning A=0 and T=1;

A partially compressed sequence is obtained.

```
C1G1011GG00C01GCCCC10CC1000GCC0GCCCCG
G1CGCCC10000GGC0111011GG00110000GCC0GG
G0C0C10CC111C011110GGG0CC1111011100011110
1GG0G11100CG010001CCCC111000110C0C1000GG
0G0G0C0GCC00GGG0001G00C1000CGG10001000C
GG0G0
```

Storing the position of G in PosMap & Assigning G= 1
Storing the position of C in PosMap & Assigning C= 0
The compressed sequence along with the position map is obtained.

```
0111011110000110000100010001000100001110100001
000011001110111110011000010001110000100011100 1
1110111000111101110001111011101111100010100010
000111000110000100011010100010000111000110001
000011100010000110 10
```

PosMap [ ] [ ] = { { 0,11,15, 16, 17, 18, …};{ 2, 7, 8, 14, 27, … } }

Phase II of ESBDNAS: (Implementation of EVCT)

Consider Row Vector **a** = 10001000011
Complement of a = **b** = 01110111100
Reverse (palindrome) of a = **c** = 11000010001
Complement of Reverse of a = **d** = 00111101110

Searching for **a**, **b**, **c** & **d** and assigning the respective characters. The compressed sequence is obtained where 15 encoding has been performed such that 6a+3b+4c+2d=15 replacements has been made as given below

b00c000a10a0bcc11dd001111b010a1000c1010a10001
aa010

IV. PERFORMANCE ANALYSIS

Analysis for the ESBDNAS algorithm is performed for the best case, average case and worst case [11].

Best case: Number of replacements = 15 (6a+3b+4c+2d) 6 repeats of vector, 3 complements of vector, 4 palindromes and 2 complements of the palindrome were found in the sequence. Therefore 200 bases are converted to 200 bits using BDNAS algorithm and then using EVCT technique further it is reduced to 155 bits where 3 bits are reduced for each replacement.

200 bases ➡200 bits ➡155 bits [35bits + (15*8) bits]
Compression ratio = bits / bases = 155 / 200 bpb

∴ Compression Ratio for best case = 0.775 bpb

∴ Compression Ratio for average case = 185/200
= 0.925 bpb

∴ Compression Ratio for worst case = 197/200
= 0.98 bpb

The computation for compression ratios for the three different cases has been tabulated below and the results are depicted in the graph. The compression ratio for the worst case is 0.98 bits per base and for the best case is 0.775 bits per base.

Table 1 : Comparison of compression ratio for cases

Efficiency Cases	**Best Case**	**Average Case**	**Worst case**
No. of bases (Original Sequence)	200	200	200
No. of bits (Compressed sequence)	155	185	197
No. of replacements	15	5	1
Compression Ratio bits/base	155/200 = 0.775	185/200 = 0.925	197/200 = 0.985

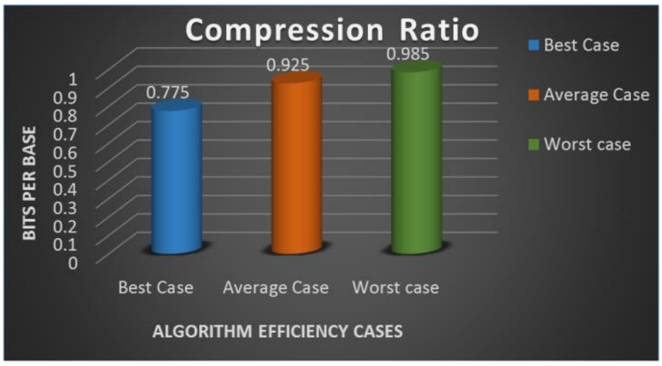

Chart 1 : Comparison of compression ratios

V. CONCLUSION

Developing new algorithms and further improvising and enhancing existing or new algorithms yield better performance and compression ratios. General two-bit based coding methods have become popular in DNA Compression, where the four bases are assigned values 00, 01, 10 and 11 in any order. In the proposed ESBDNAS algorithm the VCT technique from SBDNAS algorithm has been enhanced to search not only the repeats of the selected vector but also to search for the complements of the vector, palindromes and their complements which are replaced with unique symbols for each property. Analysis of the algorithm is performed for the best case, average case and worst case. Compression ratio is calculated and tabulated, Results are analysed using a chart for the three efficiency case scenarios.

VI. REFERENCES

1. https://en.wikipedia.org/wiki/DNA
2. https://en.wikipedia.org/wiki/Introduction_to_genetics
3. Manzini G. and Raster0 M., "A simple and fast DNA compressor, Software: Practice and Experience", MUIR support projects(ALINWEB), vol. 34(14), pp.1397-1411, 2004
4. AlamJahaan, Dr T.N. Ravi, "Scrutiny Of Lossless Compression Techniques Using A Few Quality Measures", International Journal Of Advanced Research In Computer Science And Applications Issn 2321- 872x, Volume 4, Issue 3, March 2016.
5. SR. Kodituwakku Et. Al. "Comparison Of Lossless Data Compression Algorithms For Text Data", Indian Journal Of Computer Science And Engineering, Vol 1 No 4 416-425
6. https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/
7. Nour S. Bakr et al.: "DNA Lossless Compression Algorithms: Review", American Journal of Bioinformatics Research, p-ISSN: 2167-6992 e-ISSN: 2167-6976, 2013; 3(3): 72-81, doi:10.5923/j.bioinformatics.20130303.04
8. Alam Jahaan ,Dr T.N. Ravi, , Dr. S. Panneer Arokiaraj, "Bit DNA Squeezer (BDNAS) : A Unique Technique for Dna Compression", International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307
9. Alam Jahaan ,Dr T.N. Ravi, "Single Bit Dna Squeezer (Sbdnas): An Enhancement Of BDNAS Algorithm", International Journal of Scientific Research in Computer Science, Engineering and Information Technology 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307
10. Rajeswari, P. R., and Apparao, A., 2010," Genbit Compress Tool (GBC): A Java-Based Tool To Compress DNA Sequences and Compute Compression Ratio (BITS/BASE) Of Genomes", International Journal of Computer Science and Information Technology, 2(3)
11. Rajeswari, P. R., Apparao, A., and Kumar, R. K., 2010, "HUFFBIT COMPRESS Algorithm to compress DNA sequences using extended binary tree", Journal of Theoretical and Applied Information Technology, 13(2), 101-106
12. Rajeswari, P. R., and Apparao, A., 2011, "DNABIT Compress Genome compression algorithm", Bioinformation, 5(8), 350-360
13. Satyanvesh, D., Balleda, K., Padyana, A., et al., 2012, "GenCodex - A Novel Algorithm for Compressing DNA sequences on Multi-cores and GPUs", Proc. IEEE, 19th International Conf. on High Performance Computing (HiPC), Pune, India, No 37.
14. Prasad, V. H., and Kumar, P. V., 2012, "A New Revised DNA Cramp Tool Based Approach of Chopping DNA Repetitive and Non-Repetitive Genome Sequences", International Journal of Computer Science Issues (IJCSI), 9(6), 448-454.
15. Prasad, V. H., 2013, "A new revisited compression technique through innovation partition group binary compression: a novel approach", International Journal of Computer Engineering & Technology (IJCET), 4(2), 94-101.