International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Single Bit DNA Squeezer (SBDNAS) : An Enhancement of BDNAS Algorithm

Alam Jahaan[1], Dr. T. N. Ravi[2]
[1]Research Scholar in Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India
[2]Assistant Professor in Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India

ABSTRACT

DNA Sequences are repetitive and non - repetitive concatenation of the four nucleotides namely, Adenine, Guanine, Thymine and Cytosine which are bases in DNA molecules. These nucleotides form a double stranded helical structure with each nucleotide in one strand joined to its compliment on the other with hydrogen bonds using base pairing rules. Compression in general focusses on cost effectiveness, which may be achieved by improving Space complexity, Time complexity and speed of transmission. This article presents a DNA sequence compression algorithm SBDNAS, an enhancement of BDNAS algorithm, that compresses DNA sequences by replacing the nucleotides by single bits 0 or 1. SBDNAS comprises of two Phases, Pre-processing Phase and Coding Phase. Compression ratio has been compared for the efficiency for worst case, best case and average case.

Keywords : DNA Sequence, Bit Based Method, Space Complexity, Time Complexity, DNA Compression

I. INTRODUCTION

DNA databases otherwise called DNA Data Banks are used to store DNA sequences [1]. DNA Databases are similar to text Databases, which are used to store and manipulate relevant information as files, these databases are growing exponentially posing a major task for storage, search, retrieval and transmission of data, hence paving the way for superior compression techniques and tools. Recent advances in Genetic research, Bio-medical sciences and Forensic sciences has directed the way for *DNA* compression [2].

Compression in General

Compression deals with reducing the size of the file, to make storage and transfer easy, quick, efficient and cost effective. Compression techniques may be divided into two namely, Reversible (Lossless) where the decompressed file is not in its original form and Irreversible (Lossy) techniques in which the file is exactly same as the original file after decompression [3]. Numerous compression techniques have been designed and developed pertaining to the requirements in various areas such as Image Compression, Text compression, Audio Compressions and Video Compression.

DNA Compression

Deoxyribonucleic acid (DNA) [4] is a molecule that carries the genetic instructions stored in each cell of a living organism. It is used in the growth, development and functioning of all known living organisms. DNA is essential for all known forms of life. The properties noted in most of the sequences that forms the main criteria for various compression techniques are the oft-repeated substrings, repeated palindromes and repeated reverse compliments [5].

DNA is a double helix made up of four nucleotides: Adenine (A), Thymine (T), Cytosine (C), Guanine (G). These nitrogenous bases of the two separate polynucleotide strands are bound together, according to base pairing rules (A with T and C with G), with hydrogen bonds to make double-stranded *DNA*. Depending on base pairing, only one strand of the helical structure needs to be stored as a sequence while the other pair can be regenerated [6].

General compression techniques do not perform well with biological sequences since these sequences consists of combinations and repetitions of just four nucleotides. Recently two - bit coding methods have become prevalent where the four nucleotide bases {A, T, C, G} in DNA sequences are assigned values of two bits each 00, 01, 10 and 11 respectively before the encoding process. In Most two-bit based algorithms each stage is similar in the bit pre-processing stage where the bits are assigned unique two bits each but different in the coding stage [7].

II. PERFORMANCE ANALYSIS

There are a number of factors that must be balanced when deciding which compression algorithm to use.

- **Complexity Analysis**: While analyzing an algorithm, we mostly consider time complexity and space complexity[8].
  - **Time complexity** of an algorithm quantifies the amount of time taken by an algorithm to run as a function of the length of the input. The time consumed in performing a given task (computation time or response time) is considered. In the Proposed algorithm *SBDNAS* it implicates:
    - Compression time: The time taken to compress the sequence using both phases calculated in milliseconds.
    - Decompression time is time taken to reproduce the compressed file to its original form calculated in milliseconds.
  - **Space complexity** of an algorithm quantifies the amount of space or memory taken by an algorithm to run as a function of the length of the input, space refers to the data storage consumed in performing a given task such as RAM, HDD.
    In the proposed algorithm, during Phase II, the source file size gets reduced to one bit for each nucleotide. In Phase II, the file size reduces further by implementing VCT method.
- **Compression Ratio :** The ratio between the sizes of compressed to original file[9].

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

III. ALGORITHMS ELUCIDATED

- ***BDNAS (Bit DNA Squeezer)*** Algorithm: An approach ***Bit DNA Squeezer*** (*BDNAS*) for compressing standard *DNA* Datasets, which is an improvement over two bit-based technique. It focuses on reducing the file size considerably by assigning single bit per base. The compression process involves two steps [10]

Step1: Depending on the order of the frequency of each nucleotide in the dataset, each nucleotide is assigned values as *first*, *second*, *third* & *fourth* accordingly.

Step2: The nucleotide with highest frequency (*first*) is assigned the value 0 throughout the file.

The nucleotide with second highest frequency (*second*) is assigned the value 1. For the nucleotide with third frequency (*third*), its position is recorded in a position map then it is assigned the value 0. Finally, the nucleotide with least frequency (*fourth*) its position is also recorded in the position map and its assigned the value 1 until end of file is reached.

The Decompression process uses the position map and the compressed file as input, and converts all the 0's in the compressed file according to the positions recorded in the position map to its corresponding *third* nucleotide and converts all the 1's in the compressed file according to the positions recorded in the position map to its corresponding *fourth* nucleotide. 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 Single Bit DNA Squeezer *SBDNAS* is proposed to enhance the *BDNAS* technique by using it as the pre-processing Phase and then compresses the file further using in Phase II which is the actual Coding Phase using the *Vector Coding Technique (VCT*). It which searches for a chosen vector and encodes only the exact repeats.
  - Methodology: The compression process involves two Phases: *Phase I* for pre- processing and *Phase II* for 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. The dataset is used as input and the compressed file along with the position map is the output.

Phase II: **Coding Phase** implements the *Vector Coding Technique (VCT)* using a row vector. The output bit file from the previous Phase is considered as the input file for phaseII. A row vector is selected and is searched in the input file. Then for each occurrence a unique character is assigned to it. This process of selecting – searching and assigning a unique character is repeated a number of times until E.O.F is reached.

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.

- **SBDNAS ENCODING ALGORITHM:**
  Input: Input file (INSeq) Containing A, T, G, C
  Output: Encoded file (OUTSeq), PosMap

Procedure Encode:

Begin

1. Begin compression for PhaseI (Pre-Processing Phase) using *BDNAS* algorithm
2. Calculate the frequency of each nucleotide in given input file INSeq.
3. Assign names as first, second, third and fourth respectively to the nucleotides depending on their frequency. Highest count is assigned first, second highest occurrence as second so on and so forth.
4. Start from beginning of file and assign 0 to all the first frequency and 1 to all the second frequency.
5. Again search for nucleotide with third frequency and note down its position in the position map (PosMap) then replace it with 0.
6. Similarly search for the nucleotide with fourth frequency and note down its position in the position map (PosMap) then replace it with 1.
7. Compressed file and PosMap are created as output of the source file.
8. Consider compressed file and start with second Coding Phase using Vector Coding Technique.
9. Select a single row vector with more than 8 bits.
10. Search for occurrence of the vector from BOF.
11. Replace each occurrence with a unique character ‘X’ until EOF is reached.
12. The output is the compressed file called OUTSeq is obtained.

End

- ***SBDNAS* DECODING ALGORITHM**
  Input: Input file (OUTSeq), PosMap
  Output: Decoded file (DECSeq)

Procedure Decode:

Begin

1. Decompression of Phase II (Coding Phase) is carried out first
2. Search for the unique character ‘X’
3. Replace it with the Vector until EOF is reached
4. The decompression of the Phase I (Pre-processing Phase) is carried out by reading the PosMap
5. For each entry equivalent to 0 in PosMap, go to corresponding position in compressed file and change it to 3rd Nucleotide.
6. Similarly change the entry for 1 in PosMap to 4th nucleotide at the same position in compressed file.
7. Consider the compressed file and replace all the remaining 0’s to 1st nucleotide
8. Also replace all 1’s in compressed file to 2nd nucleotide.
9. The file DECSeq which is the original file is obtained.

End

- ***IMPLEMENTATION* OF *SBDNAS***

Consider a DNA sequence which is a concatenation of the four nucleotides A,T,C,G also known as bases.

```
TTGAACGAGAAGAAAACCGTATAAAAAGGAAATGAAAATATCAA
GTACGGTTTTGTAAAGAAAAAAATGACAATTTAGGTAACTTATTT
GTCAACTTTCC
```

Phase I: *BDNAS* Algorithm is applied to the DNA sequence (Ð) above

Step 1: Total Nucleotides (bases) = **100**

Frequency of A (f_A)= 45; Frequency of T (f_T) = 27;
Frequency of G (f_C) =17; Frequency of C (f_G) = 11
Such that, $Ð = \Sigma(f_A + f_T + f_C + f_G) = \mathbf{100}$.
Since $f_A > f_T > f_C > f_G$
Let First = A; Second =T; Third = G; Fourth= C

Step 2: Assigning A=0 and T=1; A partially compressed sequence is obtained.

```
11G00CG0G00G0000CCG10100000GG0001G0000101C00G10CGG
1111G1000G00000001G0C001110GG100C110111G1C00C111CC
```

Storing the position of G in PosMap then Assigning G = 0

Storing the position of C in PosMap then Assigning C = 1

The completely compressed sequence along with the position map is obtained

110001000000000011010100000000010000010110001010011111010000000000100100111000100111011011001111111

PosMap [ ] [ ] = { { 3,7,9, 12 15 19 …};{ 6, 13, 14, 32 38 54 66 … } }

Compression ratio = bits/ bases = 100/100 = 1.00 bpb

***Phase II* of *SBDNAS*: (*Implementation of VCT*)**
Consider row vector RV= 0000000000

Searching for RV and replacing it with 'x'; A new sequence is obtained:

110001x110101x10000010110001010011110

1x0100100111000100111011011001111111

The compressed sequence is of size = 94 bits
(70bits +3*8 bits)

Compression ratio = Total bits / No. of bases
= 94/100=0.94 bits per base

IV. PERFORMANCE ANALYSIS

Separate analysis for the SBDNAS algorithm is performed for the best case, average case and worst case [11].

Best Case: Best case efficiency of SBDNAS algorithm has maximum repeats of vector. The Best-case efficiency is proved here since its compression Ratio = 0.96, which is the best among other cases.

Average Case: The Average case efficiency of this algorithm 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 repetitive vectors. In this algorithm, the worst-case compression ratio is the highest

Table 1 : Comparison of compression ratio for 3 cases.

Efficiency Cases	Best Case	Average Case	Worst case
No. bases Original File	100	100	100
No. of bits Compressed File	95	98	100
No. of repetitions	5	2	0
Compression Ratio bits/base	80/100 = 0.80	96/100 = 0.98	100/100 = 1.00

The computation for compression ratios for three different case has been tabulated below and the results are depicted in the graph. The compression ratio for the worst case is 1.0 bits per base and for best case is 0.8 bits per base.

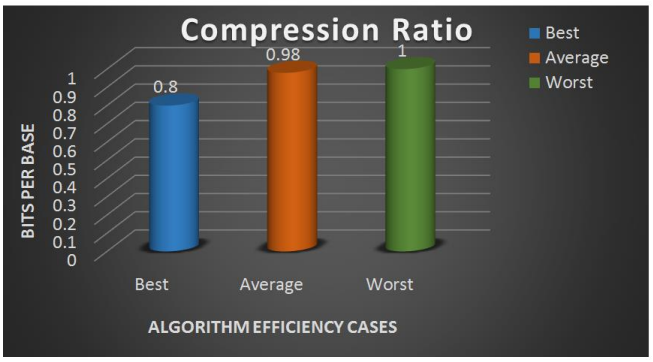

Chart1: Compression ratio for 3 different cases

V. CONCLUSION

General compression techniques do not work well with biological sequences. Recently two-bit based coding methods have become prevalent where the four nucleotide bases {A, T, C, G} in DNA sequences are assigned values of two bits each 00, 01, 10 and 11 respectively. A method BDNAS prevails over the 2 bit based techniques by assigning single bits for each base instead of 2 bits per base. An enhancement over the BDNAS algorithm is proposed called SBDNAS, which gives comparatively better compression ratio. Analysis of the algorithm is performed for the best case, average case and worst case. Complexity analysis and compressed ratio are explained. Moreover, compression ratio for selected DNA sequences is analysed and tabulated for the three efficiency cases (best, worst and average cases).

VI. FUTURE SCOPE

The Coding phase may be extended to search and replace palindromes and reverse compliments along with exact repeats of the selected vector in DNA sequences. The algorithm may be tested using enormous DNA Datasets rather than small sequences and a study on how it copes with the space and time complexity of DNA datasets may be computed and analysed.

VII. REFERENCES

[1]. https://en.wikipedia.org/wiki/DNA_database

[2]. Alam Jahaan ,Dr T.N. Ravi, Dr. S. Panneer Arokiaraj, "A Comparative Study and Survey on Existing DNA Compression Techniques", IJARCS, p-ISSN: 0976-5697, volume 8, No.3, March-April 2017

[3]. Alam Jahaan ,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.

[4]. https://ghr.nlm.nih.gov/primer/basics/dna

[5]. 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

[6]. https://en.wikipedia.org/wiki/Introduction_to_genetics

[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]. https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/

[9]. S.R. 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

[10]. 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

[11]. Alam Jahaan ,Dr T.N. Ravi,"A Relative Study On Existing Two Bit-Based DNA Compression Techniques With Bit Dna Squeezer (BDNAS)" International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307