International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

Bit DNA Squeezer (BDNAS) : A Unique Technique for Dna Compression

Alam Jahaan[1], Dr. T. N. Ravi[2], Dr. S. Panneer Arokiaraj[3]

[1]Research Scholar in Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India
[2]Assistant Professor, Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India
[3]Associate Professor, Computer Science, PERIYAR EVR College, Trichy, Tamil Nadu, India

ABSTRACT

Data compression plays a vital role in analyzing, decreasing and transferring DNA sequences, hence escalating the creation of DNA compression techniques in order to store and transfer tremendous amount of genomic data. A fresh flow of interest in the development of novel algorithms and tools for storing and managing genomic sequences highlights the increasing demand for efficient methods for DNA compression. This is ultimately, the motivating force behind the development of high-performance compression tools, which are designed specifically for genomic data. Most of the earlier DNA compression methods were essentially dictionary-based methods or statistical methods. Recently 2-bit coding methods have become prominent where the 4-nucleotide bases {A, C, G, T} in DNA sequences are assigned values 00, 01, 10 and 11 respectively. In this paper an attempt has been made to present an approach where a single bit (0 or 1) is assigned for each nucleotide base {A,C,G,T} in a DNA sequence depending on the count of each nucleotide. This proposed technique compresses large bytes of DNA sequences with the average compression ratio of approximately 1.4 bits per base.

Keywords : DNA Sequences, Bit-Based Model, Position Map, Compression, Decompression, Datasets

I. INTRODUCTION

DNA sequences have become crucial for any biological and clinical research, other research branches working with DNA sequencing are numerous everyday fields such as diagnostic, biotechnology, forensic biology, genealogy, bioinformatics, genetics, genomics, genetic engineering and criminology.

As these sequences are enormous in size, storing and sharing becomes costly and challenging, hence compression becomes mandatory. Lossless Compression Methods focus on reproducing the original data on decompression; these are suitable for DNA sequence compression as the DNA datasets cannot afford to lose any part of their data [1].

A very simple, fast, flexible and effective DNA compression algorithm named Bit DNA Squeezer (BDNAS) is proposed to compress DNA sequences which may be repetitive or non-repetitive in nature.

Organization of the paper:

The proposed algorithm may be used to compress large bytes of DNA sequences with the average compression ratio of 1.4 bits per base. In this paper Section 2 lays out related work associated to 2-bit based compression models. Section 3 provides an overview of the proposed algorithm - its elucidation, algorithms for compression and decompression. To quantify the efficiency of a given compression run, several quality measures such as compression factor, compression rate, compression time and decompression time are calculated. Section 4 describes the implementation of the proposed algorithm highlighting the hardware and software configurations and enumerates the standard data sets used. Section 5 tabulates the metrics such as compression ratio, compression and decompression time and compression factor for 9 standard DNA Sequences which have been computed, results are presented using graphs. Section 6 concludes and recommends future work.

II. RELATED WORK

DNA sequences are naturally represented as repetitive or non-repetitive strings of the four nucleotides Adenine(A), Cytosine(C), Guanine(G), & Thymine(T). However, a collection of general-purpose text compressors, such as gzip or bzip2 result in representations that require more than 2 bits per character since they fail to identify and eliminate DNA-specific redundancy. Ultimately, tailored DNA compression methods are needed and several methods have been proposed to-date [2].

Most of the earlier DNA compression methods are concerned with reducing the redundancy within a given DNA string and were essentially dictionary-based methods or statistical methods.

Two-Bit Based methods:

These algorithms implement a bit-preprocessing process by assigning 4 unique two bits (A=00, G=01, C=10, T=11) to each nucleotide before the encoding process. A DNA sequence is represented in minor fragments or sections with each being four 8 bit-characters long before the encoding process to compress both repetitive and non-repetitive DNA sequences. The following algorithms are of two phases with each phase being similar in the bit-preprocessing stage but different in the coding stage [3].

- The GENBIT Compress tool [4] was proposed by Rajeswari and Apparao for compressing DNA sequences based on a novel concept of assigning binary bits to the nucleotides.
- HUFFBIT compress was proposed by Rajeswari et al., for DNA sequences. Here a bit-preprocessing stage takes place before the encoding [5].
- Rajeswari & Apparao later on introduced the DNABIT Compress tool [6] which is a unique concept introduced in DNA compression that assigns binary bits "in the bit-preprocessing stage" to exact and reverse repeats fragments of DNA sequences.

1.1.1

1.1.2

- GenCodex introduced by Satyanvesh et al., a two-phased algorithm that produces a better compression ratio at a high throughput, it uses graphical processing units and multi-cores. In the first phase, bit-preprocessing is implemented; the next phase represents the fragments using one or two bytes [2].
- In the DNACRAMP tool by Prasad & Kumar, a DNA sequence is taken for bit-preprocessing. It performs the encoding and decoding process with the help of a two-stage index bounded linear array data structure using basic procedural language [7].
- Prasad then introduced PGBC "Partitioned group binary compression" an improvement over previous algorithms. Here the encoding process starts after bit-preprocessing where every 6 part is grouped as a partition, with two sub-partitions [8].

III. PROPOSED TECHNIQUE

3.1 Elucidation

DNA sequences are mere combinations of the 4 nucleotide bases which are both repetitive and non-repetitive. The proposed algorithm works in 2 steps during compression:

Step1: It records the total count of each nucleotide in the dataset as 1st, 2nd, 3rd & 4th occurrences.

Step2: It assigns 0 to the highest occurring nucleotide, then it assigns 1 to the 2nd highest nucleotide, the 3rd nucleotide is assigned 0 similar to the highest occurrence but its position is noted down in a position map called PosMap. Similarly the 4th nucleotide is assigned 1 and its position is recorded too in the position map (PosMap).

The Decompression process considers 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 3rd nucleotide and converts all the 1's in the compressed file according to the positions recorded in the position map to its corresponding 4th nucleotide. Finally, the remaining 0's are converted to the 1st nucleotide and 1's to the 2nd nucleotide respectively.

3.2 Algorithm

3.2.1 Compression

BDNAS ENCODING ALGORITHM:

Input: Input DataSet Containing 9 DNA sequences with repetitive and non- repetitive combinations of A,T,G,C.

Output: Compressed file and PosMap.

Procedure Encode:
Begin
1. Count all the nucleotides in the Dataset (A,C,G,T).
2. Assign names as first, second, third and fourth respectively to the nucleotides depending on their count. Highest count is assigned first, second highest occurrence as second so on and so forth.
3. Start from beginning of file and assign 0 to all the first occurrences and 1 to all the second occurrences.
4. Again search for occurrence of third nucleotide and note down its position in the position map (PosMap) then replace it with 0.
5. Similarly search for occurrence of fourth nucleotide and note down its position in the position map (PosMap) then replace it with 1.
6. Compressed file and PosMap are created.
   End

3.2.2 Decompression

BDNAS DECODING ALGORITHM:

Input : Input Compressed file and PosMap.
Output : Original (Decompressed) file

Procedure Decode:
Begin
1. For each entry equivalent to 0 in PosMap, go to corresponding position in compressed file and change it to 3rd Nucleotide.
2. Similarly change the entry for 1 in PosMap to 4th nucleotide at the same position in compressed file.
3. Consider the compressed file and replace all the remaining 0's to 1st nucleotide
4. Also replace all 1's in compressed file to 2nd nucleotide.
5. The original file is obtained.

End.

3.3 QUALITY METRICS

Compression Performances may be measured by comparing the quality of the original file with the compressed file using quality metrics as discussed below [9]:

3.3.1 COMPRESSION FACTOR: The ratio between the original file size to the compressed file size

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

3.3.2 COMPRESSION RATIO: The ratio between the compressed file size to the original file size.

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

3.3.3 SAVING PERCENTAGE: The percentage of the size reduction of the DNA sequence, after compression is calculated as given below:

$$\text{Saving Percentage} = \frac{\text{Original File Size} - \text{Compressed file size}}{\text{Original File Size}} \%$$

3.3.4 COMPRESSION TIME: The time taken by the proposed algorithm to compress each DNA sequence is calculated in milliseconds.

3.3.5 DECOMPRESSION TIME: The time taken by the proposed algorithm to decompress and retrieve the original DNA sequence. It is calculated in milliseconds too [10].

IV. IMPLEMENTATION

Implementation of the proposed algorithm BDNAS is carried out by compressing a standard set of DNA sequences. A test prototype has been implemented to evaluate the capability of the algorithm. The code has been written, compiled and run in VC++.

Tests have been carried out on a system with following configuration:
Processor : intel® Core™ (i5-4210U) CPU@ 1.70 GigaHz 2.40GigaHz
Ram : 4.00 GB
OS : Windows 8.1 Single Language.

A dataset of DNA sequences usually used in DNA compression studies has been tested using the proposed algorithm BDNAS and the results have been tabulated. Most DNA compression algorithms use the standard benchmark datasets [11]. The dataset used in our research includes 9 standard sequences from a variety of sources; such as one chloroplast genome CHMPXX; Four Human Genes - HUMDYSTROP, HUMHBB, HUMHDABCD and HUMHPRTB; One mitochondria genome MPOMTCG; and 2 Virus genome HEHCMVCG and VACCG [3].

V. RESULTS AND DISCUSSION

Quality Measures such as Compressed Size, Compression Ratio & factor, Compression & Decompression Time for the above stated 9 standard DNA sequences have been computed using the proposed BDNAS algorithm and experimental outcomes are tabulated in Table1. Chart1 below illustrates the relationship between original file size and compressed file size of the 9 standard DNA sequences considered for our research. Chart2 depicts the compression time and decompression time of each sequence from the chosen standard data set.

Table 1: Experimental results for selected 9 DNA sequences

DATASET	SIZE(B)	COMPRES-SED SIZE	COMPRESS-ION RATIO	COMPRESSI-ON FACTOR	COMPRES-SION TIME (mS)	DECOMPRES-SION TIME (mS)
Chmpxx-new	121024	22117	1.462	0.684	18	14
Humdystrop-new	38770	7114	1.468	0.681	6	4
Humghcsc-new	66495	12085	1.454	0.688	9	7
Humhbb-new	73308	13488	1.472	0.679	12	9
Humhdabcd-new	58864	10624	1.444	0.692	12	9
Humhprtb-new	56737	10319	1.455	0.687	9	7
Mpomtcg-new	186609	34126	1.463	0.684	27	21
Vaccg-new	191737	35207	1.469	0.681	29	22
Hehcmvcg-new	229354	42172	1.471	0.680	33	25

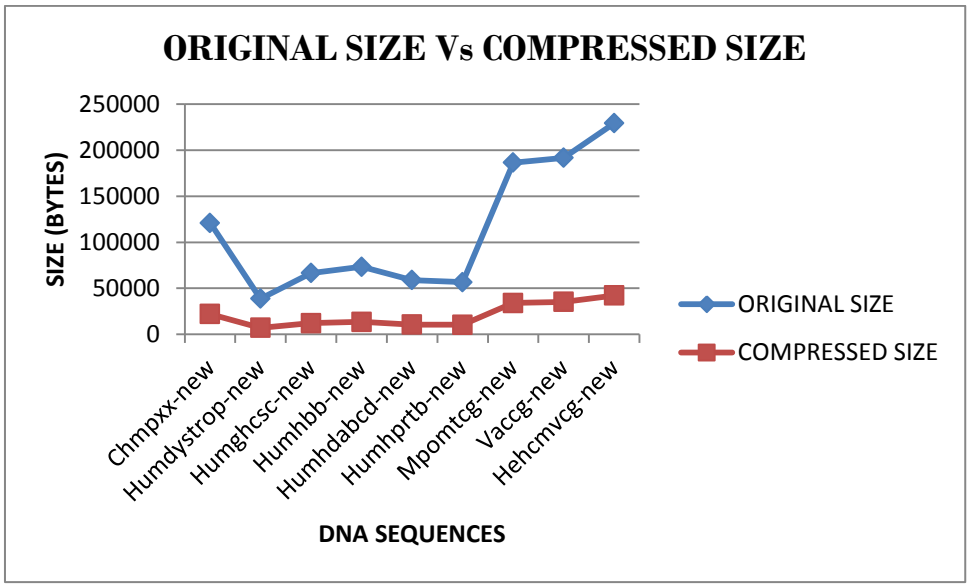

Chart 1 : Comparison of compressed size to the original size of the DNA sequences.

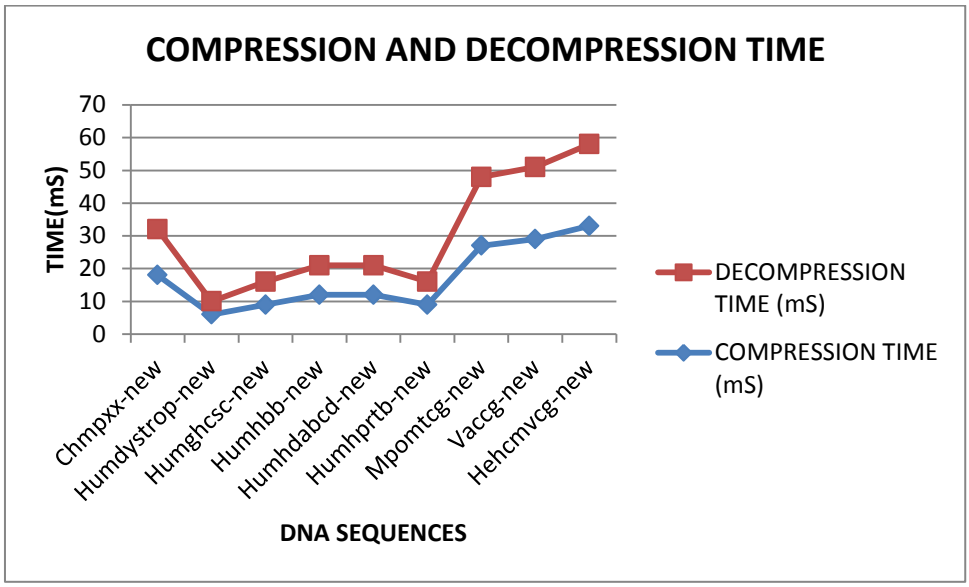

Chart 2 : Compression Time & Decompression Time

VI. CONCLUSION & FUTURE WORK

An attempt has been made to present an approach on compression of DNA sequences analogous to bit based methods. A novel algorithm has been illustrated which compresses a DNA sequence with the average compression ratio of 1.4 bits per base. Unlike existing general DNA compression techniques, Bit-based compression is used for compressing repetitive and non-repetitive DNA sequences without the use of complex dynamic programming. Our proposed algorithm is simple, fast and flexible and is mainly centered on assigning a single bit to each nucleotide unlike a 2-bit based compression technique where 2 unique bits are assigned to each nucleotide. Metrics such as compression ratio, compression time, decompression time and compression factor for 9 standard DNA Sequences have been computed and tabulated.

Our algorithm may be enhanced by using it as a pre-processing step as used in any general 2-bit based compression technique. The proposed Bit DNA Squeezer (BDNAS) algorithm is very simple and flexible hence it may be useful in several researches where large sequence analysis and comparisons may be conducted.

VII. REFERENCES

[1]. 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, volume8,No.3, March-April2017. Online:www.ijarcs.info.

[2]. 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

[3]. 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

[4]. 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), 181-191

[5]. Afify, H., Islam, M., Abdel-Wahed, M., et al., 2010, Genomic Sequences Differential Compression Model, Proc., 27th National Radio Science Conf., Egypt

[6]. Rajeswari, P. R., and Apparao, A., 2011, DNABIT Compress - Genome compression algorithm, Bioinformation, 5(8), 350-360

[7]. 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.

[8]. 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.

[9]. 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.

[10]. 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

[11]. S. Grumbach and F. Tahi, "Compression of DNA Sequences," in Proc. of the Data Compression Conf., (DCC '93), 1993, 340-350.