International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET218529

Bit Reduction based Compression Algorithm for DNA Sequences

Rosario Gilmary[1], Murugesan G[2]

[1]Department of Computer Science and Engineering, Pondicherry Engineering College, India
[2]Department of Computer Science and Engineering, St. Joseph's College of Engineering, India

Article Info
Volume 8, Issue 5
Page Number : 270-277

Publication Issue :
September-October-2021

Article History
Accepted : 10 Oct 2021
Published: 30 Oct 2021

ABSTRACT

Deoxyribonucleic acid called DNA is the smallest fundamental unit that bears the genetic instructions of a living organism. It is used in the up growth and functioning of all known living organisms. Current DNA sequencing equipment creates extensive heaps of genomic data. The Nucleotide databases like GenBank, size getting 2 to 3 times larger annually. The increase in genomic data outstrips the increase in storage capacity. Massive amount of genomic data needs an effectual depository, quick transposal and preferable performance. To reduce storage of abundant data and data storage expense, compression algorithms were used. Typical compression approaches lose status while compressing these sequences. However, novel compression algorithms have been introduced for better compression ratio. The performance is correlated in terms of compression ratio; ratio of the capacity of compressed file and compression/decompression time; time taken to compress/decompress the sequence. In the proposed work, the input DNA sequence is compressed by reconstructing the sequence into varied formats. Here the input DNA sequence is subjected to bit reduction. The binary output is converted to hexadecimal format followed by encoding. Thus, the compression ratio of the biological sequence is improved.

Keywords : Data compression, lossy and lossless compression, DNA, bases,bit reduction, hexa decimal format, variable length code, huffman codes

I. INTRODUCTION

Presently, the enormous amount of DNA is being sequenced and is stored. They are used in varied research purposes in Bioinformatics discipline. Storing the generated genomic data is a challenging task. These data are stored in online repositories. Expense of storing the genomic data increases with the increase in the amount of DNA being sequenced. In order to overcome this issue, productive and efficient compression algorithm must be introduced to compress the biological sequences.

Deoxyribonucleic acid DNA, which bears the genes and other nucleotides dwell in 23 pairs of chromosomes, whole of 46. The two strands are called polynucleotides. These biopolymer strands coil around each other to form a double helix structure. DNA is a combination of four bases, bound in pairs. The four bases are Adenine (A), Thymine (T),

Cytosine (C), Guanine (G). The genome is made of three billion bases in a decisive order. The bases of two distinct strands are linked by hydrogen bonds. The bases are linked to each other by covalent bonds. DNA is present in every living cell apart from RBC.

The research laboratories abide their data to the databases such as GenBank, the Gene Expression Omnibus etc. By October 2017 statistics, GenBank has 24494705468 bases and 203953682 sequences.

In this work, the various existing compression mechanisms that are particular for compressing the DNA sequences are analyzed followed by introduction of contemporary and persuasive compression algorithm which provides better compression ratio. So that the memory size of the storage is reduced and speed of transmission is improved. In the proposed work, the bits of the input DNA sequence are reduced. The binary format of the DNA sequence is converted to hexadecimal format followed by encoding using Huffman codes.

This paper is systematized as follows. Section 2 involves the survey of existing works which are brought in to compress the DNA sequences and motivation of the present work. Section 3 expresses the proposed work. Section 4 exemplifies the achieved experiments. It also describes the comparison of existing algorithms and proposed algorithm. This is followed by the conclusion in section 5.

II. RELATED WORK

DNA sequences are capable to incorporate repeated substrings between them. The immeasurable genomic data are stored in nucleotide databases. In order to maintain such databases, number of compression algorithms has been designed. Nevertheless, nearly all the existing algorithms are ill-suited. Data compression algorithms can be classified as loss algorithms and lossless algorithms. In loss compression algorithms, the original input is not recovered fully during decompression. A part of data is lost for all time. Whereas, in the lossless compression algorithm, the actual data is retrieved without loss when the file is being decompressed. Most of the DNA compression algorithms are lossless compression algorithms because losing a single base will misdirect the entire sequence.

DNA sequences can be compressed by two modes, namely horizontal and vertical mode. Vertical modes include compression using file formats. Biological sequences can be compressed by considering only the substrings of the entire genome. This procedure falls under horizontal mode wherein the substrings are made as reference.

GenCompress[6] is a substitution based lossless compression technique by searching the approximate repeats from the DNA sequence. This procedure was introduced specially for genomic sequences. GenCompress seeks the average repeats present in the sequence. Here, an ideal prefix is determined followed by encoding. It also explains the amplitude of similarity or relevance between two DNA sequences.

Biocompress-2[7] is a combination of statistical and substitutional procedure. It was specially designed to compress biological sequences without any loss in original data. Here, the regularities present in the sequence is discovered. One of the regularity considered is existence of palindromes. It determines the repeats and non repeats present in the DNA sequences and encodes them.

DNACompress program[5] is persuasive, faster and has better running time when compared with the previous compression algorithms. It applies a software called Pattern Hunter [12] to determine the typical average repeats followed by encoding.

Statistical compression algorithm[4] is designed specifically to compress the genomic sequences. It

considers the repeats present in the sequence as well as the statistical properties of the sequence. The algorithm anticipates the next symbol to be encoded. Arithmetic coding is used to encode the symbols.

CDNA algorithm[10] is a DNA compression technique which is pure statistical and considers the entropy estimates. Each of the symbol to occur is anticipated by considering the average partial matches. Each match is done between subsequences of the genome that have low hamming distance.

NML[11] is called Normalized Maximum Likelihood. It was introduced to compress the DNA sequences by selecting the models. NML uses the Minimum Description Length (MDL) principle. In this procedure, the input data is recognized as codes. These codes are then compressed by the model selected. The model that provides data with least description length is chosen from other candidate models.

Biological sequence compression algorithm[13] uses the distinctive structure of the genomic sequences. The two major features considered here are the average repeats and palindromes and it is done by dynamic and hash programming. This approach provides higher compression ratio when compared with canonical compression procedures.

DNAPack[3] is a compression algorithm for DNA sequences that uses dynamic programming rather than greedy approach. The procedure is less expensive and yields better compression ratio. First, the repeats, complementary palindromes and non-repeats of the substrings of the genome is determined. The repeats and complementary palindromes uses hamming distance whereas the non-repeat parts uses arithmetic 2 compression or Context Tree Weighting.

GenBitCompress[17] is a compression Tool for compressing the genomic sequences. A new principle applied here is allocating binary bits for parts of DNA sequences. This approach differs from other approach by considering only the exact repeats and encoding them rather than considering the approximate repeats.

Differential compression algorithm[1] is done by considering the likeliness of the genetic sequence repository. Every sequence is not stored separately but a storage is created only for particular data. The data encloses cited sequences, differences and their locations.

DNABIT compress[16] involves removal of redundancy from the genomic sequences so that storage is made competent. Here, both the repetitive and non repetitive regions are compressed by allocating binary bits for smaller fragments.

GenCodex[18] was introduced to compress the genomic sequences present in multi cores and GPUs. The prime target of this approach is produce prominent throughput. GenCodex yields a speed up of 11 , 23 on multi cores and GPUs , respectively. It is better than GenBit and DNABit. It produce a compression ratio of 0.017 bpb and 2.25 bpb for best and worst case, respectively.

DNACRAMP tool[15] is a technique proposed for compressing DNA sequences with or without duplicates. Here, the DNA sequences are encoded in bits. The sequence is partitioned into n/4 sections. The quadrupled sections are partitioned into sub partitions followed by assignment of header and trailer. The terminals are grouped to form a cluster. DNACRAMP does not use dynamic programming and encodes each base by 1.19 bits.

Biocompress[8] is a lossless compression algorithms for biological sequences. It is based on regularities which determines and analyze the duplicates of substring that occur in the prior . It is followed by encoding with repeat length and position of prior occurrence.

Seed based compression technique[14] was designed to compress DNA sequences that utilize the substitution procedure. Initially, the repeat structure

present in the DNA sequences are determined by forming a offline dictionary. The dictionary possess the knowledge of duplicates and mismatches present in the sequence. This technique considers only the promising mismatches.

High throughput compression[19] classifies and provides an idea of existing compression mechanisms designed particularly for biological sequences. This paper will also provide the achievements of those techniques.

Referential compression algorithm[9] introduces an innovative procedure to compress the genomic sequences by references considered. Here, set of input sequences are chosen for which reference is determined. A reference is combination of value and key.

DNA sequence compression algorithm[2] uses Extended -ASCII depiction. Here, the DNA sequences considered are represented by extended ASCII codes . The processed sequences are encoded using Run length procedure.

III.PROPOSED WORK

Proposed work represents the bit reduction based compression of given DNA sequence. Here, the DNA sequence is reconstructed to varied formats. Later, the sequence of data is encoded. The whole process of the system can be defined in three phases. The DNA sequence is bit reduced and expressed in binary format. The binary form of representation is converted to hexadecimal format. It is followed by encoding using Huffman codes.

Algorithm

Step 1: Start.
Step 2: Read the Input DNA sequence.
Step 3: Assign A=00, T=01, G=10, C=11.
Step 4: Read the first base and give its corresponding bit code.
Step 5: Compare the next base with its prior base. If same, indicate it by '0' else the corresponding binary code.
Step 6: Repeat Step 5 till the end of DNA sequence.
Step 7: Partition the entire string of binary numbers into groups of four bits.
Step 8: Convert four binary digits into one hexadecimal unit.
Step 9: Determine the frequency of each hexadecimal unit.
Step 10: Assign variable-length code to the hexadecimal characters such that the most frequent hex character gets the smallest code and the least frequent hex character gets the largest code.
Step 11: Stop.

The whole process is divided as 3 modules. The modules of the proposed system are bit reduction of DNA sequence, Conversion of Binary format to Hexadecimal format and Encoding by Huffman procedure.

Bit reduction of DNA sequence

This module explains the bit reduction of DNA sequence and representing them in binary format. Binary code uses the digits of 0 and 1 (binary numbers) to represent the DNA bases. Each base or the symbol gets a bit value assignment. The bit string can correspond to the DNA bases.

The four DNA bases are : {A,T,G,C}

Assign: A=00; T=01; G=10; C=11.

Initially, for the first base of the DNA sequence the assigned bit code is given. Then, the next base is compared with its prior base. If they are same, it is represented by '0' else it is represented by its corresponding binary code. Thus, we get a stream of binary output.

Conversion of Binary Format to Hexadecimal Format

This sector describes the conversion of binary values to hexadecimal values. It is useful and effective approach for compressing long binary strings. Here, both bases are powers of 2. Thus, it is a simple procedure than other general conversions. For converting long binary strings, partition the entire string of binary numbers into groups of four bits each. Hexadecimal converts 4 bits into one hexadecimal unit. So, in order to convert the number, first divide the entire bit sequence. For each four digit group, convert the 4 bit binary number to its equivalent hexadecimal value.

In binary to hexadecimal Conversion, the binary values for 0 to 9 will take the same values as their hex values. The binary values from 10 to 15 are represented as characters from A to F.

Example:

Conversion of binary number 10110101 to a hexadecimal number

Divide into groups of 4digits: 1011 0101

Convert each group to hex digit: B 5

D. Encoding by Huffman Codes

This module explains the encoding of hexadecimal string by Huffman codes. It is a lossless data compression algorithm. The procedure is to allocate variable-length codes to input hexadecimal characters. The lengths of the assigned codes are based on the frequencies of the corresponding characters. The character that appears the most gets the smallest code and the character that appears the least gets the largest code. In a variable-length code words may have different length as shown in TABLE I.

TABLE I. VARIABLE LENGTH CODE

HEX-BASES	A	1	9	C
FREQUENCY	7	60	85	20
CODE WORD	111	10	0	110

Given a hexadecimal string and its corresponding variable code as shown in TABLE I, it is simple to encode the hex string just by replacing the hex characters by the code words.

Input : 1A99CA1

Output : 10 111 0 0 110 111 10

E. **Applications**

- It reduces the storage space required by the biological sequence (DNA) in the nucleotide database.
- Processing costs of biological sequences can be economized.
- Transmission costs of genetic data can be diminished.
- Provision of quick access to any record and superior functionality.

IV.RESULT AND COMPARISON

Result Analysis of sample sequence

Here, a sample DNA sequence is considered. The step wise procedure of proposed DNA compression is explained keeping the sample DNA sequence as base. A model DNA fragment with A,T,G and C bases are considered. Each base is given a corresponding bit code. TABLE II shows the Bit reduction of sample DNA sequence. It is followed by restructuring the sequence in hexadecimal format and Huffman encoding.

Example

The four DNA bases are : {A,T,G,C}

Assign: A=00 ; T=01 ; G=10 ; C=11

Sample DNA Sequence (Best case scenario)

GAAATGAAATATAACCAGAATTGAATTAAG
TAATATATAGGGTTTGGTTCCGTTAGAGCT

Bit Reduction

TABLE II. BIT REDUCTION OF BASES

G	A	A	A	T	G	A	A	A	T	A	T	A	...
10	00	0	0	01	10	00	0	0	01	00	01	00	...

Reduced Binary format of sample DNA sequence

10 00 0 0 01 10 00 0 0 01 00 01 00 0 11 0 00 10 00 0 01 0 10 00 0 01 0 00 0 10 01 00 0 01 00 10 00 01 00 10 0 0 01 0 0 10 0 01 0 11 0 10 01 0 00 10 00 10 11 01

Binary Format In Partitions

1000 0001 1000 0001 0001 0001 1000 1000 0010 1000 0010 0001 0010 0001 0010 0001 0010 0001 0010 0010 1101 0010 0010 0010 1101

Hexadecimal Format : The Hexadecimal format for sample DNA sequence is as follows

8 1 8 1 1 8 8 2 8 2 2 2 1 2 1 2 2 D 2 2 2 D

Huffman Codes

TABLE III. HUFFMAN CODE

HEX-BASES	8	1	2	D
FREQUENCY	5	8	10	2
A VARIABLE CODE	111	10	0	110

From TABLE III the corresponding Huffman Codes for the hexadecimal values were obtained.

Huffman Codes of the sample DNA sequence:

000 1101 0110 0011 1000 00 110 010 101 000 1100 11101 1101

Input DNA Sequence = 60 Bases

= 60* 8 Bits

= 480 Bits

Proposed Work = 47 Bits

60 bases = 47 bits

1 base = 47/60

= 0.78 bit per base (best case)

Compression ratio = Uncompressed Volume/ Compressed Volume

= 480/47

= 10.21 units (best case)

IV. Comparison of Results

TABLE IV infers the comparison of compression ratios of various existing techniques followed by compression ratio achieved by the proposed method for the five different input DNA sequences.

TABLE IV. COMPARISON OF COMPRESSION RATIOS

Sequence	Length	GeMNL	Bioc	CTW +L Z	GenC	DNAC	DNAP	XM	Seed Based	Proposed Method
HUMGHCSA	66496	1.00	1.30	1.09	1.09	1.02	1.63	0.98	1.52	1.55
HUMHBB	73308	-	1.88	1.8	1.82	1.78	1.77	1.75	1.73	**1.65**
HUMDYSTROP	38770	1.90	1.92	1.91	1.92	1.91	1.90	1.9	1.86	**1.64**
VACCG	191737	1.76	1.76	1.76	1.76	1.75	1.75	1.67	1.64	1.66
MPOMTCG	186609	1.88	1.93	1.9	1.90	1.89	1.89	1.87	1.76	**1.62**
AVERAGE		1.64	1.76	1.69	1.70	1.67	1.79	1.63	1.70	**1.62**

The bit reduction based compression of DNA sequences was experimentally confirmed on standard DNA sequences expressed in FASTA format. The proposed approach was evaluated on the standard benchmark data. The typical input DNA sequences considered are as follows. HUMGHCSA is a growth hormone present in humans, VACCG is a genome of complete Copenhagen vaccinia virus, MPOMTCG is a complete genome of Marchantiapolymorpha mitochondrial DNA, HUMHBB is present in human beta globin region of chromosome 11 and HUMDYSTROP is present in a dystrophin gene of Homo Sapiens. The proposed algorithm yields prominent compression ratio for three DNA sequences. This compression algorithm grant better compression ratio of 1.65034 bpb, 1.6466 bpb and 1.624664 bpb for the sequences HUMHBB, HUMDYSTROP and MPOMTCG, respectively. On an average for the five sequences considered, the proposed approach contribute the compression ratio of 1.627021bpb. From the comparison table, it is deduced that Bit reduction based compression algorithm is better and efficient than existing compression techniques.

V. CONCLUSION

The compression of biological sequences particularly for DNA sequences is designed with significant improvement in the compression ratio. In this paper, the existing work related to the compression of biological sequences are discussed and a new algorithm to compress the substantial genetic code by bit reduction and encoding technique is designed. The proposed algorithm yields a better compression ratio when compared to the existing compression mechanisms.

VI. REFERENCES

[1]. Afify, H., Islam, M., Abdel-Wahed, M., et al., 2010, Genomic Sequences Differential Compression Model, Proceeding of 27th National Radio Science Conferenec, Egypt.

[2]. Bacem Saada, Jing Zhang, " DNA Sequences Compression Algorithm Based on Extended-ASCII Representation in Proceedings of the world congress on engineering and computer science 2015 Vol II WCECS 2015, October 21-23, 2015, San Francisco, USA.

[3]. Behzadi B and Le Fessant F, "DNA compression challege revisited: a dynamic programming approach",in Proceedings of the Annual Symposium on Combinatorial Pattern Matching, pp. 90-200, Springer, Berlin,Germany,2005.

[4]. Cao M D, Dix T I, Allison L, and Mears C, "A simple statistical algorithm for biological sequence compression," in Proceedings of the Data Compression Conference (DCC'07), pp. 43-52, IEEE, Snowbird, Utah, USA, March 2007.

[5]. Chen X, Li M, Ma B, and Tromp J, "DNACompress: fast and effective DNA sequence compression,"Bioinformatics, vol. 18, no. 12, pp. 1696-1698, 2002.

[6]. Chen X, Kwong S, and Li M, "Compression algorithm for DNA sequences and its applications

in genome comparison," in Proceedings of the 4th Annual International Conference on Computation Molecular Biology (RECOMB'00), p. 107, ACM, Tokyo, Japan, April 2000.

[7]. Grumbach S and Tahi F, "A new challenge for compression algorithms: genetic sequences", Information Processing and Management, vol. 30, no.6, pp. 875-886,1994.

[8]. Grumbach S and Tahi F," Compression of DNA sequences", in Proceedings of the IEEE Symposium on Data Compression, pp. 340- 3550, Snowbird, Utah, USA, 1993

[9]. Kanika Mehta and Satya Prakash Ghrera," DNA compression using referential compression algorithm",in Contemporary Computing (IC3), 2015 Eighth International Conference.

[10]. Loewenstern D and Yianilos P N, "Significantly lower entropy estimates for natural DNA sequences,"Journal of Computational Biology, vol. 6, np. 1,pp. 125-142, 1999.

[11]. Myung J I, Navarro D J, and Pitt M A, "Model selection by normalized maximum likelihood", Journal of Mathematical Psychology, vol.50, no. 2, pp. 167-179,2006

[12]. Ma B, Tromp J, and Li M, "PatternHunter: fast and more sensitive homology search", Bioinformatics, vol. 18, no. 3, pp. 440-445, 2002.

[13]. Matsumoto T, Sadakane K, and Imai H, "Biological sequence compression algorithms", Genome Informatics, vol. , pp. 43-52, 2000.

[14]. Pamela Vinitha Eric, Gopakumar Gopalakrishnan and Muralikrishnan Karunakaran, " An Optimal Seed Based Compression Algorithm for DNA Sequences", Advances in Bioinformatics, vol 2016 (2016), Article ID 3528406, 7 pages.

[15]. 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.

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

[17]. 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.

[18]. Satyanvesh, D., Balleda, K., Padyana, A., 2012, GenCodex- A Novel Algorithm for Compressing DNA seuences on Multi-cores and GPUs, Proc. IEEE, 19th International Conf. on High Performance Computing (HiPC), Pune, India, No 37.

[19]. Zhu Z, Zhang Y, Ji Z, He S, Yang X," High - throughput DNA sequence data compression",in Briefings in bioinformatics. 2015 Jan; 16 (1)

Cite this article as :

Rosario Gilmary, Murugesan G, "Bit Reduction based Compression Algorithm for DNA Sequences", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 8 Issue 5, pp. 270-277, September-October 2021. Available at doi : https://doi.org/10.32628/IJSRSET218529
Journal URL : https://ijsrset.com/IJSRSET218529