European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 07, Issue 10, 2020

# Codon Based Compression Algorithm for DNA Sequences with Two Bit Encoding

Murugesan G

*Department of Computer Science and Engineering*

*St. Joseph's College of Engineering*
*murugesh02@gmail.com*

***Abstract: Deoxyribonucleic acid called DNA is the littlest essential unit that bears the hereditary directions of a living being. Due to increase in genomic data, increases the storage capacity of that data being stored. In order to process the massive amount of genomic data, there is a need of an effectual depository for quick transposal and achieve preferable performance while processing that data. In this paper, introduced a novel compression algorithm without using much complicated computation for compression of DNA sequences.The result in terms of compression magnitude relation in bits/base and also the compression/decompression time has been compared with the present approaches conferred within the literatures. Our algorithmic rule produces higher compression quantitative relation and utilizes terribly less time to compress and decompress the genomic sequences and additionally there's no further storage needed for compression and decompression.***

***Keywords: Compression Algorithm, DNA Compression, Genomic sequence compression, Bit encoding, Computational Biology***

## 1. INTRODUCTION

DNA is composed of precise ordering of the four nucleotides as Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). It represents the physical medium within which all properties of living organisms are enclosed. Deoxyribonucleic acid DNA, that bears the genes and alternative nucleotides dwell in twenty three pairs of chromosomes, whole of forty six. The two strands are referred to as polynucleotides called biopolymer and these biopolymer strands coil around one another to create a spiral structure.The bases of two distinct strands are linked by hydrogen bonds and are linked to each other by covalent bonds. In addition, there are repeats in several forms inside a DNA sequence known as motifs that have important biological implications and are seems abundant higher frequency. Here I tend to use the terms deoxyribonucleic acid sequence, biological sequence and genomic knowledge alternatively. Due to the next generation sequencing machine for genomes generates voluminous amount of data of length ranging from megabases to gigabases, there is an increasing need of effective storage spaces to store, process and transmit the genomic data in online.The two major online repositories for storing biological sequences are GenBank and Whole Genome Shotgun (WGS). From 1982 to the current, the amount of bases in GenBank has doubled close to each eighteen months. By October 2018 statistics from World Health Organization (WHO), GenBank has 279668290132 bases and 209656636 sequences, and also WGS has 3444172142207 bases and 722438528 sequences. Storing the generated genomic data is a challenging task. In order to overcome this issue, productive and efficient compression algorithm must be introduced to compress the biological sequences.

Data compression is a method of modifying, coding or changing the bits structure of information in the simplest way that it consumes less area on disk. In literatures it has huge variety of standard compression algorithms for data compression. Since the DNA sequence composed of solely four characters {a, g, c, t} and also the sequence are rich in repeats and palindrome nature, the general purpose compression algorithms cannot compress as well. So there is a need of specialized compression algorithms to handle the DNA sequence compression. Although there is a collection of DNA sequence compression algorithms are proposed by researchers with varying compression ratios stills there is a scope to improve the compression ratio with respect to storage and data transfer rate.

This paper presents a codon based binary encoding algorithm for compressing a DNA sequence to get better compression ratio compared with the prevailing algorithms. A codon is a set of three DNA nucleotides corresponds to an amino acid or stop signal during protein synthesis. The full sets of codon for a DNA sequence are called as genetic codes and contain 64 possible combinations. Among the 64 codons, 61 codons are amino acids and the remaining three are stop signals. In this work, initially read the DNA sequence codon by codon and it is encoded with 1 bit, 2 bit, 4 bit, 5 bit or 6 bit binary value with respect to the codon representation. The proposed algorithm is based on pattern matching technique without using any predefined dictionary structure and carried out with very few computations.

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 compression algorithm. 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

## 2. RELATED WORKS

Although the general purpose data compression algorithms like gzip4 and bzip4 are widely used for compressing DNA sequences, they do not produce better results on DNA sequence compression due to smaller set of alphabets, massive amount of exact repeats and also rich in palindromes in nature. Researchers developed a rich set of compression algorithms for compressing the DNA sequences. In general, the compression algorithms are mainly classified into two classes' viz., lossy and lossless, where lossy algorithms are most suitable for video compression and the lossless algorithms are used in data compression. Lossy compression is also called as irreversible compression where there is a partial loss in the data stored. In lossy compression, the original data cannot be retrieved fully from the compressed file. They mostly discard the redundant information and are widely used in compression of images and videos and also it can follow prediction, transformation or quantization mechanisms. In lossless compression, the actual information is retrieved from the decompressed file without loss. Most of the DNA compression algorithms are lossless compression algorithms because losing a single base will misdirect the entire sequence.

The lossless compression techniques designed specifically for DNA sequence can be either vertical mode or horizontal mode. Vertical mode compression methods mainly deal with compressing the data that exists in the file. In this kind of compression mechanism the data amid two sequences are utilized. Among the two sequences, one of them is considered to be a reference sequence. With the development in techniques like Next-Generation Sequencing, there is high growth in usage of file formats. Vertical modes of DNA compression are widely used in genetic tasks where the length of the genomic sequence is considerably large.This paper mainly focuses on horizontal mode of compression technique. In horizontal mode, the compression is carried out by considering only the substrings. It can be canonical, directory based, directory free or 2 - bit confinement.

GenCompress [5] 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 [8] 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 are 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 [6] is persuasive, faster and has better running time when compared with the previous compression algorithms. It applies software called Pattern Hunter [11] 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 symbols 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 [13] 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 [12] 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 where as the non repeat parts uses arithmetic 2 compression or Context Tree Weighting. GenBit Compress [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 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 produces a compression ratio of 0.017bpb 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. Biocompress [7] is a lossless compression algorithm 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

## 3. COMPRESSION ALGORITHM

In this proposed work, a compression algorithm based on two bit substitution method for compressing a DNA sequence with considerable improvement in compression ratio compared with the existing approach have been developed. Also there is no dictionary is used to encode/decode the codon, so there is no additional memory is required for both compression and decompression process. The following procedures describe the sequence of operations carried out in the proposed algorithm. There are two subroutines are used in this algorithm; the procedure display() is used to return the binary value for a nucleotide and the other one compress() is used to read the DNA sequence three character at a time (codon) and check the cases discussed in the previous section. The method concat() is used to combine the strings in the argument in the same order as the principle of string operation of concat in programming languages.

```
Procedure display(char)
    begin
        a ←“00”
        g ←“01”
        c ←“01”
        t ←“11”
    if char =’a’ then
        pchar:=a
    else if char = ’g’ then
        pchar ← g
    elseif char = ’c’ then
        pchar ← c
    else
        pchar ← t
    end
    returnpchar
end
```

Procedure **Compress**(file)

```
begin
    pchar1←null
    while(! end of file)
        do begin
            read three character and assign it to word
            len ← length(word)
            iflen = 1 then
                pword ← display(word)
            else
                iflen =2 then
                    pchar1← display(word[0])
                    pchar2← display(word[1])
                    pword ←concat(pchar1, pchar2)
            else
                iftwo adjacent words are identical then
                    pword←'0'
            else
                iftwo adjacent words are palindrome then
                    pword←'1'
            else
                if all characters in the word are same then
                    pword ←  display(word[0])
            else
                if first two character in the word are same then
                    pchar1← display(word[0])
                    pchar2← display(word[2])
                    pword ← concat('0', pchar1, pchar2)
            else
                if last two character in the word are same then
                    pchar1← display(word[0])
                    pchar2← display(word[1])
                    pword ← concat('1', pchar1, pchar2)
            else
                pchar1← display(word[0])
                pchar2← display(word[1])
                pchar3← display(word[2])
                pword ← concat(pchar1, pchar2, pchar3)
            end
        writepword into output file
    end
end
```

## 4. RESULT AND DISCUSSION

The proposed algorithm was implemented with python programming language in windows 7 environment with the system configuration of Intel core i5-4570S CPU@2.90Ghz and 8GB RAM. To test the proposed algorithm I have taken a portion of DNA sequence form vaccg and the compressed binary form is shown below. Let us assume a DNA Sequence

S= taaaattaaattaattataaaattatgtatatgatttactaactttagttagataagttagtaatacataaattttagtatattaatattatatttt

*l* = 98.

No. of codons = *l*/3 (32 codons and 2 nucleotides)

Until end of the sequence, the sequence is read by three characters at a time (codon) and is assigned to $C_i(1 \leq i \leq l/3)$ where $l$ is the length of the sequence and it is test with the three cases (identical, partially identical and distinct) as discussed in the previous section.

$C_1$ = gta

Here $C_1$ belongs to third case (ie., distinct - all the characters are not identical), and its equivalent binary value is 011100. The next codon in the sequence is

$C_2$ = aaa

Here $C_2$ belongs to first case (ie identical - all the three characters are identical), and its equivalent binary value is 00 and the other codons are represented as follows.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| gta | aaa | tta | aat | Taa | tta | taa | aat | tat | gta |
| 011100 | 00 | 01100 | 00011 | 1 | 01100 | 11100 | 1 | 1100 | 011100 |
| tat | gat | tta | cta | act | tta | gtt | aga | taa | gtt |
| 1100 | 010011 | 01100 | 101100 | 001011 | 01100 | 10111 | 0001 | 11100 | 10111 |
| agt | aat | aca | taa | att | tta | gta | tat | taa | tat |
| 000111 | 00011 | 0010 | 11100 | 10011 | 1 | 011100 | 1100 | 11100 | 1100 |
| tat | att | tt | | | | | | | |
| 0 | 10011 | 1111 | | | | | | | |

No. of bits = 146

$$Compression\ ratio = \frac{No.of\ bits}{Length\ of\ the\ sequence}$$
$$= \frac{146}{98}$$
= 1.49 bits/base

The same principle of operations is applied to the benchmark dataset with different sequences of FASTA format from ncbi database.

Here it is considered the gene sequence of VACCG for the implementation of the proposed algorithm. It has 191737 bases and is the longest sequence which is considered for the experimentation.

$l$=191737

$$\text{No. of codons} = \frac{l}{3}$$
$$= \frac{191737}{3}$$
$$= 63912$$

The number of codons in the sequence is 63912 and the number of bases is 191737 (63912 x 3 + 1). Here +1 denotes the remaining single base from the bases in the sequence (ie 191737 mod 3). When executing our proposed algorithm gives the results as shown in Table 1

Let $B_i$ be the number of bits assigned for a codon for the i$^{th}$ category ($1 \leq i \leq N$); where N is the number of categories listed in the Table 1 (N=7), $NC_i$ be the number of codons in i$^{th}$ category and $NB_i$ be the number of bits assigned for i$^{th}$ category codons.

$$Total\ No.of\ bits = \sum_{i=1}^{N} NB_i + 2$$
$$= \sum_{i=1}^{N} B_i NC_i + 2$$

= 300725 + 2

=300727

$$Compression\ ratio = \frac{Total\ No.of\ bits}{Length\ of\ the\ sequence}$$

$$= \frac{300727}{191737} = 1.568\ bits/base$$

Table 1: Number of bits with respect to each category

| Category | No. of bits assigned/codon ($B_i$) | No. of codons ($NC_i$) | Total No. of bits ($NB_i$) |
|---|---|---|---|
| Consecutive codons are identical | 1 | 1861 | 1861 |
| Consecutive codons are palindrome | 1 | 748 | 748 |
| Three bases in a codon are identical | 2 | 5201 | 10402 |
| First and last bases of a codon are identical | 4 | 13223 | 52892 |
| First two bases in a codon are identical | 5 | 10987 | 54935 |
| Last two bases in a codon are identical | 5 | 11465 | 57325 |
| All the three bases are distinct | 6 | 20427 | 122562 |
| | Total | 63912 | 300725 |

Table 2: Comparison of Compression ratios of proposed algorithm against existing algorithm

| Algorithm | | **HUMDYSTROP** (38770 bases) | **HUMHPRTB** (56737 bases) | **HUMHBB** (73323 bases) | **MPOMTCG** (186609 bases) | **VACCG** (191737 bases) |
|---|---|---|---|---|---|---|
| WinRAR | | 2.37 | 2.23 | 2.22 | 2.30 | 2.33 |
| Gzip | | 2.36 | 2.26 | 2.24 | 2.32 | 2.25 |
| Bzip | | 2.18 | 2.09 | 2.14 | 2.17 | 2.09 |
| Gzip4 | | 1.94 | 1.92 | 1.89 | 1.97 | 1.87 |
| Bzip4 | | 2.07 | 2.00 | 1.99 | 2.01 | 1.95 |
| BioCompress2 | | 1.92 | 1.90 | 1.88 | 1.93 | 1.76 |
| GenCompress2 | | 1.92 | 1.84 | 1.82 | 1.90 | 1.76 |
| DNACompress | | 1.91 | 1.82 | 1.79 | 1.89 | 1.76 |
| DNAPack | | 1.91 | 1.79 | 1.78 | 1.89 | 1.76 |
| Seed Based | | 1.86 | 1.69 | 1.74 | 1.76 | 1.64 |
| DNABit | | 1.57 | 1.57 | 1.61 | 1.57 | 1.65 |
| **Codon Based** (proposed) | **Ratio (bpb)** | **1.55** | **1.54** | **1.55** | **1.55** | **1.57** |
| | **Time (Sec.)** | **0.095** | **0.115** | **0.156** | **0.281** | **0.297** |

Table 2 infers the comparison of compression ratios of various existing techniques with the proposed codon based compression algorithm for the five standard DNA sequences. The codon based two bit compression of DNA sequences was experimentally confirmed on standardbenchmark DNA sequences expressed in FASTA format. The typical input DNA sequences considered are VACCG is a genome of complete Copenhagen vaccinia virus, MPOMTCG is a complete genome of Marchantia polymorpha mitochondrial DNA, HUMHBB is present in human beta globin region of chromosome 11, HUMDYSTROP is present in a dystrophin gene of Homo Sapiens and HUMHPRTB is a human hypoxanthine phosphoribosyl transferase gene of homosapiens. From the literature DNABit algorithm shows the significant improvement compared to the other algorithms with the average compression ratio of 1.59. The proposed algorithm yields prominent compression ratio for all the five DNA sequences. Our compression algorithm grant better compression ratio of around 1.55 bits per base. From the comparison table, it is deduced that the codon based two bit compression algorithm is better and efficient than existing compression techniques. Moreover, the proposed algorithm utilizes very less amount of time for compressing the DNA sequence. The decompression is the reverse process and it is not discussed in this paper.

## 5. CONCLUSION

The compression of biological sequences particularly for DNA sequences is designed with significant improvement in the compression ratio. In this paper, the existing works related to the compression of biological sequences are discussed and a new algorithm to compress the substantial genetic code by codon based two bit encoding technique is designed. The proposed algorithm yields a better compression ratio when compared to the existing compression mechanisms. In future it can be extended by increasing the length of the word (more than 3 bases) instead of codon, comparing the partial palindrome and also comparing the jumbled words so that there is a possibility of improve the performance of the algorithm.

## 6. REFERENCES

[1] Afify, H., Islam, M., Abdel-Wahed, M. and Kadah, Y.M.,"Genomic Sequences Differential Compression Model", Proceeding of 27th National Radio Science Conference, Egypt., 2010

[2] Bacem Saada and Jing Zhang," DNA Sequences Compression Algorithm Based on Extended-ASCII Representation", Proceeding of the world congress on engineering and computer science (WCECS 2015), II, San Francisco, USA., 2015

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

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

[5] Chen, X., Kwong, S. and Li, M.,"Compression algorithm for DNA sequences and its applications in genome comparison", Proceedings of the 4th Annual International Conference on Computation Molecular Biology (RECOMB'00)", Tokyo, Japan,2000

[6] Chen, X., Li, M., Ma, B. and Tromp, J.,"DNACompress: fast and effective DNA sequence compression", Bioinformatics, 18(12), 1696-1698, 2002

[7] Grumbach, S and Tahi, F.,"Compression of DNA sequences", Proceedings of the IEEE Symposium on Data Compression, 340-350, Snowbird, Utah, USA,1993

[8] Grumbach, S. and Tahi, F.,"A new challenge for compression algorithms: genetic sequences", Information Processing & Management, 30(6), 875-886, 1994.

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

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

[11] Ma, B., Tromp, J. and Li, M., "PatternHunter: fast and more sensitive homology search", Bioinformatics, 18(3), 440-445, 2002.

[12] Matsumoto, T., Sadakane, K. and Imai, H.,"Biological sequence compression algorithms", Genome Informatics, 11, 43-52, 2000.

[13] Myung, J. I., Navarro, D. J. and Pitt, M. A.,"Model selection by normalized maximum likelihood", Journal of Mathematical Psychology, 50(2), 167-179, 2006.

[14] Pamela Vinitha Eric, Gopakumar Gopalakrishnan and Muralikrishnan Karunakaran (2016), "An Optimal Seed Based Compression Algorithm for DNA Sequences", Advances in Bioinformatics, 1-7, 2016

[15] Prasad, V. H., and Kumar, P. V., “A New Revised DNA Cramp Tool Based Approach of Chopping DNA Repetitive and Non- Repetitive Genome Sequences”, IJCSI International Journal of Computer Science Issues, 9(6), 448-454, 2012.

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

[17] Rajeswari, P. R., and Apparao, A.,“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, 2010.

[18] Satyanvesh, D., Balleda, K. and Padyana, A.,“GenCodex - A Novel Algorithm for Compressing DNA seuences on Multi-cores and GPUs”, Proc. IEEE, 19th International Conf. on High Performance Computing (HiPC), 37, Pune, India, 2012

[19] Zhu, Z, Zhang, Y, Ji, Z., He, S. and Yang, X."High - throughput DNA sequence data compression", Briefings in Bioinformatics, 16 (1), 1-15, 2015.