Shan e Zahra et al LGURJCSIT 2019

ISSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

LGU Research Jounral for
Computer Sciences & IT

Vol. 3 Issue 3, July - September 2019

Using codes in place of DNA Sample in Databases to reduce Storage

[1]Shan e Zahra,[2]Sabir Abbas,[3]Tayyab Altaf
[1,3]Department of Computer Science
[2]Faculty of Information Technology
Lahore Garrison University, Lahore, Pakistan
[1]shanezahra@lgu.edu.pk, [2]Sabirabbas@lgu.edu.pk,[3]Tayyab.altaf@lgu.edu.pk

Abstract:

Biological data mainly comprises of Deoxyribonucleic acid (DNA) and protein sequences. These are the biomolecules that are present in all cells of human beings. Due to the self-replicating property of DNA, it is a key constituent of genetic material that exists in all breathing creatures. This biomolecule (DNA) comprehends the genetic material obligatory for the operational and expansion of all personified lives. To save DNA data of a single person we require 10CD-Rom's. In this paper, A lossless three-phase compression algorithm is presented for DNA sequences. In the first phase the dataset is segmented having tetra groups and then the resultant genetic sequences are compressed in the form of unique numbers (e.g Array Index) and in the second phase binary code is generated on the bases of array index numbers and in the last phase the modified version of Run Length Encoding (RLE) is applied on the dataset.
The newly proposed technique has been implemented and its performance is also measured on samples. It has achieved the best average compression ratio. After Storing different DNA Samples.

Keywords: Adenine, Arithmetic Coding, Base pair, Bits per base, Cytosine, Guanine, Run length encoding (RLE) and Thymine.

1. Introduction:

Daily 2.5 quintillion bytes of information is being generated everywhere. This is sorted as large information. Datasets increment in size to some extent since they are progressively being assembled. As information expands computational expense likewise increments for handling this information. The issue of putting away enormous information produces two issues, the area needed to save and the ideal opportunity for encoding and deciphering. In few years, 90% of information has been produced quickly. To save and process information in an effective manner pressure is the need of the day. Document pressure includes encoding data utilizing fewer bits than the first portrayal. Pressure strategies are firmly related with sort of documents, for example, content, picture, sound, and video. The methods used to pack the information of content records may not be relevant for picture or sound documents.

The most ordinarily utilized human genomic information accessible in the open database, for example, GenBank is tremendous information and expanding exponentially. GenBank is a piece of the International Nucleotide Sequence Database Collaboration, which includes the DNA Databank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank is given by at NCBI (National Center for Biotechnology Information). These three associations trade information every day. Deoxyribonucleic Acid (DNA) is a particle that encodes the hereditary data. DNA groupings are immense in size makes its pressure a difficult assignment. The DNA strand contains four nucleotide bases adenine, cytosine, guanine, and thymine. In this way, DNA groupings are the

mixes of just four bases (A, C, G, T). DNA successions pass on important data about various life forms. These organic groupings are either dreary or non-tedious however, they don't contain totally irregular information and, in this way, offer an open door for compaction. General content pressure calculations can adversely affect the size of organic successions. Explicit calculations have been created for the pressure of DNA groupings.

DNA was firstly identified by Frederic Miescher in 1869.he was a German organic chemist. For more long Term, other experts did not accept the presence of DNA Atom in the blood clot. James Watson, Francis Crick, Maurice Wilkins, and Rosalind Franklin made sense of the DNA in 1953. Then they proved DNA a twofold string structure and it could store organic data. In 1962 Crick Wilkins and Watson got the Nobel Prize in medicine for their revelations regarding the atomic formation of nucleic acids and their equality for data move in existent facts." Franklin was excluded towards the honor, despite the fact that her work was basic to the examination [1]. A DNA arrangement is an invasion that entails an examiner to conclude the appeal of bases in a DNA succession. The modernization can be handled to agree with the appeal of bases in qualities, chromosomes, or a whole genome [2]. The experts accomplished the vital fully organized human genome, as determined by a report by the National Human Genome Research Institute in 2000, DNA arranged pattern in the form of a chromatogram, a series of tetrad differently colored points, with each color correlative to a disparate DNA base as shown in Figure 1.1 [3]

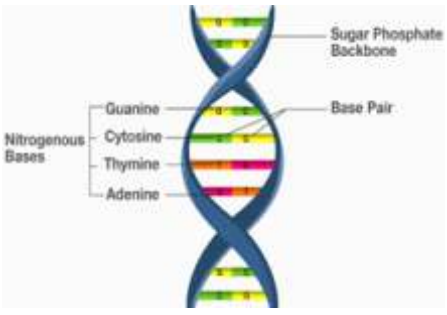

Figure 1: DNA double helix for sequence (GCGCTAGC)

DNA strings are made out of a letter in order of tetrad Symbols: A, T, C, and G as shown in Each three-letter sub-siding inside the DNA successions are termed codons (for example AGC and CGA). There is a maximum of 64 perceived codons. These codons produce 20 distinctive amino acids on the grounds that diverse codons can create a similar amino corrosive [7]. In a human DNA structure, there are such huge numbers of obscure nucleotides as shown in These concealed nucleotides are spoken to by N (space) and connected with each other to create dual long strands that twining to make an interrelation termed as diploid twist very towering, doubtlessly, without the correct binding they can't be able to merge with the cells of DNA. To fix under the roof of the cells, DNA is crook to shape up the conformation of chromosomes. Each chromosome contains a separate DNA atom. People have 23 sets of chromosomes in DNA, which are found surround the cells. Human DNA has 3 billion base pairs. It is the major carrier of genetic information and there are 20,000-25000 genes within DNA. Long DNA base pair sequence data can be represented graphically for better understanding by using Huffman Coding. Compression of DNA Sequence data has been of interest for many decades. DNA is organized in chromosomes, which are located in the center of cells. The nucleus of hominid cells comprises 46 chromosomes, each of which contains a particular lined molecule of deoxyribonucleic acid (DNA), which are confidentially multiplexes with proteins in the form of chromatin. DNA is the building block of life, which comprehends programmed genetic information for breathing creatures. A DNA is transliterated to convert a predecessor mRNA, which is then interwoven to become an mRNA, which is interpreted to become a protein. Because, except for some cells all the cells in a humanoid physique comprehend an indistinguishable set of genes, for different cells of the human body the appearance level of every gene must be different [8]. It is a major issue to store a large amount of DNA sequence data, which consists of long-chain of DNA base pair sequence data. DNA and protein sequence databases amount to hundreds of gigabytes (GB) storage space. This makes data compression in genomics a very important and challenging task. For storing the DNA and protein sequence, publicly available databases are EMBL, Gene bank, and DDBJ. The size of these databases is increasing exponentially [9]. A few qualities of DNA groupings demonstrate that they are not

arbitrary arrangements. On the off chance that these arrangements were absolutely arbitrary, the most productive and coherent approach to save them would utilize two bits for every base. Be that as it may, DNA is handled for the statement of proteins in living lives, and along with these proteins lines it must contain some consistent association [10].

1.1. Issue Statement:

In both consistent and business arranges there is uncommon activity centered at sequencing the DNA of various species and looking at the capriciousness of DNA between individuals of comparative species, which produces gigantic proportions of information that ought to be secured and passed on to endless. Subsequently, there is an uncommon necessity for the brisk and capable weight of DNA courses of action. Data weight right now has an extensively progressively critical activity in lessening the costs of data transmission since the DNA archives are customarily shared and scattered over the Internet. It is accepted that DNA successions encode life in a nonrandom manner, some organic confirmations for that are the presence of numerous duplicates of fundamental qualities, and just 1000 protein collapsing designs in nature. The size of worldwide DNA arrangement Databases surpassed 100 Gig humbles in 2005, the bend in Figure 1.4 demonstrates an exponential development of GenBank [18] database.

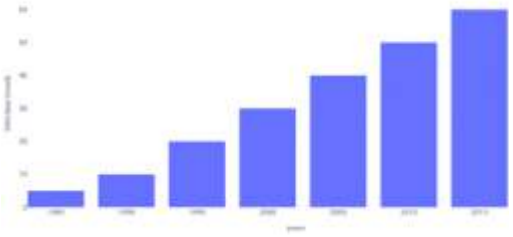

Figure 2: The trend of genetic data growth

Data compression also gained more importance because of the relation between the speed of processors and main memory as shown in Figure 1.5 that the speed of business chip has been expanding generally 70% consistently, while the speed of product DRAM has improved by minimal over half over the previous decade [19]. As data compression reduces data size it is considered trading memory cycles by CPU cycles. Compression and decompression are CPU intensive operations.

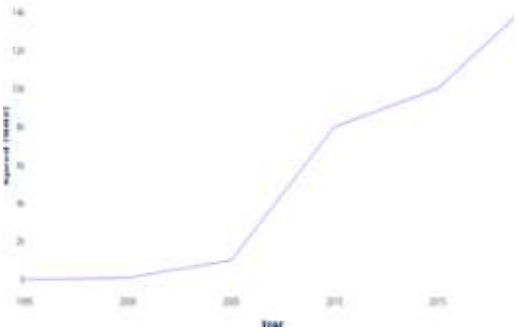

Figure 3: Hardware trends in DRAM and CPU

2. Background and Related Work

Particular techniques are reasonable considering the composition of the DNA fragment. The very first DNA-specific lossless compression wereBioCompress and Biocompress-2. Numerous DNA compression algorithms such as Cfact, Gencompress, CTW+LZ, DNA Compress, DNAC, DNA sequitur, DNA Pack, DNA Compress have also been constructed since then. Biocompress is based on an approximate search for duplicate and palindromes. Bio compress computes the non-repeat subsequence utilizing two sections. Biocompress-2 produces an order-two Markov model to encode the sequence's non-repeat zones. Cfact looks for the precise repeats of the sequence that are the shortest. It develops over the first pass the suffix tree of the series and does the official encoding in the sequence's second pass. The sequence's non-repeat zones are encoded by2-bits per symbol. Gencompress uses recording activities such as mutation, insertion, deletion to search approximate repeats. CTW+LZ computes the substitution technique for the repeated repeats. It encodes the sequence's brief and non-repeat regions by weighing context tree. DNA Compress implements the compression method of Ziv Lempel. It's operating in two phases. It experiences every single surmised rehash in the progression in the main stage, including complimentary palindromes, are utilizing tracker design programming. Surmised arrangement rehashes and non-rehash districts are encoded in the subsequent stage. DNAc is a four-stage DNA pressure gadget. It produces a

postfix tree in the principal stage to locate a particular grouping repeats.in the second stage, all particular succession rehashes are extended by powerful programming into precise rehashes. It extricates from covering ones in the second stage the suitable non-covering rehashes. It encodes every one of the pushups in the last stage. DNA sequitur is a pressure calculation dependent on language. To speak to include information, it actualizes setting free language structure. For rehash areas and complimentary grouping palindromes, DNA pack actualizes hamming separation. For non-rehash areas, it utilizes either CTW or number-crunching 2 pressure. It uses a programming approach that will be dynamic. DNABIT pack allocates double bits to pack dull and non-tedious zones for the upper area of the grouping.

3. Methodology:

There are only four bases, for instance, A, T, G, C found in the DNA sequence. when A, C, T, G combine with each other they generate an array of 0 to 255 indexes as shown in Table 1.1. Possible combinations of four letters are 256. At the first stage, the dataset is divided into segments (n/4) and each four letters have a unique index number in the array. In this manner, a unique index number is used to replace DNA sequence.

This a one pass count It takes a commitment of a DNA progression of length n, and allotments into (n-r)/4 divides where n = r mod 4. using Algorithm system create another array and store all index no's of same combinations. The index no's is then replaced with binary code. In the final step, we apply run-length encoding on the binary bits and we achieve a compressed sample.

3.1. Statement of the Algorithm:

Our estimation relies upon the parallel depiction of nucleotides. In any case, our figuring records the total incorporate of each nucleotide in the DNA Sequence and pack the DNA gathering reliant on the repeat of each nucleotide. Every DNA course of action is a mix of nucleotides {A, C, G, T}, and these alphabets generates 0-255 possible combinations of these four letters. The System generates these possible combinations as shown in Table 1.1.
Where each severe is named as BASE and tetra letter encoded in a decimal bit on the bases of array index no.

Table 1.1: Possible combinations of ACTG Patterns

[0] AAAA	[51] AGAG	[102] CTCT	[153] TCTC	[204] GAGA
[1] AAAC	[52] AGCA	[103] CTCG	[154] TCTT	[205] GAGC
[2] AAAT	[53] AGCC	[104] CTTA	[155] TCTG	[206] GAGT
[3] AAAG	[54] AGCT	[105] CTTC	[156] TCGA	[207] GAGG
[4] AACA	[55] AGCG	[106] CTTT	[157] TCGC	[208] GCAA
[5] AACC	[56] AGTA	[107] CTTG	[158] TCGT	[209] GCAC
[6] AACT	[57] AGTC	[108] CTGA	[159] TCGG	[210] GCAT
[7] AACG	[58] AGTT	[109] CTGC	[160] TTAA	[211] GCAG
[8] AATA	[59] AGTG	[110] CTGT	[161] TTAC	[212] GCCA
[9] AATC	[60] AGGA	[111] CTGG	[162] TTAT	[213] GCCC
[10] AATT	[61] AGGC	[112] CGAA	[163] TTAG	[214] GCCT
[11] AATG	[62] AGGT	[113] CGAC	[164] TTCA	[215] GCCG
[12] AAGA	[63] AGGG	[114] CGAT	[165] TTCC	[216] GCTA
[13] AAGC	[64] CAAA	[115] CGAG	[166] TTCT	[217] GCTC
[14] AAGT	[65] CAAC	[116] CGCA	[167] TTCG	[218] GCTT
[15] AAGG	[66] CAAT	[117] CGCC	[168] TTTA	[219] GCTG
[16] ACAA	[67] CAAG	[118] CGCT	[169] TTTC	[220] GCGA
[17] ACAC	[68] CACA	[119] CGCG	[170] TTTT	[221] GCGC
[18] ACAT	[69] CACC	[120] CGTA	[171] TTTG	[222] GCGT
[19] ACAG	[70] CACT	[121] CGTC	[172] TTGA	[223] GCGG
[20] ACCA	[71] CACG	[122] CGTT	[173] TTGC	[224] GTAA
[21] ACCC	[72] CATA	[123] CGTG	[174] TTGT	[225] GTAC
[22] ACCT	[73] CATC	[124] CGGA	[175] TTGG	[226] GTAT
[23] ACCG	[74] CATT	[125] CGGC	[176] TGAA	[227] GTAG
[24] ACTA	[75] CATG	[126] CGGT	[177] TGAC	[228] GTCA
[25] ACTC	[76] CAGA	[127] CGGG	[178] TGAT	[229] GTCC
[26] ACTT	[77] CAGC	[128] TAAA	[179] TGAG	[230] GTCT
[27] ACTG	[78] CAGT	[129] TAAC	[180] TGCA	[231] GTCG
[28] ACGA	[79] CAGG	[130] TAAT	[181] TGCC	[232] GTTA
[29] ACGC	[80] CCAA	[131] TAAG	[182] TGCT	[233] GTTC
[30] ACGT	[81] CCAC	[132] TACA	[183] TGCG	[234] GTTT
[31] ACGG	[82] CCAT	[133] TACC	[184] TGTA	[235] GTTG
[32] ATAA	[83] CCAG	[134] TACT	[185] TGTC	[236] GTGA
[33] ATAC	[84] CCCA	[135] TACG	[186] TGTT	[237] GTGC
[34] ATAT	[85] CCCC	[136] TATA	[187] TGTG	[238] GTGT
[35] ATAG	[86] CCCT	[137] TATC	[188] TGGA	[239] GTGG
[36] ATCA	[87] CCCG	[138] TATT	[189] TGGC	[240] GGAA
[37] ATCC	[88] CCTA	[139] TATG	[190] TGGT	[241] GGAC
[38] ATCT	[89] CCTC	[140] TAGA	[191] TGGG	[242] GGAT
[39] ATCG	[90] CCTT	[141] TAGC	[192] GAAA	[243] GGAG
[40] ATTA	[91] CCTG	[142] TAGT	[193] GAAC	[244] GGCA
[41] ATTC	[92] CCGA	[143] TAGG	[194] GAAT	[245] GGCC
[42] ATTT	[93] CCGC	[144] TCAA	[195] GAAG	[246] GGCT
[43] ATTG	[94] CCGT	[145] TCAC	[196] GACA	[247] GGCG
[44] ATGA	[95] CCGG	[146] TCAT	[197] GACC	[248] GGTA
[45] ATGC	[96] CTAA	[147] TCAG	[198] GACT	[249] GGTC
[46] ATGT	[97] CTAC	[148] TCCA	[199] GACG	[250] GGTT
[47] ATGG	[98] CTAT	[149] TCCC	[200] GATA	[251] GGTG
[48] AGAA	[99] CTAG	[150] TCCT	[201] GATC	[252] GGGA
[49] AGAC	[100] CTCA	[151] TCCG	[202] GATT	[253] GGGC
[50] AGAT	[101] CTCC	[152] TCTA	[203] GATG	[254] GGGT
				[255] GGGG

3.2. Procedure to Encode:

The computation works in three segments.

In segment-I potential mixes of tetra, letters are supplanted with decimal bits on the bases of array index no. In the absolute first stage, data is assembled in numbers. This tetra gathering is supplanted with the number and looking at digit's worth is secured in the yield record. This estimation has been minded the planned System. The segment-II deals with the numbers and the algorithms convert numbers into their binary code.

In segment-III run-length encoding on the past record was applied. In the changed version of RLE, the continues running of data are secured in the detached record and the document of the reiterated character in another report rather than keeping one record of RLE game plan. In fact, changed RLE goes about as a bit of a bonus point out this weight out of the possibility of the DNA gathering.

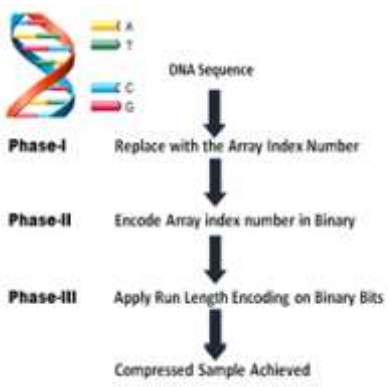

Figure.4. Encoding Flow

Figure the length of each character. On the off chance that a mix of four characters is available, by then dislodge the social event of conceivable blend with the letter set generally make the character just once and its redundancy and its summary onto the ensuing character plan. Get-together the characters into a string of length four and dole out twofold bits for each base of DNA.

3.3. Procedure to Decode:

Method for unraveling is just the reversal of encoding, for instance initially make an interpretation of the decimal and along these lines to twofold. Parallel code is decoded to DNA gathering character and by using rundown record and check to report the primary data is cultivated. Scrutinize the ASCII character; convert this decimal number into a twofold number system.

Consign the base of DNA (A, C, G, T) for every four gatherings like ATTG, ATGT, AAAA and AATC Write the characters in the yield record up to the given rundown in the rundown archive, by then make the character number out of times in the check record. The single and twofold occasions of a base aren't encoded using changed RLE as it doesn't preserve on getting a good deal on memory space since size additions for saving a base, its record, and its count. In the wake of encoding and unwinding all of the groupings, the result is moreover affirmed for every circumstance.

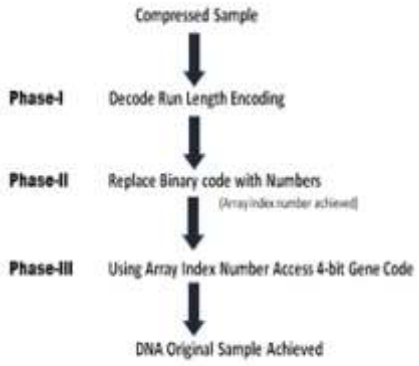

Figure. 5. Decoding Flow

4. Results and Discussion:

The proposed strategy was checked on different DNA sequences. DNA Data sets downloaded from the site NCBI [14,21]. A detailed description of the tried Datasets is given at ncbi site[21].

4.1. Encoding Analysis:

In the first phase assume that DNA Sample 'AA797726' having the length of 107 so n=r (mod 4) And repetitive A, T, G, C sequences are (e.g., ATCC GCGG GCGT AAAC GCTT CGAG ATGT TACG GGAG CTGA ATGA GGCC TTAG AGTT AAAG GAAG CCCA

TGCT ACAG AAGA GTCT GGAG ACAG CAAG GCTC ACTC CAG).

Each of these four repetitive sequences requires unique index no and if single or double alphabets left at the end of the sample then these samples will be as it is written on the array index as a combination. We cannot discard any alphabet from the DNA sequences because it will cause Data loss in the sequence and we will be unable to decode or achieve the original sample.

After segmentation the array(a) sequence for the above sample will be [0] ATCC, [1] GCGG , [2] GCGT , [3] AAAC , [4] GCTT , [5] CGAG , [6] ATGT , [7] TACG , [8] GGAG , [9] CTGA , [10] ATGA ,[11] GGCC , [12] TTAG , [13] AGTT , [14] AAAG , [15] GAAG , [16] CCCA , [17] TGCT , [18] ACAG , [19] AAGA , [20] GTCT , [21] GGAG , [22] ACAG , [23] CAAG , [24] GCTC , [25] ACTC , [26] CAG.

The system generates another array (b) as shown in Table 1.1. of possible combinations of A, C, T and G letters. We match combinations of the array (a) in the second array (b) and the result will be:

[0]37 , [1]223 , [2]222 , [3]1 , [4]218 , [5]115 , [6]46 , [7]135 , [8]243 , [9]108 , [10]44 , [11]245 , [12]163 , [13]58 , [14]3 , [15]195 , [16]84 , [17]182 , [18]19 , [19]12 , [20]230 , [21]243 , [22]19 , [23]67 , [24]217 , [25]25 , [26]CAG

In the second phase, we compress these numeric array indexes into binary no's and we achieve 3157 total lengths. The binary code is generated for the first 26 (e.g. 107/4=26.75) segments and the last 3 letters came as it is in the end of binary code.
In the third and last phase we apply run length encoding on the binary sample and we get the string of these letters (e.g. 1, 37, 0, 223, 1, 222, 0, 1, 1, 218, 0, 115, 1, 46, 0, 135, 1, 243, 0, 108, 1, 44, 0, 245, 1, 163, 0, 58, 1, 3, 0, 195, 1, 84, 0, 182, 1, 19, 0, 12, 1, 230, 0, 243, 1, 19, 0, 67, 1, 217, 0, 25, C, 1, A, 1, G, 1). The sample length is now 58. The sample is compressed 54.205 %.

4.2. Decoding Analysis:

In the first Phase we have the n string of these letters (e.g. 1, 37, 0, 223, 1, 222, 0, 1, 1, 218, 0, 115, 1, 46, 0, 135, 1, 243, 0, 108, 1, 44, 0, 245, 1, 163, 0, 58, 1, 3, 0, 195, 1, 84, 0, 182, 1, 19, 0, 12, 1, 230, 0, 243, 1, 19, 0, 67, 1, 217, 0, 25, C, 1, A, 1, G, 1). The sample length is 58.

In the second phase we convert binary no's into numeric array index no's and we achieve 3157 total lengths. The binary code is generated for the first 26 (e.g. 107/4=26.75) segments and the last 3 letters came as it is in the end of binary code.

In the last phase, we get DNA original samples having a length of 107. Each of these four repetitive sequences has unique index no and on the bases of DNA unique indexes, we achieve the original uploaded Sample sequences (e.g., ATCC GCGG GCGT AAAC GCTT CGAG ATGT TACG GGAG CTGA ATGA GGCC TTAG AGTT AAAG GAAG CCCA TGCT ACAG AAGA GTCT GGAG ACAG CAAG GCTC ACTC CAG).

4.3. Data size and reduction after compression:

The table demonstrates the compression proportions acquired by DNA explicit pressure calculations when connected to the benchmark groupings. The numbers are bits per base (bpb). This measure is typically used to make sense of the exhibition of any DNA explicit pressure technique

Table 1.2 : Performance of algorithm in two phases on different DNA sequences

Sr.No.	Sequence	Input Data Size	Size After Compression to Binary	After applying modified RLE	Reduced to %
1.	AA797726	107	3157	58	54.20
2.	AA839925	416	13480	208	50
3.	W41461	421	13353	208	49.40
4.	NM_000207	465	15586	234	50.32
5.	BF723563	476	13718	226	47.47
6.	BE333746	480	16283	236	49.16
7.	NM_001185097	491	16055	250	50.91
8.	NM_001291897	525	17595	264	50.28
9.	BI319892	537	17260	270	50.27
10.	W91618	545	18658	278	51.00
11.	HUMHBB	550	18452	272	49.45
12.	W62008	572	18736	282	49.30
13.	GR655411	580	16712	286	49.31
14.	AW741656	588	17821	288	48.97
15.	BE457084	632	20010	310	49.05

There was a contrast between the pressure ratios of some other DNA stress calculations [14] and the calculation provided in table IV. It appears to be seen that MPOMTCG data shows excellent results specifically through the use of adapted RLE (MRLE) and ASCII stress technique. It appears to be collected from the table that this calculation's pressure ratio is commendable. However, it is much lower than CTW+LZ than the normal pressure ratio (Cr) = 1,686 bits/base and GenCompress. Comparable layout and a velocity of 2.1 GHz Core 2 couple processor using the HUMDYSTROP DNA grouping were verified for the calculation presentation.

4. Conclusion

DNA Compression is a noteworthy and testing issue. We have shown another Storing strategy to pack DNA courses of action. As most other DNA blowers, our estimation works by finding gathered repeats and endeavoring to in a perfect world encode them. We mapped the issue of picking the best encoding of a DNA progression - covering more than one harsh repeat to a base cost course issue in an organized outline, by then we used special programming to settle it. Another encoding scoring limit is proposed which gives a better estimation of the number of bits required to encode a string using an as of late experienced string. Another dialect structure addressing DNA gathered repeats is proposed, which points of interest from the known properties of DNA courses of action. Our results are all around insignificantly better than past figures. This paper introduced the information disguising plan reliant on the DNA progression framework. In this paper from the start, DNA Sample has been changed over into numeric pressed code. RLE system has been used, it will make better weight dependent upon the DNA progression. Furthermore, decompression figuring has been made to get the principal data.

I acknowledge that this encoding system could be commonly recognized as a result of its consistency, consistency, homogeneity, nature, basic programming, and its assurance from bumbles.

References

[1] Pinho, Armando J., António JR Neves, Vera Afreixo, Carlos AC Bastos, and Paulo Jorge SG Ferreira. "A three-state model for DNA protein-coding regions." IEEE Transactions on Biomedical Engineering 53, no. 11 (2006): 2148-2155.

[2] Wang, Yu, Igor V. Tetko, Mark A. Hall, Eibe Frank, Axel Facius, Klaus FX Mayer, and Hans W. Mewes. "Gene

selection from microarray data for cancer classification—a machine learning approach." Computational biology and chemistry 29, no. 1 (2005): 37-46.

[3] Fritz, Markus Hsi-Yang, RaskoLeinonen, Guy Cochrane, and Ewan Birney. "Efficient storage of high throughput DNA sequencing data using reference-based compression." Genome research 21, no. 5 (2011): 734-740.

[4] Behzadi, Behshad, and Fabrice Le Fessant. "DNA compression challenge revisited: a dynamic programming approach." In Annual Symposium on Combinatorial Pattern Matching, pp. 190-200. Springer, Berlin, Heidelberg, 2005.

[5] Sanger, Frederick, Gilian M. Air, Bart G. Barrell, Nigel L. Brown, Alan R. Coulson, John C. Fiddes, C. A. Hutchison, Patrick M. Slocombe, and Mo Smith. "Nucleotide sequence of bacteriophage φX174 DNA." nature 265, no. 5596 (1977): 687.

[6] Sanger, Frederick, A. R. Coulson, T. Friedmann, G. M. Air, B. G. Barrell, N. L. Brown, J. C. Fiddes, C. A. Hutchison Iii, P. M. Slocombe, and M. Smith. "The nucleotide sequence of bacteriophage φX174." Journal of molecular biology 125, no. 2 (1978): 225-246.

[7] Yokoo, Hiromitsu, and TairoOshima. "Is bacteriophage φX174 DNA a message from an extraterrestrial intelligence?" Icarus38, no. 1 (1979): 148-153.

[8] Church, George M., Yuan Gao, and SriramKosuri. "Next-generation digital information storage in DNA." Science 337, no. 6102 (2012): 1628-1628.

[9] Goldman, Nick, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. LeProust, BotondSipos, and Ewan Birney. "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA." Nature 494, no. 7435 (2013): 77.

[10] Grass, Robert N., ReinhardHeckel, MichelaPuddu, Daniela Paunescu, and Wendelin J. Stark. "Robust Chemical Preservation of Digital Information on DNA in Silica with Error Correcting Codes." AngewandteChemie International Edition54, no. 8 (2015): 2552-2555.

[11] Erlich, Yaniv, and Dina Zielinski. "DNA Fountain enables a robust and efficient storage architecture." Science 355, no. 6328 (2017): 950-954.

[12] Erlich, Yaniv, and Dina Zielinski. "DNA Fountain enables a robust and efficient storage architecture." bioRxiv (2016): 074237.

[13] Yazdi, SM HosseinTabatabaei, Yongbo Yuan, Jian Ma, Huimin Zhao, and OlgicaMilenkovic. "A rewritable, random-access DNA-based storage system." Scientific reports 5 (2015): 14138.

[14] Organick, Lee, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z. Racz et al. "Scaling up DNA data storage and random-access retrieval." bioRxiv (2017): 114553.

[15] Yazdi, SM HosseinTabatabaei, Ryan Gabrys, and OlgicaMilenkovic. "Portable and error-free DNA-based data storage." Scientific reports 7, no. 1 (2017): 5011.

[16] Shipman, Seth L., Jeff Nivala, Jeffrey D. Macklis, and George M. Church. "CRISPR–Cas encoding of a digital movie into the genomes of a population of living bacteria." Nature 547, no. 7663 (2017): 345.

[17] Matsoukas, Ianis G. "Commentary: CrISPr–Cas encoding of a Digital movie into the Genomes of a Population of living Bacteria." Frontiers in bioengineering and biotechnology 5 (2017): 57.

[18] Church, George M., Yuan Gao, and SriramKosuri. "Next-generation digital information storage in DNA." Science 337, no. 6102 (2012): 1628-1628.

[19] Blawat, Meinolf, Klaus Gaedke, Ingo Huetter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W. Pruitt, and George M. Church. "Forward error correction for DNA data storage." Procedia Computer Science 80 (2016): 1011-1022.

[20] Church, George M., Yuan Gao, and SriramKosuri. "Next-generation digital information storage in DNA." Science 337, no. 6102 (2012): 1628-1628.

[21] www.ncbi.nlm.nih.gov/genbank/