ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

A Survey on DNA Compression Algorithms

Govind Prasad Arya, Dr. R.K.Bharti

Ph.D Scholar, Uttarakhand Technical University, Dehradun, Assistant Professor, Sikkim Manipal University, Gangtok, India

Assistant Professor, BTKIT, Dwarahat, Almora, Uttarakhand, India

ABSTRACT: The Modern instruments used for DNA sequencing are able to generate huge amounts of genomic data. Those large volumes of data require fast transmission, effective storage, superior functionality and provision of quick access to any record. Data storage costs have anconsiderable proportion of total cost in the formation and analysis of DNA sequences. In particular, the growth in the DNA sequences is highly controllable due to a fabulous increase in the disk storage capacity. The standard compression techniques unsuccessful to compress these sequences. Several specialized techniques were introduced for this purpose. In this paper, we present a detailed survey of lossless compression algorithms that have been developed for the compression of DNA sequences.

KEYWORDS: DNA, DNA Compression, Re-sequencing, Horizontal Compression, Vertical Compression, Soft Computing, Nucleotide Sequence Compression

I. INTRODUCTION

DNA sequences are formed of a set of four bases which are A, G, T, and C. Each group of three characters within the DNA sequences are knownas codons (e.g. TGC, CGT, ACT, etc.). Total 64 codons are possible. These yield 20 dissimilar amino acids as different codons can generate same amino acid [1]. An organism has in each one of its cells the identical DNA sequences building up the individual's genome, which for higher life forms are structured in many chromosomes[2]. In a human DNA there are numerous unknown nucleotides. These unknown nucleotides are denoted by N (space) so, the human genome structure contains five characters A, G, C, T andN [3]. Some features of DNA sequences show that these are not random sequences. If the sequences were entirely random, the most efficient & logical approach to store them would be using two bpb (bits per base). However, DNA is used for the expression of proteins in living organisms& thus must hold some logical organization [4]. There are several repeats within a given DNA sequence for example, AAACGT which may occur multiple times in a given DNA sequence. However, there might exist an approximate repeat of AAACGT in thesequence (e.g. AAACTG and ATACGT). In DNA, A & T are the complements of each other. Similarly G & C are the complement of each other. The complement of the DNA sequence AATCGT would be TTAGCA. If we then reversed TTAGCA we get ACGATT. ACGATT is defined as the reverse complement or palindrome of AATCGT. DNA sequences belonging to the same species are both large & highly repetitive. Consider that approximately 0.1 percent of the 3GB human genome is specific and the rest is common to all humans [5]. The size of the genomes differs wildly in range; viruses may have few thousand symbols, bacteria have millions of symbols, humans is having about 3 billion & amphibian species have around 120 billion bases[2].

The nucleotide sequences are stored by some big archival databases such as GenBank at the National Center for Biotechnology Information (NCBI)[9], the DNA Data Bank of Japan (DDBJ)[11], the European Bioinformatics Institute EMBL database[10], the Gene Expression Omnibus (GEO)[13], the Short Read Archive (SRA)[12] and the microarray database Array Express[14]. These databases organize, maintain and distribute the sequenced data. The three buddies (GenBank, EMBL Bank & DDBJ) of the International Nucleotide Sequence Database Collaboration (INSDC) set out to preserve ,capture & present the permanent scientific record for nucleic acid sequencing & associated information[15, 16]. The collected sequences from the sequencing schemes can range from terabytes to petabytes in size [8]. In the GenBank database in August 2013[22] there were approximately 154,192,921,011 bases

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

(154.2 billion) from 167,295,840 (167.3 million), In a year the size of database is getting two or three times bigger. Therefore efficient compression techniques (lossless)& data structures efficiently store, communicate, access and search these large datasets are required. The standard compression techniques are not suitable to compress the biological sequences well. These algorithms do not consider the special structures of biological sequences. Recently, numerous algorithms have been proposedto compression the DNA sequences based on the special structures of DNA. Grümbach and Tahiin[23] propose two modes for DNA compression, i.e. horizontal and vertical.

In this paper we made a survey for the main issues and results of lossless compression algorithms which were developed for DNA sequences with a comparative view. This survey would be helpful to individuals who are interested in DNA compression tools or algorithms for keeping up on their field & may be most beneficial for those who are not specialists in this field. We organize the content of the paper as follows: In second section we report the general compression algorithms & their performances with DNA sequences. In third section we present the DNA compression algorithms in horizontal mode and their performance, while the vertical mode's algorithms are presented in forth section. Finally, in fifth section, we conclude various research proposals to be explored further relating to DNA compression.

II. STANDARD ALGORITHMS FOR DNA COMPRESSION

For the general compression algorithms it is a difficult task to compression DNA sequences because these algorithms are specially designed for English text compression, while there are limited regularities among the bases in DNA sequences.DNA sequences have only four bases {A, T, C, G}. Thus, each base symbol can be represented by two bits (i.e. A-00, T-01, C-10, G-11). However, the standard text compression algorithms, such as gzip (Lempel–Ziv "LZ" + Huffman), compress (Lempel–Ziv–Welch "LZW"), and bzip2 (Huffman + Burrows–Wheeler transform "BWT" + Move-to-Front "MTF")are not able to compress these DNA sequences [4]. Using standard benchmark data, the average compression ratio is 2.271 bits per base (bpb) for gzip, 2.185 bpb for compress and 2.138 bpb for bzip2 except only one sequence "HUMGHCSA" for which the average compression ratio is 1.729 bpb[26]. Huffman's code also failsbadly to compress DNA sequences both in the static & adaptive model because of less probabilities of occurrence of the four bases [27]. The compression algorithm Prediction with Partial Match (PPM) is also not suitable to compress DNA sequences having less than two bits per base (bpb)either [27]. On the other hand, Context Tree Weighting (CTW) and arithmetic coding are known to compress the DNA data well [27]. In spite of the good compression ratio, CTW &arithmetic coding have shortcomings, such as low decompression speed [28].

A. Algorithms for DNA Compression in Horizontal Mode

The horizontal mode uses the information enclosed only in the sequence, usually by making reference only to its sub-strings. Most of the compression tools or methods used today including DNA compression come into the following categories:

1) Substitutional Based Methods: These algorithms compresses data by replacing long sequences with short pointer information or data to the same sequences in a dictionary. The first special purpose DNA compression algorithm wasBioCompresswhich was developed by Grumbach&Tahi[23] in 1993. This technique was based on the sliding window algorithm. First, it detects exact & reverse complement repeats in the DNA sequence and stores them into a 4-ary tree, and then it encodes them by repeat length & the position of a previous repeat occurrence. For non-repetitive regions, it is encoded as2bpb. The improved version of BioCompress wasBioCompress 2[29] is identical to BioCompress, but to encode non-repeat regions it uses order-2 arithmetic coding. As per theresultsthe average compression ratio for Biocompress is 1.850 bpband 1.783 bpb for Biocompress 2 which was better as compared to the general-purpose algorithms which provide at max.2 bpb compression ratio. The Cfact[30] is similar to Biocompress& uses a two pass algorithm to search for the longest exact & reverse complement repeats. For this purpose, in the first pass it builds the suffix tree of the sequence & does the actual encoding using LZ in the second pass. Non-repeat regions are encoded by 2bpb. The results using Cfact algorithm may be similar to Biocompress, but it takes more compression time because Cfact uses two passes. A similar substitution approach was used by Chen et al. in GenCompress[31], except the use of approximate repeat. An inexact repeat sub-sequence was encoded by a pair of

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

integers, as for BioCompress-2 and a list of edit operations for mutations. There are two variants: one is GenCompress-1 which uses the Hamming distance for the repeats& another is GenCompress-2, which uses the edition distance (insertion, deletion and substitution) for the encoding of the repeats. While GenCompressachieves better compression ratios with an average 1.742bpb for standard benchmark data than BioCompress-2 &Cfact.
Chen et al. propose a further modification of GenCompresswhich led to a two-pass algorithm named as DNACompress[32]. During the first pass, using a specific software called PatternHunter[33], it finds all approximate repeats including complementary palindromes. In the second pass, the approximate repeats & non-repeat regions are encodedby using the LZ compression scheme. DNACompressyields a better compression ratio with an average of 1.725 bpb for standard benchmark data with faster compression than GenCompress& CTW + LZ algorithms. DNACompress was capable to deal with large sequences (e.g. about 4.6 Megabases) approximately in a minute, where GenCompress required approximately half an hour.
Lee et al. proposed a DNAC algorithm in 2004[36] as an improvement of the Cfact algorithm, but it works in four phases: During the first phase, it constructs a suffix tree to identify exact repeats. During the second phase, by using dynamic programming, all exact repeats are extended into approximate repeats. In the third phase, the extraction of the optimal non-overlapping repeats are donefrom the overlapping ones, and in the last phase, it uses the Fibonacci encoding to encode the repeats in a self-defined way. This algorithm can be used to compress long sequences with millions or more bases. Behzadi& Le Fessant in DNAPack[4] detect a better set of repeats than those found by GenCompress&DNACompress by using dynamic programming in place of greedy approaches which others do. It uses the Hamming distance for the repeats & inverted repeats. Non-repeat regions are encoded by the best choice from ancontext tree weighting, order-2 arithmetic coding& naive 2 bpbmethods. DNAPack consistently achieved better than the GenCompress, Biocompress-2, CTW+LZ and DNA Compress in the terms of average compression ratio of 1.714 bpb for standard benchmark data. It is expected that the algorithm will be slow due to the long calculation& will not be suitable for long sequences.

2) Statistical Based Methods: There are number of statistical methods for DNA compression, such as Approximate Repeats Model (ARM), expert-Model (XM), CDNA and the finite-context model algorithm by replacing a popular symbol by a shorter code.
Loewenstern &Yianilospresented the first algorithm to combine statistical compression & approximate repeat for DNA compression called CDNA algorithm[37]. The algorithm is based on the probability distribution of each symbol or basewhich is obtained by approximate partial matches from the history. Each approximate match is applied to a previous sub-sequence having a small Hamming distance to the context preceding the symbol which has to be encoded. Predictions are joint using a set of weights which are learnt adaptively. The ARM algorithm is also a full statistical DNA compression algorithmwhich was proposed by Allison et al. in 1998[38] that procedures the probability of a sub-sequence by adding the probabilities over all explanations of how the sub-sequence is generated. The ARM & CDNA algorithms produce significantly better compression ratios than those in the substitutional methods&can alsogenerate information content sequences. The last pure statistical DNA compression algorithm is XM algorithm which was introduced by Cao et al. in 2007[39] that relies on a mixture of experts for providing symbol or bases by symbol probability estimates which are then used for driving an arithmetic encoder. The compression results of XM are outstanding as average compression ratio was 1.714 bpb for standard benchmark data. These algorithms are computationally intensive but practical only for compressing small sequences.

3) Substitutional and Statistical Based Methods:
Numerous DNA compression algorithms are the combination of substitution & statistical methods. An inexact repeat is coded using a pointer to its previous occurrence and the probabilities of symbols are changed, copied, inserted or deleted. The off-line algorithm proposed by Apostolico&Lonardi in 2000[40-41], uses repeated regions for compression. During each iteration, only exact repeats are considered& the algorithm selects a sub-string which leads to the largest contraction suffix tree used to find the sub-string with the maximum possible count of non-overlapping occurrences. The comparatively poor compression is achieved by off-line where average compression ratio is 1.938 bpb for standard benchmark data, is attributed to its consideration of exact repeats only. Certainly, we can say that off-line is not a DNA specific compression technique but a general purpose compression which works well for DNA sequences

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

[34]. Off-line is a time consuming algorithm because its create a suffix tree. Matsumoto et al. formed a new algorithm CTW + LZ[27] whichis based on the method of context tree weighting.This method uses a weighting of various models to determine the probabilities of next symbol. The algorithm identifies approximate repeats by using dynamic programming,and then encodes long exact & approximate repeats using a LZ77-type encoding. Short repeats & non-repeats are encodedby using CTW. The algorithm achieves better compression than Biocompress-2 &GenCompress, and average compression ratio is 1.738 bpb for standard benchmark data. Although this algorithm obtained good compression ratios but its execution time is too high for long sequences. Chen et al.[32] display that a small sequence (Approx. 229 K bases) required many hours to compress. Tabus et al. in[42] suggested a refined DNA sequence compression techniquewhich is based on normalized maximum likelihood, in short NML, discrete regression for approximate block matching. NML algorithm splits the input sequence into fixed size blocks. This work was later improved in GeNML[43]in 2005.The algorithm encodes fixed-size blocks equal to 24, 32, 40, or 48 by referencing a formerly encoded sub sequence of equal size of 218) with a minimum hamming distance. To edit the reference sub sequence, only replacement operations are acceptable, which therefore always has the same size as the block, although it may be situated in anrandom position inside the already encoded sequence. The algorithm performs better compression than GenCompress, Biocompress-2, CTW + LZ and DNAPack algorithms. The average compression ratio is 1.69 bpb forthe standard benchmark data except HUMHBB, because the authors were not able to find the correct version. Also, GeNMLrecovers genome size compression &speed. Mishra et al. proposed a DNA Sequence Compressor called DNASC in 2010[3]. This algorithm compresses the DNA sequence horizontally first using extended Lempel-Ziv (LZ) style illustration for the five basic symbols A, G, C, T, and N. Then the sequence is vertically compressedby using a block size equals to six&a window size of equals to 128. To compress every two digits of a block they used 7 bits. Hence, each block is compressed using 21 bits only, because every block has 6 digits & each 2 digits are represented by 7 bits. The DNASC algorithm provides better compression results as compared to Gencompress, Biocompress2, DNA compress, CTW+LZ and GeNML 2005. By comparing the results of DNASC algorithm with GeNML, the previous algorithm has an average compression ratio equals to 1.54 bpb for standard benchmark data except HUMHBB sequence. The DNASC algorithm also used to compressed the human genome.

4) Transformational Based Methods: Any sequence is subject to transformations before the actual compression takes place to achieve a better compression ratio such as Burrows Wheeler Transform (BWT). Based on the transformation from a DNA sequence to a codon sequence there are three methods that concentrate in the compression of biological sequences. Bao et al. presented a new algorithm based on fixed length LUT &LZ77[44]. First, they created a fixed size look up table (LUT) which contains a combination of three bases without 'N', 'space' or unknown nucleotide forming of 64 combinations called codons, resulting in a fixed size of the table. Also they met a series of successive N's, between two actual DNA strings in the destination file. Later these combinations were replaced by the symbols from lookup table resulting in encoding using LZ77 algorithm. Applying LZ77 to the pre-coded file improves the compression ratio. The authors compared with Chen's algorithm[32] and found the compression ratio of this algorithm is 0.2 bpbhigher in general. But the cost of the tiny 0.2 bpb the improvement was very high. This algorithm executesabout 103 times faster than DNAcompress[32].
The Differential Direct Coding algorithm (2D) by Vey in[45] suggests that compression strategies must accommodate large data sets &consist of multiple sequences & auxiliary data. The set of predictable symbols for the 2D model are (A, G, T, C, and U), which removes the load of explicit declaration of sequence type like DNA or RNA. The 2D model accommodates a total of 125 different triplets according to any of the nucleotide bases at any of the three triplet locations, such that the set of codons { AAC , AAA, , . . . , UUU, UUT }. These instances are accommodated in order to afford simplified arithmetic translation. Also, 128 different ASCII symbols are sustained as extra symbols and a single unknown flag is involved to denote a symbol that belongs to neither set. By using numerous bacterial genomes, the results display that gzip provided the best compression ratios while 2D had the fastest execution times. In 2011 Bharti and Singh[46] suggested a two phase compression algorithm based on a Look up Table (LUT) using a complementary palindrome of fixed size. During the first phase of the algorithm the algorithm searches for all palindromes in a precise length (3 Base Character). This is supported by checking all the probable places in the sequence. Then the algorithm finds a palindrome whichrelates with its demands & prints it to the output. In second phase they apply the LUT base variable length compression algorithm. The proposed algorithm has a high compression

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

ratio as compared to other existing Biological Sequence Compression algorithms, such as GenCompress [31], DNACompress[32], 2D coding[45], and Fixed length LUT[44]. Also it uses less memory as compared to the other existing algorithms and it is easy to implement. Later in 2012 Roy et al. proposed a finite LUT[47] in two algorithms using a four-step coding rule. The first step is based on the LUT, contains a combination of three characters among 'A', 'C, 'T', 'G' or 'N' (algorithm-1) and a combination of four characters ('A', 'C','T', and 'G') (algorithm-2). The second step is based on ASCII characters (in the form of algorithm-1 which is based on 125 chosen ASCII character 8 bits each and algorithm-2 which includes total 256 ASCII characters of 8 bit each. The third step is based on tandem repeats only for algorithm-1. The fourth step is based on a segment that consists of 1 or 2 characters.

5) Grammar Based Method: These algorithms infer context-free grammars to represent the input data. The grammar is then transformed into a symbol stream and finally encoded in binary.DNASequitur[48] is a grammar-based compression algorithm explored by Cherniavsky and Ladner for DNA sequences, which infers a context-free grammar to represent the input sequence. The novelty of DNASequitur is its ability to recognize reverse complements when creating rules and during substitutions beside the exact repeats. Results show that other methods achieve better compression ratios. Grammar-based compression is ideal for detecting repeats that occur across multiple sequences in a collection. Another advantage is the ability to support random access and search in the compressed collection. Therefore, grammar-based DNA compression should not be ruled out.

6) Two bits Based Methods: These algorithms make first bit-preprocessing by assigning 4 unique two bits (A = 00, G = 01, C = 10, and T = 11) to different four DNA bases before the encoding process. A DNA sequence is represented in smaller segments before the encoding process to compress both repetitive and non-repetitive DNA sequences. The input sequence is divided into fragments, where each fragment is four 8 bit-characters long. The following algorithms are similar in the bit-preprocessing stage but different in the coding stage. Rajeswari and Apparao proposed the GENBIT Compress tool[49] for DNA sequences based on a novel concept of assigning binary bits. After the bit-preprocessing stage in this coding scheme, 256 combinations can be represented. Hence every DNA segment containing four bases is replaced by an 8 bit binary number “XXXXXXXX”. If the consecutive fragments are the same, then a specific bit “1” is introduced as a 9th bit. If the consecutive fragments are different, then a specific bit “0” is introduced as a 9th bit to the 8 bit unique number. It takes an n-fragment long input of a DNA sequence, and divides into n/4 fragments. The left out individual bases (fragment length < 4) are assigned 4 unique "2" bits. The authors assume DNA sequences with a length of 64 bases to test this algorithm, not real biological sequences. They got a compression ratio for the best case (maximum repetitive fragments) of 1.125 bpb, while it was 1.727 bpb for the average case (random input which is not given by either the worst-case or the best-case efficiency), and 2.238 bpb for the worst case (there are no repetitive fragments), even for larger genomes (nearly 2,000,000 characters). Also, the GENBIT Compress tool program significantly improves the running time of all previous DNA compressors but expand the DNA sequence (with average = 2.23 bpb - standard benchmark data, except MPOMTCG, HEHCMVCG, HUMGHCSA and HUMHDABCD).
Rajeswari et al. introduced HUFFBIT compress[60] for DNA sequences. In this algorithm a bit-preprocessing stage to the sequence takes place. Then the extended binary tree principle is applied to derive a special class of variable length codes that satisfies the prefix property (Huffman Codes). The authors assume DNA sequences with length 1000 bases to test this algorithm, not real biological sequence. They got a better compression ratio than GENBIT. For the best case they got 1.006 bpb, for the average case they got 1.611 bpb, and for the worst case they got 2.109 bpb.
Rajeswari and Apparao presented the DNABIT Compress tool (DBC)[51] that assigns binary bits "in the bit-preprocessing stage" to exact and reverse repeats fragments of DNA sequences. This a unique concept introduced in this algorithm for the first time in DNA compression. DNABIT achieves the best compression ratio for DNA sequences for larger genomes. It significantly improves the running time and achieves better compression (with an average of 1.58 bpb - standard benchmark data, except MIPACGA and HUMGHCSA). Results show that DBC is the best among the remaining compression algorithms (Biocompress[29], CTW+LZ[27], GenCompress [31], DNA Compress[32], and DNAPack[4]). GenCodex[52] introduced by Satyanvesh et al. yields a better compression ratio at a high throughput by using graphical processing units (GPUs) and multi-cores in two phases.The compression ratio of GenCodex is better than GENBIT[49] and DNABIT[51] algorithms for the best and average cases. Results show that this method achieves

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

a good compression ratio along with a better throughput compared to other existing methods such as GENBIT[49], GenCompress[31], and DNA Compress[32]. Prasad and Kumar in the DNACRAMP tool[53] took a sequence of DNA for bit-preprocessing. Then they used basic procedural language to perform the encoding and decoding process with the help of a two-stage index bounded linear array data structure. This technique is applicable on repetitive and non-repetitive sequences of DNA. DNACRAMP obtains better compression ratios (with an average of 1.143 bpb - standard benchmark data) than DNABIT[51], DNAPack[4], CTW + LZ[27], and DNASC [3]. Prasad then introduced PGBC "Partitioned group binary compression"[54] that improves the worst-case scenario of the previous algorithms. These algorithms are far better than existing ones (e.g., HUFFBIT[60], GENBIT [49]) and suitable for non-repetitive DNA sequences of genomes (they encode every base by 1.33 bits in the worst-case scenarios). In addition to that, existing techniques use dynamic programming to compress the sequence which is complex in implementation and time consuming, but these algorithms are simple and fast.

B. Algorithms for DNA Compression in Vertical Mode

This mode uses the information between two sequences typically by making one of them a reference sequences. We can say that vertical mode compression nowadays is growing faster because of the development of new DNA sequencing technologies, such as next-generation sequencing (NGS). These new high-throughput sequencing technologies have led to a large number of whole genome DNA sequencing projects. Vertical mode compression is more suitable for sequences with large size. There has been considerable work on compression of sequencing data with some research in vertical mode DNA compression in many data formats e.g. FASTQ[55] and SAM/BAM[56] formats. Christley et al. present a DNAzippackage[57], which is a series of techniques that in combination reduces a single genome to a size small enough to be sent as an email attachment. Tembe et al. in[58] presented a Huffman coding-based sequencing which reads specific representation schemes that compress data without altering the relative order. Daily et al. in[59] developed data structures and compression algorithms for high-throughput sequencing data. A processing stage maps short sequences to a reference genome or a large table of sequences. Then the integers representing the short sequence's absolute or relative addresses, their length, and the substitutions they may contain, are compressed and stored using various entropy coding algorithms. Afify et al. in[60] exploited a differential compression model based on the alignment of two similarity sequences or more, which compress one sequence by comparing it with another sequence and using renewal entropy estimation. Grabowski and Deorowiczin[5] present an LZ77-style compression scheme for relative compression of multiple genomes of the same species to improve runtime and compression performance. Kuruppu et al. proposed the RLZ algorithm in[61] and the improved RLZ in[62], which are algorithms that provide effective compression in a single pass over the collection, and the final compressed representation allows rapid random access to arbitrary substrings using a simple greedy technique, akin to LZ77 parsing. The main difficulty of relative compression is in selecting an appropriate reference sequence. Kuruppu et al. in[8], explore using the dictionary of repeats generated by Comrad [63], Re-pair[64] and Dna-x[34] algorithms as applied to reference sequences for relative compression. Deorowicz and Grabowski present DSRC[65], which is a specialized compression algorithm for genomic data in FASTQ format and which dominates its competitor, G-SQZ.

In CRAM[66], Fritz et al. present a new reference-based compression method that efficiently compresses DNA sequences for storage. This approach works for re-sequencing experiments that target well-studied genomes.

Sakib et al. present SAMZIP[67], a specialized encoding scheme, for sequence alignment data in SAM format, which improves the compression ratio of existing compression tools available. Wang and Zhang present a novel compression tool for storing and analyzing genome re-sequencing data, named GRS[68]. Mark Howison addresses this gap by a new storage model called SeqDB[69], which offers high-throughput compression of sequence data with minimal sacrifice in compression ratio. Pinho et al. describe GReEn[70] (Genome Re-sequencing Encoding), a tool for compressing genome re-sequencing data using a reference genome sequence. It overcomes some drawbacks of the proposed tool GRS, Jones et al. present Quip[71], a lossless compression algorithm for next-generation sequencing data in the FASTQ and SAM/BAM formats using reference-based compression. Popitsch and Haeseler present NGC[72], a tool for the compression of mapped short read data stored in the wide-spread SAM format.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

C. Algorithms for DNA Compression Using Soft Computing Techniques

N. Qian and T.J. Sejnowski [73] present a new method for predicting the secondary structure of globular proteins based on non-linear neural network models. Network models learn from existing protein structures how to predict the secondary structure of local sequences of amino acids. The average success rate of our method on a testing set of proteins non-homologous with the corresponding training set was 643% on three types of secondary structure (u-helix, b-sheet, and coil), with correlation coefficients of C,=O-41, C,=O*31 and CcO,, =0*41. These quality indices are all higher than those of previous methods.SushmitaMitra and Yoichi Hayashi[74] survey the role of different soft computing paradigms, like fuzzy sets (FSs), artificial neural networks (ANNs), evolutionary computation, rough sets (RSes), and support vector machines (SVMs), in this direction.M.H. Aghdam, N. Ghasem-Aghaee and M.E. Basiri[75] intheir paper presents a novel feature selection algorithm that is based on ant colony optimization. Ant colony optimization algorithm is inspired by observation on real ants in their search for the shortest paths to food sources.Swagatam Das, Ajith Abraham, and Amit Konar[76] presents the perfect harmony of statistics, biology and computational intelligence methods for analyzing and processing biological information in the form of gene, DNA, RNA and proteins. SI algorithms on the other hand, have recently gained wide popularity among the researchers, for their amazing ability in finding near optimal solutions to a number of NP hard, real world search problems.Dr. Tryambak A. Hiwarkar, R. Sridhar Iyer [77] survey the role of different soft computing paradigms, like Fuzzy Sets (FSs), Artificial Neural Networks (ANNs), evolutionary computation, Rough Sets (RSs), and Support Vector Machines (SVMs), biologically inspired algorithm like ant colony system, swarm intelligence and others in bioinformatics systems and problems.

Mehdi HosseinzadehAghdamAndSetarehHeidari [78] presents a novel feature selection method based on particle swarm optimization to improve the performance of text categorization. Particle swarm optimization inspired by social behavior of fish schooling or bird flocking. The complexity of the proposed method is very low due to application of a simple classifier.Bhargavi and S. Jyothi [79] This paper specified the review of current analysis work involving soft computing techniques to categorize the DNA sequences. It has been examined that analysis of knowledge type connected elements of DNA sequence classification.

III. CONCLUSIONS

Research in bioinformatics largely depends on storage and manipulation of huge amounts of data. We believe that efficient DNA "the code of life" compression remains a challenging problem and a rather difficult task. In substitution algorithms searching for some kinds of repeats such as exact, reverse repeats, complemented repeats, and complemented palindromes, then encode using an appreciable algorithm, take more compression time. The Lempel-Ziv compression algorithm has become a reasonable default choice for compression of DNA. Most of statistical algorithms are effective for DNA sequence compression and computationally intensive in practice. Compression algorithms with combined substitution and statistical modules provide better results over substitution algorithms only.

Combining a lookup table pre-coding transformation making use of three or four bases with other compression algorithms improve its performance. Also it is easy to make sequence alignment and sequence analysis between compressed sequences. Grammar-based compression is ideal for detecting repeats and has the ability to support random access and search in the compressed collection but with poor compression ratio. In the future, we may explore the use of edit grammars for DNA sequences such as insert, delete, or replace a character with another to improve the compression ratio. Bit-based compression is compressing both repetitive and non-repetitive DNA sequences. It is implemented without dynamic programming, so it is simple and fast. Finally, vertical mode compression provides an efficient storage model for the data produced by new DNA sequencing technologies, such as next-generation sequencing. Combining vertical mode compression and statistical compression may improve the compression ratio by determining other factors to choose a reference sequence.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

REFERENCES

[1] Pinho, A. J., Neves, A. J. R., Afreixo, V., et al., 2006, A Three-State Model for DNA Protein-Coding Regions, IEEE Trans. Biomed Eng., 53(11), 2148 –2155.
[2] Korodi, G., Tabus, I., Rissanen, J., et al., 2007, DNA Sequence Compression Based on the normalized maximum likelihood model, Signal Processing Magazine, IEEE, 24(1), 47-53.
[3] Mishra, K. N., Aaggarwal, A., Abdelhadi, E., et al., 2010, An Efficient Horizontal and Vertical Method for Online DNA Sequence Compression, International Journal of Computer Applications, 3(1), 39-46.
[4] A. Postolico, et al., Eds., DNA Compression Challenge Revisited: A Dynamic Programming Approach, Lecture Notes in Computer Science, Island, Korea: Springer, 2005, vol. 3537, 190–200.
[5] Deorowicz, S., and Grabowski, S., 2011, Robust relative compression of genomes with random access, Bioinformatics, 27(21), 2979–2986.
[6] Horner, D. S., Pavesi, G., Castrignanò, T., et al. , 2010, Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing, Briefings in Bioinformatics, 11(2), 181–197.
[7] Pushkarev, D., Neff, N. F., and Quake, S. R., 2009, Single-molecule sequencing of an individual human genome, Nature Biotechnology, vol. 27, 847–852.
[8] R. Grossi et al., Eds., Reference Sequence Construction for Relative Compression of Genomes, Lecture Notes in Computer Science, Pisa, Italy: Springer, 2011, vol. 7024,
[9] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., et al., 2005, GenBank, Nucleic Acids Research, vol. 33, 34-38.
[10] Brooksbank, C., Cameron, G., and Thornton, J., 2010, The European Bioinformatics Institute's data resources, Nucleic Acids Research, vol. 38, 17-25.
[11] Sugawara, H., Ogasawara, O., Okubo, K., et al., 2008, DDBJ with new system and face, Nucleic Acids Research, vol. 36, 22-24.
[12] Shumway, M., Cochrane, G., and Sugawara, H., 2010, Archiving next generation sequencing data, Nucleic Acids Research, vol. 38, 870-871.
[13] Barrett, T., Troup, D. B., Wilhite, S. E., et al. , 2009, NCBI GEO: archive for high-throughput functional genomic data, Nucleic Acids Research, vol. 37, 885-890.
[14] Kapushesky, M., Emam, I., Holloway, E., et al. , 2010, Gene expression atlas at the European bioinformatics institute, Nucleic Acids Research, 38(1), 690-698.
[15] Ahmed A., Hisham G., Moustafa G., et al., 2010, EGEPT: Monitoring Middle East Genomic Data, Proc., 5th Cairo International Biomedical Engineering Conf., Egypt, 133-137.
[16] Karsch-Mizrachi, I., Nakamura, Y., and Cochrane, G., 2012, The International Nucleotide Sequence Database Collaboration, Nucleic Acids Research, 40(1), 33–37.
[17] International nucleotide sequence database collaboration, (2013),[Online]. Available: http://www.insdc.org.
[18] Celniker, S. E., Dillon, L. A. L., Gerstein, M. B., et al. , 2009, Unlocking the secrets of the genome, Nature, 459(7249), 927–930.
[19] Guttmacher, A. E., and Collins, F. S., 2005, Realizing the promise of genomics in biomedical research, JAMA: Journal of the American Medical Association, 294(11), 1399–1402.
[20] Joosen, R. V., Ligterink, W., Hilhorst, H. W., et al., 2009, Advances in genetical genomics of plants, Current Genomics, 10(8), 540–549.
[21] Womack, J. E., 2005, Advances in livestock genomics: opening the barn door, Genome Research, 15(12), 1699–1705.
[22] Genbank size, (2013),[Online]. Available:http://ftp.ncbi.nih.gov/genbank/gbrel.txt
[23] S. Grumbach and F. Tahi, "Compression of DNA Sequences," in Proc. of the Data Compression Conf., (DCC '93), 1993, 340–350.
[24] Giancarlo, R., Scaturro, D., and Utro, F., 2009, Textual data compression in computational biology: a synopsis, Bioinformatics, 25(13), 1575–1586.
[25] Nalbantoğlu, Ö. U., Russell, D.J., and Sayood, K., 2010, Data Compression Concepts and Algorithms and their Applications to Bioinformatics, Entropy, 12(1), 34-52.
[26] Matsumoto, T., Sadakane, K., Imai, H., et al., 2000, Can General-Purpose Compression Schemes Really Compress DNA Sequences?, Computational Molecular Biology, Universal Academy Press, 76–77.
[27] Matsumoto, T., Sadakane, K., and Imai, H., 2000, Biological Sequence Compression Algorithms, Genome Informatics, vol. 11, 43–52.
[28] Sato, H., Yoshioka, T., Konagaya, A., et al., 2001, DNA Data Compression in the Post Genome Era, Genome Informatics, vol. 12, 512–514.
[29] Grumbach, S., and Tahi, F., 1994, A new challenge for compression algorithms: genetic sequences, Information Processing & Management, 30(6), 875–886.
[30] E. Rivals, M. Dauchet, J-P. Delahaye, et al., "Fast Discerning Repeats in DNA Sequences with a Compression Algorithm," The 8th Workshop on Genome and Informatics, (GIW97), 1997, vol. 8, 215-26.
[31] X. Chen, S. Kwong and M. Li, "A Compression Algorithm for DNA Sequences and It's Applications in Genome Comparison," The 10th Workshop on Genome and Informatics, (GIW99), 1999, vol. 10, 51-61.
[32] Chen, X., Li, M., Ma, B., et al., 2002, DNACompress: fast and effective DNA sequence Compression, Bioinformatics, 18(12), 1696–1698.
[33] Ma, B., Tromp, J. and Li, M., 2002, Pattern Hunter: faster and more sensitive homology search, Bioinformatics, 18(3), 440–445.
[34] Manzini, G., and Rastero, M., 2004, A Simple and Fast DNA Compressor, Software: Practice and Experience, 34(14), 1397–1411.
[35] A. J. Pinho, A. J. R. Neves, D. A. Martins, et al., Finite-Context Models for DNA Coding, Signal Processing Lab, DETI/IEETA, S. Miron , Ed., University of Aveiro, Portugal, Chapter 6, 117-130, 2010.
[36] A. J. T. Lee, C. Chang and C. Chen, "DNAC: An Efficient Compression Algorithm for DNA Sequences," National Taiwan University, Taipei, Taiwan 10617, R.O.C., 2004.
[37] D. Loewenstern, and P. N. Yianilos, "Significantly lower entropy estimates for natural DNA sequences," in Proc. of the Data Compression Conf., (DCC '97), 1997, 151–160.
[38] Allison, L., Edgoose, T., and Dix, T. I., 1998, Compression of strings with approximate repeats, Proc. ISMB, 8–16.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

[39] M. D. Cao, T. I. Dix, L. Allison, et al., "A Simple Statistical Algorithm for Biological Sequence Compression," in Proc. of the Data Compression Conf., (DCC '07), 2007, 43–52.
[40] A. Apostolico, and S. Lonardi, Compression of biological sequences by Greedy Off-Line Textual Substitution, in Proc. of the Data Compression Conf., (DCC '00), 2000, P. 143.
[41] Apostolico, A., and Lonardi, S., 2000, Off-Line Compression by Greedy Textual Substitution, Proc. IEEE, 88(11), 1733–1744.
[42] I. Tabus, G. Korodi, and J. Rissanen, "DNA sequence compression using the normalized maximum likelihood model for discrete regression," in Proc. of the Data Compression Conf. (DCC2003), 2003, 253–262.
[43] Korodi, G., and Tabus, I., 2005, An Efficient Normalized Maximum Likelihood Algorithm for DNA Sequence Compression, ACM Trans. on Information Systems, 23(1), 3–34.
[44] Bao, S., Chen, S., Jing, Z., et al., 2005, A DNA Sequence Compression Algorithm Based on LUT and LZ77, Signal Processing and Information Technology, Proc., 50th IEEE International Symposium, 23–28.
[45] Vey, G., 2009, Differential direct coding: a compression algorithm for nucleotide sequence data, Database, Oxford University Press, vol. 2009, ID bap013.
[46] Bharti, R. K., and Singh, R. K., 2011, A Biological Sequence Compression based on Look up Table (LUT) using Complementary Palindrome of Fixed Size, International Journal of Computer Applications, 35(11), 55-58.
[47] Roy, S., Khatua, S., Roy, S., et al., 2012, An Efficient Biological Sequence Compression Technique Using LUT and Repeat in the Sequence, IOSR Journal of Computer Engineering (IOSRJCE), 6(1), 42-50.
[48] N. Cherniavsky and R. Ladner, "Grammar-based Compression of DNA Sequences," UW CSE Technical Report (TR2007-05-02), presented at the DIMACS Working Group, 2004.
[49] 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.
[50] Rajeswari, P. R., Apparao, A., and Kumar, R. K., 2010, HUFFBIT COMPRESS – Algorithm to compress DNA sequences using extended binary tree, Journal of Theoretical and Applied Information Technology, 13(2), 101-106.
[51] Rajeswari, P. R., and Apparao, A., 2011, DNABIT Compress – Genome compression algorithm, Bioinformation, 5(8), 350-360.
[52] 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.
[53] 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.
[54] 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.
[55] Cock, P. J. A., Fields, C. J., Goto, N., et al., 2010, The sanger FASTQ format for sequences with quality scores, and the Solexa/fillumina FASTQ variants, Nucleic Acids Research, 38(6), 1767–1771.
[56] Li, H., Handsaker, B., Wysoker, A., Fennell, T., et al., 2009, The Sequence Alignment/Map format and SAMtools, Bioinformatics, 25 (16), 2078-2079.
[57] Christley, S., Lu, Y., Li, C., et al., 2009, Human genomes as email attachments, Bioinformatics, 25(2), 274-275.
[58] Tembe, W., Lowey, J., and Suh, E., 2010, G-SQZ: compact encoding of genomic sequence and quality data, Bioinformatics, 26(17), 2192-2194.
[59] Daily, K., Rigor, P., Christley, S., et al., 2010, Data structures and compression algorithms for high-throughput sequencing technologies, BMC Bioinformatics, vol. 11.
[60] Afify, H., Islam, M., Abdel-Wahed, M., et al., 2010, Genomic Sequences Differential Compression Model, Proc., 27th National Radio Science Conf., Egypt.
[61] E. Chavez and S. Lonardi, Eds., Relative Lempel-Ziv Compression of Genomes for Large-Scale Storage and Retrieval: Springer 2010, vol. 6393, 201–206.
[62] S. Kuruppu, S. J. Puglisi, and J. Zobel, "Optimized Relative Lempel-Ziv Compression of Genomes," in Proc. Australasian Computer Science Conf. (ACSC'11), 2011, vol. 113, 91-98.
[63] Kuruppu, S., Smith, B. B., Conway, T., et al., 2012, Iterative dictionary construction for compression of large DNA datasets, Computational Biology and Bioinformatics, IEEE/ACM, 9(1), 137–149.
[64] N. J. Larsson, and A. Moffat, "Offline dictionary-based compression," in Proc. Data Compression Conf. (DCC'99), 1999, 296-305.
[65] Deorowicz, S., and Grabowski, S., 2011, Compression of genomic sequences in FASTQ format, Bioinformatics, 27(6), 860-862.
[66] Fritz, M. H., Leinonen, R., Cochrane, G., et al., 2011, Efficient storage of high throughput DNA sequencing data using reference-based compression, Genome Research, vol. 21, 734-740.
[67] Sakib, M. N., Tang, J., Zheng, W. J., et al., 2011, Improving Transmission Efficiency of Large Sequence Alignment/Map (SAM) Files, PLOS ONE, 6(12), e28251.
[68] Wang, C., and Zhang, D., 2011, A novel compression tool for efficient storage of genome resequencing data, Nucleic Acids Research, 39(7), e45.
[69] Howison, M., 2013, High-Throughput Compression of FASTQ Data with SeqDB, Computational Biology and Bioinformatics, IEEE/ACM, 10(1), 213 – 218.
[70] Pinho, A. J., Pratas, D., and Garcia, S. P., 2012, GReEn: a tool for efficient compression of genome resequencing data, Nucleic Acids Research, 40(4), e27.
[71] Jones, D. C., Ruzzo, W. L., Peng, X., et al., 2012, Compression of next-generation sequencing reads aided by highly efficient de novo assembly, Nucleic Acids Research, 40(22), e17.
[72] Popitsch, N. and Haeseler, A., 2013, NGC: lossless and lossy compression of aligned high-throughput sequencing data, Nucleic Acids Research, 41(1), e27.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

[73] N. Qian and T.J. Sejnowski; "Predicting the secondary structure of globular proteins using neural network models", J. Mol. Biol., Vol. 202(4), pp. 865-884, 1988.
[74] SushmitaMitra and Yoichi Hayashi; "Bioinformatics with Soft Computing", IEEE transactionson systems, man, and cybernetics—part c: Applications and Reviews, Vol. 36(5), September2006.
[75] M.H. Aghdam, N. Ghasem-Aghaee and M.E. Basiri; "Application of Ant Colony Optimizationfor Feature Selection in Text Categorization", Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2872-2878, 2008.
[76] Swagatam Das, Ajith Abraham, and Amit Konar; "Swarm Intelligence Algorithms in Bioinformatics", (SCI) 94, pp. 113-147, Springerlink., 2008.
[77] Dr. Tryambak A. Hiwarkar, R. Sridhar Iyer,New Applications of Soft Computing, Artificial Intelligence, Fuzzy Logic & Genetic Algorithm in Bioinformatics, International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 5, May 2013, pg.202 – 207
[78] Mehdi Hosseinzadeh Aghdam1 And Setareh Heidari2, Feature Selection Using Particle Swarm Optimization In Text Categorization, JAISCR, 2015, Vol. 5, No. 4, Pp. 2 31-238, 10.1515/Jaiscr-2015-0031
[79]K. Bhargavi and S. Jyothi,"Classification of DNA Sequence Using Soft Computing Techniques: A Survey", Indian Journal of Science and Technology, Vol 9(47), DOI: 10.17485/ijst/2016/v9i47/89343, December 2016.