EAI Endorsed Transactions
on Pervasive Health and Technology
Research Article EAI.EU

Design and Development of Bioinformatics Feature Based DNA Sequence Data Compression Algorithm

Kakoli Banerjee[1,*] and Vikram Bali[1]

[1]JSS Academy of Technical Education, Noida, India

Abstract

INTRODUCTION: Genetic data plays a key role in the healthcare area in specific, but they are typically very large in size. Many research shows that absence of DNA information at the right time is one of the major causes of error in the healthcare area. The more genomics information that analysts secure, the better the prospects for individual and general wellbeing. Persevering and retrieving genetic information in the right form within the given time is a big challenge in the field of Healthcare. Effectively, pre-birth DNA tests screen for formative variations from the norm. Before long, patients will have their blood sequenced to detect any nonhuman DNA that may flag an irresistible illness. Later on, somebody managing malignancy will most likely track the movement of the sickness by having the DNA and RNA of single cells from various tissues sequenced every day. DNA sequencing of whole population will give a complete and better prediction of population wellbeing.
OBJECTIVES: Hereditary data is growing exponentially; hence it is hard to deal with the consistently developing hereditary database. The human genome in its base configuration occupies almost thirty terabyte of storage space. Computational assets are constrained. Not just storage, transmission abilities and run time memory is likewise constrained. Data Compression is a test when the hereditary information is exponentially expanding. It is critical to save the integrity of hereditary information while packing it. Hence the main objective of this paper is to develop a lossless DNA compression algorithm that not only gives better compression but also help in retrieval of Information for efficient use in the area of Healthcare.
METHODS: In this paper a lossless hereditary data compression method is being proposed. The proposed calculation works in a horizontal mode and utilization a reference based substitution technique for compression. The principle thought of this paper is in the kind of similarity scanned. All the predominant hereditary Compression methods search for similarity within the chromosome. These calculations either pursue flat mode or vertical mode for accomplishing compression. But whichever method the existing genetic compression algorithms use, they are all based on searching similarities within the chromosome i.e. they exploit only inter chromosomal similarities. The current studies focus will show that compression ratio achieved by analyzing individual chromosome is always less than the method in which we analyze and compress intra chromosomal similarities.
RESULTS: This study shows that by simply using exactly matching repeats amongst all the chromosomes of the same genome, not only the compression ratio is improving but also a detailed study of all the similarities and differences between two genomes of the same species can be conducted.
CONCLUSION: In this study, a new compression algorithm is being proposed for compressing DNA. Along with Inter chromosomal similarities, Intra chromosomal similarities are considered for this method. The results clearly shows that intra chromosomal matches are bigger and more than inter chromosomal matches which helps us to achieve better compression ratio.

Keywords: Genetic Data, Genetic Data Compression, DNA, Health Care, Compression Algorithms.

Received on 01 October 2019, accepted on 30 October 2019, published on 13 November 2019

doi: 10.4108/eai.13-7-2018.164097

*Corresponding author. Email: Kakolibanerjee80@gmail.com

1. Introduction

Need of Compression in Health Care Industry

As the Health care industry is shifting from the classical methods to more prediction based methods [16-19], the need of storing biological information efficiently is becoming a more critical issue. One of the most important biological information is the DNA sequences. By the detailed study of DNA Sequences mutating over a period of time, many diseases can be predicted and stopped from occurring. Biological sequences are like blue prints of all living organisms. Special biological sequences like DNA are functional description of cells that are the basic building units of each organism. Research shows that DNA information is a very important Data Set, in the Health Care Industry. Getting the right data at the right time is of utmost importance. But as the genomic database is increasing exponentially, storing and retrieving the data is becoming a challenge. Hence there is requirement of such type of Genetic Data Compression Algorithm, which can yield best compression ratio and also retrieve the data in minimal time.

DNA

DNA successions are made of four chemical bases to be specific – Adenine (A), Thymine (T), Guanine (G) and Cytosine(C). The human DNA itself comprises of three billion such bases. The extraordinary information bases like GenBank which stores hereditary data shared by analysts everywhere throughout the world, doubles itself every ninety months. As effectively expressed computational assets are constrained.

Genetic Data Set Used

This research focuses on the genome of Saccharomyces Cerevisiae (Baker's Yeast) also known as Budding Yeast. The budding yeast Saccharomyces cerevisiae is one of the major model organisms for understanding cellular and molecular processes in eukaryotes. This single-celled organism is also important in industry, where it is used to make bread, beer, wine, enzymes, and pharmaceuticals. The Saccharomyces cerevisiae yeast genome is organized in 16 chromosomes.

In this research, the study and compression of these 16 chromosomes have been carried out. But this compression algorithm is not restricted to only Saccharomyces Cerevisiae genome, it can be used to compress any chromosomal sequence of a given genome.

What is Data Compression

Capacity is restricted, so are runtime memory and transmission abilities. With these constrained assets, the treatment of such exponentially developing information is a test. In this situation the main thing that comes as a main priority is – "Compression". What is Compression – an unavoidable issue to think about – is it simply lessening the size information. No. Compression is substantially more than that. Compression is "Modeling + Coding". Modeling is the place we discover diverse kind of techniques to discover repetition in information and coding is the place we supplant these redundancies by some sort of references. Subsequently for dealing with such enormous volume information Compression is must.

Issue till now

The main issue is that, regardless of whether the standard compression algorithm can deal with such unique hereditary information. The standard compression algorithms flop in compacting the DNA successions as well as wind up yielding negative compression proportion.

The standard pressure calculations don't be able to misuse the unique qualities of DNA groupings, and this is the motivation behind why the size of the packed record is more than that of the first document. There are some unique qualities that DNA arrangements hold, which can be misused to get positive pressure.

Need of Compression

Compression and analysis of DNA sequences is very important as it can result into better and more customized medical treatment, disease diagnosis and finding drug based solutions. Compression not only helps in efficient storage and retrieval but also in the process of querying, transfer, comparison and analysis.

The main aim of this paper is not only DNA compression but also to find out the commonness between two different chromosomes.

Table 1, Figure 1 and Table 2, Figure 2 demonstrates the exponential growth of the GenBankdatabase in terms of sequences and base pairs.

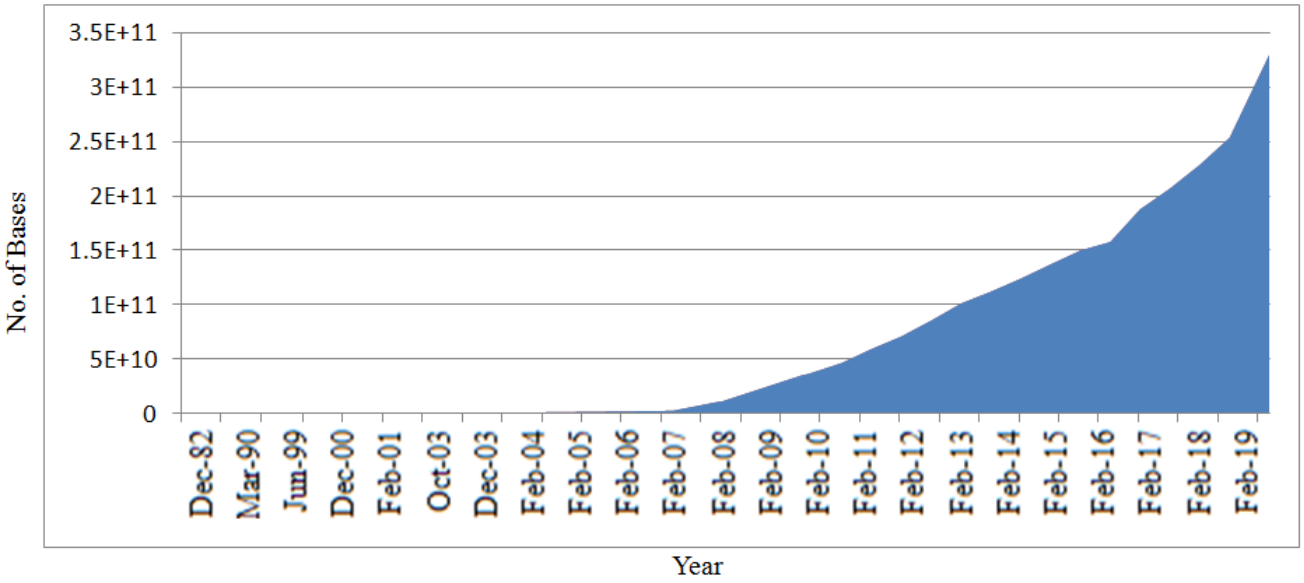

Figure 1. The Exponential Growth Of Genetic Bases in the Genbank Database
Source -http://www.ncbi.nlm.nih.gov/genbank/statistics

Table 1. The Exponential Growth of Genetic Bases in the GenBank Database
Source -http://www.ncbi.nlm.nih.gov/genbank/statistics

Year	No. of Bases
December-82	680338
March-90	40127752
June-99	2974791993
December-00	11101066288
February-01	11720120326
October-03	35599621471
December-03	36553368485
February -04	37893844733
February -05	46849831226
February -06	59750386305
February -07	71292211453
February -08	85759586764
February -09	101467270308
February -10	112326229652
February -11	124277818310
February -12	137384889783
February -13	150141354858
February -14	157943793171
February -15	187893826750
February -16	207018196067
February -17	228719437638

February -18	253630708098
February -19	329835282370

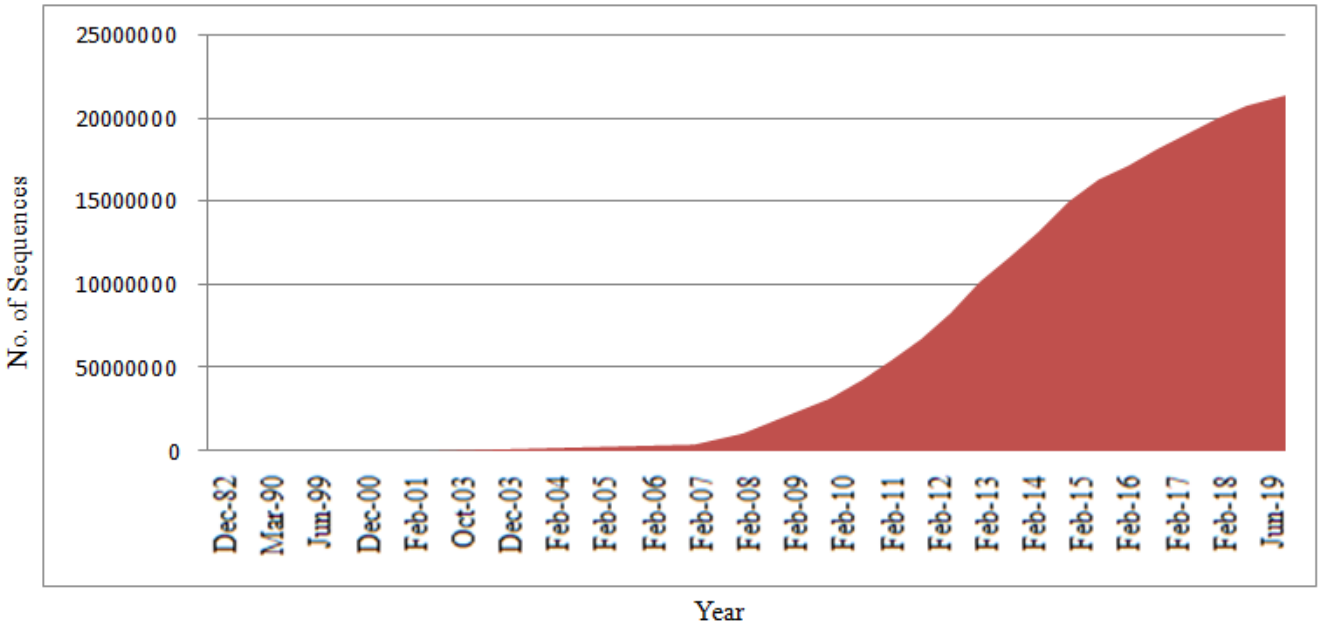

Figure 2. The Exponential Growth of Genetic Sequences in the GenBank Database
Source -http://www.ncbi.nlm.nih.gov/genbank/statistics

Table 2. The Exponential Growth of Genetic Sequences in the GenBank Database
Source -http://www.ncbi.nlm.nih.gov/genbank/statistics

Year	Sequences
December-82	606
March-90	33377
June-99	4028171
December-00	10106023
February -01	10896781
October-03	29819397
December-03	30968418
February -04	32549400
February -05	42734478
February -06	54584635
February -07	67218344
February -08	82853685
February -09	101815678
February -10	116461672
February -11	132015054
February -12	149819246
February -13	162886727

February -14	171123749
February -15	181336445
February -16	190250235
February -17	199341377
February -18	207040555
June-19	213383758

The above statistics shows the exponential growth of genetic database and this is why we need compression.

2. Literature Survey

Compression of a DNA fragment is a challenging task for current compression algorithms since these algorithms are primarily intended for compression of English text, whereas the observable behaviors in DNA samples are low [20-40]. There are four bases in DNA sequences { G, C, T, A}. Thus, two bits can represent each base. These DNA sequences can not be compressed by conventional text compression instruments such as compress, gzip and bzip2.

Thai and Grumbach introduced the first hereditary Compression Algorithm, which were known as BioCompress and its subsequent adaptation BioCompress-2 [4]. BioCompress and its subsequent adaptation BioCompress-2 utilize the LZ Compression methods. BioCompress-2 accompanied the additional component of looking through exact repeats in 2010. The repeats were encoded by rehash length and event position. The rest of the non repeat districts are encoded utilizing Arithmetic-2 Method.

The next in the row is Cfact, It's a two part calculation. and it uses a postfix tree to find the biggest accurate match [3]. In the principal part the biggest accurate match is found and in the second part the matches are coded.

GenCompress-1 and GenCompress-2 were released in 2010. GenCompress-2 utilizes addition, erasure and replacement for encoding the rehashes. GenCompress-1 utilizes the strategy of hamming distance or replacement just for matches [1].

DNABIT was presented in 2011. It is likewise a dual stage calculation. In the main stage paired bits are allotted to each nucleotide and in the next stage aged piece strategy was utilized which replaced 3,5,7 and 9 bits dependent on the length of matches [5].

CTW+LZ was presented with a combination of CTW + LZ77 [2]. The calculation utilized the context tree weight for encoding long matches though LZ77 was utilized to code short matches. It was introduced in 2012.

A two-phase lossless chromosome compression algorithm that shows supplementary qualitative design synthesis to enhance compression efficiency [13] was introduced, in 2013. DNA-COMPACT was a algorithm in which the suggested structure could manage compression of genetic code with and with-out referred frames and showed efficiency benefits over best current algorithms.

In 2014, SEQCOMPRESS was introduced in which the compression algorithm copes with genetic pattern spatial complexity. The algorithm is focused on lossless compression of information and utilizes both mathematical model and calculation coding to encode Chromosome sequences [14]. In 2018 Fatigue Detection of Workers using Supervised Learning was suggested. [15] [16]

An Ideal Seed Based Compression Algorithm for DNA data was launched in 2016, offering a seed-based lossless compression algorithm to compress a DNA pattern using a compression-like substitute technique from LZ [17].

3. Observation

The test results demonstrate that when the calculation scans for intra chromosomal matches – the quantity of matches and the length of matches increase. In the compression procedure, the algorithm is going to look for precise and inexact rehashes. Saccharomyces Cerevisiae genome has sixteen chromosomes. For discovering investigative outcomes, the Saccharomyces Cerevisiae (Budding Yeast) genome is being used in this paper. First the examination begins with discovering definite matches in chromosome I and VIII independently and after that breaking down them two together to discover the intra chromosomal similitudes too. For this examination chromosome I and chromosome VIII are being investigated as Chromosome I indicates most extreme likeness with Chromosome VIII.

Table 3. S. cerevisiae (Brewer's Yeast) First Chromosome's Repeats Having 100% Similarity

Length's taken under consideration	Total Repeats	Repeat with Longest Length	Repeat with Smallest Length	Repeat Regions
100 bases	58 repeats	337 bases	101 bases	29 regions

500 bases	Nil	Nil	Nil	Nil
1000 bases	Nil	Nil	Nil	Nil

Table 4. S. cerevisiae (Brewer's Yeast) Eighth Chromosome's Repeats Having 100% Similarity

Length's taken under consideration	Total Repeats	Repeat with Longest Length	Repeat with Smallest Length	Repeat Regions
100 bases	32 repeats	1988 bases	105 bases	16 regions
500 bases	2 repeats	1988 bases	1988 bases	1 region
1000 bases	2 repeats	1988 bases	1988 bases	1 region

Table 5. S. cerevisiae (Brewer's yeast) First and Eighth Chromosome's Repeats Having 100% Similarity

Length's taken under consideration	Total Repeats	Repeat with Longest Length	Repeat with Smallest Length	Repeat Regions
100 bases	206 repeats	3232 bases	101 bases	103 regions
500 bases	28 repeats	3232 bases	503 bases	14 regions
1000 bases	12 repeats	3232 bases	1011 bases	6 regions

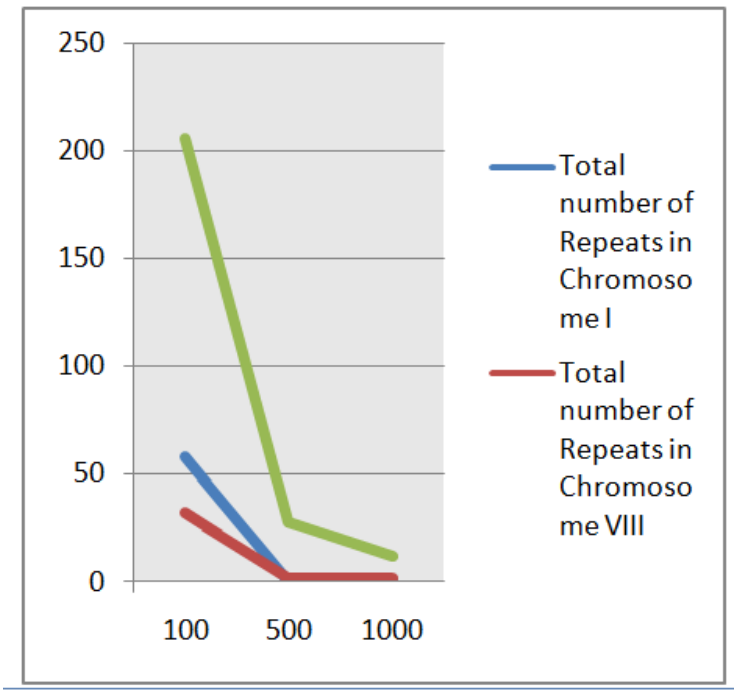

Figure 5. Enhancement in the Amount of Repeats when Searching or Self- and Cross-Chromosomal Similarities.

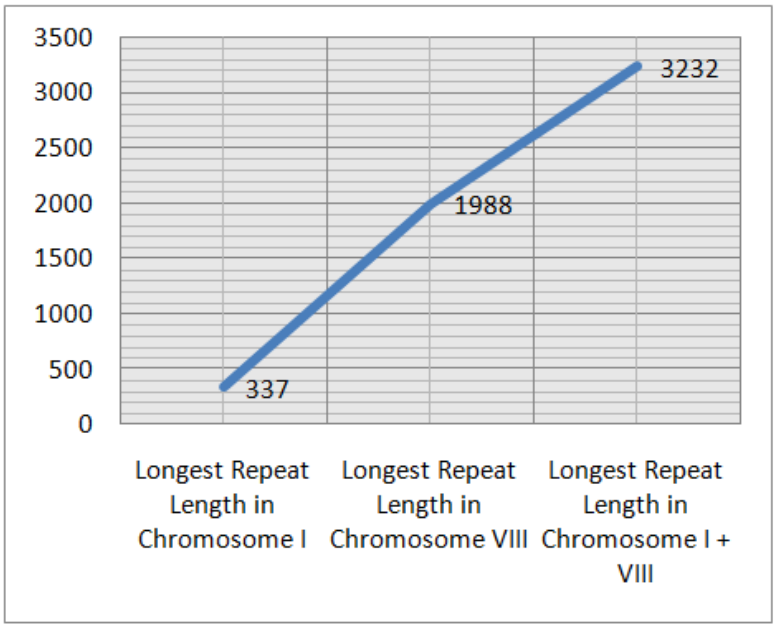

Figure 6. Enhancement in Repeat Length when Searching for Self- and Cross-Chromosomal Similarities

4. Methodology

The process starts with search for exact matches of base 100 and 100% similarity.

The complete number of bases that fall under these matches are 7860, which are being supplanted by 29 eight piece ASCII code. That implies 62880 bits will be supplanted by 232 bits. The primary chromosome is comprised of 230119 bases. After analysis, 58 matches are discovered, that implies they can be supplanted by 29 eight piece ASCII code. Exploratory outcomes demonstrate that various matches take place between the lengths of 101-337. We have 74087 codon. For coding these 74087 codons (beginning Size – 1778072 bits) just 444518 bits are required. Without compression each base involves 8 bpb. While after compression each base possesses 1.98 bpb. The remaining 222259 bases are converted into three base codon and every codon is supplanted by six piece binary code.

In chromosome I+VIII, 206 matches are found. The non matching 732112 bases are changed over into codons and every codon is supplanted by a six piece binary code. The absolute number of bases between the length of 101-3232 are 60440, which are supplanted by 16 eight piece ASCII code. Chromosome I+VIII is compression to 1.84bpb.

In chromosome VIII, 32 matches are found. The non matching 554511 bases are changed over into codons and every codon is supplanted by a six piece binary code. The bases between the length of 105-1988 are 7912, which are supplanted by 16 eight piece ASCII code. It is compressed to 1.97.

5. Development Model/ Algorithm Developed for Compression

The basic idea behind this research paper is to create a comp ression algorithm that can successfully compress the sequen ce of DNA and offer a comparatively better compression rati o than other existing algorithms and provide the much-needed partial decompression function as well.

The design of the compression algorithm has been divided into four different phases.

(i) Phase 1 – Search for Inter and Intra Chromosomal Repeats.
(ii) Phase 2 – Create a Mature Dictionary and Assigning ASCII codes to each Repeats. Also Create a CODON Dictionary.
(iii) Phase 3 – Compression Phase I – Repeat area are detected and replaced with ASCII Codes of the Mature Dictionary
(iv) Phase 4 – Compression Phase II – Non repeat areas which can also be called Non coding Areas are replaced with CODON Codes.

6. Development Model/ Algorithm Developed for Decompression

The design of the Decompression algorithm has been divided into two different phases.

- Phase 1 – Decompression Phase I – Decompression where complete sequence was decompressed in one go.
- Phase 2 – Decompression Phase II – Decompression where only a part of the compressed Sequence was decompressed.

7. The Mathematical Results

The following table shows a comparison of Intra Vs Inter Chromosomal Repeats, which clearly shows the improvement in the number of repeats.

Table 6. Comparison of Intra Vs Inter Chromosomal Repeats of Saccharomyces Cerevisiae S288C, which clearly shows the Improvement in the Number of Repeats.

SEQ	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI
I	**58**	100	118	484	178	62	238	206	128	124	72	328	184	128	144	218
II	100	**36**	178	394	134	120	134	104	136	286	48	394	306	156	192	262
III	118	178	**48**	416	162	88	182	90	124	152	72	378	210	114	210	212
IV	484	394	416	**352**	526	460	514	436	430	418	372	618	444	460	492	570
V	178	134	162	526	**62**	118	324	246	166	190	88	404	216	166	202	344
VI	62	120	88	460	118	**2**	96	62	96	96	14	264	152	116	124	134
VII	238	134	182	514	324	96	**88**	222	196	214	110	390	184	212	286	390
VIII	206	104	90	436	246	62	222	**32**	124	170	70	316	168	116	146	266
IX	128	136	124	430	166	96	196	124	**62**	134	78	274	178	130	140	204
X	124	286	152	418	190	96	214	170	134	**50**	66	422	270	178	276	252
XI	72	48	72	372	88	14	110	70	78	66	**12**	222	110	48	82	108
XII	328	394	378	618	404	264	390	316	274	422	222	**198**	478	396	364	438
XIII	184	306	210	444	216	152	184	168	178	270	110	478	**80**	218	196	250
XIV	128	156	114	460	166	116	212	116	130	178	48	396	218	**18**	196	172
XV	144	192	210	492	202	124	286	146	140	276	82	364	196	196	**66**	242
XVI	218	262	212	570	344	134	390	266	204	252	108	438	250	172	242	**94**
SUM	2770	2980	2754	7386	3526	2004	3780	2774	2600	3298	1572	5884	3644	2824	3358	4156

The current study not only focuses on development of a novel DNA compression Algorithm, but also aims at improving the compression ratio of already existing algorithms. As already concluded – "More number of repeats better is the compression". The Genetic compression Algorithms that have been developed till now and discussed in the literature survey, don't use Inter Chromosomal Similarities. All the Algorithms have been using only intra Chromosomal repeats till now. The novelty of this algorithm is in the use of Inter Chromosomal Repeats along with Inta Chromosomal Repeats of the same Genome. Table Number 6 shows the huge difference between Intra and Inter Chromosomal Repeats in number. This study is not only end up developing a better compression algorithm but will also change the approach of already existing repeat based DNA compression algorithms, hence forth improve the compression ratio of all existing algorithms.

8. Implication of Data and Findings

The proposed technique is not only going to improve the compression ratio of this algorithm but can improve the compression ratio of other existing algorithms also.

In the proposed technique the mature dictionary is created only for inter and intra chromosomal exact repeats. The compression ratio can further improve if other features like approximate repeat tandem repeats and palindromes of DNA sequences are also exploited.

Hence it can be concluded that in the proposed technique inter and intra chromosomal exact repeats have been used for creating a mature dictionary, which in turn is going to compress and decompress the DNA sequence. This method is giving more than 75% of compression and adding the feature of Random Access and partial decompression. The compression ratio can further be improved if other features of DNA sequences are also used.

9. Conclusion

In this paper, a new compression algorithm is being proposed for solving DNA sequence compression problem. Another idea is being presented in this paper. This idea is about intra chromosomal similarities and sub strings. The numerical calculation results demonstrate that intra chromosomal matches are bigger and more than inter chromosomal matches. For the current paper, only uniquely exact matches are considered. This work can be stretched out for rough matches for better compression. This work can be utilized for successfully compress DNA arrangements and can likewise be utilized to discover the degree of comparability between two unique chromosomes of a similar genome.

For Inter Chromosomal Sequences, this compression algorithm achieves more than 75% compression ratio by assigning ASCII values to the repeats and CODONS found in the original database and swapping them to generate compressed sequence. During compression, the input files stores the data sequence and assign ASCII from the dictionary table. The file can be accessed from client side and server side as well. Partial Decompression can be done by accessing random string from the Database Dictionary. The input file should contain a starting space then ASCII code and then the ending space in order to distinguish the repeats and decompress.

The compression algorithm proposed in this research work does not compress biological sequences randomly. This algorithm requires the complete genome data with all the chromosomes to prepare the mature dictionary. Complete compression of the sequences can take place only if the mature dictionary is ready for the replacement process.

References

[1] F. Claude, A. Farina, M. Martínez-Prieto and G. Navarro, "Compressed q-Gram Indexing for Highly Repetitive Biological Sequences", 2010 IEEE International Conference on BioInformatics and BioEngineering, 2010. Available: 10.1109/bibe.2010.22.

[2] S. Kuruppu, B. Beresford-Smith, T. Conway and J. Zobel, "Iterative Dictionary Construction for Compression of Large DNA Data Sets", IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 9, no. 1, pp. 137-149, 2012. Available: 10.1109/tcbb.2011.82.

[3] S. Merino et al., "Protein patterning on the micro- and nanoscale by thermal nanoimprint lithography on a new functionalized copolymer", Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures, vol. 27, no. 6, p. 2439, 2009. Available: 10.1116/1.3264687.

[4] A. Pinho, D. Pratas and P. Ferreira, "Bacteria DNA sequence compression using a mixture of finite-context models", 2011 IEEE Statistical Signal Processing Workshop (SSP), 2011. Available: 10.1109/ssp.2011.5967637.

[5] P. Rajarajeswari and A. Apparao, "DNABIT Compress-Genome Compression Algorithm", Bioinformation, vol. 5, no. 8, pp. 350-360, 2011. Available: 10.6026/97320630005350.

[6] C. G. Nevill-Manning and I. H. Witten, "Protein is incompressible", DCC, pages 257-266, 1999.

[7] M. Miyazaki, A. Shinohara, M. Takeda, "An improved pattern matching algorithm for strings in terms of straight-line programs", Proc. Combinatorial Pattern Matching, vol. 1264, pp. 1-11, 1997.

[8] P. G. Howard, "The design and analysis of efficient lossless data compression systems", June 1993.

[9] J. Ziv, A. Lempel, "Compression of individual sequences via variable-rate coding", IEEE Trans. Inform. Theory, vol. IT-24, pp. 530-536, September 1978.

[10] J. Rissanen, G. G. Langdon, Jr., "Arithmetic coding", IBM J. Res. Devel., vol. 23, no. 2, pp. 149-162, March 1979.

[11] P. A. Chou, M. Effros, R. M. Gray, "A vector quantization approach to universal noiseless coding and quantization", IEEE Trans. Inform. Theory, vol. 42, pp. 1109-1138, July 1996.

[12] M. Arimura, H. Yamamoto, "Asymptotic optimality of the block sorting data compression algorithm", IEICE Trans. Fundamentals, vol. E81-A, no. 10, pp. 2117-2122, October 1998.

[13] P. Li, S. Wang, J. Kim, H. Xiong, L. Ohno-Machado and X. Jiang, "DNA-COMPACT: DNA COMpression Based on a Pattern-Aware Contextual Modeling Technique", PLoS ONE, vol. 8, no. 11, p. e80377, 2013. Available: 10.1371/journal.pone.0080377.

[14] M. Sardaraz, M. Tahir, A. Ikram and H. Bajwa, "SeqCompress: An algorithm for biological sequence compression", Genomics, vol. 104, no. 4, pp. 225-228, 2014. Available: 10.1016/j.ygeno.2014.08.007.

[15] Yadav, N., Banerjee, K. and Bali, "A Survey on Fatigue Detection of Workers using Machine Learning", International Journal of E-Health and Medical Communications (IJEHMC), IGI Publications, Vol. 11, Issue 3, pp. 1-8, 2020. Available: 10.4018/IJEHMC.2020070101.

[16] Yadav, N., Banerjee, K. and Bali, "Fatigue Detection of Workers using Supervised Learning", Biological Forum-International Journal, Vol. 11, No. 1, pp. 236-242, 2019.

[17] P. Eric, G. Gopalakrishnan and M. Karunakaran, "An Optimal Seed Based Compression Algorithm for DNA Sequences", Advances in Bioinformatics, vol. 2016, pp. 1-7, 2016. Available: 10.1155/2016/3528406.

[18] P. Kaur and M. Sharma, "Diagnosis of Human Psychological Disorders using Supervised Learning and Nature-Inspired Computing Techniques: A Meta-Analysis", Journal of Medical Systems, vol. 43, no. 7, 2019. Available: 10.1007/s10916-019-1341-2.

[19] M. Sharma, G. Singh and R. Singh, "Stark Assessment of Lifestyle Based Human Disorders Using Data Mining Based Learning Techniques", IRBM, vol. 38, no. 6, pp. 305-324, 2017. Available: 10.1016/j.irbm.2017.09.002.

[20] R. Gautam, P. Kaur and M. Sharma, “A comprehensive review on nature inspired computing algorithms for the diagnosis of chronic disorders in human beings”, Progress in Artificial Intelligence, 2019. Available: 10.1007/s13748-019-00191-1.
[21] P. Kaur and M. Sharma, “A Survey on Using Nature Inspired Computing for Fatal Disease Diagnosis”, International Journal of Information System Modeling and Design, vol. 8, no. 2, pp. 70-91, 2017. Available: 10.4018/ijismd.2017040105.
[22] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression”, IEEE Transactions on Information Theory, vol. IT-23, No. 3, May 1977, pp. 337–343.
[23] J. Ziv and A. Lempel, “Compression of individual sequences via variable rate coding”, IEEE Transactions on Information Theory, Vol. IT-24, No. 5, September 1978, pp. 530–535.
[24] T. Bell, I. H. Witten, and J.G. Cleary, “Modeling for text compression”, ACM Computing Surveys, vol. 21, No. 4, December 1989, pp. 557–589.
[25] J.L. Bentley, D.D. Sleator, R.E. Tarjan, and V.K. Wei “A locally adaptive data compression algorithm”, Communications of the ACM, vol. 29, No. 4, April 1986, pp. 320–330.
[26] E.M. McCreight, “A space economical suffix tree construction algorithm”, Journal of the ACM, Vol. 32, No. 2, April 1976, pp. 262–272.
[27] C. Shannon, “A Mathematical Theory of Communication”, Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.
[28] Mark Daniel Ward, “Exploring Data Compression via Binary Trees1”, International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No.8, 2012.
[29] Manjeet Gupta Brijesh Kumar, “Web Page Compression using Huffman Coding Technique”, International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012) Proceedings published in International Journal of Computer Applications® (IJCA), 2012.
[30] Zhenhai Duan, Xin Yuan, Jaideep Chandrashekar, “Controlling IP Spoofing Through Inter-Domain Packet Filters”, Ieee Transactions on Dependable and Secure Computing, Vol. 5, No. 1, January-March 2008.
[31] M. Nelson, “Data Compression With Arithmetic Coding”, Marknelson.US, 2017. [Online]. Available: http://marknelson.us/2014/10/19/data-compression-with-arithmetic-coding/.
[32] Williams, Ross N, “An extremely fast Ziv-Lempel data compression algorithm”, In Data Compression Conference, 1991. DCC'91., pp. 362-371. IEEE, 1991
[33] T.C. Bell, J.G. Cleary, I.H. Witten, “Text Compression, New Jersey”, Englewood Cliffs:Prentice Hall, 1990.
[34] T Bell, “Longest Match String Searching For Ziv-Lempel Compression”, New Zealand, Christchurch:Department of Computer Science, University of Canterbury.
[35] E.R. Fiala, D.H. Greene, “Data Compression with Finite Windows”, Communications of the ACM, vol. 32, no. 4, pp. 490-505.
[36] B.W. Kernighan, D.M. Ritchie, “The C Programming Language”, New Jersey, Englewood Cliffs:Prentice Hall, 1988.
[37] D.E. Knuth, “Sorting and Searching in The Art of Computer Programming”, Massachusetts, Reading:Addison-Wesley Publishing Company, vol. 23, 1973.
[38] G.G Langdon, ”On Parsing Versus Mixed-Order Model Structures for Data Compression”, 1984.
[39] P.K. Pearson, “Fast Hashing of Variable-Length Text Strings”, Communications of the ACM, vol. 33, no. 6, pp. 677-680, June 1990.
[40] J.A. Storer, Data Compression: Methods and Theory, Maryland, Rockville:Computer Science Press, 1988.
[41] T.A. Welch, “A Technique for High-Performance Data Compression”, IEEE Computer, vol. 17, no. 6, pp. 8-19.
[42] R. Horspool, “Improving LZW (data compression algorithm)”, Proceedings. Data Compression Conference.