Trends in Bioinformatics

ISSN 1994-7941

science alert

ANSInet
an open access publisher
http://ansinet.com

Trends in Bioinformatics 8 (3): 93-98, 2015
ISSN 1994-7941 / DOI: 10.3923/tb.2015.93.98

A Compression Algorithm for DNA Sequences Based on R^2G Techniques with Security

[1,2]Syed Mahamud Hossein, [1]P.K. Das Mohapatra and [2]Debashis De
[1]Department of Microbiology, Vidyasagar University, West Bengal, Midnapur, 721102, India
[2]Department of Computer Science and Engineering, West Bengal University of Technology, BF-142, Sector-I, Kolkata, 700064,West Bengal, India

Corresponding Author: Syed Mahamud Hossein, Department of Microbiology, Vidyasagar University, West Bengal, Midnapur, 721102, India

ABSTRACT

A lossless compression algorithm, for genetic sequences, based on searching the exact repeat, reverse and genetic palindromes is reported. The compression results obtained in the algorithm show that the exact repeat, reverse and genetic palindromes are one of the main hidden regularities in DNA sequences. The proposed DNA sequence compression algorithm is based on repeat, reverse and genetic palindrome substring and creates online library file acting as a Look Up Table (LUT). The repeat, reverse and genetic palindrome substring is replaced by ASCII character where repeat of ASCII character start from 33-33+72, for reverse 33+73-33+73+72 and for genetic palindrome 179-179+72. It can provide the data security, by using ASCII code and on line Library file acting as a signature. The compression results obtained in the algorithm show that the exact repeat, reverse and genetic palindromes are one of the main hidden regularities in DNA sequences. The algorithm can approach a compression rate of 3.851273 bit/base.

Key words: Repeat, reverse, genetic palindrome and security

INTRODUCTION

Biological sequence should be very compressible. There are also strong biological evidences in supporting this claim. It is well-known that DNA sequences, especially in higher eukaryotes, contain many repeat, reverse and genetic palindromes. It is well recognized that the compression of DNA sequences is a very difficult task (Curnow and Kirkwood, 1989; Grumbach and Tahi, 1994; Lanctot *et al*., 2000; Rivals *et al*., 1995).

The DNA sequences only consist of 4 nucleotide bases, each nucleotide base required 8 bits for storage. It is our purpose to study such subtleties in DNA sequences. We will present a DNA compression algorithm, based on exact matching that gives the best compression results on standard benchmark DNA sequences. However, searching for all exact repeat, reverse and genetic palindromes in a very long DNA sequence is not a trivial task. These algorithms take a long time (essentially a quadratic time search or even more) in order to find approximate repeat, reverse and genetic palindromes that are optimal for compression. Proposed algorithm consists of two phases: (1) Find all exact repeat, reverse and genetic palindromes and (2) Encode exact repeat, reverse and genetic palindrome regions and non-match regions. We have developed for fast and sensitive homology search as our exact repeat, reverse and genetic palindrome search engine (Ma *et al*., 2002). This will present a DNA compression algorithm, based on repeat, reverse and genetic

palindrome substring. This substring of repeat, reverse and genetic palindrome is placed in a file, called library file, also it acts as a dynamic Look Up Table (LUT). At the time of decompression this ASCII character is placed in appropriate places on source file. This gives the best compression results on standard benchmark DNA sequences. Now the details of the algorithm are discussed, experimental results are shown and this result is compared with the most effective compression algorithm for DNA sequence (gzip-9) (Matsumoto *et al*., 2000).

We have changed the sequence order as reverse, complement and reverse complement and find out the result on it. Also, we can find the compression rate, compression ratio of randomly generated of equivalent length of artificial DNA sequence and compared with each other.

In this study, if not otherwise mentioned, we will use lower case letters u, v to denote finite strings over the alphabet {a, t, g, c}, $|u|$ denotes the length of u, the number of characters in u. u_i is the i-th character of u. $u_{i:j}$ is the substring of u from position i to position j. The first character of u is u_1. Thus, $u = u_{1:|u|-1}$, where $u_{i:j}$, represent the original substring and $|v|$ denotes the length of v, the number of characters in v. v_i is the i-th character of v. $v_{i:j}$ is the another substring of v from position i to position j. The first character of v is v_1. Thus $v = v_{1:|v|-1}$. $u_{i:j}$ match with $v_{i:j}$. The minimum different between u-v is of substring length. The $v_{i:j}$ represent the repeat/reverse /genomic palindrome substring .The match found if $u_{i:j} = v_{i:j}$ and count exact maximum repeat, reverse and genetic palindrome of $u_{i:j}$. It is use e to denote empty string and $\varepsilon = 0$.

MATERIALS AND METHODS

DNA sequences store on a text file:

Searching for exact repeat, reverse and genetic palindromes: Consider a finite sequence *s* over the DNA alphabet {a, c, g, t}. An exact repeat, reverse and genetic palindrome is a substring in *s* that can be transformed from another substring in *s* with edit operations (repeat, reverse and genetic palindrome, insertion). We only encode those substring match approx maximum that provide profits on overall compression.

This methods of compression is as follow:

- Run the program and output all exact repeat, reverse and genetic palindromes into a list s in the order of descending scores
- Extract a repeat, reverse and genetic palindrome r with highest score from list s, then replace all r by corresponding ASCII code into another intermediate list o and place r in library file. Where r is repeat/reverse/genetic palindrome
- Process each repeat, reverse and genetic palindrome in s so that there's no overlap with the extracted repeat, reverse and genetic palindrome r
- Goto step 2 if the highest score of repeat, reverse and genetic palindromes in s is still higher than a pre-defined threshold; otherwise exit

Encoding repeat, reverse and genetic palindromes: An exact repeat, reverse and genetic palindrome can be presented as two kinds of triples. First is (l, m, p), where l means the repeat, reverse and genetic palindrome substring length, m and p show the starting positions of two substrings in a repeat, reverse and genetic palindrome respectively. Second replace : this operation is expressed as (r, p, char) which means replacing the exact repeat/reverse/genetic palindrome substring at position p by ASCII character char.

In order to recover an exact repeat, reverse and genetic palindrome correctly the following information must be encoded in the output data stream.

Encoding algorithm

Algorithm for compression:

- Check for replaced character, if found just in shift in right direct ran
- Replace the first three consecutive replaceable symbol by the available special symbol in sequential order
- Check for the repeat, reverse and genomic palindrome for the rest of the part of the string it repeat found replace it by the symbol used for the replacement of the first three symbol for reverse and genomic palindrome respectively use the equivalent character of additive ASCII value 72 and 144, respectively
- During each pass place only one entry in the library file against the original replaceable characters with the replaced one rest, means reverse and genomic palindrome can be calculated during replacement by adding 72 and 144, respectively
- Continue step 1-4 until no three consecutive replaceable symbol exit
- Stop

Decoding algorithm

Algorithm for decompression:

- Extract the character
- Check if it is within a, t, g and c just directly put if not among those character replace by equivalent umbination reading from a, t, g and c by checking it with all replace character entry from library file
- If direct match replace exactly with the entrees available in the library else replace by reverse or genomic palindrome of that if match found with the 72 and 144 additive value ascii character of the give in library
- Continue until full string lossy either of a, t, g and c

Algorithm evaluation

Accuracy: As to the DNA sequence storage, accuracy must be taken firstly in that even a single base mutation, insertion, deletion would result in huge change of phenotype as we see in the sicklemia. It is not tolerable that any mistake exists either in compression or in decompression.

Efficiency: That the internal repeat, reverse and genetic palindrome algorithm can compress original file from substring length (l) into 1 character for any DNA segment and destination file uses less ASCII character to represent successive DNA bases than source file.

Space occupation: This algorithm reads characters from source file and writes them immediately into destination file. It costs very small memory space to store only a few characters. The space occupation is in constant level. In our experiments, the OS has no swap partition. All performance can be done in main memory which is only 512 MB on our PC.

RESULTS AND DISCUSSION

This algorithm tested repeat, reverse and genetic palindrome techniques on standard benchmark data used in (Grumbach and Tahi, 1994). For testing purpose we use two sets of data. The definition of the compression ratio (Chen *et al*., 2001); 1- (|O|/2|I|), where |I| is number of bases in the input DNA sequence and |O| is the length (number of bits) of the output sequence, the compression rate, which is defined as (|O|/|I|), where |I| is number of bases in the input

DNA sequence and |O| is the length (number of bits) of the output sequence. The improvement (Matsumoto *et al*., 2000) over gzip-9, which is defined as (Ratio_of_gzip-9-Ratio_of_LUT-3)/Ratio_of_gzip-9 *100. The compression ratio and compression rate for reverse, complement and reverse complement are presented in Table 1-2 and Fig. 1-2.

The result in Table 1-2 and Fig. 1-2 show compression ratio which vary from each other as the data set (first data set and second data set) come from different sources. This algorithm is very useful in database storing (Whiteford *et al*., 2008). You can keep sequences as records in database instead of maintaining them as files. By just using the exact repeat, reverse and genetic palindrome, users can obtain original sequences in a time that can't be felt. Additionally, this algorithm can be easily implemented.

Table 1: Comparative study within the first data set among normal, reverse, complement and reverse complement data

		Normal sequence		Reverse sequence		Complement sequence		Reverse complement sequence	
Sequence names	Base pair/ File size	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)
MTPACGA	100314	-0.79082	3.581634	-0.78715	3.574297	-0.79082	3.581634	-0.78715	3.574297
MPOMTCG	186608	-0.79223	3.584455	-0.79857	3.597145	-0.79223	3.584455	-0.79857	3.597145
CHNTXX	155844	-0.79736	3.594723	-0.80224	3.604476	-0.79736	3.594723	-0.80224	3.604476
CHMPXX	121024	-0.7852	3.570399	-0.77905	3.558104	-0.7852	3.570399	-0.77905	3.558104
HUMGHCSA	66495	-0.79526	3.590526	-0.79923	3.598466	-0.79526	3.590526	-0.79923	3.598466
HUMHBB	73308	-0.80733	3.614667	-0.795	3.590004	-0.80733	3.614667	-0.795	3.590004
HUMHDABCD	58864	-0.80735	3.614705	-0.79308	3.586165	-0.80735	3.614705	-0.79308	3.586165
HUMDYSTROP	38770	-0.80789	3.615785	-0.80872	3.617436	-0.80789	3.615785	-1.63833	5.276657
HUMHPRTB	56737	-0.80693	3.613867	-0.80284	3.605689	-0.80693	3.613867	-0.80284	3.605689
VACCG	191737	-0.77848	3.556956	-0.79296	3.585912	-0.77848	3.556956	-0.79296	3.585912
HEHCMVCG	229354	-0.79072	3.581433	-0.78116	3.562319	-0.79072	3.581433	-0.78116	3.562319

Table 2: Comparative study within the second data set among normal, reverse, complement and reverse complement data

		Normal sequence		Reverse sequence		Complement sequence		Reverse complement sequence	
Sequence names	Base pair/ File size	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)	Compression ratio	Compression rate (bits/base)
atatsgs	9647	-0.84762	3.69524	-0.84679	3.69358	-0.84762	3.695242	-0.84679	3.69358
atef1a23	6022	-0.87911	3.75822	-0.8738	3.74759	-0.87911	3.75822	-0.8738	3.74759
atrdnaf	10014	-0.83703	3.67406	-0.83383	3.66767	-0.83703	3.674056	-0.83383	3.66767
atrdnai	5287	-0.88689	3.77378	-0.88387	3.76773	-0.88689	3.773785	-0.88387	3.76773
celk07e12	58949	-0.80563	3.61126	-0.82029	3.64057	-0.80563	3.611257	-0.82029	3.64057
hsg6pdgen	52173	-0.78805	3.5761	-0.80001	3.60002	-0.78805	3.576103	-0.80001	3.60002
mmzp3g	10833	-0.8115	3.623	-0.83883	3.67765	-0.8115	3.623004	-0.83883	3.67765
xlxfg512	19338	-0.01479	2.02958	-0.8006	3.6012	-0.84238	3.684766	-0.8006	3.6012

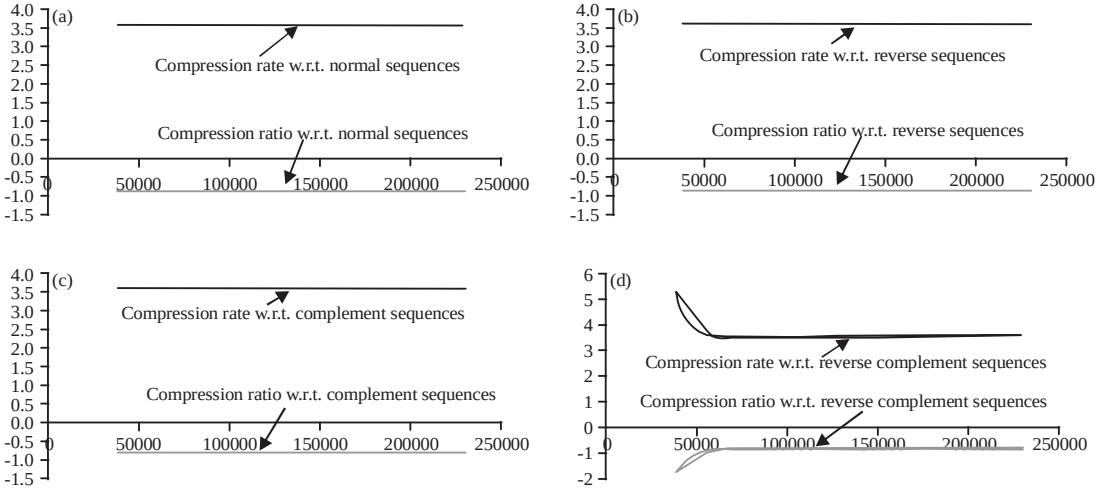

Fig. 1(a-d): Compression ratio and rate for (a) Norma sequences, (b) Reverse sequence, (c) Complement sequence and (d) Reverse complement sequences

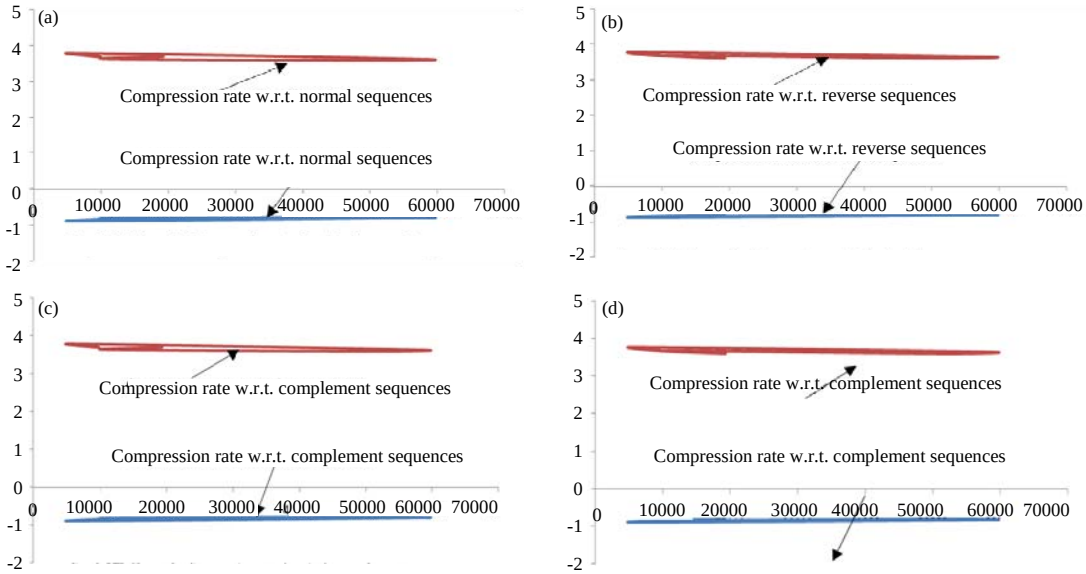

Fig. 2(a-d): Compression ratio and rate for (a) Normal sequences, (b) Reverse sequence, (c) Complement sequence and (d) Reverse complement sequences

From these experiments, we conclude that internal repeat, reverse and genetic palindrome matching patter (Karaca *et al*., 2005) are same in all type of sources and Look Up Table (LUT) plays a key role in finding similarities or regularities in DNA sequences. Output file contain ASCII character (Venugopal and Prasad, 1998) with unmatched a, u, g and c so, it can provide information security (Kahate, 2008), which is very important for data protection over transmission point of view. These techniques provide the moderate security to protect nucleotide sequence in a particular source.

CONCLUSION

In this study, a new DNA compression algorithm has been discussed, whose key idea is internal repeat, reverse and genetic palindrome. This compression algorithm gives a good model for compressing DNA sequences that reveals the true characteristics of DNA sequences. The compression results of repeat, reverse and genetic palindrome DNA sequences also indicate that this method is more effective than many others. This method is able to detect more regularities in DNA sequences, such as mutation and crossover and achieve the best compression results by using this observation. This method fails to achieve higher compression ratio than other standard methods but it has provided moderate information security.

Important observation are:

- Repeat, reverse and genetic palindrome substring length vary from 2-5 and no match found in case the substring length becoming six or more
- The substring length is three of highly repeat, reverse and genetic palindromes than substring length of four and five. That is why substring length of three is highly compressible over substring length of four and five
- Normal sequence is highly compressible than reveres, complement and reverse complement sequences

- Cellular DNA sequences compression rate and compression ratio are distinguishable different due each sequence that come into different sources where as artificial DNA sequences compression rate and compression ratio are same in all time in all data sets

ACKNOWLEDGMENT

Authors are grateful to all of the colleagues for their valuable suggestion, moral support, interest and constructive criticism of this study. The author offer special thanks to Ph.D guides for helping in carrying out the research work also like to thank their PCs.

REFERENCES

Chen, X., S. Kwong and M. Li, 2001. A compression algorithm for DNA sequences. IEEE Eng. Med. Biol., 20: 61-66.

Curnow, R.N. and T.B.L. Kirkwood, 1989. Statistical analysis of deoxyribonucleic acid sequence data: A review. J. Royal Stat. Soc., 152: 199-220.

Grumbach, S. and F. Tahi, 1994. A new challenge for compression algorithms: Genetic sequences. Inform. Process. Manage., 30: 875-886.

Kahate, A., 2008. Cryptography and Network Security. McGraw-Hill, New York, USA.

Karaca, M., M. Bilgen, A.N. Onus, A.G. Ince and S.Y. Elmasulu, 2005. Exact Tandem Repeats Analyzer (E-TRA): A new program for DNA sequence mining. J. Genet., 84: 49-54.

Lanctot, K., M. Li and E.H. Yang, 2000. Estimating DNA sequence entropy. Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA., pp: 409-418.

Ma, B., J. Tromp and M. Li, 2002. PatternHunter: Faster and more sensitive homology search. Bioinformatics, 18: 440-445.

Matsumoto, T., K. Sadakane and H. Imai, 2000. Biological sequence compression algorithms. Genome Info., 11: 43-52.

Rivals, E., J.P. Delahaye, M. Dauchet and O. Delgrange, 1995. A guaranteed compression scheme for repetitive DNA sequences. Technical Report IT-285, LIFL Lille I University.

Venugopal, K.R. and S.R. Prasad, 1998. Mastering C. McGraw-Hill, New York, USA.

Whiteford, N.E., N.J. Haslam, G. Weber, A. Prugel-Bennett, J.W. Essex and C. Neylon, 2008. Visualizing the repeat structure of genomic sequences. Complex Syst., 17: 381-398.