Chapter 6

A COMPRESSION ALGORITHM FOR DNA SEQUENCES BASED ON R^2G TECHNIQUES WITH SECURITY

Abstract

A lossless compression algorithm, for genetic sequences, based on searching the exact Repeat, Reverse and Genetic palindromes (R^2Gp) is reported. The compression results obtained in this algorithm shows 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 R^2Gp substring and creates an online library file acting as a Look Up Table (LUT). The R^2G substring is replaced by ASCII character where repeat of ASCII character starts 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 with online Library file acting as a signature. The compression results obtained in this algorithm show that the exact R^2Gp 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

1 Introduction

Biological sequence should be very compressible. There are also strong biological evidences in

support of this claim. It is well-known that DNA sequences, especially in higher eukaryotes, contain many repeats, reverse and genetic palindromes. It is well recognized that the compression

of DNA sequences is a very difficult task [102,48,103,89].The DNA sequences consist only 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 unmatch regions. We have developed for fast and sensitive homology search as our exact repeat, reverse and genetic palindrome search engine [88]. 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 the 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)[29].

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 equivalent length of artificial DNA sequence and compared with each other.

2 Proposed tri-combination method of Repeat, Reverse & Genetic Palindrome technique

2.1 Method of Repeat, Reverse and Genetic Palindrome (R^2GP) technique

atgatagacgatagttaccagataatgttacatgtacatgatacag

In combined technique, the principal idea is s_1=ata repeat sub-string (consider size of sub sequence is 3) is repeated in how many places, is shown by red color. The s_2=cag (reverse of gac) sub-string repeated in how many places is shown by green color and s_3=tac (genetic palindrome of atg) is repeated in how many places is shown by purple color , continue at the end of file.

Replace maximum number repeat of R^2Gp in descending order of substring by corresponding ASCII code.

Introduction of Repeat, Reverse & Genetic Palindrome technique

Repeat, Reverse and Genetic Palindrome substring is reported and replaced by corresponding ASCII character and also creates a corresponding library file.

This algorithm consists of two phases: i) find all exact Repeat, Reverse and Genetic Palindrome and ii) encode match and non-match regions.

2.2 Basic terminology of proposed try-combination of Repeat, Reverse & Genetic Palindrome technique

Searching for exact repeat, reverse and genetic palindromes: Consider a finite sequence *s* over the DNA alphabet {a, c, g, t}. An exact R^2Gp 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 approximate maximum that provide profits on overall compression.

This methods of compression is as follow:

● Run the program and output all exact R^2Gp into a list s in the order of descending scores

● First find the highest match score from list s and extract match substring r of R^2Gp after that all sub string r is replaced by a corresponding ASCII symbol, store in another intermediate list o. Place all highest match substring r into library file, where substring r is define repeat or reverse or genetic palindrome

● All substring r of R^2Gp are finally store in list s where no overlap is observed.

● Repeat the step for finding highest score of R^2Gp in s is still higher than a pre-defined threshold; otherwise exit

2.3 Encoding and decoding algorithm

Algorithm for compression:

● Find match substring which is replaced, if match found just move forward

● Then replace all match substrings by ASCII symbol in sequential order

● Check for the R^2Gp for the rest of the part of the string, if repeat found replace it by the symbol. It is done by the replacement of the first three symbols of R^2Gp respectively and place 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 by replacing one ASCII symbol, other two- 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

Algorithm for decompression:

● Extract the character

● Check if it is within a, t, g and c just directly put if not among those characters replace by

equivalent combination reading from a, t, g and c by checking it with all replace character entry

from library file

● If match found, replace exactly with the entries available in the library else replace by other symbol by adding 72 and 144 with previous ASCII value.

- Continue the process until the library file is empty

3 Results and discussion of Repeat, Reverse and Genetic Palindrome technique

This R^2Gp algorithm tested on standard benchmark data used in paper [48] and two sets of data are used for testing.

The compression ratio and rate for reverse, complement and reverse complement result are presented in Table 6.1 for cellular and table 6.2 for artificial sequences

Table 6.1 cellular DNA sequence compression ratio and rate

			Cellular DNA sequences							
			Normal Sequences		Reverse Sequences		Complement Sequences		Reverse Complement Sequences	
Data set	**Sequence Name**	**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)
Data set-I	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
	Average			3.59265		3.58909		3.59265		3.73993
Data set-II	atatsgs	9647	-0.84762	3.69524	-0.84679	3.69358	-0.84762	3.695242	-0.84679	3.69358
	atef1a23	6022	-0.8791	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
	Average									

Table 6.2 artificial sequence compression ratio and rate

Data set	Sequence Name	Base pair/ File size	Artificial sequences							
			Normal Sequences		Reverse Sequences		Complement Sequences		Reverse Complement Sequences	
			Compression ratio	Compression rate(bits /base)	Compression ratio	Compression rate(bits /base)	Compression ratio	Compression rate(bits /base)	Compression ratio	Compression rate(bits /base)
Data set-I	X1	100314	-0.808202	3.616404	-0.80860	3.617202	-0.80820	3.616404	-0.80860	3.617202
	X2	186608	-0.81332	3.62664	-0.80611	3.612235	-0.81332	3.62664	-0.80611	3.612235
	X3	155844	-0.805164	3.610328	-0.80449	3.608994	-0.80516	3.610328	-0.80449	3.608994
	X4	121024	-0.804964	3.609929	-0.79927	3.598559	-0.80496	3.609929	-0.79927	3.598559
	X5	66495	-0.808978	3.617956	-0.80933	3.618678	-0.80897	3.617956	-0.80933	3.618678
	X6	73308	-0.80395	3.607901	-0.81158	3.623179	-0.80351	3.607028	-0.81180	3.623615
	X7	58864	-0.808304	3.616608	-0.81061	3.621229	-0.80830	3.616608	-0.81061	3.621229
	X8	38770	-0.812432	3.624865	-0.81036	3.620738	-0.81243	3.624865	-0.81036	3.620738
	X9	56737	-0.810741	3.621482	-0.81765	3.6353	-0.81074	3.621482	-0.81765	3.6353
	X10	191737	-0.799298	3.598596	-0.80267	3.605355	-0.79929	3.598596	-0.80267	3.605355
	X11	229354	-0.803867	3.607733	-0.80271	3.605431	-0.80386	3.607733	-0.80271	3.605431
	Average			3.61440		3.61517		3.61432		3.61521
Data set-II	XX1	9647	-0.88841	3.77682	-0.86172	3.72344	-0.84513	3.69027	-0.86172	3.72344
	XX2	6022	-0.84901	3.69802	-0.90037	3.80073	-0.89372	3.78745	-0.90037	3.80073
	XX3	10014	-0.90202	3.80405	-0.86339	3.72678	-0.84901	3.69802	-0.86339	3.72678
	XX4	5287	-0.81662	3.63324	-0.89143	3.78286	90202	3.80405	-0.89143	3.78286
	XX5	58949	-0.80752	3.61505	-0.80807	3.61614	-0.81662	3.63324	-0.80807	3.61614
	XX6	52173	-0.8499	3.69981	-0.80967	3.61934	-0.80752	3.61505	-0.80967	3.61934
	XX7	10833	-0.83494	3.66987	-0.83809	3.67617	-0.8499	3.69981	-0.83809	3.67617
	XX8	19338	-0.808202	3.69839	-0.82418	3.64836	-0.83494	3.66987	-0.82418	3.64836
	Average			3.616404		3.69922		3.69972		3.69922

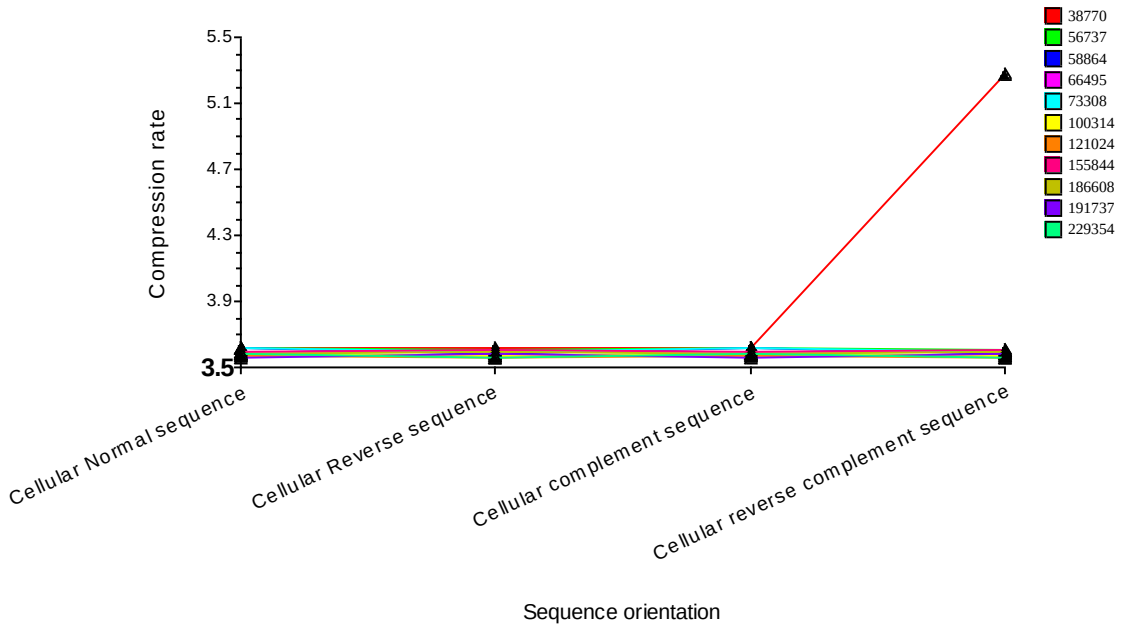

Fig.6.1 cellular sequence orientation vs. compression rate of different file size(data set-I)

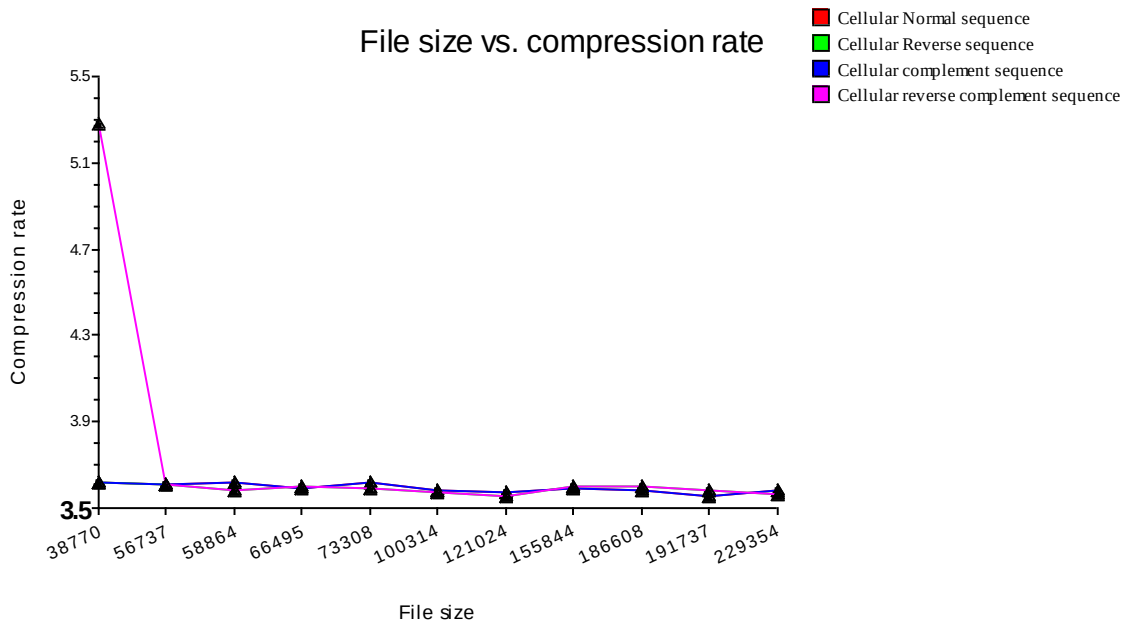

Fig.6.2 file size of cellular sequence vs. compression rate of different orientation(data set-I)

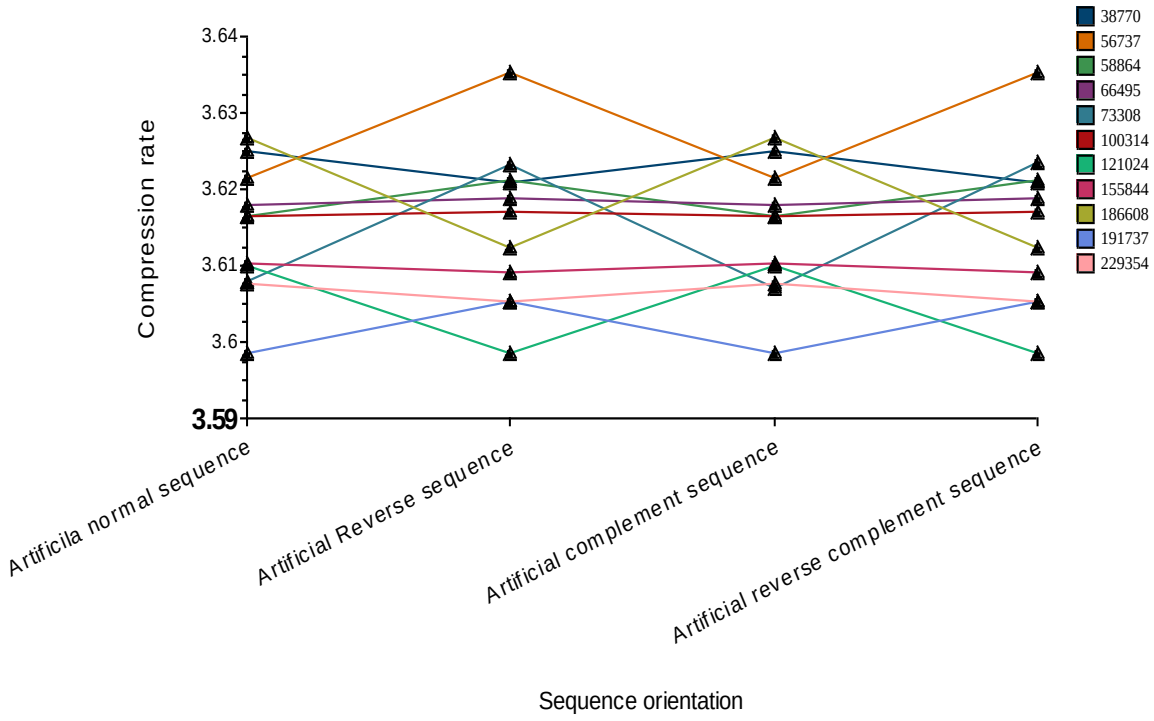

Fig.6.3 artificial sequence orientation vs. compression rate of different file size(data set-I)

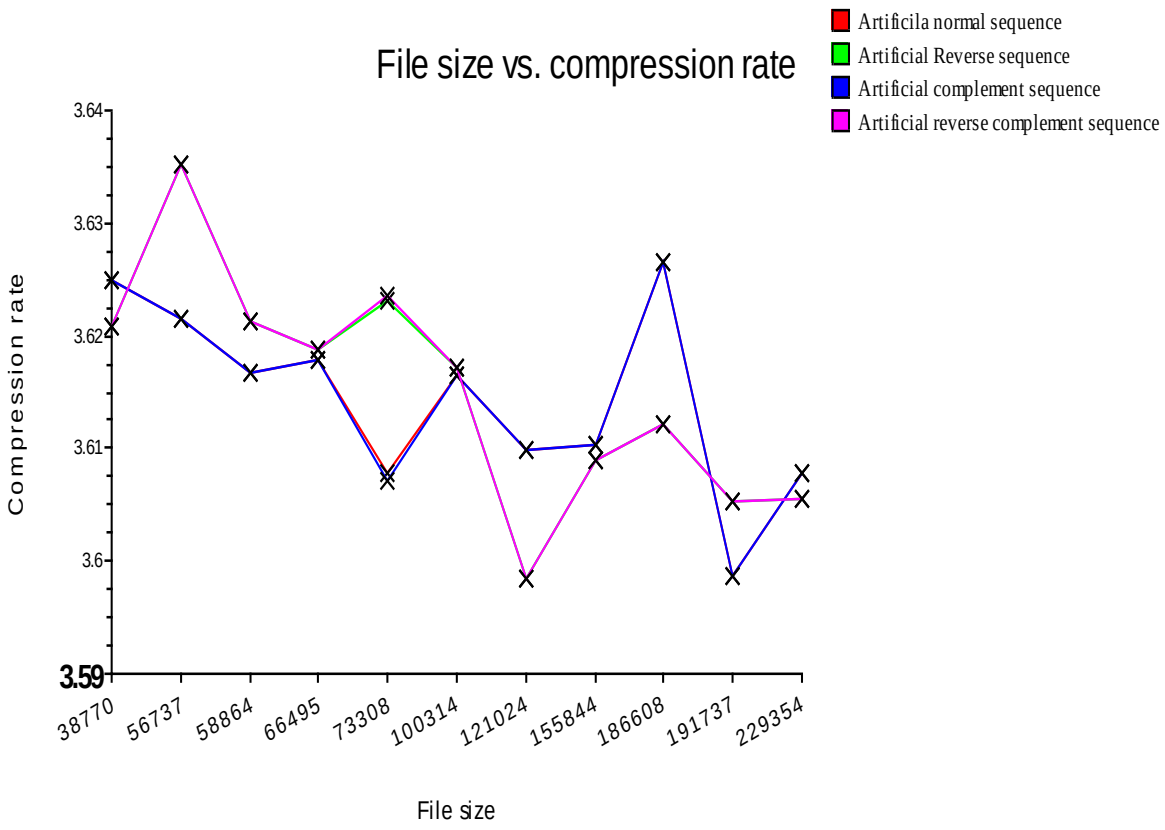

Fig.6.4 artificial sequence file size vs. compression rate of different orientation (data set-I)

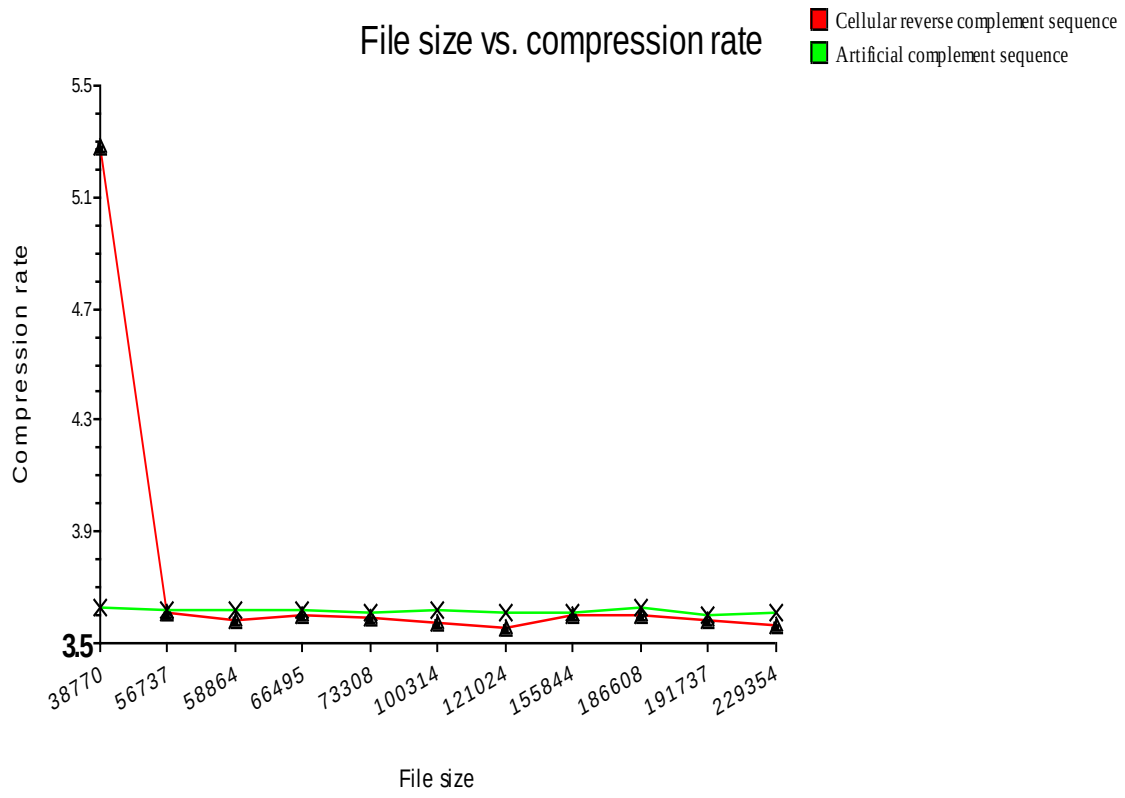

Fig.6.5 file size vs. compression rate of cellular & artificial sequences (data set-I)

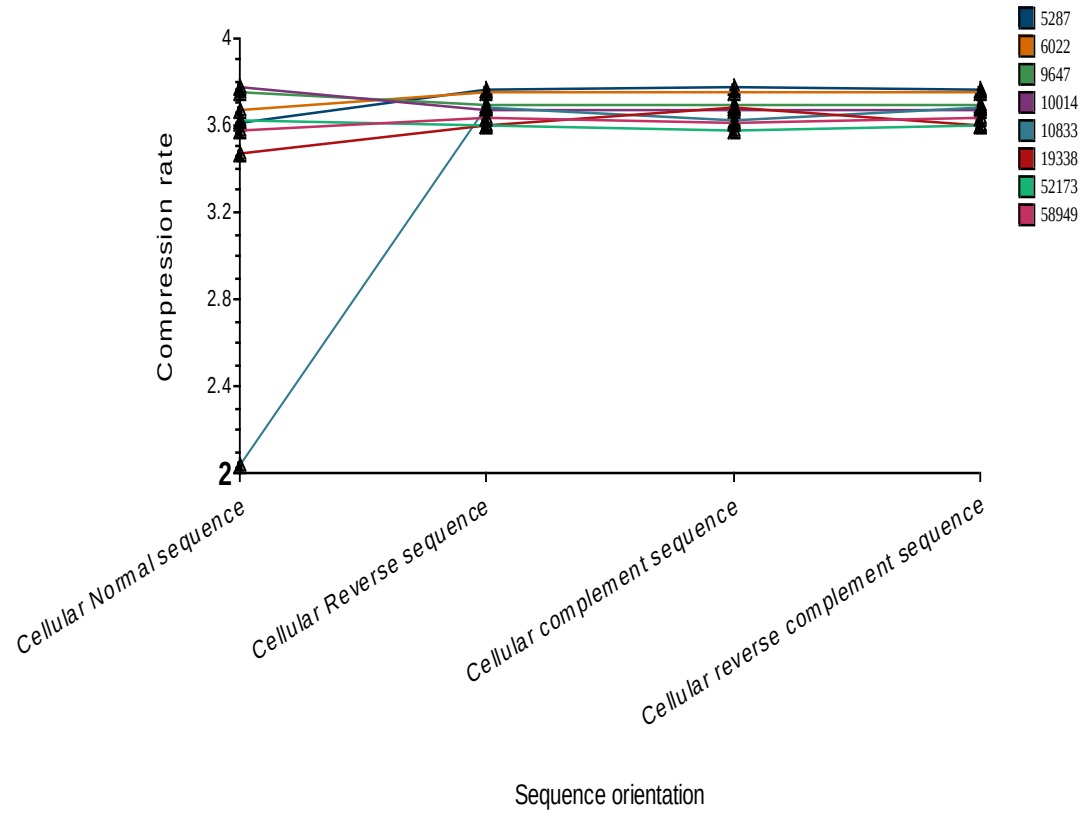

Fig.6.6 cellular sequence orientation vs. compression rate of different file size(data set-II)

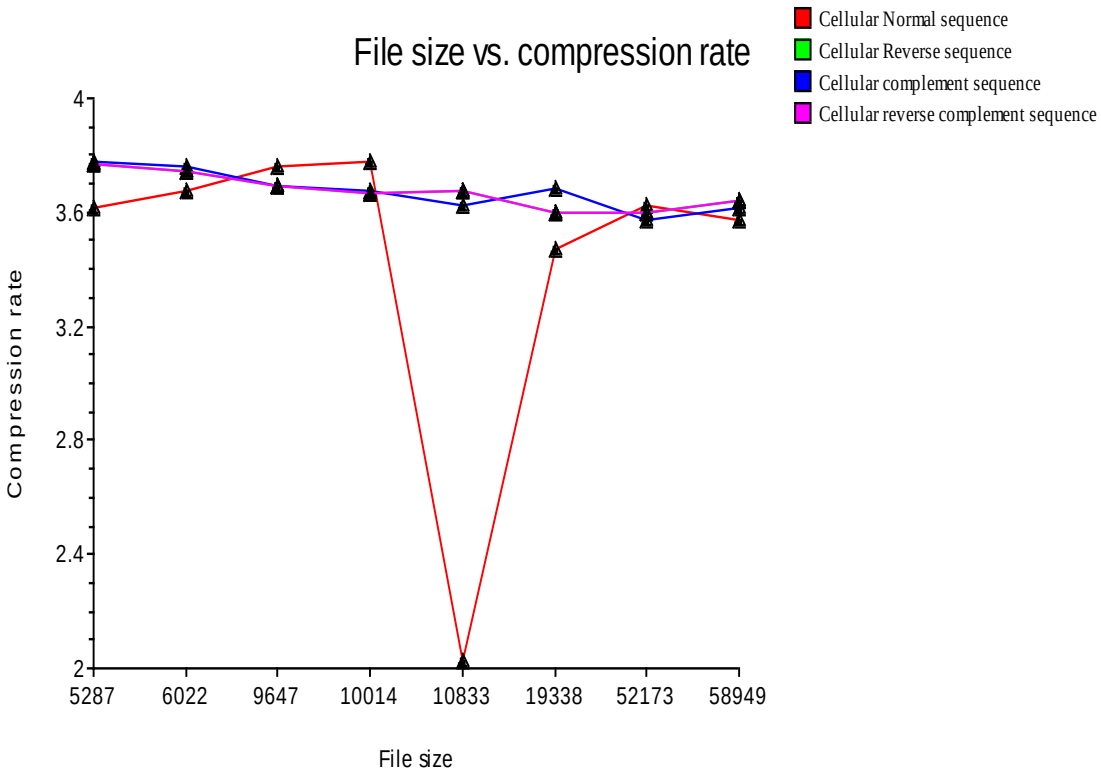

Fig.6.7 file size of cellular sequence vs. compression rate of different orientation(data set-II)

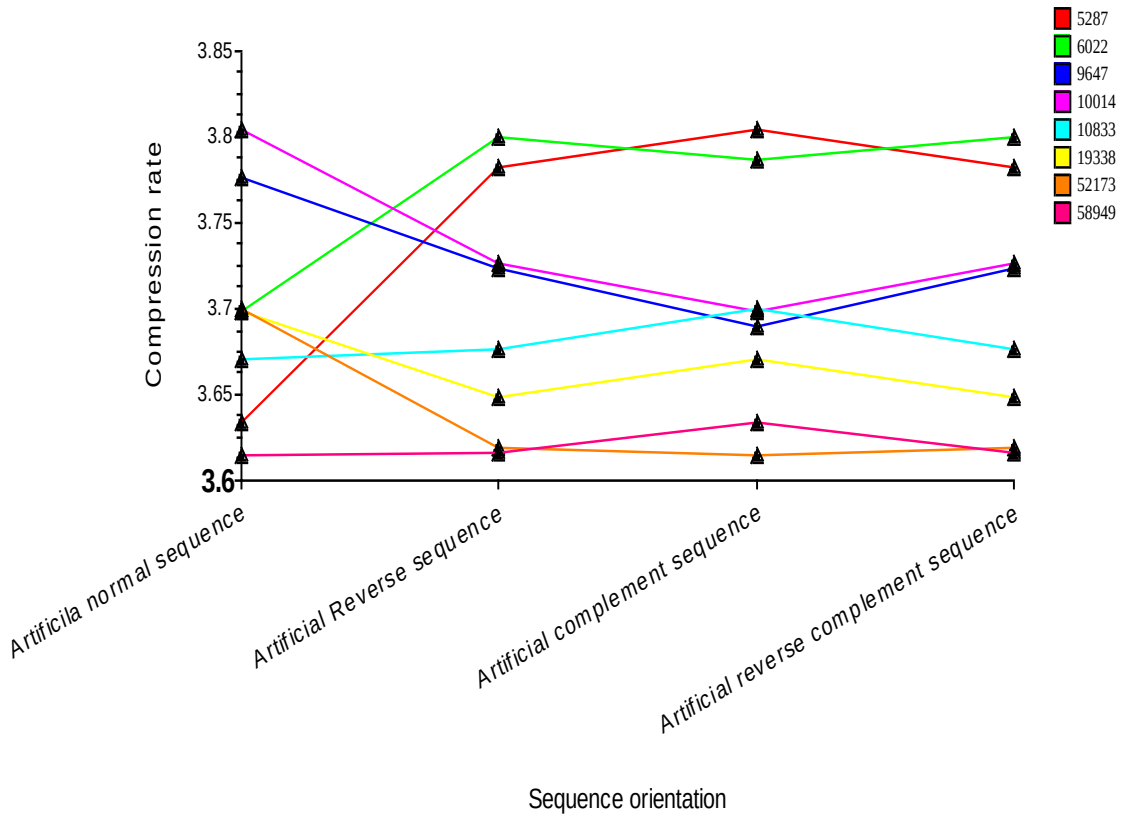

Fig.6.8 artificial sequence orientation vs. compression rate of different file size(data set-II)

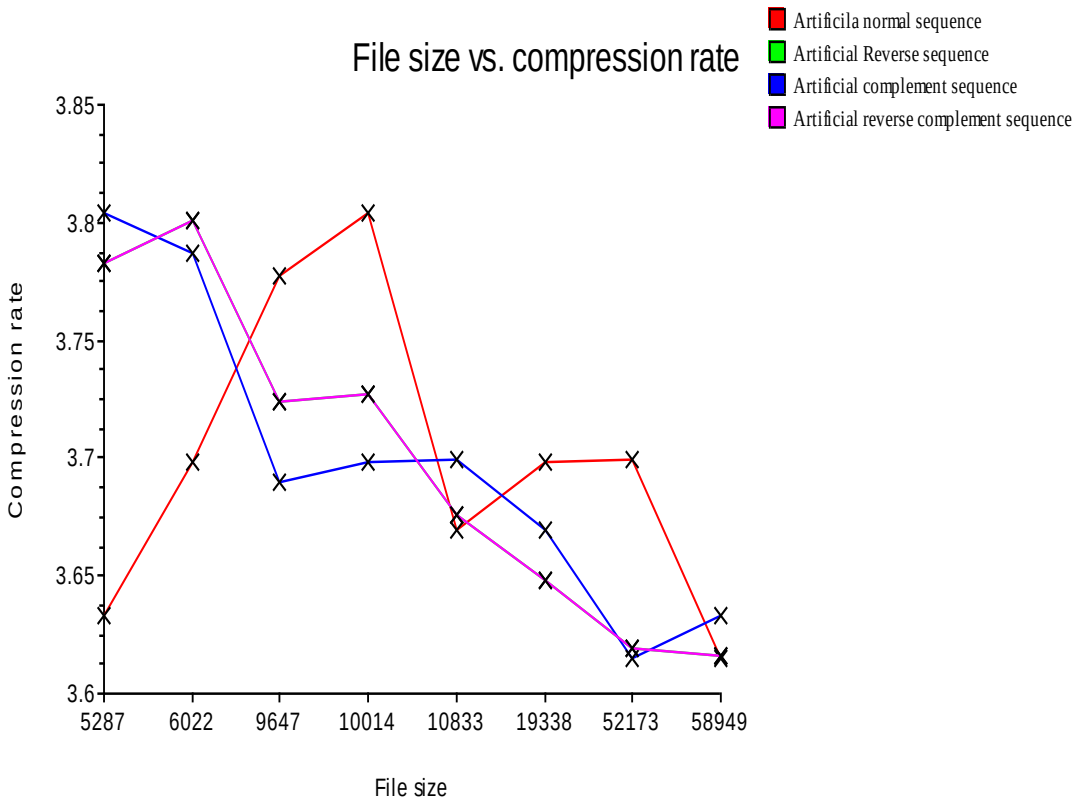

Fig.6.9 artificial sequence file size vs. compression rate of different orientation (data set-I)

Fig.6.10 file size vs. compression rate of cellular & artificial sequences (data set-I)

This data is graphically presented in fig.6.1 & 6.2 for data set-I and 6.6 & 6.7 for data set-II. The fig.6.1 & 6.2 shown the compression rate & ratio varies with word size and independent of file size. The fig.6.1 & 6.7 shows the compression rate & ratio are different in different orientations, if the word size is 4 , the minimum compression rate is 3.58909 bits/base for data set-I & 3.581634 bits/base for data set-II of cellular sequences and 3.61432 bits/base for data set-I & 3.616404 bits/base for data set-II of artificial sequences. The artificial sequence compression rate & ratio are graphically shown in the fig. 6.3 & 6.4 for data set-I and 6.8 & 6.9 for data set-II. The fig. 6.5 & 6.10 shows the artificial compression rate and cellular sequence compression rate are completely different, also graphical nature is different. It is also observed that the compression rate is similar in a particular word size because the sequence comes under different species and matching pattern are same but in case of artificial data the compression rate is dissimilar because this data is random.

The result shows the compression ratio which varies from each other as the data set (first data set and second data set) comes from different sources. This algorithm is very useful in database storing [104]. We 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.

From these experiments, we conclude that internal repeat, reverse and genetic palindrome matching pattern [105] are same in all types of sources and Look Up Table (LUT) plays a key role in finding similarities or regularities in DNA sequences. The output file contains ASCII character [106] with unmatched a, u, g and c so, it can provide information security [107], 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 observations 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 reverse, complement and reverse complement Sequences.

●Cellular DNA sequences compression rate and compression ratio are distinguishable different due to each sequence that come from different sources where as artificial sequences compression rate and compression ratio are same for all time in all data sets.