OPEN ACCESS

International
Journal
Of Modern Engineering Research (IJMER)

A Compression & Encryption Algorithms on DNA Sequences Using R^{2}P & Selective Technique

Syed Mahamud Hossein[2,3], A. Mitra[1],Pradeep Kumar Das Mohapatra[2], Debashis De[3]

[2,3]Research Scholar, Vidyasagar University, Midnapur-721102
[1]Department Of M.C.A.,Hit, Haldia
[2]Department Of Microbiology, Vidyasagar University, Midnapur-721102, West Bengal
[3]Department Of Computer Science And Engineering, Maulana Abul Kalam Azad University Of Technology, Bf-142, Sector-I, Kolkata, India.

ABSTRACT: *The size of DNA (Deoxyribonucleic Acid) sequences is varying in the range of millions to billions of nucleotides and two or three times bigger annually. Therefore efficient lossless compression technique, data structures to efficiently store, access, secure communicate and search these large datasets are necessary. This compression algorithm for genetic sequences, based on searching the exact repeat, reverse and palindrome (R^{2}P) substring substitution and create a Library file. The R^{2}P substring is replaced by corresponding ASCII character where for repeat, selecting ASCII characters ranging from 33 to 33+72, for reverse from 33+73 to 33+73+72 and for palindrome from 179 to 179+72. The selective encryption technique, the data are encrypted either in the library file or in compressed file or in both, also by using ASCII code and online library file acting as a signature. Selective encryption, where a part of message is encrypted keeping the remaining part unencrypted, can be a viable proposition for running encryption system in resource constraint devices. The algorithm can approach a moderate compression rate, provide strong data security, the running time is very few second and the complexity is O(n^2). Also the compressed data again compressed by renounced compressor for reducing the compression rate & ratio. This techniques can approach a compression rate of 2.004871bits/base.*

***Keyword**: DNA Sequence, Lossless Compression, ASCII code, Repeat, Reverse, palindrome, Substitution and Encryption*

Abbreviation of R^{2}P : Repeat, Reverse and Palindrome

I. INTRODUCTION

The DNA database are too large [1-8], complex, must contain some logical organization [9-10], hence data structure to store, access, process this data efficiently is a difficult & very challenging task [11-12]. So it needs an efficient compression algorithm to store these huge mass of data. The standard compression techniques [13-14] cannot compress the biological sequences well because the regularities in DNA sequences are much subtler [15]. The two bit encoding is efficient if the bases are randomly distributed in the sequence, but the life of an organism is non-random, hence DNA sequences which appear in a living organism are expected to be nonrandom and have some constraints [15]. Huffman's code also fails badly on DNA sequences both in the static and adaptive model, because the probabilities of occurrence of the four symbols are not very different [11,15]. There are many repeats[11] within a given DNA sequence (e.g. ATGC), which may occur more than once in a given DNA sequence. Recently, several algorithms have been proposed for the compression of DNA sequences based on DNA sequence special structures[11,16-17].Though a lot of works have to be done on selective encryption of images, videos, speech etc, and not much work has been done on the selective encryption on compressed DNA sequences[18-19]. But comparing to DNA computing, the research of biological cryptology attracted less attention [20-24].

This DNA sequences Compression algorithm achieves a moderate compression ratio and runs significantly faster than any existing compression program on benchmark DNA sequences. This algorithm developed on the basis of fast and sensitive homology search [25], as our exact R^{2}P search engine. Proposed algorithm consists of three phases: i) finding all exact repeat, reverse and palindrome ii) encode R^{2}P regions and non-match regions and iii) Encrypt the library file, compress file or in both. Now a day's information security is a most challenging question, how to protect the DNA data from the hackers [26-31]. Selective encryption is the process of selecting a part of a whole message, to begin through the process of encryption, keeping the remaining portion of the message clear in such a way that the security is not compromised. In the selective encryption process only a fraction (r) of the whole message or plain text is selected for encryption and the remaining part is kept in the clear. Selection of the ‘r’ part is vital for the security point of view in case of selective encryption; the criteria for selection for ‘r’ vary according to the type of medium. Intuitively, as ‘r’ increases, the security level also increases at the cost of increased time of encryption.

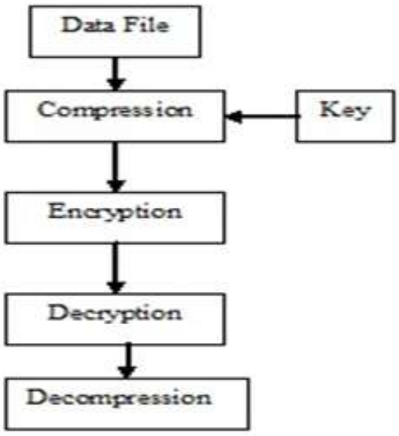
This compression method provides two tier security i) the data are compressed, generates two separate files individually and each file contains ASCII characters ii) Apply selective encryption on library file or compress file or both. This selective encryption approach not only reduces the time complexity for encryption and decryption due to encryption of the part of the compressed data where reconstruction information are mostly concentrated and but also it reduces the storage and communication cost. Also developed specific programm of our requirement and finding the result on AES, DES and RSA [32-34].

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, and the number of characters in u.u_i is the ith character of u. $u_{i: j}$ is the substring of u from position I to position j. The first character of u is u_i. Thus $u=u_{1:|u|-1}$, where $u_{i: j}$ represents the original substring and |v| denotes the length of v, the number of characters in v. v_i is the ith 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 $v=v_{1:|u|-1}$. $u_{i:j}$ match with $v_{i:j}$. The minimum difference between u-v is of substring length. The $v_{i: j}$ represents the repeat, reverse, palindrome substring. The match found if $u_{i: j}= v_{i: j}$ and count exact maximum R^2P of $u_{i:j}$. We use Є to denote empty string and Є =0.

This paper discuss details of the algorithm, provide exponential results and compare the compression rate, execution time and encryption time [35-42]. Other related algorithms are file size measurement, file mapping, DNA sequences orientation changing and random string generation. The overall compression process is two pass where first step output of R^2P is again compress by FreeArc[43] compressor and finally getting final result.

II. METHODS

2.1: Process diagram



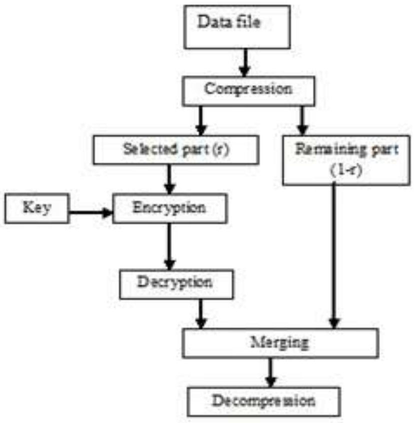


Fig-1: Fig-2

Fig-1 & II : show how to apply compression followed by encryption on compressed file

2.2: File format : File type is text file and blank space ahead the end of file. The output file also text file, contains the information of both unmatch four base pair and a coded value of ASCII character.

2.3 Generating the substring from input sequence

a t g g t a g t a a t gtacatg …… ...n_n
It is clear that for ith substring W_i .
i, is the starting position of the substring and.
j= (i-1) + l, is end position of the substring; where l is the substring length.

The substring length is less than 3(three) has no importance in matching context therefore we consider the substring size in the range: 3<=1<=n.
Therefore range for I and j are as 1<=i<=n-1+1 and 1<= j<=n respectively.

2.4: searching for exact R^2P

Consider a finite sequence s over the DNA alphabet {a, t, g, c}. As exact R^2P is a substring in s that can be transferred from another substring in s with edit operations (on repeat, reverse and palindrome, insertion). Encode these substrings only to match approximate maximum that provides profit on overall compression.

This method of compression is as below:

1. Run the program and output all exact R^2P into a list s in the order of descending scores.

2. Extract a repeat, reverse and palindrome r with highest score from list s, and then replace all r by corresponding ASCII code into another intermediate list o and place r in library file. Where r is repeat, reverse and palindrome substring.

3. Process each R^2P in s so that there's no overlap with the extracted repeat, reverse and palindrome r.

4. Goto step 2 if the highest score of repeat, reverse and palindrome in s is still higher than a pre-defined threshold; otherwise exits.

2.5 : Encoding R^2P

An exact R^2P can be presented as two kinds of triplets, first is (l,m,p), where l means the repeat, reverse and palindrome substring length, m and p show the starting position of two substrings in a R^2P respectively. Second: replace this operation as expressed (r; p; char), which means replacing the exact repeat, reverse & palindrome substring at position p by ASCII character char. In order to recover an exact R^2P correctly the following information must be encoded in the output data stream:

2.6: Decoding

Decoding time first requires online Library file, which was created at the time of encoding the input file. On this particular value, the encoded input string is decoded and produces the original files.

2.7 Information security

This technique can provide two tier information securities.

This technique can provide two tier information securities.

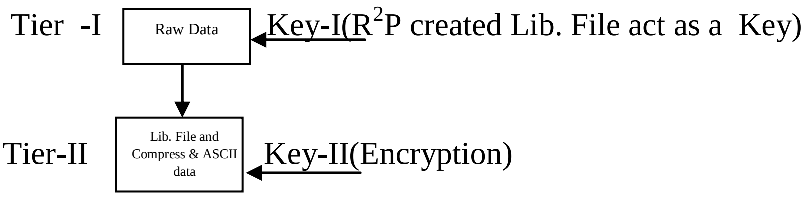

Tier -I
Raw Data
Key-I(R^2P created Lib. File act as a Key)
Tier-II
Lib. File and Compress & ASCII data
Key-II(Encryption)

Fig -2 : Show the Label I & II security Technique

2.7.1: In tire one; the input sequence contain only 4 bases (a, t, g, c), after compression reduce the file size, converted from 4 letters to 256 characters, with unmatch a,t,g & c and one substring contains 3 characters, is replaced by single ASCII characters, so the output file is information secure than input file.

2.7.2: In tire two; Apply selective encryption technique on compressed output file. Selective encryption are applied in three ways i) Select only single character (any character from 1-256) ASCII characters ii) Select numeric numbers only iii) Pattern selection. For selection encryption purpose, generate private and public key.

3. Algorithms

3.1: Compression Algorithm:

1. Check for replaced character, if found just shift in right, direct run.

2. Replace the first three consecutive replaceable symbols by the available special symbols in sequential order.

3. Check for the R^2P for the rest of the part of the string, if repeat found replace it by the symbols used for the replacement of the first three symbols, for reverse and palindrome respectively use the equivalent character of additive ASCII value 72 and 144 respectively.

4. During each pass place one entry in the library file against the original replaceable characters with the replaced one. Rest, means reverse and palindrome can be calculated during replacement by adding 72 and 144 respectively.
5. Continue step 1 to 4 until no three consecutive replaceable symbol exists.
6. Stop.

3.2. Decompression Algorithm

1. Extract the character
2. Check if it is within 'a','t','g','c' just directly put it, if it not among these characters , replace by equivalent rumination reading from 'a','t','g','c' by checking it with all replaceable characters entered from library file.
3. If direct matched replace exactly with the entries available in the library, else replace by reverse or palindrome of that, if match found within the 72 and 144 additive values ASCII character of the given in library file.
4.Continue until full string lossy either of 'a','t','g' and'c'.

3.3: Selective Encryption & Decryption Algorithm

3.3.1: Selective Encryption Algorithm

1. Input filename with path.
2. Select number or a specific string.
3. Use RSA algorithm for encryption of the selected number or specified string.
4. Generate an auxiliary file to keep the flag for the specific regions of the encrypted data.
5. Generate the encrypted output file.
6. Generate the Public Key and Private Key ultimately.

3.3.2: Selective Decryption Algorithm:

1. Open Encrypted and Auxiliary file.
2. Input Encryption option.
3. Read encrypted data from Auxiliary file.
4. Use Private Key to decrypt data using RSA module.
5. Get the Decrypted output file.

III. ALGORITHM EVALUATION

4.1: Accuracy

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. It is not tolerable that any mistake exists either in compression or decompression. For accuracy purpose develop one by one file mapping algorithm.

4.2 : Efficiency

The initial R^2P algorithm can compress original file from substring l to l characters for any DNA segment and destination file uses less ASCII character to represent successive DNA bases than source file.

4.3: Space Occupation

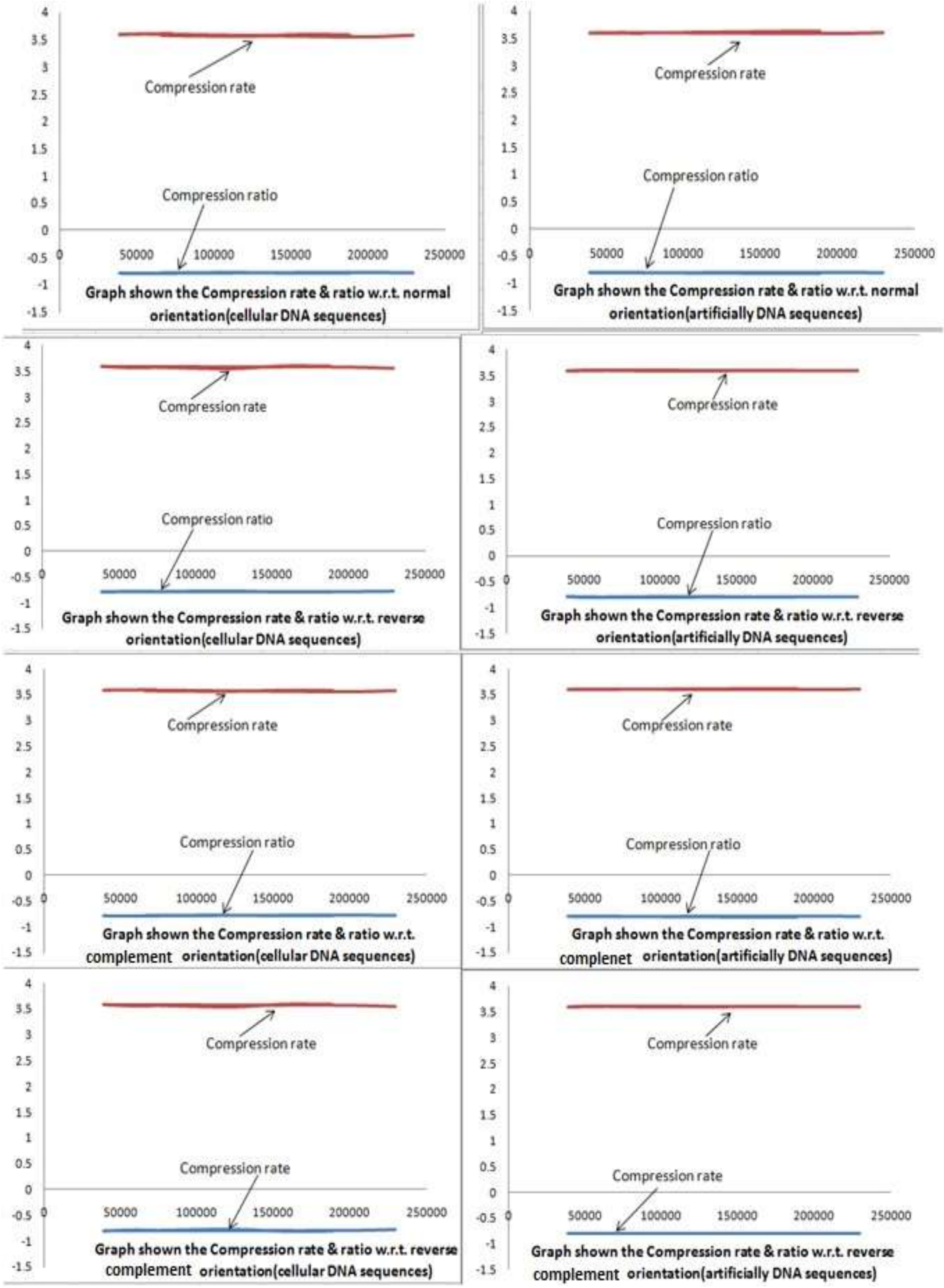
Our algorithm reads characters from source file and writes them immediately into destination file. It costs very small memory space to store only a few characters. The space occupation is in constant level.

5. Experimental result

We tested R^2P technique on standard benchmark data, used in [12,44], definition of the compression ratio, rate and improvement are also used in [44]. The compression ratio and rate of R^2P are presented in table-I, including the result of artificial DNA data and Graph-I shown the same. The last two columns show the average compression and decompression speed in seconds(10^{-1}) per input byte (average computed over five runs for each sequence)."encode" means compression while "decode" means decompression. Also apply the selective encryption algorithm on compress data. In table-II showing the AES, DES & RSA result.

Sequence Orientation	Sequence Name	Sequence Size	Cellular Dna Sequences									Artificial Dna Sequences								
			Reduce File Size Byte(C)	Lib. File Size(L)	Compression Ratio	Compression Rate(Bits /Base)	Apply Selective Encryption Algorithm	Encrypt Time		Compression Time		Reduce File Size Byte(C)	Lib. File Size(L)	Compression Ratio	Compression Rate(Bits /Base)	Apply Selective Encryption Algorithm	Encrypt Time		Compression Time	
								Encode Time	Decode Time	Encode Time(10-1)	Decode Time						Encode Time	Decode Time	Encode Time	Decode Time
Normal Orientation	Mtpacga	100314	44784	128	-0.790857	3.581713	No Change	<100	<100	33.35	0.329	45220	128	-0.808242	3.616484	No Change	<100	<100	33.02	0.329
	Mpomtcg	186608	83484	128	-0.792249	3.584498	No Change	<100	<100	61.09	0.604	84468	128	-0.813341	3.626683	No Change	<100	<100	59.78	0.604
	Chntxx	155844	69900	128	-0.797387	3.594774	No Change	<100	<100	50.27	0.494	70204	128	-0.80519	3.61038	No Change	<100	<100	51.70	0.494
	Chmpxx	121024	53886	128	-0.785233	3.570465	No Change	<100	<100	37.19	0.384	54484	128	-0.804997	3.609995	No Change	<100	<100	37.14	0.439
	Humghcsa	66495	29717	128	-0.795323	3.590646	No Change	<100	<100	21.86	0.219	29945	128	-0.809038	3.618077	No Change	<100	<100	21.70	0.219
	Humhbb	73308	32996	128	-0.807388	3.614776	No Change	<100	<100	23.62	0.219	32926	128	-0.803569	3.607137	No Change	<100	<100	23.68	0.219
	Humhdabcd	58864	26470	128	-0.80742	3.614841	No Change	<100	<100	18.29	0.219	26486	128	-0.808508	3.617015	No Change	<100	<100	19.12	0.164
	Humdystrop	38770	17396	128	-0.807996	3.615992	No Change	<100	<100	12.41	0.109	17440	128	-0.812535	3.625071	No Change	<100	<100	12.58	0.109
	Humhprtb	56737	25503	128	-0.807004	3.614008	No Change	<100	<100	18.73	0.164	25557	128	-0.810811	3.621623	No Change	<100	<100	18.79	0.219
	Vaccg	191737	85123	128	-0.778499	3.556997	No Change	<100	<100	55.60	0.604	86121	128	-0.799319	3.598638	No Change	<100	<100	61.04	0.604
	Hehcmvcg	229354	102550	128	-0.790734	3.581468	No Change	<100	<100	73.35	0.769	103304	128	-0.803884	3.607768	No Change	<100	<100	73.62	0.714
	Average					3.592744									3.614442					
Reverse Orientation	Mtpacga	100314	44692	128	-0.787188	3.574376	No Change	<100	<100	29.06	0.329	45230	128	-0.808641	3.617282	No Change	<100	<100	33.02	0.329
	Mpomtcg	186608	83780	128	-0.798594	3.597188	No Change	<100	<100	57.41	0.604	84132	128	-0.806139	3.612278	No Change	<100	<100	59.83	0.604
	Chntxx	155844	70090	128	-0.802264	3.604528	No Change	<100	<100	46.75	0.494	70178	128	-0.804522	3.609045	No Change	<100	<100	48.84	0.494
	Chmpxx	121024	53700	128	-0.779085	3.55817	No Change	<100	<100	34.94	0.384	54312	128	-0.799313	3.598625	No Change	<100	<100	37.19	0.384
	Humghcsa	66495	29783	128	-0.799293	3.598586	No Change	<100	<100	21.64	0.219	29951	128	-0.809399	3.618798	No Change	<100	<100	21.20	0.219
	Humhbb	73308	32770	128	-0.795056	3.590113	No Change	<100	<100	23.40	0.219	33074	128	-0.811644	3.623288	No Change	<100	<100	23.84	0.219
	Humhdabcd	58864	26260	128	-0.79315	3.586301	No Change	<100	<100	19.12	0.219	26518	128	-0.810682	3.621365	No Change	<100	<100	19.01	0.219
	Humdystrop	38770	17404	128	-0.808821	3.617643	No Change	<100	<100	12.52	0.109	17420	128	-0.810472	3.620944	No Change	<100	<100	12.63	0.164
	Humhprtb	56737	25445	128	-0.802915	3.60583	No Change	<100	<100	18.02	0.219	25655	128	-0.81772	3.635441	No Change	<100	<100	17.96	0.274
	Vaccg	191737	85817	128	-0.792977	3.585954	No Change	<100	<100	57.41	0.604	86283	128	-0.802698	3.605397	No Change	<100	<100	61.64	0.604
	Hehcmvcg	229354	102002	128	-0.781177	3.562353	No Change	<100	<100	74.94	0.714	103238	128	-0.802733	3.605466	No Change	<100	<100	74.50	0.769
	Average					3.589186									3.61527					
Complement Orientation	Mtpacga	100314	44784	128	-0.790857	3.581713	No Change	<100	<100	34.28	0.384	45220	128	-0.808242	3.6164842	No Change	<100	<100	32.19	0.329
	Mpomtcg	186608	83484	128	-0.792249	3.584498	No Change	<100	<100	62.04	0.604	84468	128	-0.813341	3.6266827	No Change	<100	<100	58.46	0.604
	Chntxx	155844	69900	128	-0.797387	3.594774	No Change	<100	<100	50.05	0.494	70204	128	-0.80519	3.6103796	No Change	<100	<100	50.38	0.494
	Chmpxx	121024	53886	128	-0.785233	3.570465	No Change	<100	<100	37.08	0.384	54484	128	-0.804997	3.6099947	No Change	<100	<100	37.08	0.384
	Humghcsa	66495	29717	128	-0.795323	3.590646	No Change	<100	<100	21.70	0.219	29945	128	-0.809038	3.6180765	No Change	<100	<100	21.70	0.219
	Humhbb	73308	32996	128	-0.807388	3.614776	No Change	<100	<100	23.46	0.219	32926	128	-0.803569	3.607137	No Change	<100	<100	23.73	0.219
	Humhdabcd	58864	26470	128	-0.80742	3.614841	No Change	<100	<100	18.13	0.219	26484	128	-0.808372	3.6167437	No Change	<100	<100	19.12	0.219
	Humdystrop	38770	17396	128	-0.807996	3.615992	No Change	<100	<100	12.25	0.109	17440	128	-0.812535	3.6250709	No Change	<100	<100	12.52	0.164
	Humhprtb	56737	25503	128	-0.807004	3.614008	No Change	<100	<100	10.73	0.164	25557	128	-0.810811	3.6216226	No Change	<100	<100	18.73	0.164
	Vaccg	191737	85123	128	-0.778499	3.556997	No Change	<100	<100	55.71	0.604	86121	128	-0.799319	3.5986377	No Change	<100	<100	61.09	0.604
	Hehcmvcg	229354	102550	128	-0.790734	3.581468	No Change	<100	<100	73.35	0.714	103304	128	-0.803884	3.6077679	No Change	<100	<100	73.68	0.714
	Average					3.592744									3.614418					
Reverse Complement Orientation	Mtpacga	100314	44692	128	-0.787188	3.5743765	No Change	<100	<100	29.175	0.274	45230	128	-0.808641	3.6172817	No Change	<100	<100	33.24	0.32967
	Mpomtcg	186608	83780	128	-0.798594	3.5971877	No Change	<100	<100	57.52	0.604	84132	128	-0.806139	3.6122781	No Change	<100	<100	59.89	0.604
	Chntxx	155844	70090	128	-0.802264	3.6045276	No Change	<100	<100	46.75	0.494	70178	128	-0.804522	3.6090449	No Change	<100	<100	49.06	0.494
	Chmpxx	121024	53700	128	-0.779085	3.5581703	No Change	<100	<100	35	0.384	54312	128	-0.799313	3.5986251	No Change	<100	<100	32.25	0.439
	Humghcsa	66495	29783	128	-0.799293	3.5985864	No Change	<100	<100	21.70	0.219	29951	128	-0.809399	3.6187984	No Change	<100	<100	21.20	0.219
	Humhbb	73308	32770	128	-0.795056	3.5901129	No Change	<100	<100	23.46	0.274	33074	128	-0.811644	3.623288	No Change	<100	<100	24.01	0.274
	Humhdabcd	58864	26260	128	-0.79315	3.5863006	No Change	<100	<100	19.06	0.219	26518	128	-0.810682	3.6213645	No Change	<100	<100	19.06	0.219
	Humdystrop	38770	17404	128	-0.808821	3.6176425	No Change	<100	<100	12.58	0.109	17420	128	-0.810472	3.620944	No Change	<100	<100	12.63	0.109
	Humhprtb	56737	25445	128	-0.802915	3.6058304	No Change	<100	<100	18.02	0.219	25655	128	-0.81772	3.6354407	No Change	<100	<100	18.02	0.164
	Vaccg	191737	85817	128	-0.792977	3.5859537	No Change	<100	<100	57.63	0.604	86283	128	-0.802698	3.605397	No Change	<100	<100	61.81	0.604
	Hehcmvcg	229354	102002	128	-0.781177	3.5623534	No Change	<100	<100	75.05	0.714	103238	128	-0.802733	3.6054658	No Change	<100	<100	74.94	0.714
	Average					3.5891856									3.6152662					

Table-I : Showing the Compression ratio, rate, selective encryption and speed for the DNA sequences



Graph shown the Compression rate & ratio w.r.t. normal orientation(cellular DNA sequences)

Graph shown the Compression rate & ratio w.r.t. normal orientation(artificially DNA sequences)

Graph shown the Compression rate & ratio w.r.t. reverse orientation(cellular DNA sequences)

Graph shown the Compression rate & ratio w.r.t. reverse orientation(artificially DNA sequences)

Graph shown the Compression rate & ratio w.r.t. complement orientation(cellular DNA sequences)

Graph shown the Compression rate & ratio w.r.t. complenet orientation(artificially DNA sequences)

Graph shown the Compression rate & ratio w.r.t. reverse complement orientation(cellular DNA sequences)

Graph shown the Compression rate & ratio w.r.t. reverse complement orientation(artificially DNA sequences)

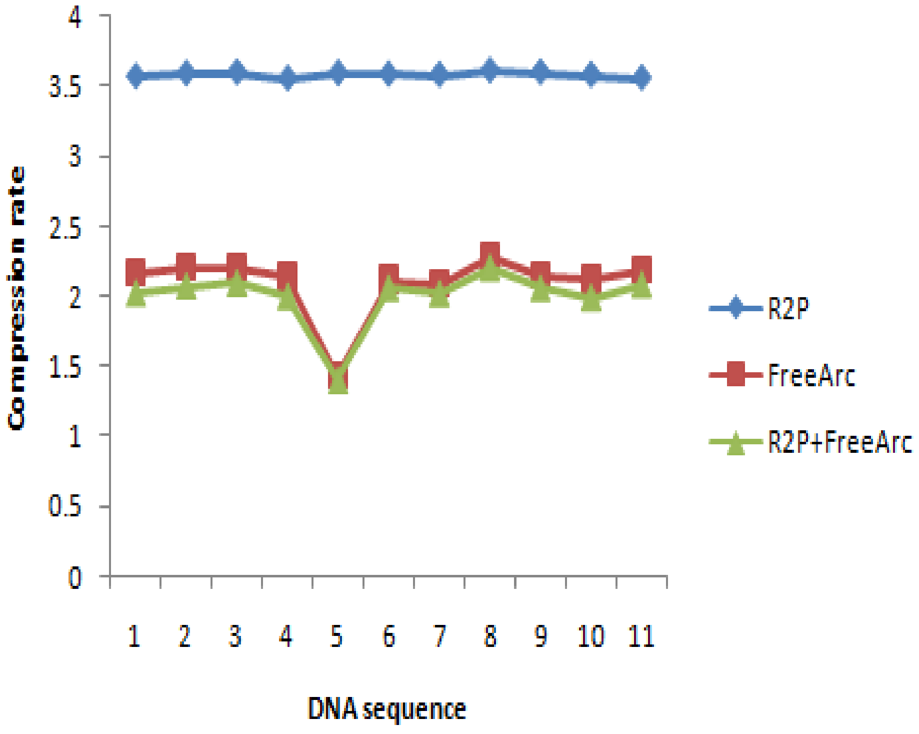
Graph-I : Shown the Compression rate both in cellular & artificial DNA sequences in above algorithm mentioned table1

Sequence Orientation	Sequences name	Sequence size	Cellular DNA Sequence											
			AES algorithm				DES Algorithm				RSA Algorithm			
			Compression ratio	Compression rate(bits /base)	Encode Time	Decode Time	Compression ratio	Compression rate(bits /base)	Encode Time	Decode Time	Compression ratio	Compression rate(bits /base)	Encode Time	Decode Time
Normal Orientation	MTPACGA	100314	-0.785753	3.571505	33	<100	-0.785753	3.571505	33	<100	-0.785753	3.571505	33	<100
	MPOMTCG	186608	-0.789505	3.579011	62	<100	-0.789505	3.579011	62	<100	-0.789505	3.579011	62	<100
	CHNTXX	155844	-0.794102	3.588204	51	<100	-0.794102	3.588204	51	<100	-0.794102	3.588204	51	<100
	CHMPXX	121024	-0.781002	3.562004	38	<100	-0.781002	3.562004	38	<100	-0.781002	3.562004	38	<100
	HUMGHCSA	66495	-0.787623	3.575246	22	<100	-0.787623	3.575246	22	<100	-0.787623	3.575246	22	<100
	HUMHBB	73308	-0.800404	3.600808	25	<100	-0.800404	3.600808	25	<100	-0.800404	3.600808	25	<100
	HUMHDABCD	58864	-0.798722	3.597445	19	<100	-0.798722	3.597445	19	<100	-0.798722	3.597445	19	<100
	HUMDYSTRO	38770	-0.79479	3.58958	13	<100	-0.79479	3.58958	13	<100	-0.79479	3.58958	13	<100
	HUMHPRTB	56737	-0.79798	3.59596	19	<100	-0.79798	3.59596	19	<100	-0.79798	3.59596	19	<100
	VACCG	191737	-0.775828	3.551657	58	<100	-0.775828	3.551657	58	<100	-0.775828	3.551657	58	<100
	HEHCMVCG	229354	-0.788502	3.577003	76	<100	-0.788502	3.577003	76	<100	-0.788502	3.577003	76	<100

Table –II : shown the encryption result of AES,DES & RSA

Sequence orientation	Sequence Name	Sequence Size	R^2P		FreeArc		R^2P+FreeArc		Improvement
			Compression ratio	Compression rate(bits /base)	Compression ratio	Compression rate(bits /base)	Compression ratio	Compression rate(bits /base)	
Reverse Orientation	MTPACGA	100314	-0.787188	3.574376	-8.10056	2.16201	-1.39362	2.02787	4.25%
	MPOMTCG	186608	-0.798594	3.597188	-10.1946	2.20389	-3.8755	2.07751	
	CHNTXX	155844	-0.802264	3.604528	-9.90221	2.19804	-4.91517	2.09830	
	CHMPXX	121024	-0.779085	3.55817	-6.6334	2.13266	-0.17517	2.00350	
	HUMGHCSA	66495	-0.799293	3.598586	28.48184	1.43036	29.97368	1.40052	
	HUMHBB	73308	-0.795056	3.590113	-5.53828	2.11076	-3.19747	2.06394	
	HUMHDABCD	58864	-0.79315	3.586301	-3.94808	2.07896	-1.11443	2.02228	
	HUMDYSTROP	38770	-0.808821	3.617643	-13.9644	2.27928	-10.4669	2.20933	
	HUMHPRTB	56737	-0.802915	3.60583	-6.97076	2.13941	-3.16372	2.06327	
	VACCG	191737	-0.792977	3.585954	-5.676	2.11352	0.188279	1.99623	
	HEHCMVCG	229354	-0.781177	3.562353	-9.17795	2.18355	-4.53884	2.09077	
	Average	----	----	3.589186		2.09386	-----	2.00487	

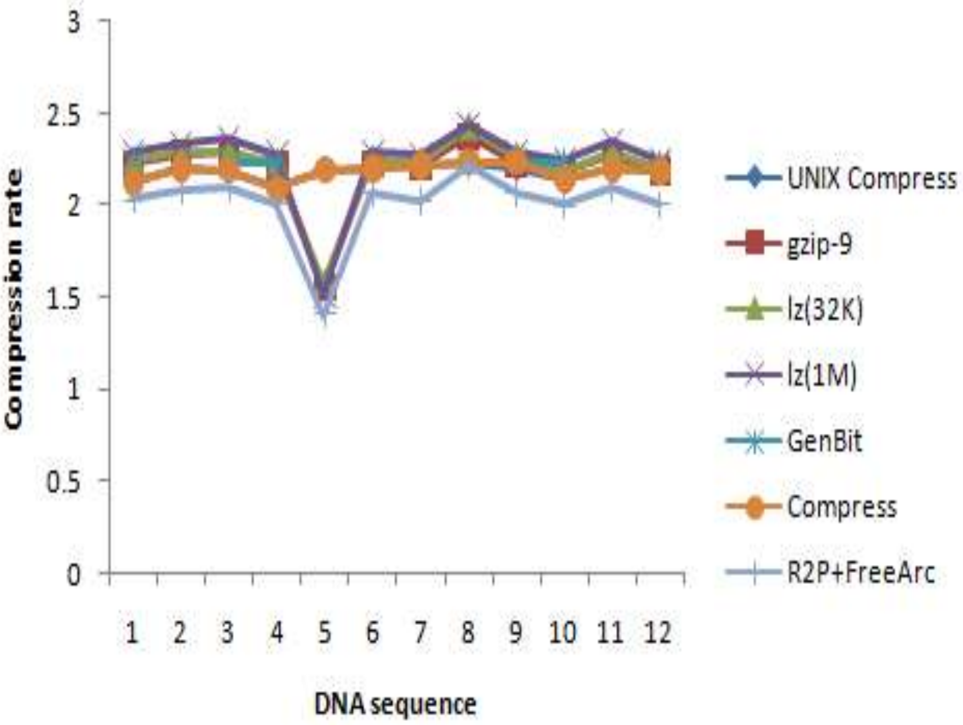
Table-III : Comparison of Compression rate



Graph-II: Line chart Shows the comparison of compression ratio of above algorithm in table1II

Data set	Sequence Name	Base pair/ File size	UNIX Compress	gzip-9	lz(32K)	lz(1M)	GenBit	Compress	R^2P+FreeArc	Improvement over lz(1M)
Data set-I	MTPACGA	100314	2.12	2.232	2.249	2.285	2.243	2.12	2.02787	10.37%
	MPOMTCG	186608	2.20	2.280	2.289	2.326	---	2.20	2.07751	
	CHNTXX	155844	2.19	2.291	2.300	2.352	2.232	2.19	2.09830	
	CHMPXX	121024	2.09	2.220	2.234	2.276	2.225	2.09	2.00350	
	HUMGHCSA	66495	2.19	1.551	1.580	1.513	---	2.19	1.40052	
	HUMHBB	73308	2.20	2.228	2.255	2.286	2.226	2.20	2.06394	
	HUMHDABCD	58864	2.21	2.209	2.241	2.264	---	2.21	2.02228	
	HUMDYSTROP	38770	2.23	2.377	2.427	2.432	2.234	2.23	2.20933	
	HUMHPRTB	56737	2.20	2.232	2.269	2.287	2.238	2.23	2.06327	
	VACCG	191737	2.14	2.190	2.194	2.245	2.237	2.14	1.99623	
	HEHCMVCG	229354	2.20	2.279	2.286	2.344	--	2.20	2.09077	
	Average	----	2.1790	2.189	2.211	2.237	2.2335	2.18	2.00487	

Table-IV : Comparison of Compression rate



Graph-III: Line chart Shows the comparison of compression ratio of above algorithm in table1V

III. RESULT DISCUSSION

Normal sequence is highly compressible than reveres, complement and reverse complement. Cellular DNA sequences compression rate and ratio are distinguishable different due to each sequence that come into different sources (showing in the graph-I) where as artificial DNA sequences compression rate and ratio are same in all time in all data sets. The AES, DES & RSA encryption algorithm tested on normal cellular DNA sequences only. Also it was showing that internal R^2P matching patter for cellular DNA sequences are same in all type of sources and library file plays a key role in finding similarities or regularities in DNA sequences. Output file contain encrypted data with unmatched a, u, g and c so, it can provide high information security which is very important for data protection over transmission point of view & to protect nucleotide sequence in a particular source.

IV. CONCLUSION

This compression algorithm gives a good model for compressing DNA sequences that reveals the true characteristics of DNA sequences and very useful in database storing. This method is fails to achieve higher compression rate & ratio than others standard method, but it has provide very high information security.
Important observation are :

a) R^2P substring length vary from 2 to 5 and no match found in case the substring length becoming six or more. The substring length, three is highly compressible over substring length of four and above.

b) The cellular DNA sequence encode codon/amino acid, here library file of size three are play key role to formation of codon table.

c) This algorithm provide the better data security than other methods. If apply security directly on the cellular DNA sequence, getting very low label security because DNA sequence contain only four bases, anyone can hack the data by trial an error methods where as our result show that after compression it has created two separate file, first one is compress data contain 256 different characters second, file is library life, which is also contains more than four characters. At the time of transmission if two files are transmit one by one it is very hard to hack the data. The compressed output contains more characters than input file, in that situation apply selective encryption technique, enjoy strong of information security and selection encryption options are more.

d) Compressing the genome sequence will help to increase the effect of their uses. Speed of encryption and security levels are two important measurements for evaluating any encryption system.

e) The R^2P technique convert the DNA sequence into 256 ASCII characters with unmatch a,t,g and c, in that situation the Huffman's & two bit encoding algorithm is easily apply on DNA sequences.

f) No change in file size before and after selection encryption process applied.

FUTURE WORK

We try to reduce the time complexity, improve compression rate & security.

ACKNOWLEDGEMENT

Above all, author are grateful to all our 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 our PCs.

REFERENCES

[1]. International nucleotide sequence database collaboration, (2013),[Online]. Available: http://www.insdc.org.

[2]. Karsch-Mizrachi, I., Nakamura, Y., and Cochrane, G., 2012, The International Nucleotide Sequence Database Collaboration, Nucleic Acids Research, 40(1), 33–37.

[3]. Deorowicz, S., and Grabowski, S., 2011, Robust relative compression of genomes with random access, Bioinformatics, 27(21), 2979–2986.

[4]. Brooksbank, C., Cameron, G., and Thornton, J., 2010, The European Bioinformatics Institute's data resources, Nucleic Acids Research, vol. 38, 17-25.

[5]. Shumway, M., Cochrane, G., and Sugawara, H., 2010, Archiving next generation sequencing data, Nucleic Acids Research, vol. 38, 870-871.

[6]. Kapushesky, M., Emam, I., Holloway, E., et al. , 2010, Gene expression atlas at the European bioinformatics institute, Nucleic Acids Research, 38(1), 690-698.

[7]. 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.

[8]. 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.

[9]. Mr Deepak Harbola1 et al. State of the art: DNA Compression Algorithms, International Journal of Advanced Research in Computer Science and Software Engineering, 2013, pp 397-400.

[10]. 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.

[11]. Nour S. Bakr1, Amr A. Sharawi, 'DNA Lossless Compression Algorithms: Review ', American Journal of Bioinformatics Research, 2013 pp 72-81

[12]. S. Grumbach and F. Tahi, "A new challenge for compression algorithms: Genetic sequences," J. Inform. Process. Manage., vol. 30, no. 6, pp. 875-866, 1994.

[13]. X. Chen, S. Kwong and M. Li, "A Compression Algorithm for DNA Sequences and its Applications in Genome Comparison,*Genome Informatics*, 10:52–61, 1999.

[14]. Bell, T.C., Cleary, J.G., and Witten, I.H., *Text Compression*, Prentice Hall, 1990.

[15]. Matsumoto, T., Sadakane, K., and Imai, H., 2000, Biological Sequence Compression Algorithms, Genome Informatics, 2000,pp 43–52.

[16]. Giancarlo, R., Scaturro, D., and Utro, F., 2009, Textual data compression in computational biology: a synopsis, Bioinformatics, 25(13), 1575–1586.

[17]. Nalbantoḡlu, Ö. U., Russell, D.J., and Sayood, K., 2010, Data Compression Concepts and Algorithms and their Applications to Bioinformatics, Entropy, 12(1), 34-52.

[18]. H. Cheng and X. Li, “Partial Encryption of Compressed Images and Video,” IEEE Transactions on Signal Processing, 48(8), 2000, pp. 2439-2451.

[19]. Nidhi S Kulkarni et al. Selective Encryption of Multimedia Images, XXXII National System Conference, NSC 2008, pp 467-470

[20]. Gehani A, LaBean T H, Reif J H. DNA-based cryptography. In: DNA Based Computers V. Providence, USA: American Mathematical Society, 2000. 233–249

[21]. Clelland C T, Risca V, Bancroft C. Hiding messages in DNA microdots. Nature, 1999, 399: 533–534

[22]. Leier A, Richter C, Banzhaf W, et al. Cryptography with DNA binary strands. Biosystems, 2000, 57: 13–22

[23]. Xiao G Z, Lu M X, Qin L, et al. New field of cryptograhy: DNA cryptography. Chinese Sci Bull, 2006, 51: 1413–1420

[24]. Lu M X, Lai X J, Xiao G Z, et al. A symmetric-key cryptosystem with DNA technology. Sci China Ser F-Inf Sci, 2007, 50: 324–333

[25]. Ma,B., Tromp,J. and Li,M. (2002) PatternHunter—faster and more sensitive homology search. Bioinformatics, 18, 440–445.1698

[26]. Deepak singh chouhan,R.P.Mahajan, “An architectural framework for encryption and generation of digital signature using DNA cryptography”,International Conference on Computing for Sustainable Global Development(INDIACom),2014.

[27]. Grasha Jacob,A.Murugan, “DNA based cryptography An overview & analysis”, International Journal of emerging science 3(1),(36- 42) march 2013

[28]. L.Xuejia,L.Mingxin,Q.Lei,H.Junsong and F.Xinven, “Assymmetric encryption and signature method with DNA technology”,science in china: Information Science,vol.53,2010,pp.506-514.

[29]. Xing wang,Qiang zang “DNA computing based cryptography”,Fourth international conference on bio-inspired computing BIC-TA2009 pp1-3.

[30]. Zheng Zhang,Xiaolong shi,JieLiu,”A method to encrypt information with DNA computing”,3rd international conference on bio-inspired computing.Theories and application 2008.pp.155-160

[31]. G.Cui,L.Cuiling,L.Haobin and L.Xiaoguang, “DNA computing and its application to information security field”,IEEE 5 th international conference in National computation,Tianjian china,Aug 2009,pp.148-152.

[32]. William stallings, “cryptography and network security principles and practise”, 5th edition 2011

[33]. Bell, T.C., Cleary, J.G., and Witten, I.H., Text Compression, Prentice Hall, 1990.

[34]. Atul Kahate(2008), Cryptography and Network Security, Tata McGraw-Hill Publicating Company Ltd.

[35]. Jie Liu et al. A Fixed-Length Coding Algorithm for DNA Sequence Compression (Draft, Using Bioinformatics LATEX template), 2005, pp1-3

[36]. [36] Sheng Bao et al. A DNA Sequence Compression Algorithm Based on LUT and LZ77, Conference: Signal Processing and Information Technology, 2005, pp 1-14

[37]. [37]Alok Kumar Shukla et al., Data Encryption and Decryption using Modified RSA Cryptography Based on Multiple Public Keys and ‘n’ Prime Number, International Journal Of Engineering Sciences & Research Technology, 2014, pp 713-720

[38]. [38]Singh et al., Comparative Analysis Of Cryptographic Algorithms,International Journal of Advanced Engineering Technology, 2013, pp 16-18

[39]. [39] Chem,X., et al., DNAcompress : fast and effective dna sequence compression, Bioinformatics, 2002, 18, 1696-1698

[40]. [40]sansan.phy.ncu.edu.tw/~hclee/ref/CompressingDNAsequence.pdf

[41]. [41]Matsumoto et al.,Can General-Purpose Compression Schemes Really Compress DNA Sequences?, Current in Computational Molecular Biology, 2000, pp 76-77

[42]. [42]P.Raja Rajeswari et al.,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, 2010, pp 181-191

[43]. [43] http://freearc.org/

[44]. [44] Xin Chen, San Kwong and Mine Li, “A Compression Algorithm for DNA Sequences Using Approximate Matching for Better Compression Ratio to Reveal the True Characteristics of DNA”, IEEE Engineering in Medicine and Biology, 2001, pp 61-66