British Journal of Mathematics & Computer Science
11(5): 1-11, 2015, Article no.BJMCS.19561
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Hiding Secret Information in DNA Sequences Using Silent Mutations

Amal Khalifa[1,2*] and Safwat Hamad[2]

[1]Department of Computer Science, College of Computer and Information Sciences, Princess Nora Bint Abdulrehman University, Riyadh, Saudi Arabia.
[2]Department of Scientific Computing, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt.

Article Information

DOI: 10.9734/BJMCS/2015/19561
Editor(s):
(1) Neculai Andrei, Research Institute for Informatics (ICI), Romania.
Reviewers:
(1) Anonymous, University of Calcutta, India.
(2) Anonymous, Universiti Sains Malaysia, Malaysia.
Complete Peer review History: http://sciencedomain.org/review-history/11335

Original Research Article

Received: 16 June 2015
Accepted: 12 August 2015
Published: 09 September 2015

Abstract

People are always looking for secure methods to protect valuable information against unauthorized access or use. That's why; disciplines like cryptography and steganography are gaining a great interest among researchers. Although the origin of steganography goes back to the ancient Greeks, recent steganographic techniques hide data into digital media such as sound, images, and videos. However, steganography took a step further to utilize the DNA as a carrier of secret information. DNA-based steganography techniques could be for either authentication or data storage. In this paper, we propose an original idea of hiding data in DNA or RNA called LSBase (Least Significant Base Substitution). It uses a remarkable property of codon redundancy to introduce silent mutations into DNA sequences. In this way, the DNA sequence can be altered without affecting the type or the structure of protein it produces. When compared with other techniques, the proposed algorithm showed to be the only blind technique that is capable of conserving the functionality of the carrier DNA while maintaining a reasonable data payload.

Keywords: Information hiding; steganography; DNA; codons; codon redundancy; mutation.

1 Introduction

The significant advances in computers and communication created a strong need to secure valuable information against unauthorized access. Cryptography is one of the earliest techniques and yet efficient solutions to many security problems. Contrary to encryption, which encrypts the messages to make them

**Corresponding author: E-mail: askhalifa@pnu.edu.sa;*

incomprehensible, steganography hides the messages into some innocent looking cover media in such a way that the resultant "stego" media is perceptually indistinguishable from the original one. However, cryptography protects the data only during the transmission stage, and this protection cannot be effective after subsequent decryption while steganography techniques actually merge the message in another such that it is almost impossible to discover the concealed information or even suspect its existence.

The first confirmed use of steganography is in "Histories" of Herodotus and dates back to the fifth century BC: a certain Histio, wanting to make secret contact with his superior, the tyrant Aristagoras of Miletus, chose a believing slave, shaved his head and wrote in skin the message you wanted to send. He waited for the hair to grow and ordered the slave to meet Aristagoras with the instruction that they should shave their hair.

For the past two decades, researchers are focusing on new techniques for hiding information in binary files like text, images [1], audio tracks, video, file systems [2], networks [3] and more interestingly 3D Objects [4]. Recently, researchers looked at DNA as a coding medium that can contain information just like a disk or RAM. This makes biological data; stored in DNA, a perfect candidate for hiding messages.

Genes are the equivalent of words that, together, make up the "recipe", or genome, of a living being. Genes are made of DNA or deoxyribonucleic acid. The DNA, in turn, is formed by pairs of molecules called nitrogenous bases. The nitrogenous bases are of four types: Adenine (A), thymine (T), cytosine (C) and guanine (G). What the researchers did was to group the three nitrogenous bases in three. Each sequence was equivalent to a letter of the alphabet. For example, the letter sequence to the equivalent cytosine, adenine and guanine, or CGA. The letter B was coded as CCA (cytosine, adenine and cytosine), and so on, until the entire alphabets, punctuation marks, and the numbers had its equivalent in trios of nitrogenous bases.

During the past few years, researchers gave a great attention to encoding messages amongst millions of other similar looking DNA molecules. This inventive technology can be used for protecting genetic discoveries such as genetically engineered organisms, gene therapy, transgenic crops, tissue cloning, and DNA computing [5,6,7]. Another interesting application for information hiding in DNA is the bacteria-based storage systems. Jiao et al. [8] proved that digital data can be stored in the genome of a living organism for thousands of years while protecting it against nuclear explosions.

One of the earliest tries in applying information hiding schemes on biological DNA appeared in 1999 [9]. In 1999, Catherine Taylor Clelland, Viviana Risca and Carter Bancroft published in Nature "Hiding messages in DNA microdots" (hiding messages in DNA microdots). In fact, any genetic material is formed by chains of four nucleotides (Adenine, Cytosine, Guanine, and Thymine) that can compare to a four letter alphabet: A, C, G and T. In addition, scientists are currently capable of producing DNA strands with a predetermined set of nucleotides. Nothing prevents you assign to a group of three nucleotides an alphabet letter, number or punctuation marks (e.g. "A" = CGA, "B" = CCA, etc.) and compose a "genetic message." To cover the tracks, power would mix a few other random nucleotide sequences. The result is visible only under a microscope. The authors synthesized a DNA strand that encrypts the secret message. The message sequence is then copied and camouflaged within an enormous number of similarly sized fragments of human DNA. In their proof-of-principle experiment, the researchers dropped a small quantity of a DNA-containing solution onto a little dot printed on filter paper. They cut out the dot, taped it over the period in a typed letter, and mailed the letter. The recipient succeeded to recover the secret message after laboratory analysis. Furthermore, they proved that such DNA-based steganography is essentially unbreakable [10].

Another category of the proposed steganography algorithms regards DNA sequences as a pure information coding model. This kind of algorithms has shown provide an efficient solution for a number of applications. In the field of cryptography, for example, several algorithms have been proposed to forms encrypted data as sequences of DNA nucleotides [11]. In addition, DNA-based steganography techniques have been utilized to hide the encryption key into a DNA sequence and hence the parties of the communication will be able to share it through an unsecure channel [12]. Other examples include: data hiding in text documents [13], and smart information management [14].

In this context, a number of algorithms have been developed to embed messages into DNA sequences for the purpose of data hiding. One of these methods applied the arithmetic encoding on binary messages using the feature of codon redundancy [15]. The length of the resultant stego-DNA depends on the precision of the embedded fraction and obviously affects the accuracy of the blind retrieval process. More methods were introduced in [16]. The authors claimed that there is almost no difference between a real DNA sequence and a fake one. According to their proposed model, the sender, and the receiver agree on a reference sequence before the transmission takes place. The sender then embeds the secret message into that sequence producing another DNA sequence. Thus, the retrieval process cannot be done without the help of the reference sequence. However, there are two problems with this set of techniques. First, the extraction process cannot be done blindly. That is; sender and the receiver have secretly to communicate both the reference sequence and the Stego-DNA. In fact, communicating such information could be suspicious and reveal the secrecy of the steganographic channel itself. Secondly, the reference sequence is randomly modified without any consideration of the biological interpretation of the DNA information. Therefore, the resultant DNA sequence may encode for an entirely different protein and hence can't be employed in any biological operation instead of the reference sequence.

In this paper, we will present a novel technique for embedding secret information in DNA. The algorithm utilizes the idea of codon degeneracy and silent mutations in order to embed the secret information into the genetic sequence without changing its functionality. Furthermore, the extraction process can be done blindly without any need to the reference sequence. Hence, the sender and the receiver don’t have to exchange anything in advance but the secret key. Furthermore, it is important to distinguish between live DNA and chemical DNA. Cellular DNA (or live DNA) ultimately ends up in a living organism [8], while chemical DNA sequences are just chemical messages such as DNA computing solutions and are not intended to undergo processing by cellular machinery. For the purpose of this paper, DNA is regarded as a theoretical construct of chemical DNA where changes can be wherever necessary to hide data.

The rest of the paper is as follows: the next section gives a glimpse on the genetic code which forms the basis of the proposed algorithm illustrated later in section 3. Next, a detailed discussion of the experimental results is given in section 4. Finally, we summarize our conclusions in Section 5.

2 The Genetic Code

Each cell of the human body contains a nucleus, in which the genetic material known as Deoxyribose Nucleic Acid (DNA) is into chromosomes. The DNA molecule is structured as a double helix that is made up of building blocks called nucleotides. Nucleotides can contain either a purine or a pyrimidine base. The purine bases are adenine (A) and guanine (G), while the pyrimidines are thymine (T) and cytosine (C). A pairs with T and G pairs with C. The sequence of DNA bases is read manually after exposure of bases identified in an X-ray film. To do this four reaction tubes are prepared, each containing the template DNA, DNA polymerase and a primer. Each tube receives a small amount of dNTP (dATP, dTTP, dGTP and dCTP) with one of the dNTPs (dATP, dTTP, dGTP or dCTP) labeled with radioactive phosphorus.

In any of the 4 tubes are produced various chain lengths, each corresponding to the point at which the corresponding ddNTP is incorporated into this tube and causing the end of the chain growth. Samples of each tube are subjected to electrophoresis and subsequent exposure to X-ray film so that the position of the corresponding bases can be determined by reading along the four gel column.

Hence, DNA is viewed as the sequence of base pairs: AAGTCGATCGATCATCGAT. Three adjacent nucleotides constitute a unit known as the codon that codes for an amino acid. Through a long and complicated process; called the Central Dogma (shown in Fig. 1), these codons are read and eventually translated into chains of amino acids, which form a protein (Genetic code). In other words, genes code for the amino acid sequence of protein molecules where the arrangement of the amino acids dictates the structure and function of the resultant protein. Thus, the genetic code defines a regular mapping of two entirely different kinds of molecules: nucleic acid bases; in DNA or RNA, and amino acids that form the building blocks of protein molecules. So, the mRNA bases (C, A, U, G) can be viewed as triplets that are mapped to individual amino acids.

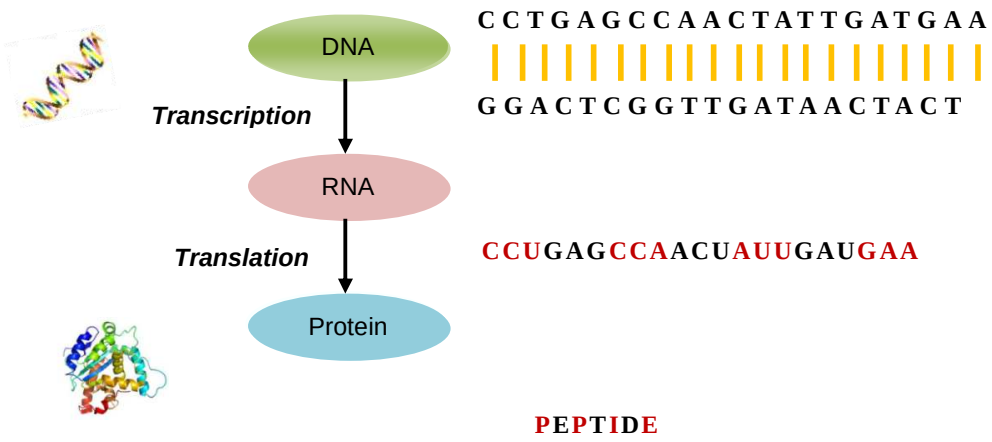


Fig. 1. The central dogma

In Fig. 2, the codons are organized as a quad-tree. In this codon tree, the nodes have no labels and the edges are annotated with the bases U, C, A, and G. With a three-level depth, the codon tree offers a space for exactly 64 (4^3) unique codes at the leave nodes. However, for the sake of compact representation, the codons are combined into 16 groups of four surrounded by a rectangle. An edge-wise traversal of the tree is starting from the root, taking one of the paths until a leaf node reached dictates a codon. With 64 distinctive ways to do this, you will end up generating all the possible combinations of codons.

The three-letter abbreviations (shown in figure) such as “Phe” and “Leu” indicate the types of amino acid molecules. Although we have 64 different codons, there are only 20 amino acids that build up proteins. This means that some amino acids are coded by more than one codon in a feature called degeneracy [17]. This codon redundancy can be effectively exploited to change the genetic sequence without changing its functionality.

3 The LSBase Algorithm

Viewing DNA as a coding medium means that the information contained in DNA strands can be interpreted, copied, and maybe modified or mutated. However, rather than a binary representation of zeroes and ones, DNA data consists of sequences of 4 nucleic acids (A, T, G, and C).Taking this analogy a step further, would make DNA a suitable medium for hiding secret data. In fact, information hiding can be accomplished by intelligently altering some portions of the cover such that the data is not functionally or perceptibly altered. The proposed algorithm; called LSBase, applies an efficient way to hide secret information in DNA sequences without affecting the type or structure of the protein it codes for. The following subsections will illustrate the details of both the embedding and extraction processes. The data payload offered by the algorithm will be as well.

3.1 The Embedding Module

The question now is how can we change the genetic sequence of DNA and still produce the same protein? Well, the answer is Silent mutations. Silent mutations are as DNA mutations that do not result in a change in the amino acid sequence of a protein. For example, the sequence: CCTGAGCCAACTA will turn into: CCUGAGCCAACU after the transcription process. During the translation step, this code is processed in triplets to produce the protein chain: Pro-Glu-Pro-Thr-Ile. Changing the third codon to become CCG instead of CCA can be regarded as a silent mutation. That is since both codons code for the (Pro) amino acid, and the resultant protein chain remains unchanged. Silent mutations may occur in a non-coding region (outside of a gene or within an intron), or they may occur within an exon (Silent mutation).

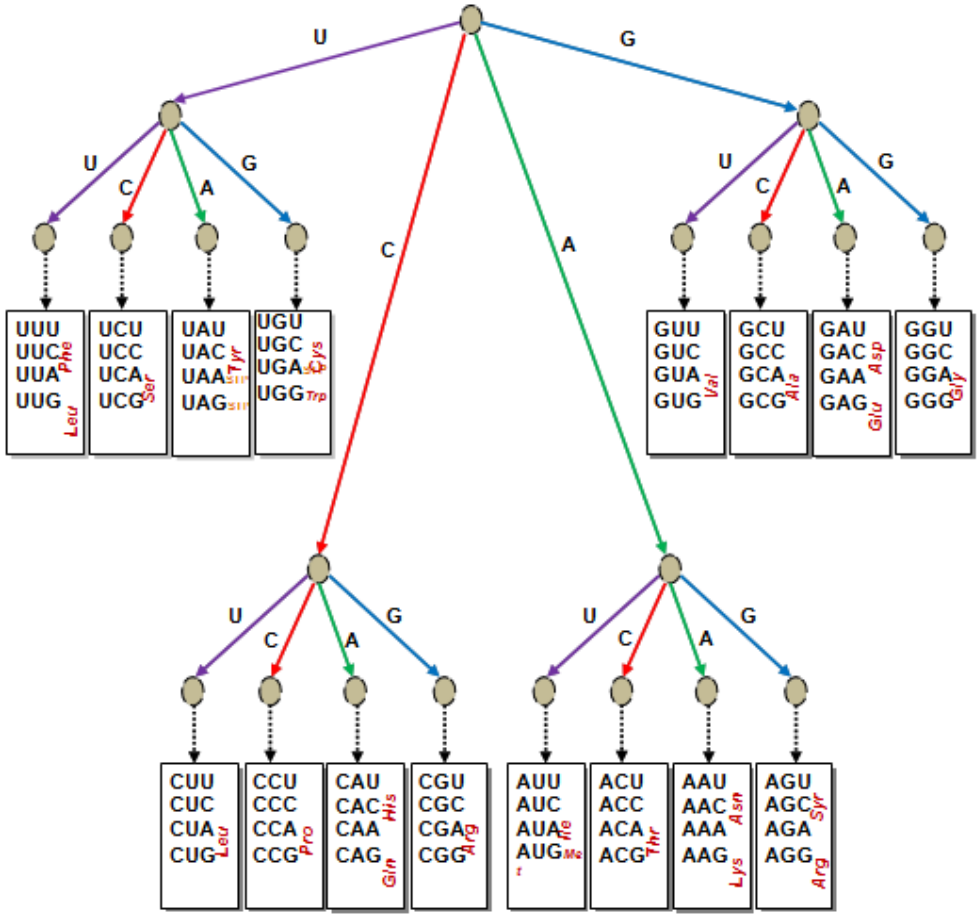


Fig. 2. The codon tree

Looking carefully at the codon tree (shown in Fig. 2) you will discover that the codons inside each group begin with the same two bases. In eight instances, all the four codons in a group specify the same amino acid. In the remaining groups, the two codons that end with the pyrimidines (U and C) often specify one amino acid, whereas the two codons that end with the purines (A and G) specify another. The idea of the proposed algorithm relies on changing the codon's last base while keeping its type (either pyrimidine or purine), in order to code for the same amino acid. In other words, the codon's third nucleotide base will be substituted according to the value of the secret bit as shown in the coding scheme in Table 1. That's why we called this algorithm LSBase, which stands for Lease Significant Base substitution.

Table 1. A binary coding scheme for the lease significant base of codons

Secret bit	Pyrimidines	Purines
0	U	A
1	C	G

The substitution process can be done as explained on all codons except for four cases. That is, tryptophan (Trp) and methionine (Met) have only one codon, so they can't be utilized for embedding at all. The same applies on the stop codon UGA. The fourth case appears with the amino acid Isoleucine (Ile) which is coded by three codons: AUU, AUC, and AUA. The codons AUU and AUC are interchangeable because their least

significant base is pyrimidine while the codon AUA remains singular and hence will be excluded from the embedding process.

The process of embedding starts by dividing the DNA sequence into codon triplets. The least significant nucleotide of each triplet is checked to decide whether it is pyrimidine or purine. In the case of pyrimidine, the least significant base is changed to (U) if the secret bit is (0) otherwise it is replaced by (C). Similarly, the least significant base of purines is substituted by (A) if the embedded bit is (0) and by (G) otherwise. This substitution scheme can be applied uniformly on the whole sequence ignoring the codons UGA, UGG, AUA, or AUG as explained above.

Fig. 3 shows the steps of the hiding process on a sample DNA sequence. Notice that the secret key was used randomly to scatter the message across the DNA sequence. This step is necessary to enhance the security of the technique such that only the one who has the key can know the positions and the order of the nucleotides used for embedding. Furthermore, a simple comparison between the modified sequence and the original one would show that the changes made to the DNA sequence don't affect its amino-acid chain and hence would not change its function.

S : ATG GCG CAG GTA TTA GAC GTA CGC CTA GCT CCA TGG ACC AUA GTT GCC CTG CCC
Msg : 0100011001110101
Key : EGYPT_25JAN

1. Convert ***S*** from DNA to mRNA:
   S =AUG GCG CAG GUA UUA GAC GUA CGC CUA GCU CCA UGG ACC AUA GUU GCC CUG CCC

2. Use Key to shuffle the message bits:
   NewMsg : 0101001001101101

3. Detect codon triplets***:***
   S = AUG |GCG |CAG |GUA |UUA |GAC |GUA |CGC |CUA |GCU |CCA |UGG |ACC | AUA |GUU |GCC |CUG |CCC

4. Apply the LSB substitution:
   S′ = AUG |GCA|CAG |GUA |UUG|GAU|GUA |CGC |CUA |GCU |CCG|UGG |ACU| AUA | GUC|GCC |CUA|CCC

5. Convert ***S′*** back to DNA:
   S′ = ATG GCA CAG GTA TTG GAT GTA CGC CTA GCT CCGTGG ACTATA GTC GCC CTA CCC

Fig. 3. A detailed example illustrating the steps of the proposed hiding process

3.2 The Extraction Module

This part of the algorithm is usually carried out by the receiver, where he/she have to be able to extract the embedded message correctly from the stego-DNA. The steps are the inverse of the embedding process. That is; the DNA string is first converted into RNA triplets that their least significant nucleotide base (LSBase) are checked to discover the hidden bits. That is, if the LSBase is either (U) or (A), then the embedded bit was 0; otherwise it was 1. Once again, the codons UGA, UGG, AUA, or AUG are skipped since they can't hide any data.

Fig. 4 shows the steps of the extraction process of the sample DNA used in the embedding example in Fig. 3. Notice that the extraction process is done "blindly" without the need to refer to the original cover DNA sequence. The only thing that the sender and the receiver must share is the key to make sure that no other party can easily guess or extract the hidden message even if he/she was aware of its existence and/or the algorithm used for embedding. Furthermore, the actual DNA sequences may have hundreds of thousands of nucleotides while the message can be much shorter than that. In this case, not all the codons will carry secret bits. Therefore, header information should be embedded in the message itself in order to provide metadata for the length and maybe the type of the embedded message.

S' = ATG GCA CAG GTA TTG GAT GTA CGC CTA GCT CCGTGG ACTATA GTC GCC CTA CCC
Key : EGYPT_25JAN

1. Convert ***S'*** from DNA to mRNA:
   S' = AUG GCA CAG GUA UUG GAU GUA CGC CUA GCU CCG UGG ACUAUA GUC GCC CUA CCC

2. Detect LSBase of carrier codons***:***
   S' = AUG|GCA|CAG|GUA|UUG|GAU|GUA|CGC|CUA|GCU|CCG|UGG|ACU|AUA| GUC|GCC|CUA|CCC

3. Extract Shuffled message
   ShufMsg : 0101001001101101

4. Use Key to reverse the shuffling process:
   Msg : 0100011001110101

Fig. 4. A detailed example illustrating the steps of the proposed extraction process

3.3 Data Payload

Generally, the payload of a steganographic technique is measured by the maximum size of bits that can be embedded in the cover media. In the case of DNA media, the hiding capacity is measured in bit-per-nucleotide (bpn). Assume that the length of the cover DNA (S) is denoted by |S| reflecting the number of nucleotides composing its sequence. Since the (LSBase) algorithm can hide only one bit per codon, the overall hiding payload of the algorithm can be expressed as follows:

$$Capacity = \frac{size\ of\ message\ in\ bits}{size\ of\ cover\ in\ bases} = \frac{\frac{1}{3} * |S|}{|S|} = \frac{1}{3}\ bpn$$

4 Experimental Results

4.1 Experimentations on Sample DNA

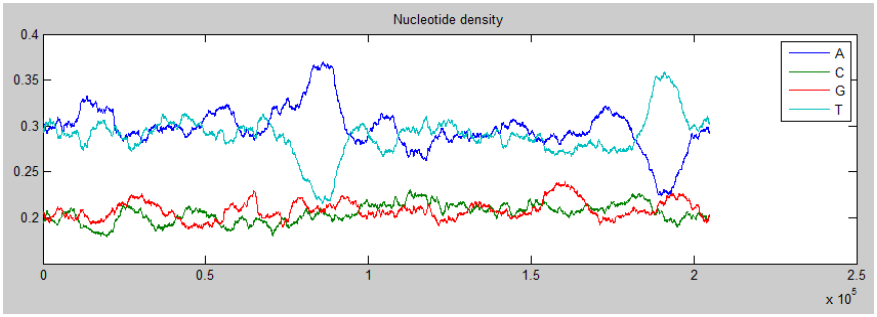
The purpose of this set of experiments is to evaluate the performance of the proposed method. As shown in Table 2, ten DNA sequences were used as test samples. Each sequence is identified by an accession number as drawn from the database of the Gen bank. In addition, the secret message used consists of approximately 3.7 kilobytes of textual data. The code was implemented using Matlab bioinformatics toolbox. In addition, Figs. 5, 6 and 7 were also produced using the Matlab sequence visualization tools.

Table 2. Results of hiding 3.7 kilo bytes of text messages into different DNA sequences

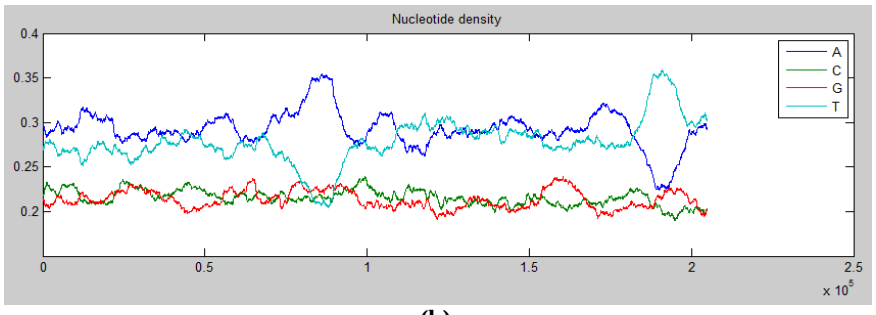
Sequence	Length (bp)	Max capacity (KB)	Unused codons (%)	Actual Capacity (KB)	Changed codons (%)
AC153526	200,117	8.14	10.73	7.27	10.56
AC167221	204,841	8.34	10.91	7.42	11.5
AC171183	211,697	8.61	10.78	7.69	9.92
AC168274	177,020	7.20	10.88	6.41	12.72
AC168276	186,675	7.60	9.74	6.86	10.69
AC166256	181,903	7.40	10.41	6.63	11.04
AC167229	197,711	8.04	11.2	7.14	12.50
AC166259	198,972	8.1	12.03	7.12	12.32
AC153853	194,037	7.9	9.88	7.11	10.9
AC158656	194,657	7.92	12.01	6.97	13.65

In each case, the table lists the maximum capacity offered by the cover sequence (measured in kilobytes) and the percentage of the special case codons that were unsuitable for hiding and hence could not be used for embedding. Knowing the percent of these codons can be helpful for computing the expected actual capacity offered in each case. Furthermore, the numbers listed under the last column reflect the fraction of codons that were changed or substituted by other codons due to the embedding process. The results show that all of the test cases, no more than 14% of the nucleotides composing the sequence has been replaced by others due to the embedding process. Furthermore, the amino-acid structure is strictly maintained which means that both the cover sequence and the stego-sequence still encode for the same protein chain.

Figs. 5, 6 and 7, visualize these results for the sequence AC153526. Notice that the embedding process has slightly changed the composition of the cover sequence with respect to either nucleotides or codons. However, this change did not affect the amino-acid chain it encodes for. This is, in Fig. 7 that depicts a pair-wise alignment of the first 200 amino acids of both the cover and stego-sequences. The alignment verifies a100% matching between the two sequences.



(a)



(b)

Fig. 5. Nucleotide density of the "AC153526" sequence: (a) Before embedding, (b) After embedding

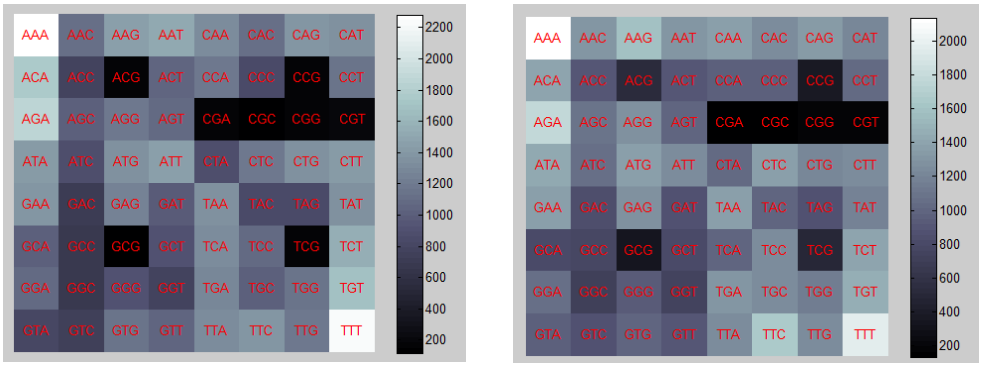


Fig. 6. Codon composition of "AC153526" sequence: (a) Before embedding, (b) After embedding

```
Identities = 200/200 (100%)
001  NSKETDSTASEGMDLLAGANRQRESKPSSSISFI*TAN*RRRTRLEIHLNTSNDLM*LRKKIPY
     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
001  NSKETDSTASEGMDLLAGANRQRESKPSSSISFI*TAN*RRRTRLEIHLNTSNDLM*LRKKIPY

065  TCAQPFVFY*F*M*SS*HKGGHHTLFGFFSTSKF*KSLLARPNSQHL*SQHQLKVTPKLGMELH
     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
065  TCAQPFVFY*F*M*SS*HKGGHHTLFGFFSTSKF*KSLLARPNSQHL*SQHQLKVTPKLGMELH

129  VCGQPWTPKA*RQKNHKFKSSWGCITRPSVSKCPSSTKHFL*TKSIQSPRSYGPWRLPEAGDIM
     ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
129  VCGQPWTPKA*RQKNHKFKSSWGCITRPSVSKCPSSTKHFL*TKSIQSPRSYGPWRLPEAGDIM

193  GVLRPNCS
     ||||||||
193  GVLRPNCS
```

Fig. 7. A pair-wise alignment for the first 600 bases of the original and embedded "AC153526" Sequence

4.2 Comparisons with Other Techniques

In this set of experiments, the proposed method was compared with other DNA-based hiding schemes. The comparison spotted a number of differences among these methods with respect to three data hiding parameters: capacity, function conservation, and blindness. With function conservation, we mean that the amino-acid chain coded by both the cover and embedded sequence should be identical. In addition, the algorithm is considered blind if the embedded data can be without reference to the cover sequence used at the sender side.

As shown in Table 3, although both the methods suggested by [18] allow blind extraction of the hidden data, they still could not preserve the functionality of the embedded sequences. The same goes for the substitution and the insertion methods suggested by [16]. In addition, although they provide a higher capacity than the proposed algorithm, they are not blind. Furthermore, despite the fact that the blind mathematical encoding algorithm proposed in [15] preserves the functionality of the embedded sequence, the LSBase still outperforms it in terms of complexity and precision. That is; the mathematical encoding algorithm embeds the binary message as an approximate binary fraction. This could cause a severe loss of data in the extraction phase.

Table 3. A comparison between the proposed hiding approach and similar ones

Provider	Approach	Capacity (*bpn*)	Functionality conserved?	Blind?
Shimanovsky [15]	Mathematical Encoding	Not defined	√	√
Chang, 2007 [18]	Lossless compression-based	0.78	X	√
	Difference expansion-based	0.11	X	√
Shiu, 2010 [16]	Insertion method	0.58	X	X
	Complementary method	0.07	X	X
	Substitution method	0.82	X	X
The authors	LSBase	0.333	√	√

5 Conclusions

Looking at DNA strands as a coding medium makes it possible for DNA to carry information that can be interpreted, copied, and maybe modified. DNA is up of sequences of 4 nucleic acids adenine (A), thymine (T), guanine (G), and cytosine (C). These nucleotides could be used to encode binary information in such a way that its functionality of producing proteins is not affected.

In this paper, we utilize a simple and yet very interesting characteristic of codon redundancy. The proposed algorithm; LSBase, hides a binary message into a DNA sequence by altering the least significant base of its codons. Depending on the type of the base as well as the hidden bit, the message is embedded in the DNA without actually changing the protein chain it codes for. Furthermore, the algorithm performs a blind extraction on the resultant Stego-DNA. That is; the hidden message can be completely recovered without reference to the original DNA sequence.

When compared with other techniques, the LSBase algorithm proved to outperform other existing techniques in terms of capacity and blindness. Furthermore, the proposed technique showed to be the only blind technique that is capable of conserving the functionality of the carrier DNA while maintaining a reasonable data payload.

The generic nature of the technique presented in this paper makes it applicable to many areas. Examples include protecting intellectual property especially in the field of gene therapy and DNA computing. The presented hiding method may be used as well for the purpose of sequence annotation and cataloguing. Another useful application can be in the field of cryptography. That is, the LSBase method can be plugged into the key encapsulation mechanism (KEM) of a hybrid cryptosystem providing an innovative solution to the key management problem as proposed in [19].

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Cheddad A, Condell J, Curran K, Mc Kevitt P. Digital image steganography: Survey and analyses of current methods. Journal of Signal Processing. 2010;90(3):727-752.

[2] Anderson R, Needham R, Shamir A. The steganographic file system, Proc. International Information hiding Workshop, London, UK. 1998;73-82.

[3] St. J. Murdoch, Lewis S. Embedding covert channels into TCP/IP. Information Hiding. 2005;3727: 247-261.

[4] Wang K, Lavoué G, Denis F, Baskurt A. A comprehensive survey on three-dimensional mesh watermarking. IEEE Transactions on Multimedia. 2008;10(8):1513-1527.

[5] Arita M, Ohashi Y. Secret signatures inside genomic DNA. Biotechnol Prog. 2004;1605–1607.

[6] Barnekow A, Heider D. DNA-based watermarks using the DNA-Crypt algorithm. BMC Bioinformatics. 2007;8(1):176. Available: http://dx.doi.org/10.1186/1471-2105-8-176
DOI: 10.1186/1471-2105-8-176.

[7] Heider D, Pyka M, Barnekow A. DNA watermarks in non-coding regulatory sequences. BMC Res Notes; 2009.

[8] Jiao S, Goutte R. Hiding data in DNA of living organisms. Natural Science. 2009;1(3):181-184.

[9] Clelland CT, Risca V, Bancroft C. Hiding messages in DNA microdots. Nature. 1999;399:533-534.

[10] Risca VI. DNA-based steganography. Cryptologia. 2001;25(1):37–49.

[11] Hamad S. A novel implementation of an extended 8x8 playfair cipher using interweaving on DNA-encoded data. International Journal of Electrical and Computer Engineering (IJECE). 2014;4(1): 93-100.

[12] Torkaman MRN, Kazazi NS, Rouddini A. Innovative approach to improve hybrid cryptography by using DNA steganography. International Journal on New Computer Architectures and Their Applications (IJNCAA) the Society of Digital Information and Wireless Communications. 2012;2(1): 225-236. (ISSN: 2220-9085).

[13] Lin HLD, Kadir A. A novel data hiding method based on deoxyribonucleic acid coding. Computers and Electrical Engineering. 2013;39:1164–1173.

[14] Ogiela MR, Ogiela U. DNA-like linguistic secret sharing for strategic information systems. International Journal of Information Management. 2012;32:175– 181.

[15] Shimanovsky B, Feng J, Potkonjak M. Hiding data in DNA. Information Hiding, Springer. 2002;2578:373-386.

[16] Shiu HJ, Ng KL, Fang JF, Lee RCT, Huang CH. Data hiding methods based upon DNA sequences. Information Sciences. 2010;180:2196–2208.

[17] Crick F. Central dogma of molecular biology. Nature. 1970;227:561–563.

[18] Chang CC, Lu TC, Chang YF, Lee RCT. Reversible data hiding schemes for deoxyribonucleic acid (DNA) medium. International Journal of Innovative Computing, Information and Control. 2007; 3(1):16.

[19] Amal Khalifa. LSBase: A key encapsulation scheme to improve hybrid crypto-systems using DNA steganography. 8th International Conference on Computer Engineering & Systems (ICCES); 2013. DOI: 10.1109/ICCES.2013.6707182.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/11335