# Finding Data in DNA: Computer Forensic Investigations of Living Organisms

Marc B. Beck*, Eric C. Rouchka, Roman V. Yampolskiy

Cybersecurity Lab, Department of Computer Engineering and Computer Science, Speed School of Engineering, University of Louisville, Louisville, KY 40292

**Abstract.** Recent advances in genetic engineering have allowed the insertion of artificial DNA strands into the living cells of organisms. Several methods have been developed to insert information into a DNA sequence for the purpose of data storage, watermarking, or communication of secret messages. The ability to detect, extract, and decode messages from DNA is important for forensic data collection and for data security. We have developed a software toolkit that detects the presence of a hidden message within a DNA sequence, deciphers that message. In order to decode a message we are modifying several existing cryptanalysis tools that have been developed for solving simple substitution ciphers and compare their performance.

**Keywords:** Bioinformatics, Cryptography, DNA computing, Natural languages, Computer forensics, Steganalysis

## 1 Introduction

Deoxyribose Nucleic Acid (DNA) is the carrier of hereditary information for every living organism. DNA is a double helix with two anti-parallel strands containing four different nucleotides, which are distinguished by one of the four bases adenine (A), cytosine (C), guanine (G), and thiamine (T). The two strands form base pairs of interacting complementary bases (A-T and C-G) held together by hydrogen bonds. DNA has the potential to store vast amounts of data using combinations of those four nucleotides within genomes that can range to several billion bases in length [1].

Contained within genomic sequences are regions that code for genes that produce proteins which are collections of amino acids. In the process of translation, an mRNA sequence that has been transcribed, or copied, from the gene coding region is used as a template to transform from the four base code of DNA to the 20 base code of amino acids. The process by which this transformation occurs, known as the genetic code, was first uncovered by Marshall Nirenberg [2]. In gene coding regions, a codon refers to a sequence of three nucleotides that determines which amino acid will be

incorporated next during protein synthesis. With four nucleotides, this method allows $4^3$=64 possible combinations. Each codon encodes for one of 20 amino acids, with exception of the three STOP codons TAA, TAG, and TGA [3], thus allowing for degeneracy where multiple codon sequences code for the same amino acid. For the purposes of DNA steganography, characters of messages may be encoded by variable lengths of DNA sequences that may or may not be three bases in length. While not codons in the strict biological sense, we will refer to these encoding patterns as codons for the purpose of this manuscript.

### 1.1 DNA Computing

DNA computing is an emerging new research field that uses DNA molecules instead of traditional silicon based microchips. The first researcher to demonstrate the computing capability of DNA was Leonard Adelman, who in 1994 developed a method of using DNA for solving an instance of the directed Hamiltonian path problem [4]. In 1997, Ogihara and Ray demonstrated that DNA computers can simulate Boolean AND and OR gates [5]. The advantage of DNA computers is that they are smaller and faster than traditional silicon computers, and they can be easily used for parallel processing. DNA has also been used as a tool for cryptography and cryptanalysis, using molecular techniques for its manipulation [3]. Bogard et al. describe how multiple sequence alignment can be used for error reduction in DNA computing [6].

### 1.2 DNA as Storage Medium

DNA has recently been investigated as an ultra-compact, long term data storage medium (Table 1) and a stegomedium for hiding messages. Instead of expressing a message as a series of ones and zeros, it is represented in DNA as a series of As, Cs, Gs, and Ts. A number of algorithms have been developed to encode a message in DNA characters and either disguise these messages as novel DNA sequences or encapsulate them within existing ones. It has been proven that it is possible to insert artificial DNA components that contain encoded information into the genomes of living organisms [3,7-15].

Craig Venter, who led the private effort to sequence the human genome, managed to create the first cell with a synthetic genome in 2010. The J. Craig Venter Institute (JCVI) took a computer file containing the DNA sequence of the bacterium *Mycoplasma mycoides*, modified it, produced physical DNA from this sequence, and inserted this DNA into a cell, which then reproduced under control of the new DNA to create a new bacterium [15]. This led to the creation of a company, Synthetic Genomics, which focuses on the creation of synthetic genomes for applications including vaccine design, bioenergy, and biofuels.

Using DNA as storage medium has many advantages, such as long life, redundancy, and high density. According to Bancroft et al. [16] about 200 novels or other data each equivalent in size to “A Tale of Two Cities” could be stored in a DNA microchip with the area of a postage stamp.

**Table 1.** Life expectancy and storage capacity of various data storage media compared to DNA

| Type | Life Expectancy | Capacity |
|---|---|---|
| DNA | Millions of years | $10^8$ TB per 1 gram [1] |
| Hard disk | ~10 years | Up to 4 TB (2011) |
| CD | ~10 years | 800 MB |
| DVD | <10 years | Up to 17GB |
| USB flash drive | ~10 years, depending on usage | Up to 256GB (2011) |
| Tape | ~30 years | Up to 35 TB |

Yachie et al. [8] demonstrated the possibility to use DNA of living organisms as a data storage medium by inserting the message "E=mc^2 1905!" into the genome of *B. subtilis*. Over 99% of the encoded data was later recovered using sequence alignment methods.

Living organisms are a great storage medium when it comes to preserving data over timespans ranging in millions of years. When an organism reproduces, it automatically creates a backup copy of the data contained in its DNA. In addition, selective pressure and DNA error correction reduce the risk of the data being destroyed by random mutations. It has been suggested to use cockroaches, which are known for their resilience and high reproduction rate, as living time capsules for storing every issue of The New York Times Magazine for a certain year in their DNA which could theoretically be retrieved 1000 years later [17].

### 1.3 Error Correcting Approaches

Even though mutations are rare, occurring at a rate between $10^{-11}$ and $10^{-7}$ per base per replication in bacteria and higher eukaryotes [18], it is necessary to consider some form of error detection and error correction since a mutation can destroy the encrypted message in the DNA sequence. According to Yachie et al. [8], inserting the data redundantly into multiple loci of the genome is sufficient to allow the retrieval of stable and compact data without the need for template DNA, parity checks, or error correcting algorithms.

The comma code and the alternating code provide a form of error detection capability by encoding the message in a distinguishable pattern [19]. Arita [9] developed a comma-free code that has error correction capabilities. The message is translated into binary as an intermediary step. A parity bit is used in the binary code to keep the respective number of ones and zeroes odd.

The DNA-Crypt software developed by Heider and Barnekow [10] also translates messages into binary before encoding it in DNA code. It uses a very thorough approach to error detection by employing two error correction codes: the 8/4 Hamming-code and the WDH-code. The 8/4 Hamming-code is more compact, but it can correct fewer errors than the WDH code. DNA-Crypt has an integrated fuzzy controller using singleton fuzzyfication. The fuzzy controller decides which of the

two error detecting codes should be used, or none at all. This decision is based on the individual mutation rate of the DNA sequence that contains the secret message, the length of the sequence, and its stability over time. An answer is determined from those three factors by a set of rules based on heuristics [10].

## 2 Hiding Data in DNA

### 2.1 Overview

Steganography is the science of hiding information by transmitting secret messages through unsuspicious cover carriers in a way that makes the presence of any embedded messages undetectable. The term has its origins in the Greek language and means, "covered writing". While the goal of cryptography is to make a message unreadable, steganography aims at avoiding suspicion to the existence of a hidden message [20]. Due to its properties as a data storage medium, DNA can be used for steganography (stegomedium).

One of the most important problems in espionage is how to get the obtained information out of the target country without the information being detected by the enemy. With the appropriate knowledge and technology, a spy could have the information inserted into the DNA of an organism, and send it out of the country as an unsuspicious biological sample. It is possible to insert not only text, but also images and many other forms of digitizable data into a DNA sequence. For that reason, it is important to develop forensic tools that can detect hidden information in DNA.

**Table 2.** Genetic code for protein translation (codons that code for the same amino acid regardless the third position are highlighted) [2, 3].

| | | Second Position of codon | | | | | |
|---|---|---|---|---|---|---|---|
| | | **T** | **C** | **A** | **G** | | |
| **First Position** | **T** | TTT [F] | TCT [S] | TAT Tyr [Y] | TGT [C] | **T** | **Third Position** |
| | | TTC [F] | TCC [S] | TAC Tyr [Y] | TGC [C] | **C** | |
| | | TTA [L] | TCA [S] | TAA [end] | TGA [end] | **A** | |
| | | TTG [L] | TCG [S] | TAG [end] | TGG [W] | **G** | |
| | **C** | CTT [L] | CCT [P] | CAT His [H] | CGT [R] | **T** | |
| | | CTC [L] | CCC [P] | CAC His [H] | CGC [R] | **C** | |
| | | CTA [L] | CCA [P] | CAA Gln [Q] | CGA [R] | **A** | |
| | | CTG [L] | CCG [P] | CAG Gln [Q] | CGG [R] | **G** | |
| | **A** | ATT [I] | ACT [T] | AAT Asn [N] | AGT [S] | **T** | |
| | | ATC [I] | ACC [T] | AAC Asn [N] | AGC [S] | **C** | |
| | | ATA [I] | ACA [T] | AAA Lys [K] | AGA [R] | **A** | |
| | | ATG [M] | ACG [T] | AAG Lys [K] | AGG [R] | **G** | |
| | **G** | GTT [V] | GCT [A] | GAT Asp [D] | GGT [G] | **T** | |
| | | GTC [V] | GCC [A] | GAC Asp [D] | GGC [G] | **C** | |
| | | GTA [V] | GCA [A] | GAA Glu [E] | GGA [G] | **A** | |
| | | GTG [V] | GCG [A] | GAG Glu [E] | GGG [G] | **G** | |

An obvious choice of a location for inserting a message into a genome would be a noncoding genomic region. However, those regions might have other critical, unknown functions [7] and thus, inserting data there might possibly kill the organism. Therefore Arita et al. [7] suggested that it may be a more reliable solution to encode the message in the protein coding regions of genes. There are 20 amino acids and one stop symbol using a total of 64 possible codons [7]. Two or more codons often code for the same amino acid. Many of these redundant, or synonymous, codons typically differ in their third position, also known as the wobble base [3]. In Table 2, codons encoding the same amino acid regardless of the base in the third position are highlighted. These are the codons that can be used to embed messages.

**Table 3.** Research on data hiding in DNA

| Researcher | Year | Coding | Message | Location | Organism |
|---|---|---|---|---|---|
| Clelland et al.[11] | 1999 | Clelland | June 6 invasion: Normandy | artificial | human |
| Brenner et al.[12] | 1999 | Comma code | Not reported | Bsp120I | *E.coli* |
| Wong et al. [20] | 2003 | Clelland variant | Not reported | Not reported | *Deinococcus radiodurans* |
| Arita and Ohashi [7] | 2004 | Arita | AO2KEIO1-F | ftsZ gene | *B. subtilis RIK8* |
| Tanaka et al. [21] | 2005 | Similar to Clelland | MESSAGE | Artificial sequence | Artificial DNA strand |
| Yachie et al. [8] | 2007 | Keyboard scan | E=mc^2 1905! | metB and proB | *B.subtilis BEST2136* |
| Heider and Barnekow[14] | 2008 | DNA-Crypt | TB | Vam7 sequence | *Saccharomyces cerevisiae CG783* |
| Jiao and Gouette [13] | 2009 | ASCII 8 bit binary | CODING | tatAD gene | *B. subtilis* |
| J. Craig Venter Institute[15] | 2010 | Clelland variant | Multiple messages | Not reported | Artificial bacterium |

## 2.2 Coding Schemes

A code is an algorithm which uniquely represents symbols from some source alphabet, by symbols or strings of symbols in a target alphabet. In our case, the source alphabet is the English alphabet plus digits and punctuation characters, and the target alphabet consists of the four nucleotides. A coding scheme is a set of rules that determines which symbol of the source alphabet is represented by which symbol in the target alphabet.

**Types of coding schemes.** The coding schemes for inserting messages into DNA that have been developed can be grouped into three categories: schemes using direct translation, schemes that use intermediate steps for error detection, and schemes that have been optimized for detectability or efficiency.

The first category uses a straightforward approach by substituting a sequence of nucleotides of length n for each alphanumeric symbol [10, 12]. Since the codon in this case is of length n, up to 4n distinct characters can be encoded. Given the codon length of n, there are 4n! possible coding schemes. The coding schemes developed by Clelland [11] and Wong [20] fall into this category.

The second category of coding schemes consists of more complex schemes that use several intermediate steps, such as translating a message into binary before using a coding table to translate it into nucleotides. Intermediate steps like this are often used for error detection, since there are many proven error detection algorithms for binary messages.

The third category of coding schemes consists of schemes that were designed to meet certain criteria, such as providing error detection capability, being economical, or being easy to detect. The comma code, the alternating code, and a coding scheme based on the Huffman code [18] fall into this category.

**Clelland's coding scheme and Wong's coding scheme.** The coding scheme developed by Clelland et al. [11] is very similar to the one developed by Wong et al. [20]. They are both extensions of the three base codon encoding used by the genetic code. Since there are $4^3$=64 possible distinct characters that can be encoded, this scheme allows for all 26 characters of the English alphabet, the digits 0-9, and special characters. Both coding schemes do not use all possible codons.

**DNA-Crypt.** The DNA-Crypt coding scheme developed by Heider and Barnekow [10] translates a message into a five bit sequence, where one bit serves as parity bit to keep the respective number of ones and zeros odd. The other four bits are translated into nucleotides, with two bits per nucleotide. As mentioned earlier, it employs two error correction codes, the 8/4 Hamming-code and the WDH-code.

**ASCII based coding scheme.** Another coding scheme implements the algorithm described by Jiao and Gouette [3] which inserts a message into the noncoding region of an existing DNA sequence. This method consists of several steps:

1) Convert each character in the message into its ASCII representation.
2) Convert the ASCII code from decimal into binary.
3) Converting binary to DNA by replacing 00 with A, 01 with C, 10 with G, and 11 with T.
4) Insert message into a carrier DNA sequence.

Steps 1-3 are referred to as the ASCII coding scheme throughout the remainder of this paper. The fourth step can be applied to other coding schemes if the message is to be inserted into a coding DNA region. This insertion is performed by replacing the last bits of redundant codons in the carrier sequence with characters from the message sequence. The ASCII coding scheme makes it possible to encode uppercase letters, lowercase letters, numbers, and special characters. Each character is represented by a sequence of four bases.

There are 4! = 24 different ways to choose which two-bit binary sequence is translated into which nucleotide. The DNA-Crypt coding scheme uses 00=T; 01=G; 10=C; 11=A [10], whereas the ASCII coding scheme uses 00= A; 01= C; 10= G; 11= T [3].

**Yachie coding scheme.** Yachie et al. [8] developed a coding scheme similar to ASCII encoding. Instead of ASCII it uses the keyboard scan code for each character. The keyboard scan code is the data that the keyboard sends to the computer to indicate which key has been pressed. This code, which is hexadecimal, is converted into binary, and then translated into DNA using a coding table.

**Arita coding scheme.** Arita and Ohashi [7] translated each letter of the English alphabet as well as an empty space and the characters ‘‘’, ‘.’, ‘&’ into a 6-bit binary sequence. One of the bits serves as parity bit by keeping both the number of 0s as well as the number of 1s odd for error detection. When a message encoded with this coding scheme is inserted into a coding region of a DNA sequence, a 0 indicates to leave the 3rd base of a codon unchanged, while a 1 indicates that it needs to be changed. In order to extract the encoded message, one needs to compare the sequence that contains the message with the original, unchanged sequence to determine if a base was changed or not [10].

**Coding Scheme based on the Huffman code.** Another coding scheme is based on the Huffman code developed by David A. Huffman [22] and the frequency of letters in the English language from “The Code Book” by Simon Singh [23]. The Huffman code is an entropy encoding algorithm used for lossless data compression. The coding scheme based on this code currently only encodes letters, but not numbers or special characters. The average codon length is 2.2 bases. There are 4! possible ways to generate a Huffman code for encoding the 26 letters of the English alphabet, but it is also possible to create a Huffman code based scheme that includes numbers and special characters.

**Table 4.** Letter frequency in English language and DNA coding scheme using Huffman code [19].

| Letter | Freq (%) | Codon | | Letter | Freq (%) | Codon |
|---|---|---|---|---|---|---|
| e | 12.7 | T | | w | 2.4 | AAT |
| t | 9.1 | AG | | m | 2.4 | ACA |
| a | 8.2 | AT | | f | 2.2 | ACG |
| o | 7.5 | GA | | y | 2.0 | ACC |
| i | 7.0 | GG | | g | 2.0 | ACT |
| n | 6.7 | GC | | p | 1.9 | CCA |
| s | 6.3 | GT | | b | 1.5 | CCG |
| h | 6.1 | CA | | v | 1.0 | CCT |
| r | 6.0 | CG | | k | 0.8 | CCCA |
| d | 4.3 | CT | | j | 0.2 | CCCG |
| l | 4.0 | AAA | | x | 0.2 | CCCC |
| c | 2.8 | AAG | | q | 0.1 | CCCTA |
| u | 2.8 | AAC | | z | 0.1 | CCCTG |

**Comma Code.** The comma code uses four base codons consisting of combinations of A, C, G and T, where G serves as a separator between the different characters. The term comma code may be misleading. It does not mean that G is the encoding for the comma character, but that it separates the encodings for each character. Smith et al. [19] suggest using five base codons with a separator every sixth base, but the original paper by Brenner et al. [12] is more descriptive and recommends the use of four bases

per codon and a vocabulary made up of eight four-base ‘‘words’’ for biochemical reasons. The gaps between the Gs are filled with TTAC, AATC, TACT, ATCA, ACAT, TCTA, CTTT, or CAAA. Since this method would only allow the encoding of eight characters, combinations of two such words separated by a G are used for each character. This approach results in a total of 64 possible characters consisting of ten nucleotides each. The comma code encodes lowercase letters, numbers from 0-9, and special characters. The mapping of codons to characters was arbitrarily constructed. A sequence in comma code can easily be identified as containing a message, due to the occurrence of G every five bases, including the beginning and the end of the sequence. The comma code is the least efficient coding algorithm.

**Alternating Code.** The alternating code uses 64 codons with six bases per codon, alternating between purines (A or G) at odd positions and pyrimidines (C or T) at even positions. This coding scheme creates a pattern that does not occur naturally and can easily be recognized. For the same reason, the bases could be arranged for example in a pattern that has three purines followed by three pyrimidines or vice versa. The alternating code encodes the same characters as the comma code. The decision which codon codes for which character was made arbitrarily [19].

**Summary of coding schemes.** The coding schemes differ in codon length, detectability, number of characters that can be encoded, and the number of steps involved in encoding a message. The Huffman code is the most economical in terms of codon length, while the comma code is the least economical. The Clelland coding scheme and the Wong coding scheme are the easiest to implement. The comma code, alternating code and DNA-Crypt are the easiest to detect, and DNA-Crypt offers the best error correction.

Inserting a message into a coding region only replaces bases, but does not add new ones. Therefore the size of the genome is only affected if the message is inserted into a non-coding region. The length of the encoded message is the length of the unencoded message multiplied by the codon length of the coding scheme. For example, the message “UNIVERSITY OF LOUISVILLE” is 24 characters long, including spaces. It would be

CTGCCTCAGCTTATGCGTCTACAGCTCGAGCGAATTCCCCGACTGCAGC
TACTTCAGCCCCCCATG, which is 72 characters in Wong’s coding scheme and
GTACTGACATGAATCGACATGAATCGTTACGTACTGTCTAGTTACGACATG
TACTGAATCGTACTGTACTGAATCGTTACGTACTGATCAGATCAGTTACGA
ATCGCTTTGTTACGTCTAGAATCGATCAGAATCGCTTTGTACTGACATGAA
TCGTTACGTACTGTACTGTACTGTCTAGAATCGTTACGAATCGATCAGAAT
CGATCAGTTACGACATG, which is 240 characters in Comma Code.

For inserting pictures, audio, and video files into a DNA sequence it would be best to translate the binary representation of the file into DNA code, with each base encoding two bits, for example A=00, C=01, G=10, and T=11.

### 2.3 Encryption and Watermarking of DNA Messages

To make detection even more difficult, it is possible to encrypt a message using modern encryption algorithms such as Data Encryption Standard (DES), RSA, and Number Theory Research Unit (NTRU) before encoding it into DNA.

One application for inserting messages into DNA is watermarking. This method can help establish brand names for engineered bacteria strains in order to resolve legal disputes regarding gene related patents [7]. Watermarking infectious agents can be useful for tracking them back to their source after an accidental release [24].

Researchers at the JCVI inserted four watermarks using a Category 1 coding scheme similar to Clelland's and Wong's into their artificial genome. The first watermark consists of a copyright like statement; the coding table for Ventner's coding scheme, and a hidden HTML page. The second, third, and fourth watermarks consist of a list of the authors and three quotations.

The coding scheme created by Arita and Ohashi [7] and the DNA-Crypt algorithm developed by Heider and Barnekow [10] were both designed for watermarking short trademarks or signatures into genomic DNA.

### 2.4 Messages Finding Data in DNA

Steganalysis is the process of discovering hidden messages [25]. There are two main categories of steganalytic methods: blind steganalysis and specific steganalysis. Blind steganalysis can be used to detect a variety of different steganographic algorithms, including previously unknown ones. The goal of specific steganalysis is to detect a specific known steganographic algorithm by exploring how this particular algorithm works and how it changes the statistics of the cover media [26].

The research on steganalysis is important for several reasons: First, detecting the presence of secret messages can help intercept communication between members of terrorist organizations or other illegal groups. Second, improvements in steganalysis also help to develop better methods for information hiding. Third, better statistical methods for multimedia contents can emerge as a byproduct of steganalysis research. These can then be applied in other related research fields, such as digital forensics [26], or bioinformatics.

Most existing steganalysis approaches focus on images as a stegomedium, especially JPEG images as well as audio and video files. Text documents are not used as often since they can only hold a smaller amount of information than a graphic document with same amount of carrier data. However, text files are still used because they are easily edited, stored, and transferred [27].

### 2.5 Experiments Performed *in Silico*

Wang and Zhang [28] have developed a software called WordSpy to detect certain biological features within a genome. This software regards these biological features of a genome as a message hidden in a cover-text of genomic sequences. A Hidden Markov Model is used to decipher the message and to extract over-represented motifs.

WordSpy combines word counting and statistical modeling to detect frequently occurring sub-sequences [28].

Since many different coding schemes for inserting messages into DNA have been developed, we decided to develop a software toolkit that would enable us to insert and extract messages from DNA sequences, allow us to compare different coding schemes, and serve as basis for research into developing methods to find and extract messages encoded with unknown coding schemes.

### 2.6 Solving Substitution Ciphers

The way messages are encoded in DNA is typically through the use of substitution ciphers, for example, the letter 'a' is substituted by the sequence 'AAA', the letter 'b' by 'AAC', and so on. Several methods have been developed for breaking substitution ciphers. One of our goals is to adapt an algorithm for breaking substitution ciphers to decode a message written in DNA symbols. Almost all approaches use n-grams of letters.

Spillman et al. [29] developed a Genetic Algorithm and although they report good results for their Genetic Algorithm, Delman [30] found Genetic Algorithms to be unreliable for solving substitution ciphers and was unable to reproduce the results.

Another software called Quipster has been developed by Hasinoff [31]. The software decodes a median of 94% of the cipher letters correctly.

A Particle Swarm Optimization (PSO) algorithm has been developed by Uddin and Youssef [29]. Their results show that PSO provides a very powerful tool for the cryptanalysis of simple substitution ciphers using a ciphertext only attack. Uddin and Youssef [32] also investigated the use of Ant Colony Optimization (ACO) for automated cryptanalysis of classical simple substitution ciphers and found them to be very effective on various sets of encoding keys.

Lucks [33] developed an algorithm which employs an exhaustive search in a dictionary for words that satisfy constraints on word length, letter position and letter multiplicity. His method is not restricted to English and can be used for any language.

It is especially difficult to decode short ciphers, because they have different distribution statistics than larger texts. Hart [34] developed a method that addresses these problems by using whole words instead of n-grams and by employing a maximum likelihood estimator.

Jakobsen [35] developed a fast algorithm that is based on a process where an initial key guess is refined through a number of iterations. Each step of this algorithm evaluates the plaintext corresponding to the current key and the result is used as a measure of how close the algorithm is to discovering the correct key. The author claims that only knowledge of the bigram distribution in the ciphertext and the expected bigram distribution in the plaintext is necessary in order to decipher the message. The algorithm currently only uses bigrams, but the author suggests using trigrams or whole words for future research.

Forsyth and Safavi-Naini [36] approached the solving of substitution ciphers as a combinatorial optimization problem and developed an algorithm that uses simulated annealing. This algorithm is very complicated and difficult to implement, but it is very successful at decrypting ciphertexts, especially ones with over 5000 letters.

Peleg and Rosenfeld [37] address it as a probabilistic labeling problem and assigned probabilities of representing plaintext letters to every code letter. This approach was done by using joint letter probabilities. These probabilities were updated in parallel for all code letters, and using this scheme iteratively, they were able to break the cipher.

# 3 Description of DNA-Steg

### 3.1 Encoding and inserting messages

The DNA steganography and steganalysis Toolkit DNA-Steg we developed currently consists of two programs. One for encoding a message in a DNA sequence (steganography), and the other for detecting and extracting a hidden message from a DNA sequence (steganalysis).

DNA-Steg offers a choice of several different coding schemes. It reads in the coding table for the selected coding scheme from a file and then prompts the user to either type the message to be encoded on the keyboard or to read it in from a file. Since the ASCII coding scheme is the only one that distinguishes between uppercase and lowercase characters, the program converts all characters in the message into uppercase characters for all coding schemes other than the ASCII coding scheme. The steganography program then encodes the message using the appropriate coding table.

The toolkit implements the following coding schemes:
- Huffman code based scheme [19]
- Alternating code [19]
- Comma code [12, 19]
- Wong's coding scheme [20]
- Clelland's coding scheme [11]
- DNA-Crypt [10]
- ASCII coding scheme [3]

Coding tables for comma code and alternating code were created arbitrarily, since the original researchers did not provide any.

The message can either be directly written to a file by itself if it is to be stored in a noncoding region, or be inserted into the coding region of an existing DNA sequence file. For inserting a message in a coding region, the algorithm described by Jiao and Gouette [3] is used. DNA sequences can be chosen from a folder where they are stored in FASTA format [24], which is widely used in bioinformatics. The program displays the maximum number of characters a message can have, depending on the coding scheme and the sequence it is to be inserted into.

### 3.2 Approaches to Detecting Messages in DNA

Finding a message that has been inserted into the coding region of a DNA sequence is relatively simple if the original sequence is known. We have developed a program which compares a modified DNA sequence with its original. Since the message is

assumed to have been inserted into wobble bases, the first step is to identify wobble base codons in both sequences and to compare them to each other. The first codon where the wobble base is different from the one in the original is identified as the beginning of the message. The last codon where it differs is marked as the end of the message.

The limitation in this approach is that there are codons where the wobble base does not change. This is not a problem if it happens in the middle of the message. The program therefore assumes it contains one long message instead of several smaller ones. Problems can arise if this happens at the beginning or end of the message, but if the message can be decoded and it is seen that pieces are missing, the program can expanded to go back and fix it.

In order to test the program, the message "THIS IS A TEST" was inserted into ftsZ using the Wong coding scheme. The program then compared the modified sequence with the original one. It correctly identifies the beginning codon and the end codon of the message and extracts the modified wobble bases.

Finding a message in a noncoding DNA region is much more difficult. But there are ways to determine if a DNA sequence is artificial by statistical analysis. For example, if a certain base is significantly overrepresented, underrepresented, or not present at all, it can be assumed that the sequence is artificial and should be further analyzed to determine if it may contain a message.

Messages that have been encoded using a variation of the alternating code or the comma code are more likely to be detected than messages that were encoded with a different coding scheme. The reason for that is that they have a repeating pattern, which can be detected by a human or a computer program. If every $n^{th}$ base is the same, this hints at the possibility that comma code or a variation thereof has been used to create this sequence.

Another coding scheme that is easy to identify is the DNA-Crypt coding scheme because the low occurrence of As in a message encoded with this scheme.

However, as a countermeasure against attempts to detect messages by counting the occurrence of nucleotides, a coding scheme such as the one developed by Modego [38] can be used. Modego's coding scheme uses two codons to encode each letter. Which codon is used is determined by the GC content of the carrier sequence. For example, if a message was to be inserted into a sequence with a high GC content, the letter L would be encoded as CTG, but in a sequence with low GC content it would be TTA [38]. The obvious tradeoff is the number of characters that can be encoded is cut in half.

In order to detect the alternating code, the program stores all odd position characters in one list and all even position characters in another and then compares both of them. If none of the even characters appears in the list with the odd ones and vice versa, the program has detected a message in alternating code with the pattern XYXYXY, where X is either an A or a G and Y a C or a T, or vice versa. The program can easily be extended to detect alternating codes with pattern XXYYXXYY or XXXYYY.

Currently the steganalysis part of DNA-Steg reads messages from a DNA sequence by executing the algorithm that was used for encoding the message in reverse order. Since it does not know which coding scheme was used, it uses a brute force approach to test all supported coding schemes and displays the resulting message on the screen.

The user can choose if the message was hidden in a noncoding region or in an existing DNA sequence, which can be selected from a folder. If a sequence is chosen, the program will display the number and the percentage of occurrences of each nucleotide in the altered sequence as well as in the original sequence. Usually there is not much difference in the statistics of both sequences, since the inserted messages are fairly small.

One possible way to detect unknown encoding schemes might be to use the WordSpy algorithm developed by Wang and Zhang [28].

## 4 Further Research

We are currently working on a way to modify an existing approach for solving simple substitution ciphers where each letter in the English alphabet is substituted for a different English letters to solving simple substitution ciphers in which each letter of the English alphabet, numbers from 0-9, and several special characters such as spaces, commas, and periods are each substituted by a combination of three DNA bases. While the original program searches over the space of 26! possible keys, our program will have a search space of 64! possible keys.

The goal is to modify several existing approaches and then use the Wisdom of Artificial Crowds (WoAC) [39, 40] post-processing algorithm instead of brute force guessing in order to find out with which coding scheme the message has been encoded in. As a proof of concept, a program will be developed that will be able to decode any message that has been encoded with a category 1 coding scheme of codon length 3. This approach can be adapted to be used for other coding schemes as well.

Currently the steganalysis tool can only detect and extract messages that have been encoded with the previously described coding schemes. For example, it only has two coding tables for coding schemes of category 1, namely the ones developed by Clelland [11] and Wong [20]. But since there are four nucleotides, a category 1 coding scheme with a codon length of three, which can encode 64 characters, can be generated in 64! possible ways. And that is only if the same 64 characters are being used. For example, one coding scheme can start with A=AAA, B=AAC, C=AAG... while another one could be A=AGT, B=CCG, C=CTG... Brute force guessing which variant has been used to encode the message would take an enormous amount of time and would therefore not be feasible.

Knowing the characteristics of several previously developed coding schemes and their different variations makes it possible to develop a more optimized coding scheme. For example, coding scheme that is based on the Huffman code could be expanded to include numbers and special characters.

We also need to develop methods to detect and decode a message if the encoded message is also encrypted.

## Conclusion

DNA steganography is a new field and therefore it offers many opportunities to improve upon existing approaches for steganography as well as steganalysis.

## References


1. Anam B, Sakib K, Hossain A, Dahal K. Review on the Advancements of DNA Cryptography. International conference on Software, Knowledge, Information Management and Application; August 25-27, 2010; Paro, Bhutan.(2010)
2. Nirenberg M. Historical review: Deciphering the genetic code – a personal account. Trends in Biochemical Sciences.29(1):46-54.(2004)
3. Jiao S-H, Goutte R, editors. Code For Encryption Hiding Data Into Genomic DNA. International Conference on Software Process; 2008.
4. Adleman LM. Molecular Computation Of Solutions To Combinatorial Problems. Science, New Series.266(5187):1021-4.(1994)
5. Ogihara M, Ray A, editors. Simulating Boolean Circuits on a DNA Computer. RECOMB; 1997.
6. Bogard CM, Rouchka EC, Arazi B. DNA media storage. Progress in Natural Science.18:603-9.(2007)
7. Arita M, Ohashi Y. Secret Signatures Inside Genomic DNA.. Biotechnology Progress.20(5):1605-7.(2004)
8. Yachie N, Sekiyama K, Sugahara J, Ohashi Y, Tomita M. Alignment-Based Approach for Durable Data Storage into Living Organisms. Biotechnology Progress.23(2):4. Epub 01/25/2007.(2007)
9. Arita M. Comma-free design for DNA words. Communications of the ACM.47(5):99.(2004)
10. Heider D, Barnekow A. DNA-based watermarks using the DNA-Crypt algorithm. BMC bioinformatics.8:176. Epub 2007/05/31.(2007)
11. Clelland CT, Risca V, Bancroft C. Hiding messages in DNA microdots. Nature. 1999:533-4.
12. Brenner S, Williams SR, Vermaas EH, Storck T, Moon K, McCollum C, et al. In vitro cloning of complex mixtures of DNA on microbeads: Physical separation of differentially expressed cDNAs. Proceedings of the National Academy of Sciences of the United States of America.97(4):1665-70.(2000)
13. Jiao S-H, Goutte R. Hiding data in DNA of living organisms. Natural Science.01(03):181-4.(2009)
14. Heider D, Barnekow A. DNA watermarks: a proof of concept. BMC molecular biology.9:40. Epub 2008/04/23.(2008)
15. Gibson DG, Glass JI, Lartigue C, Noskov VN, Chuang RY, Algire MA, et al. Creation of a bacterial cell controlled by a chemically synthesized genome. Science.329(5987):52-6. Epub 2010/05/22.(2010)
16. Bancroft C, Bowler T, Bloom B, Clelland CT. Long-Term Storage of Information in DNA. Science, New Series.293(5536):1763-5.(2001)
17. A Y3K bug.pdf. nature biotechnology.18.(2000)
18. Drake JW, Charlesworth B, Charlesworth D, Crow JF. Rates of Spontaneous mutation. Genetics.148(4):20.(1998)
19. Smith GC, Fiddes CC, Hawkins JP, Cox JPL. Some possible codes for encrypting data in DNA. Biotechnology Letters.25(14):1125-30.(2003)
20. Wong PC, Wong K-K, Foote H. Organic Data Memory Using the DNA Approach. Communications of the ACM.46(1):95-8.(2003)
21. Tanaka K, Okamoto A, Saito I. Public-key system using DNA as a one-way function for key distribution. Bio Systems.81(1):25-9. Epub 2005/05/27.(2005)
22. Huffman DA. A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the IRE.1098–102.(1952)
23. Singh S. The Code Book: The Evolution of Secrecy from Mary, Queen of Scots to Quantum Cryptography. New York: Doubleday; 1999.
24. Jupiter DC, Ficht TA, Qin Q-M, de Figueiredo P. DNA Watermarking of Infectious Agents Progress and Prospects. Public Library of Science Pathogens.6(6):1-3.(2010)
25. Johnson NF, Jajodia S, editors. Steganalysis: The Investigation of Hidden Information. IEEE Information Technology Conference; 1998; Syracuse, NY.
26. Li B, Huang J, Shi YQ. Steganalysis of YASS. IEE Transactions on Information Forensics and Security.4(3):369 - 82 (2009)
27. Xin-guang S, Hui L, Zhong-liang Z, editors. A Steganalysis Method Based on the Distribution of Characters.pdf. 8th International Conference on Signal Processing; 2006; Beijing, China.
28. Wang G, Zhang W. A steganalysis-based approach to comprehensive identification and characterization of functional regulatory elements. Genome biology.7(6):R49. Epub 2006/06/22.(2006)

29. Spillman R, Janssen M, Nelson B, Kepner M. Use of a Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers. Cryptologia.17(1):31-44.(1993)
30. Delman B. Genetic Algorithms in Cryptography. Rochester, New York: Rochester Institute of Technology; 2004.
31. Hasinoff S. Solving Substitution Ciphers. A Technical Report, University of Toronto.(2003)
32. Uddin MF, Youssef AM. An Artificial Life Technique for the Cryptanalysis of Simple Substitution Ciphers. CCECE+CCGEI; May 7 to 10, 2006; Ottawa, Canada: IEEE. p. 1582-5.(2006)
33. Lucks M, editor. A Constraint Satisfaction Algorithm for the Automated Decryption of Simple Substitution Ciphers. CRYPTO '88; 1988 August 21-25, 1988; Santa Barbara, California, USA.
34. Hart GW. To decode short Cryptograms. Communications of the ACM.37(9):102-8.(1994)
35. Jakobsen T. A fast method for cryptanalysis of substitution ciphers. Cryptologia.19(3):265-74.(1995)
36. Forsyth WS, Safavi-Nani R. Automated Cryptanalysis of substitution ciphers. Cryptologia.17(4):407-24.(1993)
37. Peleg S, Rosenfeld A. Breaking Substitution Ciphers Using a Relaxation Algorithm. Communications of the ACM.22(11):598–605.(1979)
38. Modegi T. Watermark Embedding Techniques for DNA Sequences Using Codon Usage Bias Features. 16th International Conference on Genome Informatics; Yokohama, Japan.(2005)
39. Yampolskiy RV, El-Barkouky A. Wisdom of artificial crowds algorithm for solving NP-had problems. International Journal of Bio-inspired computation.3(6).(2011)
40. Yampolskiy RV, Ashby LH. Genetic Algorithm and Wisdom of Artificial Crowds Algorithm Applied to Light Up. The 16th International Conference on Computer Games; Louisville, KY. p. 27-32.(2011)