Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Sequence Analysis

Data Storage in Cellular DNA: Contextualizing Diverse Encoding Schemes

Anthony P. Machado , Gaby G. Dagher , and Eddie C. Davis

Computer Science Department, Boise State University, Boise, 83725, USA

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: DNA, the macromolecule used by organisms to store and transmit genomic data, has attracted the attention of privacy researchers as a channel for secure data transfer. DNA's small size and abundance in nature makes it an ideal steganographic medium for hiding messages. Already, artificially synthesized DNA has been used to store text, audio, and images. Encoded cellular DNA is not far behind, with much research being done on ways to safely embed data without harming the cell.
Results: In this survey we provide the first systematic comparison of cellular encoding schemes proposed in the literature. Different DNA regions in the cell have their own unique bio-restrictions that must be satisfied for DNA storage. Drawing from a wide array of schemes, we compare the novel techniques used to meet these bio-restrictions. This contextualization of the research creates a bigger picture that can help guide the design of future schemes. We also survey the compression methods and error detection techniques used by the encoding schemes, and their effect on error rate and density. Finally, we propose future directions for research by examining gaps in the literature and suggesting novel approaches with epigenetics and mitochondrial storage.
Contact: anthonymachado@u.boisestate.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Biosteganography is an emerging field in privacy research that combines techniques from genetic engineering, bioinformatics, cryptography and forensics to secretly transfer data within a living cell's DNA [10]. Data ranging from simple text messages to audio recordings and color images can potentially be inserted into cellular DNA for secure transmission. Because DNA is information dense and occurs abundantly in nature, it makes an ideal medium for sending messages secretly. Basic steganographic principles dictate that if an attacker doesn't know where to find a confidential message in the first place, the message is far more secure [5]. Locating encoded DNA would be harder than finding a needle in a haystack. A needle, at least, can be seen by the human eye.

Traditional digital storage systems such as hard drives and SD cards have detectable emanations, which makes them vulnerable to side attacks [48]. Modified DNA, however, has no measurable emanations. If an agent needed to carry a confidential message through a tight security checkpoint, any form of electronic storage could be easily detected, while physical recordings like paper or tape could be visibly located by the security guard. A DNA encoded message, however, located in the cells of the agent's thumb, or in bacteria under the agent's nail, could be brought through without detection. The receiver of the message would need to know how to decode the data, and have the lab equipment to amplify and sequence the DNA in order to retrieve the message.

Inserting hidden data into DNA is also crucial for watermarking patented genes. Advances in gene editing technology have led to a rise in genetically modified organisms being developed in a variety of sectors such as agriculture, healthcare, and energy. For added security of these novel genomes, many companies have begun embedding hidden watermarks in their DNA so that ownership can be established in the case of theft. For this purpose too, secrecy is essential. The watermark must be embedded in such a way that the thief cannot find it, else the thief would simply remove it. Thus, biosteganography is an overlapping goal both of individuals wanting to send data privately and companies wanting to watermark genetic inventions.

Several encoding schemes have been proposed to accomplish data embedding in cellular DNA. In this survey, we will provide the first systematic analysis of these diverse encoding schemes, in which:

- We define the unique bio-restrictions associated with cellular encoding and compare how the proposed encoding schemes meet these restrictions;
- We contrast the error rates and densities of each of the schemes, highlighting the effects of error correction and compression techniques on both; and
- We propose new directions for research incorporating novel epigenetic techniques.

The rest of the survey is organized as follows. In Section 2 we provide a background on genetics and explain the biological constraints of encoding with cellular DNA. Section 3 illustrates related work done in artificial encoding. A systematic comparison of cellular encoding schemes is presented in Section 4. Next, Section 5 proposes new directions for future encoding schemes that use epigenetics and alternative genetic regions. Finally, we conclude in Section 6.

2 Background

2.1 Genetics

Every living cell contains DNA molecules encoded with instructions for making the proteins necessary for the cell to function. DNA takes the form of a double helix made of two antiparallel strands. Each strand is composed of a sequence of 4 nucleotides: adenine (A), guanine (G), cytosine (C), and thymine (T). The purines adenine and guanine, and the pyrimidines cytosine and thymine, form hydrogen bonds with each other across the double helix. Adenine binds to thymine with two hydrogen bonds, and guanine binds to cytosine with three. Every three nucleotides forms a codon that the cell reads and processes to create an amino acid, the building block of a protein [52]. Proteins can be encoded on either strand.

There are only 20 amino acids, despite the fact that $4^3 = 64$ possible codon variations exist. This is because many amino acids are correlated with up to five redundant codons. Also, three codons are used specifically as stop signals to indicate the end of a protein chain. The redundancy in codon to amino acid mapping is called codon degeneracy [51]. Table 1 illustrates codon to amino acid mapping.

2.2 Cellular Encoding Restrictions

Cellular DNA encoding must not harm the carrier organism, either by removing cellular functionality or adding mutative behavior. To avoid this, encoding schemes must ensure that modified DNA strands remain biologically equivalent to their wild-type form. This section defines the bio-restrictions that exist for two distinct areas of cellular DNA, protein coding DNA (pcDNA) and noncoding DNA (ncDNA). As our knowledge of genetics continues to expand, more particular restrictions may become known.

2.2.1 pcDNA Constraints

The protein coding region of DNA contains the codons that are eventually translated to amino acids, which are concatenated into proteins. Any data insertions in this area must meet the following constraints.

Protein Preservation The structure of the protein coded by the region must remain unchanged. Nucleotide insertions and modifications must not alter the codons in such a way that would change the original amino acid sequence.

Codon Bias Preservation Individual cells have specific ratios of cytoplasmic tRNA associated with their genomic codons, which can be disrupted if the codon balance is changed, and have a negative effect on the cell. Therefore, it is important that codon bias is preserved by maintaining roughly equal numbers of each codon after encoding is complete.

Table 1. Codon to Amino Acid Mapping

Amino Acids	Codons
Ala	GCT GCC GCA GCG
Arg	CGT CGC CGA CGG AGA AGG
Asn	AAT AAC
Asp	GAT GAC
Cys	TGT TGC
Gln	CAA CAG
Glu	GAA GAG
Gly	GGT GGC GGA GGG
His	CAT CAC
Ile	ATT ATC ATA
Leu	TTA TTG CTT CTC CTA CTG
Lys	AAA AAG
Met	ATG
Phe	TTT TTC
Pro	CCT CCC CCA CCG
Ser	TCT TCC TCA TCG AGT AGC
Thr	ACT ACC ACA ACG
Trp	TGG
Tyr	TAT TAC
Val	GTT GTC GTA GTG
START	ATG
STOP	TAA TGA TAG

2.2.2 ncDNA Constraints

The noncoding region of DNA is often called "junk DNA" for its apparent lack of use in the cell. Because they appear to be non-functional, data can be embedded into these regions if the following constraints are met.

Truly nonfunctional region. When ncDNA was first discovered it was assumed to have no role in cell functionality, but recent studies have shown that up to 80% of ncDNA may have biochemical functions in the cell, despite not coding for proteins [16]. Therefore, it is first imperative that the individual who wishes to insert a message into ncDNA verify that they are encoding their message in the 20% that has no biochemical use.

No start codons. When a cell's genetic machinery locates a start codon, it can begin the transcription process. To prevent unwanted transcription from happening in ncDNA with embedded data, it is important to make sure the encoded nucleotides to not create a start codon. When a DNA string is being transcribed, three-nucleotide codons can be read in six different reading frames. Therefore, there should not be a start codon in any of the six frames. The most common start codon is AUG, though some alternative start codons can also exist, particularly in bacteria [6]. If a cell contains alternative start codons, the encoding scheme should avoid all of them.

No homopolymers. A DNA homopolymer is a region where the same nucleotide is repeated multiple times. Too many repeats can cause errors during DNA replication through polymerase slippage [50]. These replication errors could quickly distort the inserted message and possibly damage the cell after a few generations. For this reason any ncDNA encoding scheme should not include homopolymers greater than length 3.

3 Related Work

3.1 Artificial Encoding

Artificial DNA encoding, or DNA data storage, is the process of storing digital information such as text, audio, or images within *in vitro* DNA base pair sequences. DNA is an appealing storage medium due to its high density. For example, the human genome contains 3.3e9 base pairs, or about 725 MB of binary data, with a mass of 3.59e-12 grams. DNA is also

incredibly resilient, as the sequencing of 45,000 year old woolly mammoth genomes indicate [40]. As with any memory device, the critical operations are read and write. Continued advances in DNA sequencing (reading) and oligonucleotide synthesis (writing) increase the viability of nucleic acid for long term archival storage.

DNA was first encoded with a secret message in 1999 by Clelland et al [15]. Inspired by the tiny, concealed microdot messages used in World War II, artificial DNA strands were constructed using a simple substitution cipher, then mixed with human DNA and pipetted onto a printed period on filter paper. The encoded DNA was later recovered from the dot and sequenced to successfully read the secret message: JUNE 6 INVASION: NORMANDY.

The primary limiting factor has been the length of the oligonucleotides that can be synthesized. Church et al. [14] developed a technique using high fidelity DNA microchips to encode 5.27 MB of data including a book by the author, 11 JPEG images, and a JavaScript program into 54,898 159 nt sequences. Each 159 nt sequence consisted of a 96 nt data block and 19 nt address. The data were recovered with only 10 bit errors.

Goldman et al. [21] were able to encode 739 KB of text, images, and audio with 100% recovery rate. The technique applied a modified Huffman code [29], converting the binary (base-2) data into ternary (base-3), and mapping each trit to a nucleotide, different from the one preceding it in order to prevent homopolymers. The sequences were capped with reverse complemented index information so that the data could be reassembled by scanning for overlaps.

Grass et al. [22] developed error correction codes for DNA stored in silica gel and exposed to high temperature (70^o C). Every two bytes of input data are mapped to three values from the Galois Field of size 47 (GF(47)). The mappings are arranged in blocks of 594x30. Each block also has 594x3 bits of index data. In the second step, Reed-Solomon (RS) codes [45] are applied to add 119 bits of redundancy A. In another round of RS, a redundancy B of 6 bits in length. This results in a final block size of 713x39. Each column is mapped to an oligonucleotide by mapping each value from GF(47) to a DNA codon (triplet).

Yazdi et al. [56] also developed a random access and rewritable DNA storage system, achieved with an addressing scheme. A 1000 bp sequence represents a 960 bp block of data flanked on either side by address regions of 20 bps. The data regions are divided into 12 blocks of 80 bps each, or 6 words of input text per block. Primer sequences correspond to the address sequences, allowing for selective PCR amplification of a desired block. The binary sequences are converted into nucleotides via DNA prefix-synchronized codes. The error correction code is based on binary running digital sums (BRDS). The encoding scheme satisfies four constraints, 1) GC content of approximately 50%, 2) large Hamming distance between any two strings of equal length, 3) uncorrelated addresses to prevent primer binding to the wrong address, and 4) absence of secondary folding structures.

Blawat et al. [7] improved upon the encoding scheme of Church et al. [14] by developing forward error correction codes. The homopolymer restriction is relaxed somewhat by requiring that the first three nucleotides must differ, as must the last two. A block of data is contained within an oligo sequence of up to 250 bp. An error detection code (EDC) is added to each. Addresses are protected with a BCH code [44] and a minimum Hamming distance of 9. Consecutive oligo sequences are protected with RS code [45] over GF(2^8) with block size of 223. The EDC parity bits employe a CCITT 16 bit cyclic redundancy check (CRC) [35].

Bornholt et al. [8] have developed a DNA-based archival storage system that also allows for random access reading. They adopt the binary to ternary Huffman encoding technique of Goldman et al. [21] for storing data. Similarly to Yazdi et al. [56], each sequence has a primer target sequence on either end of the oligonucleotide. Data are stored in the payload, which is followed by the address. Following the first primer sequence and before the last is an orientation nucleotide to determine if the oligo is being read in the 5' to 3' or 3' to 5' direction. Also like Yazdi et al. [56], random access is provided by selective PCR amplification via specific primers. The overall system acts as a key-value store with primers acting as keys. Error correction is provided via redundancy, that is more important regions are sequenced in greater quantities. A simple error correction technique is achieved by performing an XOR on strands of equal length to ensure proper alignment.

3.2 Compression and Error Correction

Smith et al. [47] were the first to suggest a data compression technique in DNA encoding, specifically the Huffman Code. The Huffman Code is a form of lossless data compression that forms a symbol table using fewer bits to encode more common characters [29]. Smith et al created a Huffman Code table mapping letters of the alphabet to nucleotide strings, where the most common English letter 'e' was mapped to the nucleotide string "T", and the least common english letter 'z' was mapped to a longer nucleotide string "CCCTG". This achieved an average encoding length of 2.2 bases per letter. The mapping is unambiguous, making only one possible interpretation of each message.

Comma encoding [9] specifies that encoded words be separated by a single nucleotide (i.e., G). The remaining four (or five in Smith et al.), are composed of the other three nucleotides. Additional constraints are that only three A-T pairs are allowed, and two G-C always on the top strand, so that the DNA molecules will have isothermal melting temperatures. This technique provides particularly good detection of insertions and deletions, unfortunately it is also space inefficient.

The alternating code, also from Smith et al. [47] consists of 64, 6 base pair codons, with nucleotides alternating between an A or G (purines) at odd positions, and C or T (pyrimidines) at evens (e.g., RYRYRYâĽ¦, YRYRYR). Like the comma encoding, the alternating code results in isothermically stable molecules with a 1:1 ratio of A-T to G-C pairs. While more space efficient, this technique is not as proficient at error detection, as only 67% of codons result in nonsense codons after mutation, compared to 83% for the comma code.

Contrast mapping is an encoding scheme developed by Mousa et al. [39]. The binary message is divided into 6-bit groups, each converted to decimal (base 10). Pairs of consecutive values (x, y) are converted into (x', y') with the the following linear transformations: $x' = 2x - y$, $y' = 2y - x$. Values are limited to the subdomain: $0 \leq 2x - y \leq L$, $0 \leq 2y$ - $x \leq L$. Values are decoded with the following equations: $x = [3x' + 3y']$, $y = [3x' + 3y']$. This technique is sufficiently flexible to be applied to DNA or image steganography.

3.3 Image Steganography

One of the inspirations of biosteganography is the practice of image steganography. The modification of the wobble codon in pcDNA encoding is analogous to the storage of information within the least significant bits (LSB) of the bytes within digital images. The concept of watermarking in particular is closely related. Image steganography can be described by three broad categories, spatial domain, frequency domain, and adaptive methods.

The target digital media is referred to as the carrier or "cover" image C, and the embedded data as the "payload", M, encoded in the stego-image C'. An optional key K can be used to encrypt the message M. The function Em represents embedding (analogous to encryption), and Ex extraction (analogous to decryption), such that:

$$Ex[C'] = Ex[Em(C, K, M)] = M$$

LSB modification falls under the spatial domain category. Up to four LSB bytes can be modified, but the image quality quickly begins to suffer, and the effects more apparent to the viewer. Potdar et al. [41] developed a fingerprinting technique to minimize these image cropping effects by dividing C into subimages C_i. The success of steganography in the spatial domain is dependent on the image format. Lossless formats are ideal (e.g., BMP, PNG), while lossy formats (e.g., GIF, JPG) are less so. [30] Histogram data hiding techniques based on the differences between adjacent pixels are less susceptible to detection. [34]

Methods in the frequency domain include the discrete cosine transform (DCT) [37], Fourier transform (FT) [38], and discrete wavelet transform (DWT) [12, 42, 49, 2]. JPEG compression applies DCT to transform image sub-blocks, data can encoded into the corresponding coefficients. Detection is difficult if chosen carefully, e.g., by a genetic algorithm. [19]. Work in the DWT domain includes artificial neural networks (ANNTS) [17] and cover image decomposition. [1]

Adaptive integration is a composite of the previous two methods. Statistics are applied to automatically select the best data hiding technique. These include the model based method using the Cauchy distribution [46], the LSB substitution method that estimates the degree of smoothness between adjacent pixels [11], block complexity based data embedding (ABCDE) [28], and genetic algorithm approaches that attempt to artificially obscure statistical features [54].

3.4 DNA Computing

The seminal paper in the field of DNA computing is Adleman's paper [**?**], in which DNA is applied as a computational device for solving the seven point Hamiltonian path problem (NP-complete). Other NP-complete problems were addressed by Boneh et al. [**?**] who solved circuit satisfiability, and Kari et al., the Bounded Post Correspondence Problem. [**?**]. Ogihara and Ray developed boolean circuits with DNA to perform massively parallel computations [**?**]. Stojanovic and Macdonald developed automata called "DNAzymes" (DNA molecules with enzymatic functions), using them to construct the MAYA [**?**] and MAYA II [**?**] computers, that were capable of playing the game of tic tac toe. DNAzymes employ a stem loop secondary structure to implement logic gates.

Benenson et al. expanded DNA computing from the *in vitro* realm to *in vivo*, by introducing a scheme for controlling gene expression with a molecular computer [**?**]. Nayebi developed a DNA-based implementation of Strassen's fast matrix multiplication algorithm [**?**]. A crucial advance in this field came with the development of the "transcriptor" by Bonnet et al. [**?**], a three terminal transistor like device implemented with serine integrases from bacteriophages to guide RNA polymerases and control gene expression. Transcriptors allow circuit designers to create AND, NAND, OR, XOR, NOR, and XNOR logic gates. This work has since been expanded to develop entire genetic circuits [**? ?**].

4 Encoding Schemes

Encoding schemes for cellular DNA can involve several components:

- Encryption algorithm
- Mapping table
- Compression
- Error correction

The following encoding schemes use some or all of these elements in their design. A comparison of the schemes is shown in Table 2. Table 3 has details of their implementation.

4.1 pcDNA Encoding

Shimanovsky et al [47] were the first to propose using codon degeneracy to encode data in pcDNA. By switching codons between their redundant forms with the modification of one nucleotide, data was inserted without altering protein translation. Arita and Ohashi [4] implemented this technique in a living cell. They did this using site-directed mutagenesis of wobble codons in the *ftsZ* gene of *Bacillus subtilis*. The university name "KEIO" was inserted by modifying the redundant nucleotides of the codons downstream of the *ftsZ* start codon. An unmodified codon represented value 0 while a codon with a wobble nucleotide changed to any of its non-wild type redundant forms represented value 1. Messages were translated using a 6-bit mapping table, with the first 5 bits corresponding to an English alphabet letter or basic punctuation, and the last used as a parity bit for error correction.

Heider et al. developed the DNA-Crypt [25] algorithm for creating DNA watermarks for marking genetically modified organisms. It is similar to the work of Arita et al. in that the encoding targets the wobble base pair in the genetic code. An encryption function E maps the plaintext (binary data) X to the ciphertext (genetic data) Y, such that $X \in \{0, 1\}$ and $Y \in \{\text{A, C, G, T}\}$. Two bits are encoded per base, or one byte for four bases. Error correction is achieved with a fuzzy controller that selects one of two algorithms, either the 8/4 Hamming code for mutations that differ in only bit (e.g., 00 to 01) or the WDH-code for those that differ by multiple bits. This encoding scheme allowed implementations of several cryptographic algorithms, including One-Time Pad, AES, Blowfish, and RSA. The accuracy was tested using the GTPase encoding Ypt gene in $S.cerevisiae$.

Liss et al. were the first to design a pcDNA encoding scheme that allowed for blind decoding [36]. First a codon usage table was created, which ranked synonymous codons according to their natural occurrence. An odd-ranked codon (1st, 3rd, or 5th) represented binary 1, while an even-ranked codon (2nd, 4th, or 6th) represented binary 0. Encoding in this way also allowed for codon bias preservation through gene optimization. The receiver who obtains the DNA would only need to know which gene had the encoded message, and the codon usage table, which itself could be encoded into the same genome. Since the receiver does not need to know the wildtype sequence of the genome to decode, this type of scheme can be blindly decoded.

Another blind technique for pcDNA, LSBase, was developed by Khalifa and Hamad [31]. Their scheme used a very simple mapping table that allowed for easy encoding and blind decoding. For every wobble codon in the encoding region, the use of a U or an A in the least significant base represented bit 0 while a C or a G represented bit 1. The researchers were able to test their scheme in silico, and found a high bits per codon capacity (0.333) compared to other schemes. However, no effort was made to account for codon bias preservation.

Haughton and Balado proposed BioCode [24], a pair of encoding algorithms, one for ncDNA, and one for pcDNA. The ncDNA algorithm expands upon DNA-Crypt by observing the no start codons in restriction. This is accomplished by defining a set of dinucleotides $D = \{$ AT, CT, TT, CA $\}$ that covers the possible eukaryotic start codons on either DNA strand. The trailing dinucleotide d is continually checked for membership in D. If found, d is replaced with a lookup table formed from a graduated mapping of the message space M_d to set S_d. The pcDNA encoding technique is similar, but enforces the additional constraint of Binary Codon Equivalency (BCE). This requires that the cardinality of the codon set ($|S_d|$) be varied during the embedding process to allow the usage of a static lookup table.

In order to better preserve codon bias, Lee developed a discrete wavelet transform (DWT) technique for pcDNA encoding [33]. The target coding sequence was divided into subsequences in which every codon was given

Table 2. Comparison of Encoding Schemes

	Scheme	Bio-Restrictions Satisfied				Scheme Components				
		No Homopolymers	No Start Codons	Preserves Codon Bias	Balances C-G	Error Detection	Error Correction	Compression	Encryption	Blind Decoding
pcDNA	Secret Signature [4]	●	●			●				
	DNA-Crypt [25]	●	●			●	●		●	
	Codon Usage Table [36]	●	●	●	●				●	●
	BioCode [24]	●	●	●	●	●	●			
	DWT Based [33]	●	●	●	●					
	LSBase [31]	●	●							●
ncDNA & Plasmids	Organic Memory [53]			●						●
	Alignment-Based [55]			●		○	○			●
	Improved Huffman [3]			●	○			●		●
	RAlign [23]			●		○	○			●
	HTAlign [23]			●		○	○			●
	DNA Barcodes [32]	●		●		●	●			●
	DNA-Courier Attack [13]			●	●				●	●

○ denotes partially satisfied whereas ● denotes completely satisfied.

Table 3. Details of Scheme Implementation

	Scheme	Organism	Method	Data	Density †	Error Detection/Correction
pcDNA	Secret Signature	Bacteria	In vivo	"KEIO"	0.167 cpc	Parity bit
	DNA-Crypt	Bacteria	In silico	"this is a test"	0.173 bpc	Hamming, WDH
	Codon Usage Table	Bacteria, yeast, plant, human	In vivo	"GENE"	0.143 bpc	None
	BioCode	Bacteria, yeast	In silico	"BioCode preserves codon bias"	0.204 bpc	Hamming, Repetition
	DWT Based	Bacteria, yeast, human	In silico	N/A	0.180 bpc	None
	LSBase	Rodent	In silico	3.7 kb text	0.333 bpn	None
ncDNA & Plasmids	Organic Memory	Bacteria	In vivo	"It's a Small World" lyrics	0.333 cpn	None
	Alignment-Based	Bacteria	In vivo	"E=mc2"	0.5 bpn	Repetition
	Improved Huffman	Theoretical	In silico	Lyrics, music & image	0.286 bpn	None
	RAlign	Virus	In silico	1000 bits	0.48 bpn	Repetition
	HTAlign	Virus	In silico	1000 bits	0.46 bpn	Repetition
	DNA Barcodes	Theoretical	In silico	73 code sets	0.2 bpn	Hamming
	DNA-Courier Attack	Bacteria	In silico	"HelloMyNameIsLHL"	0.97 bpn	None

† In this column, cpc denotes "characters per nucleotide", bpc denotes "bits per codon", and bpn denotes "bits per nucleotide". In the case of multiple storage sites, the average density is taken.

a numerical code associated with amino acid histogram rankings. DWT coefficients were calculated for synonymous codons and the optimal subsequence was found, which was then replaced with the encoded subsequence. A nonlinear congruential-pseudorandom number generator then created the watermark and picked the location in the DWT domain for embedding.

4.2 ncDNA and Plasmid Encoding

The development of encoding schemes for ncDNA and plasmids have been strongly correlated, due to two regions having nearly identical bio-restrictions. The main difference being that plasmid encoding is also confined by length. However, due to their similarities and the parallel advancement of their encoding schemes, we will consider them both in this section.

Wong et al. [53] were the first to encode data into plasmids. Letters and punctuation were mapped to nucleotide triplets, then short text snippets of 19 to 33 characters were inserted into the plasmids of *Deinococcus radiodurans*, a bacteria that is very resilient to extreme conditions. The encoded section of the plasmid was flanked by sentinel sequences 20 base pairs long containing stop codons. This was done to prevent the bacteria from transcribing the message while reading the adjacent functional parts of the plasmid. After insertion of the plasmids into the host bacteria, they were incorporated into their genomes. No error correction, encryption, or compression techniques were used.

Yachie et al. [55] came up with the idea of using multiple alignment in lieu of error detection or correction techniques. With this scheme, messages were composed using the Keyboard Scan Code Set2, which contains all keyboard inputs, and converted these messages to binary. The encryption keys mapped four bits of binary code to two nucleotides. This type of mapping created four reading frames in the binary message (C1 - C4), each of which was converted to DNA and inserted adjacently into the plasmid, creating redundancy that allowed for multiple alignment when decoding.

Repetition coding was used again, this time for ncDNA, by Haughton and Balado [23] who used two variant approaches. Both approaches required finding subsequences in the host ncDNA equal to the length of the encoded message, and replacing them with the message, as opposed to simply inserting the message and expanding the DNA strand. With the first approach, $RAlign$, the redundant subsequences were each prepended and appended with unique 24-bp long markers to aid alignment, and replaced specific subsequences in the genome that were known by both the sender and receiver. With the second approach, $HTAlign$, the prepended and appended markers were identical, which means the receiver does not need to know the location of the replaced subsequences, but can find the repeated messages by searching for the markers alone.

Wanting to improve on the very low bits per base density of previous schemes, Ailenberg and Rotstein suggested an improved Huffman code that could decrease the sequence sizes by up to 40% [3] compared to other schemes. Keyboard characters were mapped to variable length sequences, with the more common characters assigned to the shorter sequences. To help maintain C-G balance, CG-rich codons were pushed downwards on the frequency table. To illustrate the density that could be achieved, text and musical notes for "Mary Had a Little Lamb", along with an image of a lamb made with geometric shapes, were encoded into a mere 844-bp DNA fragment. It was estimated that this encoding scheme would average 0.286 characters per base, which was a noted improvement over previous schemes. No error correction was used.

Heider et al. [27] were interested in the possibility of encoding data in non-protein coding but biochemically functional DNA. 2-3 character watermarks were inserted into non-coding promoter regions and a regulatory RNA region in $Escherichiacoli$. The introduction of the watermarks

were shown to disrupt the reading of the genes associated with the promoters and changed the secondary structure of the regulatory RNA molecule. This research was significant in showing that biochemically functional ncDNA is a poor region for encoding.

Chun et al. [13] were the first to design a cyber attack model they named "DNA courier attack" using encoded DNA. To ensure secrecy, the message was first encrypted into ciphertext using the encryption algorithm AES-128 with a secret key. Then a probability distribution of the DNA fragments in the non-coding regions was created. The mapping table was formed by sorting the codons and 5-bit sequences by frequency and matching the two directly. After insertion of the message, fake data was then embedded to rebalance the original codon distribution and further hide the message data. The encoded sequence was then inserted into an isolated plasmid, which was injected into a bacterial cell that can multiply and be hidden so the sender can carry it through a tight security checkpoint if necessary.

5 Gaps and Future Directions

There are several aspects of cellular DNA storage that have seen little research, despite their substantial importance for message encoding. In this section, we illuminate why these areas deserve further study, and we suggest novel approaches for future encoding schemes.

5.1 Addressing

Artificial DNA researchers have already taken advantage of data addressing to effectively access random data sections in encoded synthetic DNA and also update those sections with new data [18]. This useful technique has yet to be implemented in cellular DNA, and should be a primary direction of future research. With addressing, a lengthy message could be split up and inserted into multiple parts of a genome, taking advantage of pcDNA, ncDNA, and plasmids simultaneously.

5.2 Memory Limits

Currently, the largest amount of data inserted in to cellular DNA has been 844 base pairs [3]. Most organisms have genomes numbering in the billions of base pairs, which begs the question of how much encoded data can cells safely maintain? And how would those maximum lengths differ between organisms?

The DNA substitutions in pcDNA schemes could hypothetically be done in every pcDNA region, but insertions into ncDNA are expected to have a limit. If a DNA strand is expanded too far beyond it's normal length, cellular disruption could occur. But so far that limit has not been tested in any genome.

The problem of single lengthy insertions could be avoided by using addressing techniques mentioned previously. If the data of a long insertion were dispersed to multiple parts of the genome and accessed via addressing, much more data could be inserted into the genome. So the question of memory limits is twofold: 1.) How much can be inserted into single regions? and 2.) How much data can be safely inserted into a genome as a whole? Knowing the limitations for different organisms would be invaluable knowledge as the technology for cellular DNA encoding continues to develop.

5.3 Compression

Advances in cellular DNA encoding have so far been made using short messages. Therefore, the need for compression techniques has been minimal. Compression also creates additional difficulties because it allows mutative errors to disrupt more of the data. But if cellular DNA is to become a more viable method of secure data encoding, more research must be done into methods that compress data while at the same time reducing damage from errors.

The clearest way to accomplish this is with compression matched with error correction. The DNA-Crypt algorithm implements a fuzzy controller that determines what type of error correction is best for optimal performance given mutation rate, sequence length, and stability [25]. We believe a similar sort of fuzzy controller could be used to determine ideal error correction methods by adding an additional parameter for compression density.

Both compression and error correction are already being used widely in artificial DNA encoding. Many of those techniques could be transferred to cellular DNA encoding with few changes. Compression would be particularly useful for transferring large private messages in cellular DNA. For the purposes of watermarking, it may not be necessary. But due to the lack of implementation in the literature and the clear usefulness of compression, we believe it is an important direction for future research.

5.4 Mitochondrial DNA

Heider et al. [26] suggested using mitochondria for secret data encoding. Mitochondria are organelles in the cell that contain a small amount of DNA that encodes translation machinery and oxidative chain components [20]. Heider et al. identified an mtDNA section 1,541 base pairs long that contains no active gene regions and suggested that it would be an ideal location for embedding data with synonymous codons. They created a program called $ProjectMito$, derived from $DNA-Crypt$, that encrypts binary files and modifies them with the Hamming code for error correction, before translating the message into an mtDNA sequence. $ProjectMito$ can also be used for decryption after the DNA is sequenced. $ProjectMito$ inserts data into the Cytochrome c oxidase subunit I gene, allowing for the insertion of up to 60 bytes.

Since the development of $ProjectMito$, little research has been done on Mitochondrial DNA encoding, and the tool itself has seen little use by other researchers. The potential for mitochondrial encoding is still largely untapped, and should be an important focus for future research. Mitochondria contain genes for both proteins and tRNA. The tRNA genes, due to their unique functionality in the cell, may have slightly different bio-restrictions than the protein coding genes. Designing encoding schemes to meet tRNA gene restrictions is another avenue requiring more research.

5.5 Methylation Encoding

Methylation is the process by which cells repress the expression of their own genes by attaching methyl groups to specific adenine and cytosine nucleotides of those genes [43]. Methylation is maintained after cell division by a family of enzymes known as DNA methyltransferase. Because the cell can preserve these methyl groups even after replicating, the authors of this survey would like to propose encoding via methylation as a completely new form of DNA encoding.

If the sender and receiver can agree on a specific start location in junk DNA for the message, the data can be encoded in binary via methylation. For every cytosine (and adenine for bacteria and plants) found after the start location, the presence of a methyl group represents bit 1 while the absence of a methyl group indicates bit 0. If the receiver of the message had an easy way to read the methyl groups ahead of the site, it would be trivial for them to decode the embedded data.

This technique of methylation encoding would be restricted only by the insertion location. If the embedding site is in truly non-functional ncDNA, the encoder does not need to be concerned about the other bio-restrictions associated with ncDNA. This would allow for raw binary encoding without the need for complex algorithms and schemes to get around the common bio-restrictions.

While the technology for site-specific methylation is still in its infancy, we can expect that soon the ability to encode and read methyl groups in DNA will be more available and economical. When this happens, we believe methylation encoding could become an efficient and useful form of cellular DNA encoding.

6 Conclusion

Great strides have been made in both synthetic and cellular DNA encoding, with a focus on long term data storage for synthetic DNA, and steganography and watermarking for cellular DNA. In the near future, computer data centers may partially replace their large storage spaces with smaller shelves of synthetic DNA containing the same amount of information. In a few years, government agents or private individuals wishing to transmit secret messages might do so by encoding the messages in bacterial DNA and hiding the bacteria on their bodies. Corporations that have created patented genes are already looking at ways to encode secret watermarks inside the genomes of their novel organisms.

The potential for cellular DNA encoding continues to grow. Data has been inserted into pcDNA and ncDNA, bypassing the biological restrictions associated with those areas through encoding schemes. Encryption, error correction, and data compression have also been implemented in a variety of ways by these schemes to make them more useful. Future research into cellular DNA encoding has many possible avenues which could provide important benefits over existing schemes. Nature has been using DNA to store biological data for millions of years, and finally humans are learning to use the same medium for our own data.

References

[1]A. A. Abdelwahab and L. A. Hassaan. A discrete wavelet transform based technique for image data hiding. In *Radio Science Conference, 2008. NRSC 2008. National*, pages 1–9. IEEE, 2008.

[2]N. Abdulaziz and K. Pang. Robust data hiding for images. In *Communication Technology Proceedings, 2000. WCC-ICCT 2000. International Conference on*, volume 1, pages 380–383. IEEE, 2000.

[3]M. Ailenberg and O. Rotstein. An improved huffman coding method for archiving text, images, and music characters in dna. *BioTechniques*, 47:747–754, 2009.

[4]M. Arita and O. Yoshiaki. Secret signatures inside genomic dna. *Biotechnology Progress*, 20:1605–1607, 2004.

[5]D. Artz. Digital steganography: hiding data within data. *IEEE Internet Computing*, 5(3):75–80, 2001.

[6]F. e. a. Blattner. The complete genome sequence of escherichia coli k-12. *Science*, 277:5331:1453–1462, 1997.

[7]M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. Pruitt, and G. Church. Forward error correction for dna data storage. *Procedia Computer Science*, 80:1011–1022, 2016.

[8]J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss. A dna-based archival storage system. In *Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 637–649. ACM, 2016.

[9]W. S. R.-V. E. H. S. e. a. Brenner, S. In vitro cloning of complex mixtures of dna on microbeads: physical separation of differentially expressed cdnas. *Proceedings of the National Academy of Sciences*, 97(4):1665–1670, 2000.

[10]T. Brunet. Aims and methods of biosteganography. *Journal of Biotechnology*, 226:56–64, 2016.

[11]C.-C. Chang, P. Tsai, and M.-H. Lin. An adaptive steganography for index-based images using codeword grouping. In *Pacific-Rim Conference on Multimedia*, pages 731–738. Springer, 2004.

[12]W.-Y. Chen. Color image steganography scheme using set partitioning in hierarchical trees coding, digital fourier transform and adaptive phase modulation. *Applied Mathematics and Computation*, 185(1):432–448, 2007.

[13]J. Chun, H. Lee, and J. Yoon. Passing go with DNA sequencing: Delivering messages in a covert transgenic channel. *IEEE CS Security and Privacy Workshop*, 14:121, 2013.

[14]G. M. Church, Y. Gao, and S. Kosuri. Next-generation digital information storage in dna. *Science*, 337(6102):1628–1628, 2012.

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

[16]E. P. Consortium et al. An integrated encyclopedia of dna elements in the human genome. *Nature*, 489(7414):57–74, 2012.

[17]L. Dailey Paulson. New system fights steganography, 2006.

[18]P. DeSilva and G. Ganegoda. New trends of digital data storage in dna. *Biomed Research International*, 2016.

[19]A. M. Fard, M.-R. Akbarzadeh-T, F. Varasteh-A, and F. Varasteh-A. A new genetic algorithm approach for secure jpeg steganography. In *2006 IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6. IEEE, 2006.

[20]R. Garesse and C. Vallejo. Animal mitochondrial biogenesis and function: a regulatory cross-talk between two genomes. *Gene*, 263:1–16, 2001.

[21]N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E. M. LeProust, B. Sipos, and E. Birney. Towards practical, high-capacity, low-maintenance information storage in synthesized dna. *Nature*, 494(7435):77–80, 2013.

[22]R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark. Robust chemical preservation of digital information on dna in silica with error-correcting codes. *Angewandte Chemie International Edition*, 54(8):2552–2555, 2015.

[23]D. Haughton and F. Balado. Repetition coding as an effective error correction code for information encoded in dna. *Internation Conference on Bioinformatics and Bioengineering*, 2011.

[24]D. Haughton and F. Balado. Biocode: Two biologically compatible algorithms for embedding data in non-coding and coding regions of dna. *BMC bioinformatics*, 14(1):1, 2013.

[25]D. Heider and A. Barnekow. Dna-based watermarks using the dna-crypt algorithm. *BMC bioinformatics*, 8(1), 2007.

[26]D. Heider, D. Kessler, and A. Barnekow. Watermarking sexually reproducing diploid organisms. *Bioinformatics*, 24:17:1961–1962, 2008.

[27]D. Heider, M. Pyka, and A. Barnekow. Dna watermarks in non-coding regulatory sequences. *BMC Research Notes*, 2:123, 2009.

[28]H. Hirohisa. A data embedding method using bpcs principle with new complexity measures. In *Proc. of Pacific Rim Workshop on Digital Steganography*, pages 30–47, 2002.

[29]D. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[30]K.-H. Jung and K.-Y. Yoo. Data hiding method using image interpolation. *Computer Standards & Interfaces*, 31(2):465–470, 2009.

[31]A. Khalifa and S. Hamad. Hiding secret information in dna sequences using silent mutations. *British Journal of Mathematics and Computer Science*, 11(5):1–11, 2015.

[32]D. Kracht and S. Schober. Insertion and deletion correcting dna barcodes based watermarks. *BMC Bioinformatics*, 16:50, 2015.

[33]S. Lee. Dwt based coding dna watermarking for dna copyright protection. *Information Sciences*, 273:263–286, 2014.

[34]Z. Li, X. Chen, X. Pan, and X. Zeng. Lossless data hiding scheme based on adjacent pixel difference. In *Computer Engineering and Technology, 2009. ICCET'09. International Conference on*, volume 1, pages 588–592. IEEE, 2009.

[35]S. Lin and D. J. Costello. *Error control coding*. Pearson Education India, 2004.

[36]M. Liss, D. Daubert, K. Kliche, U. Hammes, A. Leiherer, and R. Wagner. Embedding permanent watermarks in synthetic genes. *PLOS One*, 7:8, 2012.

[37]C. Manikopoulos, Y.-Q. Shi, S. Song, Z. Zhang, Z. Ni, and D. Zou. Detection of block dct-based steganography in gray-scale images. In *Multimedia Signal Processing, 2002 IEEE Workshop on*, pages 355–358. IEEE, 2002.

[38]R. T. McKeon. Strange fourier steganography in movies. In *2007 IEEE International Conference on Electro/Information Technology*, pages 178–182. IEEE, 2007.

[39]M.-K. A.-W. W. . H. M. M. Mousa, H. Data hiding based on contrast mapping using dna medium. *International Arab Journal of Information Technolology*, 8(2):147–154, 2011.

[40]E. Palkopoulou, S. Mallick, P. Skoglund, J. Enk, N. Rohland, H. Li, A. Omrak, S. Vartanyan, H. Poinar, A. Götherström, et al. Complete genomes reveal signatures of demographic and genetic declines in the woolly mammoth. *Current Biology*, 25(10):1395–1400, 2015.

[41]V. M. Potdar, S. Han, and E. Chang. Fingerprinted secret sharing steganography for robustness against image cropping attacks. In *INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pages 717–724. IEEE, 2005.

[42]V. M. Potdar, S. Han, and E. Chang. A survey of digital image watermarking techniques. In *INDIN'05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, pages 709–716. IEEE, 2005.

[43]D. Ratel, J. Ravanat, F. Berger, and D. Wion. N6-methyladenine: the other methylated base of dna. *Bioessays*, 28(3):309–315, 2006.

[44]D. R.-C. R.C. Bose. On a class of error correcting binary group codes. *Information and Control*, 3(1):68–79, 1960.

[45]I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.

[46]P. Sallee. Model-based steganography. In *International workshop on digital watermarking*, pages 154–167. Springer, 2003.

[47]G. Smith, C. Fiddles, J. Hawkins, and J. Cox. Some possible codes for encrypting data in DNA, volume = 25, pages = 1125-1130, year = 2003. *Biotechnology Letters*.

[48]H. Tanaka. Evaluation of information leakage via electromagnetic emanation and effectiveness of tempest. *IEICE Transactions on Information and Systems*, 91(5):1439–1446, 2008.

[49]B. Verma, S. Jain, and D. Agarwal. Watermarking image databases: a review. In *Proceedings of the International Conference on Cognition and Recognition, Mandya, Karnataka, India*, pages 171–179, 2005.

[50]E. Viguera, D. Conceill, and S. Ehrlich. Replication slippage involves dna polymerase pausing and dissociation. *The Embo Journal*, 20(10):2587–2595, 2001.

[51]J. Watson, T. Baker, S. Bell, A. Gann, M. Levine, and R. Losich. *Molecular Biology of the Gene, 6th Edition*. Pearson, 2008.

[52]J. Watson and F. Crick. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171 (4356):737–738, 1953.

[53]P. Wong, K. Wong, and H. Foote. Organic data memory using the dna approach. *Communications of the ACM*, 46:1:95–98, 2003.

[54]Y.-T. Wu and F. Y. Shih. Genetic algorithm based methodology for breaking the steganalytic systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 36(1):24–31, 2006.

[55]N. Yachie, K. Sekiyama, J. Sugahard, Y. Ohashi, and M. Tomita. Alignment-based approach for durable data storage into living organisms. *Biotechnology Progress*, 23:501–505, 2007.

[56]S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic. A rewritable, random-access dna-based storage system. *Scientific reports*, 5, 2015.