2022 IEEE 40th International Conference on Computer Design (ICCD) | 978-1-6654-6186-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCD56317.2022.00071

HL-DNA: A Hybrid Lossy/Lossless Encoding Scheme to Enhance DNA Storage Density and Robustness for Images

Yi Li†, David H.C. Du*, Li Ou**, and Bingzhe Li†
†Department of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, USA
*Department of Computer Science and Engineering, University of Minnesota, Twin Cities, Minneapolis, MN, USA
*Department of Pediatrics, University of Minnesota, Twin Cities, Minneapolis, MN, USA
{yli12, bingzhe.li}@okstate.edu; {du, ouxxx045}@umn.edu

***Abstract*—With the storage's demand for high density and long-term preservation, Deoxyribonucleic Acid (DNA) has become a promising candidate to satisfy the requirement of archival storage for rapidly increased digital volume. However, due to the biochemical constraints, DNA storage faces critical issues of low practical capacity and robustness. In this paper, we target image applications and propose to apply approximation to DNA storage to improve the overall encoding density and robustness of DNA storage by using a hybrid lossy and lossless encoding scheme (called HL-DNA). Several lossy and lossless encoding schemes (lossy and lossless codes) are proposed and used to encode incoming binary sequences. These two types of codes are coordinated to balance the encoding density and errors. The lossless codes are used to limit the errors and the lossy codes are used to improve the encoding density. Moreover, the introduced approximation and newly proposed hybrid encoding schemes in one DNA strand can improve the robustness of DNA storage. Finally, the experimental results indicate that the proposed HL-DNA improves the encoding density of DNA storage and makes it much close to the ideal case. Also, HL-DNA achieves higher robustness to the injected errors than other DNA storage codes.**

***Index Terms*—DNA Storage, Approximation, Encoding density**

I. Introduction

With the explosively increased digital data, in past decades many emerging storage devices have been investigated such as Solid-State Drive (SSD) [1], [2], Shingled Magnetic Recording (SMR) [3], [4], Interlaced Magnetic Recording (IMR) [5], [6], etc. However, the capacity of existing storage media cannot keep up with the growth of the created digital data. Also, all devices could become obsolete within several years. The data stored in the traditional storage devices are vulnerable as they will perish in a few years. Therefore, synthetic deoxyribonucleic acid (DNA) becomes an attractive alternative storage medium due to its potential of high density and long durability. DNA storage can achieve a theoretical density of 455 Exa-bytes/gram [7] and has a long-lasting property for several centuries and beyond.

[1]This work was partially supported by NSF I/UCRC Center Research in Intelligent Storage and the following NSF awards 2208317, 2204656, and 2204657.

However, the current DNA storage faces two critical issues. One is that the practical DNA storage is far from the expected capacity (e.g., terabytes per tube). As demonstrated in [8], [9], a single tube capacity of DNA storage is only about several hundreds gigabytes (e.g., only about 230 GB per tube indicated in [9]). This is because of the bio-constraints in the DNA storage. For example, to implement random access in DNA storage, primers are necessary to be used and the number of available primers is proportional to the DNA storage capacity. However, the number of available primers will be significantly decreased as the amount of digital data increases. Moreover, ideally since there are four types of DNA nucleotides (i.e., A, T, G, and C), each nucleotide can represent 2-bit binary data, that is, the encoding density is 2 bits per nucleotide (i.e., 2 bits/nt). To avoid those biochemical constraints in DNA storage, the encoding densities of previous studies are far from the ideal encoding density. Secondly, the DNA storage is error-prone. DNA storage faces different types of errors during synthesis and sequencing processes such as deletion, substitution, and insertion [8]. Those errors can significantly reduce the original digital data quality [10].

To improve the densities and robustness of DNA storage, different encoding schemes [8], [11], [12] (i.e., converting digital data into DNA sequences), error-correction codes, and biochemical technologies are proposed to improve the efficiency of synthesis and sequencing processes. Rare of them investigated how to efficiently improve the encoding density and robustness of DNA storage together based on the properties of those applications. Li et al. [10] proposed IMG-DNA, which is the most recent study to increase the reliability of DNA storage by using the properties of approximate storage systems. However, IMG-DNA faced low encoding density since they directly use the rotation code proposed by Bornholt et al. [13]. Therefore, as inspired by the approximate storage systems, there is a great potential to increase the encoding density of DNA storage for approximate applications such as images.

In this paper, we target image applications to increase the encoding density and robustness of DNA storage. We propose

2576-6996/22/$31.00 ©2022 IEEE
DOI 10.1109/ICCD56317.2022.00071

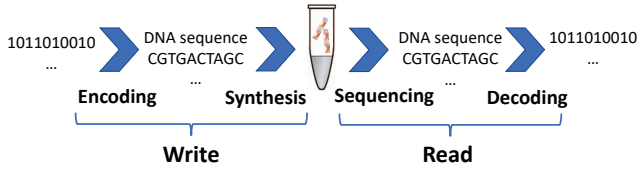


Fig. 1: Basic steps of DNA storage.

a hybrid lossy/lossless encoding scheme, called HL-DNA, to convert binary data to DNA sequences. Several lossy encoding manners are newly proposed, which have the ideal encoding density (i.e., 2bits/nt) but may introduce some approximation during the decoding process. To complement those lossy codes, we also propose lossless encoding schemes to reduce the errors in DNA storage. An adaptive selection scheme will be used for each DNA strand based on the error rate and encoding density. Moreover, a partition scheme is proposed to efficiently prevent the error propagation scenario of DNA storage and increase the robustness of DNA storage. Finally, the HL-DNA scheme significantly improves the encoding density and robustness compared to prior work.

The remainder of this paper is organized as follows: Section II describes the background of DNA storage. Section III introduces the implementation of HL-DNA scheme. The feasibility and overhead discussion is introduced in Section IV. Section V shows the experimental results of HL-DNA and compare them with some prior work. Some conclusions are drawn in Section VI.

II. Background

In this section, we mainly introduce the background of DNA storage.

DNA storage basics: In DNA, four basic nucleotides (i.e., Adenine (A), Cytosine (C), Guanine (G), and Thymine (T)) as the basic elements to construct DNA sequences. Essentially, a single DNA strand or oligonucleotide is composed of a number of nucleotides. Two complementary DNA strands are bonded together based on the complementary pairs, which form a double helix. In this helix form, A and T are a complementary pair, and C and G are aligned with each other. For DNA storage, people store binary digital data in DNA strands, which has much high density and long-term preservation. To implement the DNA storage, as shown in Fig. 1, four major processes in the DNA storage system are used: encoding, synthesis, sequencing, and decoding. The encoding and synthesis refer to the processes of writing digital data to DNA storage. The sequencing and decoding indicate the processes of reading data out from DNA storage.

Encoding and decoding: since DNA has four types of nucleotides and the digital binary has only two types of data, '1' and '0', to store binary digits to DNA storage, first of all, we need to convert digital data into DNA format (i.e., A, T, G, and C). Ideally, to map binary data to nucleotides, we can achieve 2 bits per nucleotides encoding density (2 bits/nt) since each nucleotide can represent two binary bits

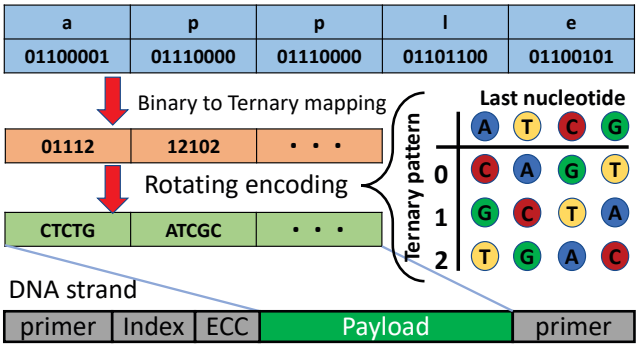


Fig. 2: DNA architecture with the rotation encoding manner [13].

(e.g., 00→A, 01→T, 10→C and 11→G). However, the ideal encoding method will introduce much high error rates during the synthesis and sequencing processes due to biochemical constraints such as homopolymers. In other words, the synthesis and sequencing processes are not friendly to some specific DNA patterns and thus we should avoid them to make write and read easily in DNA storage. Therefore, most existing studies [8], [11], [13]–[16] avoid the special patterns such as homopolymers and obtain much lower encoding density than 2 bits/nt. The decoding process is a reverse process of encoding. After reading DNA strands out from sequencing machines, DNA sequences are decoded back to binary data according to the encoding schemes.

DNA Storage Errors: In DNA storage, there are three major types of errors including deletion, insertion and substitution. More specifically, some nucleotides might be written or read by another type of nucleotides (i.e., substitution) or may not be written into the DNA strands or read out (i.e., deletion) or newly added into or read out (i.e., insertion). The substitution error rate may reach 0.8% per base, which are around 2X-10X higher than the other two types [8], [17]. Moreover, some features may increase the error probability in the synthesis and sequencing processes. For example, due to the technology limitation, with increasing DNA strand length, it becomes harder to add more nucleotides on the DNA strands. In other words, when the strand length increases, the errors happening on each nucleotide bind also exponentially increase [8], [15], [18]–[21]. Therefore, most of the existing works for DNA storage use 100˜300 bp length of a DNA strand. GC content (i.e., the percentage of bases in a DNA sequence that are either C or G) is related to the melting temperature of DNA strands during PCR. Too high and too low GC content cause too high melting point and unspecific binding with primers, respectively, which makes PCR more difficult and error-prone.

Some other features such as long homopolymers (e.g., AAAAA), hairpins/loops/secondary structures, and other forms of higher-order structures [22] also induce the difficulty of writing or reading DNA strands. Although hairpins/loops/secondary structures and higher-order structure in-

crease the difficulty of sequencing, DNA strands with those structures are possibly read out by one or several time sequencings. Therefore, most of the existing DNA storage studies [8], [11], [13]–[16], [23]–[26] proposed their encoding schemes only intentionally avoiding the long homopolymers and low/high GC content since the hairpins/loops/secondary structures and other forms of a higher-order structure are hard to avoid and might be mitigated by sequencing multiple times.

III. HL-DNA Algorithm Design

In this section, we introduce our proposed HL-DNA algorithm design. First, we describe the lossless encoding schemes, which achieve a lower encoding density but no error is introduced. Based on that, an extension of lossless encoding schemes is proposed to add approximation to the encoding scheme to increase the encoding density. After that, with a pool of encoding schemes, the HL-DNA scheme demonstrates the algorithm of combining all those proposed schemes to increase the encoding density of the DNA storage system while remaining a low error rate for those image applications. Finally, we proposed a new partition scheme for HL-DNA to improve the robustness of DNA storage.

A. Lossless Encoding Schemes

As discussed in the background section, due to the biochemical constraints, the DNA storage encoding scheme should satisfy some rules to mitigate the induced errors during synthesis and sequencing processes. The biochemical constraints include the absence of homopolymers (e.g., AAAA) and low/high GC content (i.e., the ratio of G and C in the DNA sequence). To satisfy the biochemical constraints, people [13] proposed the rotation code, in which the current encoded nucleotide is based on the current binary bits and the last nucleotide.

As inspired by the rotation code [13], we propose new lossless codes with a rotation manner. Lossless means that the digital binary data encoded into DNA sequences can be correctly recovered from the decoding process theoretically. Two encoding manners (i.e., C0 and C1) are indicated in Fig. 3. For each encoding manner, unlike the rotation code [13], the binary patterns will have different encoding densities. We define a digital pattern based on two binary digits, and thus there are four combinations (i.e., ‘00’, ‘01’, ‘10’, and ‘11’) for 2-bit patterns. Those two encoding manners (C0 and C1) achieve different encoding densities for these 2-bit patterns. For example, C0 encodes any 2-bit pattern with the first bit of ‘0’ into one nucleotide and others into two nucleotides. That is, the patterns ‘00’ and ‘01’ will be encoded into one nucleotide, and the patterns ‘10’ and ‘11’ will be encoded into two nucleotides. In contrast, C1 encodes the patterns ‘10’ and ‘11’ into one nucleotide and ‘00’ and ‘01’ into two nucleotides.

According to the C0 and C1 encoding manners, if a binary sequence contains more ‘0X’ patterns (where ‘X’ can be either ‘0’ and ‘1’), C0 can achieve a higher encoding density than C1, vice versa. Therefore, based on the pattern frequencies in a binary sequence, we can select different encoding manners

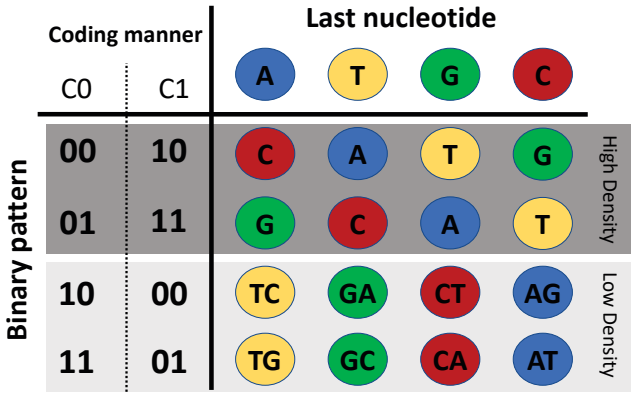


Fig. 3: Combined lossless code Design.

to achieve a higher encoding density. The details of how to achieve the hybrid encoding scheme are introduced in Section III-D.

B. Lossy Encoding Schemes

Based on the properties of image applications, a few errors in those data will not affect the data quality too much. For example, even though errors are added to some pixels of an image, we can still clearly recognize the image. Based on that, in this work, we can use lossy codes to enhance the overall encoding density of DNA storage for those approximate applications.

As observed from Fig. 3, in the lossless encoding scheme, the low-density portion can only have the encoding density of 1 bit/nt. Therefore, if an incoming binary sequence contains many ’10’ and ’11’ patterns, the encoding density will be close to 1 bit/nt by using C0 encoding scheme. Similarly, C1 encoding scheme will face low encoding density when a binary sequence includes a lot of ’00’ and ’01’ patterns. These two codes have four rows (i.e., four binary patterns). The reason is that 2-bit binary has four types of patterns but DNA only has four types of nucleotides. If we want to achieve a rotation manner that only encodes binary patterns to one nucleotide, it must satisfy that the number of patterns is one less than the number of types of nucleotides (i.e., 4). That is, the number of binary patterns needs to be 3. By doing so, each pattern will be encoded into only one nucleotide as indicated in Fig. 2. Another observation from Fig, 3 is that the binary patterns in the low-density portion of each proposed lossless coding manner have the same first bit. In other words, C0 follows ’1X’ format and C1 follows ’0X’ format, where ‘X’ indicates either ‘0’ or ‘1’.

According to the above discussion, we apply the approximation to the encoding scheme and propose lossy encoding manners to improve the encoding density. As indicated in Fig. 4, the lossless code can be extended to the lossy design. For the lossy design, each coding manner covers three binary patterns. The first two binary patterns are the same as that of lossless code design, but the last one in the lossy design has an uncertain pattern. Taking C10 as an example in Fig. 4,

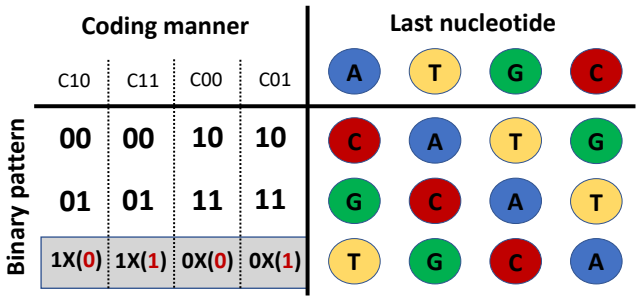


Fig. 4: Combined lossy code design. The binary pattern 'AX(C)' indicates that the patterns 'A0' and 'A1' belongs to this pattern but will be decoded to 'AC'. For example, '1X(0)' refers to the patterns '10' and '11' but will be decoded to '10'.

if a 2-bit pattern has the first bit as '1', it will belong to the pattern of '1X(0)'. Then, the digital patterns of '10' and '11' will be encoded to the same nucleotide. For the decoding process, '(0)' in '1X(0)' indicates that the nucleotides will be decoded back to '10'. Therefore, 'lossy' in this design comes from the patterns of '10' and '11' only decoded back to '10' in C10. By doing so, the encoding density of C10 is 2 bits/nt although it introduces errors. Similarly, C11, C00, and C01 follow the same concept as C10. The only difference between them is which binary pattern contributes to the lossy encoding. The reason for using multiple coding manners is that if the frequency of one specific binary pattern is really high and the pattern belongs to the lossy part, the binary sequence might be inserted with a high error rate resulting in low data quality. With multiple lossy encoding manners, these encoding manners introduce errors from different specific binary patterns. So, if we can analyze the frequency of the patterns in binary sequences first, we can reduce the errors by using the combination of multiple encoding manners. The details of the hybrid mapping scheme are introduced in Section III-D.

C. Partition Scheme

As introduced in the background section, the DNA storage is error-prone, and the sequencing results may have errors such as insertion, deletion, and substitution. Those errors may cause a significant impact on the quality of digital data. We follow similar idea from [10] to insert 'A' into one DNA strand at a specific position to fit the proposed encoding scheme. As demonstrated in Fig. 5, a new start point 'A' is inserted into one DNA strand at a specific position. The DNA chunks before and after the start point could be encoded with different encoding schemes. The advantages of using the partition are twofold. One is similar to IMG-DNA [10] the partition can improve the robustness of DNA storage due to preventing the error propagation. The other advantage is that the partition enables multiple encoding schemes used in one DNA strand. Since one DNA strand may follow different pattern distributions, multiple encoding schemes in one strand can further improve the encoding density.

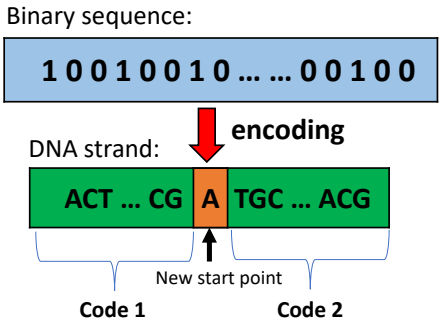


Fig. 5: An example of partition scheme by inserting a start point in a DNA strand.

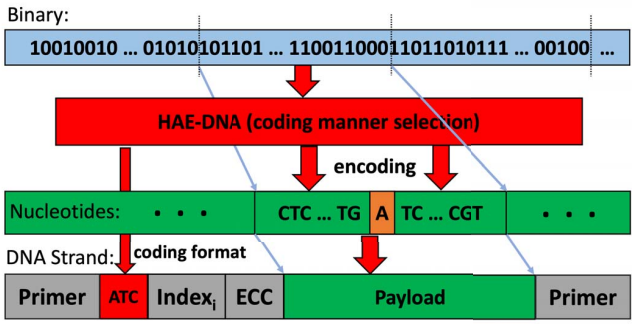


Fig. 6: The architecture of HL-DNA encoding process.

The reason to choose 'A' as a new start point is that the 'XAX' pattern ('X' is one of the other three nucleotides (i.e., T, G, and C)) has the lowest error rate compared to other patterns [8]. The position of the start point can be recorded in the metadata of DNA storage, which is used to identify the start point of the next DNA chunk. Based on the encoding schemes, the positions of the start points might be varied, and it may introduce a large overhead to record them. In this paper, we assume those starting partitions use lossy codes and thus the number of nucleotides can be pre-computed. In other words, all DNA strands will have the same position of the start point.

D. Overall HL-DNA Scheme

In this subsection, we introduce the overall architecture of the hybrid lossy/lossless encoding (HL-DNA) scheme. As indicated in the previous sections, there are two types of encoding manners, lossless and lossy. The lossy encoding manner introduces errors but improves the encoding density. The lossless manner will compensate for the lossy encoding manner to mitigate the error rate. Therefore, these two encoding manners are balanced with each other to improve the overall encoding density while remaining good data quality.

For the whole process of the HL-DNA scheme, as introduced in Fig. 6, a long bitstream will be first chunked into a fixed-length (e.g., 400 bits). If we have two partitions, the fixed-length bitstream will further partition into two bit-

streams. Then, the fixed bitstreams will go through the HL-DNA algorithm to select a code for each short bitstream, which achieves a good encoding density while remaining a low error rate (the details are shown in the following paragraph). After that, the corresponding metadata information including primer, internal index, encoding format, and error-correction code (ECC) are attached to the payload. The new start point will be also inserted at the beginning of the second partition. Then, each DNA strand will be synthesized into DNA sequences. To decode a DNA strand, the process is a reserved process of encoding. First, the DNA strand will be amplified via PCR and sequenced out. The encoding format field will first determine the encoding manners of the DNA strand. Finally, the corresponding binary sequence will be decoded following the encoding manner. For the encoding field, since we have a total of six encoding manners (i.e., C0, C1, C00, C01, C10, and C11) and two partitions in each DNA strand, it requires three nucleotides to represent them in the format field. If we have more partitions or more types of encoding manners, the coding format field needs more nucleotides to distinguish the encoding scheme for each partition in one DNA strand, and thus it will cause a higher overhead.

Algorithm 1 indicates the details of the HL-DNA scheme. First, for each incoming binary sequence, the frequencies of four binary patterns (i.e., '00', '01', '10', and '11') in the binary sequence will be computed (Line 6). Then, two encoding paths are used. The first one is to select a lossless code (Lines 7-12). Based on the densities, we will select the lossless code with a higher encoding density. The second path is to choose a proper lossy code among four codes in Fig. 4 (Lines 13-24). Since all those lossy codes have an encoding density of 2 bits/nt, the selection criterion is based on how much error the code introduces. Finally, according to the encoding density of lossless codes and error rates of lossy codes, a proper code is selected to encode the corresponding binary sequence. Since the state-of-the-art code using the rotating encoding [8] has an encoding density of about 1.6 bits/nt, we use 1.65 bits/nt as a threshold for the lossless codes to make sure that the overall encoding density of HL-DNA is higher than the state-of-the-art codes. The error rate threshold is another tuning parameter, which is defined by the number of induced wrong bits divided by the total number of binary bits. The error rate threshold balances the trade-off between data quality and encoding density. We take an empirical value of 0.1 as a default value. In the experiment, we investigate the influence of the error rate threshold on the data quality and encoding density.

Algorithm 1 HL-DNA Algorithm

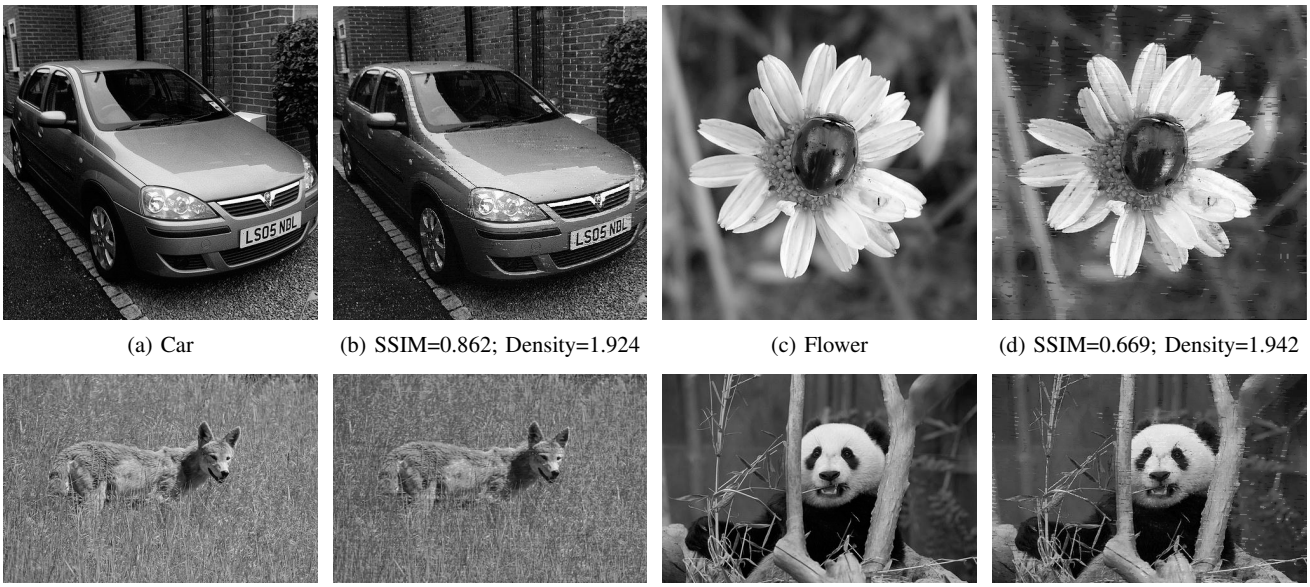
1: **Inputs:** BinarySeqs //**Binary sequences**//
2: **Outputs:** DNASeqs //**DNA sequences**//
3: **procedure** HL-DNA ENCODING ALGORITHM(binary sequence BinarySeqs)
4: binary_len = length(BinarySeqs)
5: **for** i in binary_len **do**
6: Compute frequencies f_{xx} of four binary patterns '11', '10', '01', '00'
7: **if** $f_{11} + f_{10} \geq f_{00} + f_{01}$ **then**
8: density_lossless = $\frac{binary\_len}{length(C1(i))}$
9: DNA_lossless = C1(i)
10: **else**
11: density_lossy = $\frac{binary\_len}{length(C0(i))}$
12: DNA_lossless = C0(i)
13: **if** $f_{00} == min(f_{00}, f_{01}, f_{10}, f_{11})$ **then**
14: density_lossy = $\frac{binary\_len}{length(C01(i))}$
15: DNA_lossy = C01(i)
16: **else if** $f_{01} == min(f_{00}, f_{01}, f_{10}, f_{11})$ **then**
17: density_lossy = $\frac{binary\_len}{length(C00(i))}$
18: DNA_lossy = C00(i)
19: **else if** $f_{10} == min(f_{00}, f_{01}, f_{10}, f_{11})$ **then**
20: density_lossy = $\frac{binary\_len}{length(C11(i))}$
21: DNA_lossy = C11(i)
22: **else**
23: density_lossy = $\frac{binary\_len}{length(C10(i))}$
24: DNA_lossy = C10(i)
25: $err = \frac{min(f_{00}, f_{01}, f_{10}, f_{11})}{f_{00}+f_{01}+f_{10}+f_{11}}$
26: **if** density_lossless $\leq$ 1.65bits/nt **or** $err < Threshold$ **then**
27: DNASeqs[i] = DNA_lossy
28: **else**
29: DNASeqs[i] = DNA_lossless
30: Note: C0(), C1(), C00(), C01(), C10(), and C11() are the functions of encoding manners in Fig. 3 and Fig. 4.

IV. FEASIBILITY

This section mainly focuses on the feasibility of the proposed HL-DNA scheme. Due to the large data size and high-cost sequencing and synthesis in DNA storage, the experiments of this work are based on simulation. It is common to use the simulation-based feasibility check in biological fields since the simulation can reflect real-life events to avoid potential collision/fails, and improve the success rate of future wet-lab experiments. Moreover, the price of synthesis and sequencing in DNA storage can reach hundreds of thousand of dollars per GB. So, it is impractical to use expensive experiments to investigate all possible scenarios for a large amount of data.

Based on the commercial design rules [27], [28] for synthesis and sequencing efficiency, we design a set of rules to check the feasibility of the proposed scheme and previous studies [8], [13], [15]. The design rules are shown in the following:

- Absence of long homopolymers (in this paper we use a restrictive constraint and limits to avoid two consecutive identical nucleotides) [15], [16].
- GC contents (40% - 60%) [8], [27], [29].
- Avoiding secondary structure: try to avoid sequences containing two inverted repeats, separated by at least three nucleotides [27].
- DNA strand length smaller than 1000 bp [27], [28].



(a) Car (b) SSIM=0.862; Density=1.924 (c) Flower (d) SSIM=0.669; Density=1.942

(e) Wolf (f) SSIM=0.896; Density=1.837 (g) Panda (h) SSIM=0.735; Density=1.932

Fig. 7: Graphic view of different image examples with the error threshold of 0.1. (a), (c), (e) and (g) are original image examples. (b), (d), (f) and (h) are based on the HL-DNA schemes with the results of SSIM and Encoding Density.

TABLE I: Results of design rules check for HL-DNA and the reference work [8].

	Organick et al. [8]	HL-DNA
GC content (Average ratio)	50.71%	51.38%
long homopolymer violation	No	Only have 'AA'
Long DNA strand length	No	No
Self-sequence complementarity*	0.382	0.329

*We use the self-sequence complementarity to indicate the secondary structure in the encoded DNA strands. The value indicates the ratio of DNA strands containing the self-sequence complementarity.

Table I demonstrates the feasibility results of two schemes. For the GC content, both schemes have around 50% GC contents since those two schemes use rotating codes, which uniformly make the distribution of 'G' and 'C' nucleotides close to 25%. For the long homopolymers, both schemes can avoid long homopolymers due to rotating encoding behavior. For the proposed HL-DNA, since we use the partition scheme by injecting a new start point 'A' in the middle of DNA strands, it may create a 'AA' pattern. With the investigation, around 14% of DNA strands have only one 'AA' pattern. In general, a DNA strand can have up to four consecutive identical nucleotides with reasonable successful rates of synthesis and sequencing [16]. Therefore, the proposed HL-DNA only containing the 'AA' pattern in a small number of DNA strands will not cause a problem for the synthesis and sequencing processes. For the self-sequence complementarity structure, we expected the number of the self-sequence complementarity in DNA strands as small as possible. As shown in Table I, the proposed HL-DNA achieves a little smaller ratio of self-sequence complementarity than the referenced work [8]. Finally, after comparing all design rules, the proposed HL-DNA scheme achieves similar scores with the validated reference. Therefore, we can conclude that the proposed HL-DNA will have similar difficulties to the work [8] during synthesis and sequencing processes, and be highly possible that all encoded DNA strands by HL-DNA can be implemented in the wet-lab experiments.

V. Experimental Results

This section presents the experimental results with the comparison of the encoding density between different schemes. We use three baseline encoding schemes denoted and used by Church et al. [11], Organick et al. [8] and Blawat et al. [16]. The default DNA strand length is about 300 bp for all four schemes. We run the simulation based on the system environment as indicated in Section IV. A dataset of images [30] is used with about 4GB. To quantify the data quality, a metric called SSIM (structural similarity index metric) [31] is used between images with no errors and approximate images. The SSIM values have a range of -1 to 1. The larger SSIM value indicates that the two images are more similar to each other. So, if two images are near-identical, the SSIM will be close to 1.

A. Overall Comparison

In this subsection, we compare the previous schemes with the proposed HL-DNA in terms of the encoding density. The encoding density is the average value of all images. As indicated in Fig. 8, the proposed HL-DNA achieves the best encoding density among all those schemes. HL-DNA increases the average encoding density of the previous studies by about

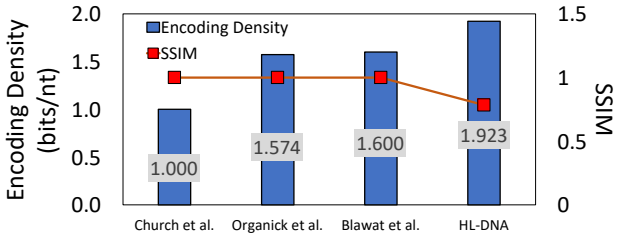


Fig. 8: Overall comparison of average encoding density and average SSIM between different schemes.

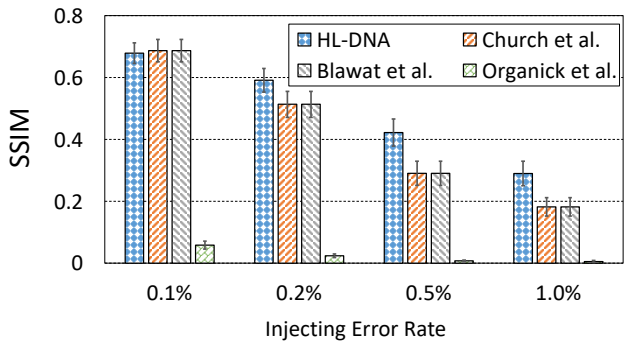
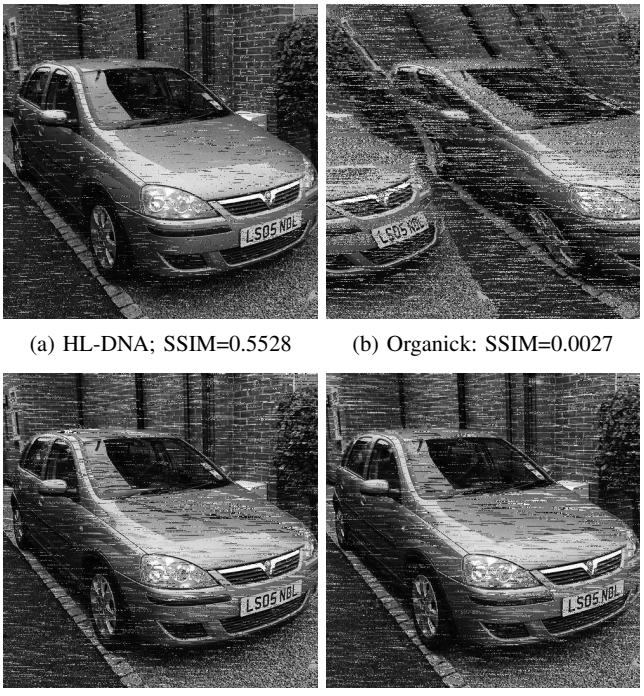


Fig. 9: Robustness comparison by injecting different error rates with 99% confidence interval.



(a) HL-DNA; SSIM=0.5528 (b) Organick: SSIM=0.0027

(c) Church: SSIM=0.4482 (d) Blawat: SSIM=0.4567

Fig. 10: Graphic view of one example with injecting an error of 0.5% for four encoding schemes.

20.2% - 89.4%. In other words, for a DNA storage system, the HL-DNA can improve the overall DNA capacity by about 21% compared to the state-of-the-art design. The reason is that HL-DNA adds the approximation to the encoding scheme to improve the encoding density. Based on the different binary patterns in images, HL-DNA can adaptively select a proper code to achieve a high encoding density while remaining a low error rate. To investigate the accuracy of different schemes, the average SSIM values are shown in Fig. 8. The three baselines have no error induced with SSIM = 1 and the proposed HL-DNA obtains SSIM=0.784 on average. Moreover, the densities of different images can be varied from 1.802 bits/nt to 1.998 bits/nt based on these examples. For the variance investigation, the proposed HL-DNA delivers much small 99% confidence interval ranges, which are (1.919bits/nt, 1.927bits/nt) and (0.775, 0.792) for encoding density and SSIM, respectively. In summary, HL-DNA attains much higher encoding densities while remaining good qualities for the approximate applications.

To graphically see how many errors are introduced in the images, several image examples are indicated in Fig. 7. We find that the HL-DNA scheme can achieve a much close vision to the original images and the major objects in those images can be clearly recognized. Moreover, for different images, HL-DNA may obtain much different SSIM values. For example, the flower in Fig. 7(d) only gets the SSIM value about 0.669, which is much lower than the wolf image (Fig. 7(f)) with SSIM=0.896.

For the overhead, there are two major aspects. One is the computation overhead since we need to scan the bitstream and select a proper encoding scheme among six. The system running the experiments has Intel i-7-47900 CPU@3.6GHz and 8GB memory. The tool used to encode binary data into DNA sequences is MATLAB2020a. HL-DNA has about 10.7ms per DNA strand and previous studies have about 2.86ms per DNA strand. Although HL-DNA has 3.7 times slower than others, compared to the DNA synthesis time (around 10 hours), the computation overhead is negligible. Moreover, those encoding times can be further decreased if changing to a more efficient computing manner such as running in C/C++ in parallel or executing in high-performance systems. The other is the overhead of the encoding field. Since we use about 300 nt as the length of DNA strands and four extra nucleotides for the encoding format indicator and the start point, it has about 4/300 = 1.33%, which can be easily complemented by the enhanced encoding density. With biotechnology developing, the DNA strand length might be longer and the overhead of encoding format indicator and the start point is reduced further.

B. Robustness

In this subsection, we investigate the robustness of HL-DNA storage system and three other baselines by randomly injecting errors from 0.1% to 1% following the distributions of the error model [8]. The errors include insertion, deletion, and substitution errors. As shown in Fig. 9, the proposed HL-DNA achieve much better robustness compared to the

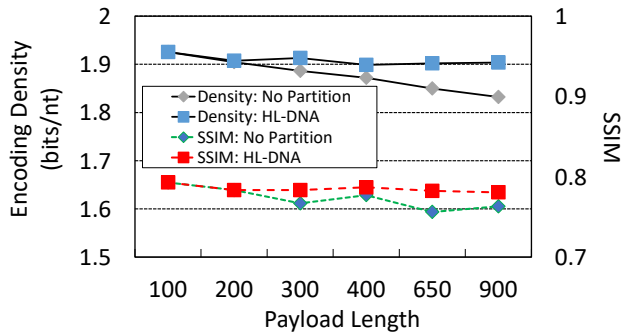


Fig. 11: The influence on encoding density and SSIM as varying payload length for HL-DNA schemes with and without the partition scheme.

other baselines by injecting various error rates. In detail, we can find that for a small error injection, the proposed HL-DNA obtain similar SSIM as the studies from Church et at. [11] and Blawat et al. [16]. The reason is that HL-DNA originally introduces approximation into DNA data and the errors from both approximation and injection in HL-DNA is quite similar to the injected error in the others. As increasing the injected error rate, HL-DNA performs much higher robustness compared to those two schemes. This is because HL-DNA uses the partition scheme and lossy encoding scheme preventing the injected errors. The work [8] obtains the worse robustness among all those schemes since it borrows the rotation scheme and aggregates the error propagation issue. The schemes of Church et at. [11] and Blawat et al. [16] gets similar robustness performance with different error rates although they use different encoding scheme. That is because both of them simply encode either one bit or one bytes for each time and thus the current bit/byte has no relationship with its subsequent bit/byte. As a result, they achieve better robustness performance than the rotation code [8]. However, in reality, those schemes of Church et at. [11] and Blawat et al. [16] may violate some bio-constraints, such as the absence of two consecutive identical nucleotides, and thus cause a higher error rates during the synthesis and sequencing processes. Fig. 10 provides an example of graphical view as injecting a 0.5% error in those four encoding schemes. As analyzed previously, the scheme [8] obtains the worse graphical view. The proposed scheme HL-DNA achieves the best view due to the approximation and the partition scheme.

C. Investigation with Varying DNA Strand Length

With the biology technology development, some of the bio-constraints may become loose such as allowing longer DNA strands for DNA storage. In this subsection, we investigate the influence of longer DNA strand length on the encoding density and the image quality of HL-DNA. Two types of schemes are used for this comparison. One is the proposed HL-DNA, and the other is HL-DNA without the partition scheme. As indicated in Fig. 11, as increasing the payload length, the SSIM values for both schemes remain similar. This is because the approximation error introduced will keep similar and is independent of the payload length. For the encoding density, HL-DNA achieves a flat trend since each partition will have the same length and thus longer payload lengths have little impact on the encoding density. However, the encoding density of the scheme without the partition is decreased. This is because with a long payload length, without the partition scheme, one DNA strand can only use a single encoding scheme and may not be proper to the pattern distributions of the whole DNA strand. In summary, by increasing the DNA strand length with the biotechnology improvement, the proposed HL-DNA scheme can still keep its advantages and maintain high encoding densities.

VI. Conclusion

In this paper, we target on image applications and store them in DNA storage. We propose to apply approximation to DNA storage to improve the overall encoding density and robustness of DNA storage by using a hybrid lossy and lossless encoding scheme (called HL-DNA). Multiple approximate encoding schemes (lossy and lossless codes) are proposed and used to encode incoming binary sequence. The lossless codes are used to limit the errors. These two types of codes are coordinate to balance the encoding density and errors. Moreover, the introduced approximation and newly proposed hybrid encoding schemes in one DNA strand can improve the robustness of DNA storage. Finally, the experimental results indicate that the proposed HL-DNA improves the encoding density of DNA storage and makes it much close to the ideal case (i.e., 2 bits/nt). Also, HL-DNA achieves lower errors than other DNA storage codes.

VII. Acknowledgement

This work was partially supported by NSF I/UCRC Center Research in Intelligent Storage and the following NSF awards 2208317, 2204656, and 2204657. Any opinions, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] J.-D. Lee, S.-H. Hur, and J.-D. Choi, "Effects of floating-gate interference on nand flash memory cell operation," *IEEE Electron Device Letters*, vol. 23, no. 5, pp. 264–266, 2002.

[2] B. Li, C. Deng, J. Yang, D. Lilja, B. Yuan, and D. Du, "Haml-ssd: A hardware accelerated hotness-aware machine learning based ssd management," in *38th IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2019*. Institute of Electrical and Electronics Engineers Inc., 2019, p. 8942140.

[3] A. Amer, J. Holliday, D. D. Long, E. L. Miller, J.-F. Pâris, and T. Schwarz, "Data management and layout for shingled magnetic recording," *IEEE Transactions on Magnetics*, vol. 47, no. 10, pp. 3691–3697, 2011.

[4] F. Wu, B. Li, and D. H. Du, "Fluidsmr: Adaptive management for hybrid smr drives," *ACM Transactions on Storage (TOS)*, vol. 17, no. 4, pp. 1–30, 2021.

[5] M. H. Hajkazemi, A. N. Kulkarni, P. Desnoyers, and T. R. Feldman, "Track-based translation layers for interlaced magnetic recording," in *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, 2019, pp. 821–832.

[6] F. Wu, B. Li, B. Zhang, Z. Cao, J. Diehl, H. Wen, and D. H. Du, "Tracklace: Data management for interlaced magnetic recording," *IEEE Transactions on Computers*, vol. 70, no. 3, pp. 347–358, 2020.
[7] R. Appuswamy, K. Le Brigand, P. Barbry, M. Antonini, O. Madderson, P. Freemont, J. McDonald, and T. Heinis, "Oligoarchive: Using dna in the dbms storage hierarchy." in *CIDR*, 2019.
[8] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen *et al.*, "Random access in large-scale dna data storage," *Nature biotechnology*, vol. 36, no. 3, p. 242, 2018.
[9] Y. Wei, B. Li, and D. H. Du, "Dna storage: A promising large scale archival storage?" *arXiv preprint arXiv:2204.01870*, 2022.
[10] B. Li, L. Ou, and D. Du, "Img-dna: approximate dna storage for images," in *Proceedings of the 14th ACM International Conference on Systems and Storage*, 2021, pp. 1–9.
[11] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.
[12] R. Lopez, Y.-J. Chen, S. D. Ang, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Seelig, K. Strauss, and L. Ceze, "Dna assembly for nanopore data storage readout," *Nature communications*, vol. 10, no. 1, pp. 1–9, 2019.
[13] 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*, 2016, pp. 637–649.
[14] 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*, vol. 494, no. 7435, pp. 77–80, 2013.
[15] Y. Erlich and D. Zielinski, "Dna fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, 2017.
[16] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for dna data storage," *Procedia Computer Science*, vol. 80, pp. 1011–1022, 2016.
[17] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the dna data storage channel," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.
[18] P. Richterich, "Estimation of errors in 'Raw' DNA sequences: A validation study," *Genome Research*, vol. 8, no. 3, pp. 251–259, 1998.
[19] "Evaluation of linear synthetic dna fragments from separate suppliers," https://www.thermofisher.com/content/dam/LifeTech/global/life-sciences/Cloning/gene-synthesis/PDF/GeneArt%20Strings%20compared%20to%20gBlocks.pdf.
[20] S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access dna-based storage system," *Scientific reports*, vol. 5, p. 14138, 2015.
[21] B. Li, N. Y. Song, L. Ou, and D. H. Du, "Can we store the whole world's data in dna storage?" in *12th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage 20)*, 2020.
[22] B. L. Nelms and P. A. Labosky, "A predicted hairpin cluster correlates with barriers to pcr, sequencing and possibly bac recombineering," *Scientific reports*, vol. 1, no. 1, pp. 1–7, 2011.
[23] 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*, vol. 54, no. 8, pp. 2552–2555, 2015.
[24] L. Anavy, I. Vaknin, O. Atar, R. Amit, and Z. Yakhini, "Improved dna based storage capacity and fidelity using composite dna letters," *bioRxiv*, p. 433524, 2018.
[25] Y. Choi, T. Ryu, A. Lee, H. Choi, H. Lee, J. Park, S.-H. Song, S. Kim, H. Kim, W. Park *et al.*, "Addition of degenerate bases to dna-based data storage for increased information capacity," *bioRxiv*, p. 367052, 2018.
[26] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Enzymatic dna synthesis for digital information storage," *bioRxiv*, p. 348987, 2018.
[27] "Difficult template sequencing," https://www.genewiz.com/Public/Services/Sanger-Sequencing/Difficult-Template-Sequencing/.
[28] "Gene synthesis handbook," https://www.genscript.com/gsfiles/gene_synthesis_handbook.pdf.
[29] "How to design a primer," https://www.addgene.org/protocols/primer-design/.
[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.