2020 Eighth International Conference on Advanced Cloud and Big Data (CBD) | 978-1-6654-2313-7/20/$31.00 ©2020 IEEE | DOI: 10.1109/CBD51900.2020.00012

MRC: A High Density Encoding Method for Pratical DNA-based Storage

Qin Liu[1,2], Pengcheng Wang[1,2], Jingsong Cui[1,2], Hao Qi[3]
[1]*Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education*
[2]*School of Cyber Science and Engineering, Wuhan University*
[3]*School of Chemical Engineering and Technology, Tianjin University*
[1, 2]*{qinliu, pc.wang, jscui}@whu.edu.cn,* [3]*haoq@tju.edu.cn*

***Abstract*—DNA data storage has become one of the most high-profile techniques for long-term data storage. In the process of synthesizing, storing, amplifying, and sequencing of DNA sequences, there are constraints on the form of the DNA sequences, namely repeated bases avoidance, proper GC content, and special DNA fragments avoidance. Previous studies met the constraints either at the cost of information density or too complicated for implementation. In this paper, we propose an encoding method called MRC (Mixed Radix Coding) that can effectively satisfy those constraints and achieve the almost theoretically optimal information density as well. MRC is very easy to implement in practice and extend to accomodate other DNA coding constraints.**

***Keywords*-DNA-based storage, Long term data storage, Next-generation information storage, Binary-to-DNA coding**

I. Introduction

To meet evolving compliance and regulatory requirements, more and more enterprises are retaining large data sets for a longer duration that is expensive and complex to manage. Current storage technologies relied on optical and semiconductor media are difficult to keep pace with exponential growth of data for long-term retention. DNA, as a promising storage medium, has been paid much attention recently due to its attractive feature of enormous storage with great capacity and durability. DNA-based storage can achieve very high information storage density (e.g., 455 EB data can be encoded in 1 gram of single stranded DNA [5]), which makes it much more compelling than traditional storage media. DNA is also one of the most stable biomolecules so that it is suitable for long-term storage.

DNA-based storage writing involves encoding binary data into the DNA sequences, synthesizing desired DNA sequences and storing the synthetic biological material. Reading the stored information requires sequencing of the DNA, which may contain some errors, and decoding DNA sequences to obtain the original digital information. Before sending samples to sequencing platform, amplification using Polymerase Chain Reaction (PCR) may be needed to scale stored DNA sequences library up. The whole DNA-based storage process is shown in Fig. 1.

In 2012, Church et al. [4] stored 0.65 MB data into 8.8 Mb DNA oligos (oligonucleotides) of 159nt (nucleotides) in synthetic DNA with a physical density of 1.28 PB/gram,

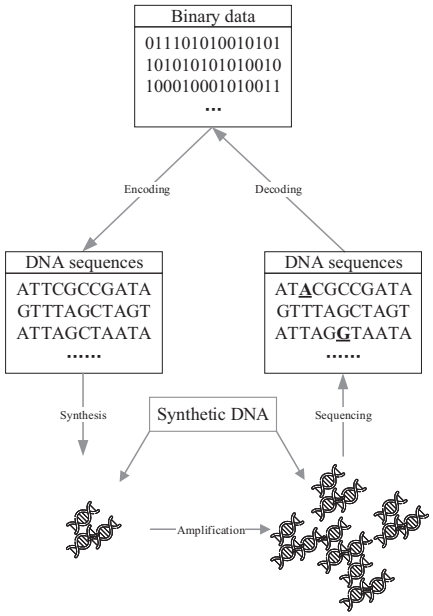


Figure 1. Process of DNA-based storage

which was considered to be a milestone study in DNA-based storage. However, only partial data was extracted from the stored DNA sequences due to lack of error tolerance method. In 2013, Goldman et al. [7] encoded and fully recovered 0.75MB of data from synthesized DNA using an encoding method that consists of data compression and error correction. Many improvements [8, 2, 6] have been done since then that promote the reality of storing extremely large amounts of digital data in DNA.

There are two major issues that should be dealt with in encoding of DNA-based storage. One is that error tolerance mechanisms need to be introduced to handle errors. Several mechanisms [9] such as RS (Reed-Solomon) codes [8], fountain codes [6] have been adopted. The other issue is to design binary-to-DNA encoding method that converts binary data into DNA sequences while satisfying the biochemical constraints. It has been found that oligos with abnormal GC content or long homopolymer runs are prone to errors [10]. Moreover, some special DNA fragments should also be avoided, e.g., primer binding sites. Besides, several recent works [1, 3] introduced additional degenerate bases or

978-1-6654-2313-7/20/$31.00 ©2020 IEEE
DOI 10.1109/CBD51900.2020.00012

Table I
COMPARISON OF PUBLISHED BINARY-TO-DNA ENCODING METHODS

Methods	Satisfaction of constraints			compatiblity with degenerate/composite bases	information density (bits/nt)
	no repeated bases	no avoided fragment	proper GC content		
Church et al. [4]	Y	Y	Y	N	1
Goldman et al. [7]	Y	N	N	N	$\log_2 3 \approx 1.58$
Grass et al. [8]	Y	N	N	N	$\log_2 47/3 \approx 1.85$
Blawat et al. [2]	Y	N	N	N	1.6
C-RLL (Wang et al. [12])	Y	N	Y	N	1.917
Hybrid coding (Wang et al. [11])	Y	N	Y	N	1.976
MRC (ours)	Y	Y	Y	Y	1.98 (natural bases) and 3.11 (degenerate bases)

composite bases to the basic natural bases. Thus, supporting of degenerate/composite bases should also be considered in binary-to-DNA encoding method.

Church et al. [4] proposed a simple encoding method that encodes one bit per base (A or C for 0, G or T for 1). Then, Goldman et al. [7] proposed an encoding method in which each byte of the resulting data was substituted by 5 or 6 bases. These encoding method can't prevent abnormal GC content and the encoding information density is far from the theoretical optimal value.

An encoding method proposed by Grass et al. [8] in 2015 achieved great success in the process of approaching the theoretical encoding information density. This method introduced a finite field of DNA nucleotide triplets as its elements. However, Grass's method can't prevent abnormal GC content either.

Blawat et al. [2] proposed an encoding method using two reference coding tables specified in advance. A 1-byte (8 bits) fundamental information block is assigned to a 5nt DNA sequence, the third and fourth nucleotide are swapped and other criteria are also applied to prevent repeated bases. Same as the previous methods, this encoding method can't prevent abnormal GC content.

In 2019, a novel content-balanced run-length limited code was proposed by Wang et al. [12]. The proposed encoding method has high effective code rate of 1.917 bits per nucleotide and low coding complexity. And then a hybrid coding method consisting of interleaved mapping and VLC (variable length constrained) mapping is developed by Wang et al. [11], exhibiting close to the theoretical information density. However, this encoding method is very complicated and can't produce fixed-length sequences, nor is it suitable for the degenerate base schemes.

As Table I shows, although these methods may have some advantages in terms of encoding information density or avoidance of repeated bases, none of them can achieve perfect encoding information density under constraints or satisfy all the constraints. Moreover, none of these methods can be used for degenerate/composite bases schemes.

In this paper, we propose a binary-to-DNA encoding method called MRC (**M**ixed **R**adix **C**oding). In MRC, the binary data to DNA conversion is no longer with fixed radix, but mixed radix. The radix of each position is determined by the context information to satisfy constraints. Using this strategy, we achieve a high average encoding information density while GC content is kept balanced, no homopolymer run is larger than specific length, and no special DNA fragments occur. To the best of our knowledge, MRC is the first one with high encoding information density that can prevent long homopolymer runs, keep GC content within a limited range and avoid DNA fragments similar to primer binding sites. More importantly, MRC can be used for degenerate/composite bases schemes.

This paper is organized as follows. The requirements of DNA-based storage encoding are described in Section II. The proposed encoding and decoding methods are presented in Section III. Simulation results are given in Section IV, and the paper is concluded in Section V.

II. DNA-BASED STORAGE ENCODING REQUIREMENTS

There are several important requirements to measure the quality of DNA-based storage to make it more stable and capable of enormous long-term storage.

A. Information density

One of the most important requirements for the capacity of DNA-based storage is information density.

Definition 1: Suppose that we can encode n bits data into b bases, then the **encoding information density** $d_e = \frac{n}{b}$ is the number of bits encoded per base.

It's easy to know that the optimal d_e is 2 bits/nt for natural bases. However, if we introduce additional degenerate bases [1, 3] to the basic four bases, the d_e will increase over the upper boundary (2 bits/nt) of the natural bases.

Higher information density can effectively reduce the cost and time of DNA synthesis and sequencing. Therefore, we should explore the approach of designing efficient encoding method with minimal loss of the information density.

B. Avoidance of repeated bases

It has been shown that DNA sequences with long homopolymer runs (DNA fragments with consecutive nucleotides or so-called repeated bases) are prone to errors during synthesis, amplification and sequencing [10]. Suppose that DNA sequences are sequences consisted of q-ary symbols, $\ldots, b_{i-1}, b_i, b_{i+1}, \ldots b_i \in B = \{0, \ldots, q-1\}$, we should keep homopolymer run lengths for certain base b

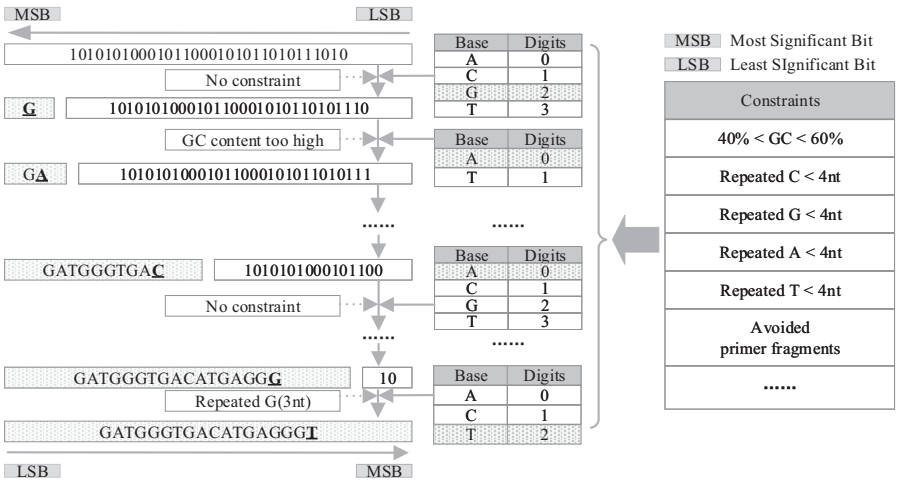


Figure 2. Mixed Radix Coding

less than L_b. That is to say, for all subsequence s composed only by b, lengths of s should be less than L_b, where L_b can be various for different base. In this paper, we assume the typical value of L_b as 4, but it also depends on the development of biotechnologies.

C. Avoidance of abnormal GC content

Same as repeated bases, abnormal (too high/low) GC content (the ratio of total number of G and C against the total number of nucleotides in a DNA sequence) can result in more errors.

Typically, the GC content of each DNA sequence should be close to 50% [6]. That is, for DNA sequence with n nucleotides, the total number of G and C is m, GC content $r = \frac{m}{n}$ should be close to 0.5.

D. Avoidance of special fragment

Beside inherent constraints of DNA-based storage, there are some special fragments that should be avoided. For example, a primer binding site is a region of a nucleotide sequence where an RNA or DNA single-stranded primer binds to start replication. Therefore, DNA fragments are similar to primer binding sites should also be avoided otherwise it may have a negative effect on amplification.

E. Support of degenerate/composite bases

Degenerate base allows to use more than one base possibility at a particular position. An oligo sequence can be synthesized with multiple bases at the same position, which is termed as degenerate base. IUB (International Union of Biochemistry) has established single letter codes for all possible degenerate possibilities. An example is "R" that is $[A, G]$ at the same position with 50% of the oligo sequence will have an A at that position, and the other 50% have G.

A composite base is a representation of a position in a sequence that constitutes a mixture of all four standard DNA nucleotides in a specified predetermined ratio [1]. Same as degenerate bases, composite bases can be used to extend the available bases and therefore allow higher capacity per synthesis cycle.

The latest researches [1, 3] show that degenerate/composite bases can also be used in DNA-based storage. It's very impressive that the times of synthesis can be reduced, thereby reducing the cost of storage, in spite of that DNA storage with degenerate/composite bases necessitate higher sequencing depths and improved synthesis techniques for practical usage compared with DNA storage with natural bases (i.e., A/T/C/G). Since related researches and technologies are in progress rapidly, it is also necessary to consider supporting degenerate/composite bases in DNA encoding.

III. MRC Encoding/Decoding

In this section, we will present an efficient encoding method with high encoding information density called MRC (Mixed Radix Coding). MRC can satisfy constraints we've discussed and apply to degenerate/composite bases as well.

A. Method Description

Since each base may appear at every position in DNA sequences, a DNA sequence can be regarded as a number with positional radix n where each base represents a digit. For binary data, it can be easily translated into DNA sequences composed of natural bases by mapping every two bits to a base. Note that this binary-to-quaternary mapping exhibits the optimal encoding information density (2 bits/nt), but it does not always meet the above mentioned constraints.

For a binary data D of n (assume that n is even) bits, it can be divided into two parts in a positional numeral system: i) radixes $r = [2, \ldots, 2]$ and ii) digits $d = [d_{n-1}, \ldots, d_0]$, where d_0 is the least significant digit and d_{n-1} is the most significant digit. If every two bits represent a quaternary digit with $r = [4, 4, \ldots, 4]$ and $d = [d'_{\frac{n}{2}-1}, d'_{\frac{n}{2}-2}, \ldots, d'_0]$, it can be expressed as follows:

$$D = \underbrace{d_{n-1} \times 2^{n-1} + d_{n-2} \times 2^{n-2} + \cdots + d_0 \times 2^0}_{n} \quad (1)$$

$$= \underbrace{d'_{\frac{n}{2}-1} \times 4^{\frac{n}{2}-1} + d'_{\frac{n}{2}-2} \times 4^{\frac{n}{2}-2} + \cdots + d'_0 \times 4^0}_{\frac{n}{2}} \quad (2)$$

$$= \underbrace{((\ldots(d'_{\frac{n}{2}-1} + 4 \times 0)\ldots) \times 4 + d'_1) \times 4 + d'_0 \times 4^0}_{\frac{n}{2}} \quad (3)$$

In MRC, we generate mapping tables (from digits to bases) at every position dynamically. However, these tables may not contain all the bases, so that the radix of each position is no longer a fixed number but determined by context, that is so-called **Mixed Radix**. In a mixed radix system, the weights form a sequence where each weight is an integral multiple of the previous one starting from the least significant position. Under this condition, $r = [r_{m-1}, r_{m-2}, \ldots, r_0]$ and $d = [d'_{m-1}, d'_{m-2}, \ldots, d'_0]$:

$$D = D'_0 \times r_0 + d'_0 \quad (4)$$

$$= (D'_1 \times r_1 + d'_1) \times r_0 + d'_0 \quad (5)$$

$$= \underbrace{((\ldots(d'_{m-1} + r_{m-1} \times 0)\ldots) \times r_1 + d'_1) \times r_0 + d'_0}_{m} \quad (6)$$

$$= \underbrace{d'_{m-1} \times \prod_{j=0}^{m-2} r_j + \cdots + d'_1 \times r_0 + d'_0 \times 1}_{m} \quad (7)$$

$$= \sum_{i=0}^{m-1} (d'_i \prod_{j=0}^{i-1} r_j), \quad (8)$$

where d'_i is *digit* under radix r_i, and $\prod_{j=0}^{i-1} r_j$ is weight at position i, $0 \le i < m$.

As Fig. 2 shows, the mapping table of each position is determined by previous bases and current GC content. When homoplymer runs is about to exceed expected length, unwanted DNA fragment is about to occur or GC content is abnormal, some bases will be removed from the next candidate bases so that no unqualified fragment will appear and the GC content can be adjusted. So if there are no constraints, MRC just equals to binary-to-quaternary conversion like above under natural bases.

Notations that will be used are shown as Table II.

B. Encoding

A binary-to-DNA encoding is a method to convert n bits binary data $D = d_{n-1} \ldots d_0, d_i \in \{0, 1\}$ to a sequence $S = b_0 \ldots b_{L-1}, b_i \in B$ of length L where B is a set of bases. B can either be natural bases or degenerate/composite bases. If each base $b_i \in B$ represents a digit, it is just like converting numbers from binary to base $|B|$ numeral system.

Table II
NATATIONS

Symbol	Description
B	Set of basic candidate bases for every position.
D	Binary data to be encoded, considered as a big decimal integer of n bits.
S	Sequence to be decoded, considered as a sequence of length L.
B_i	Set of bases that can be used at position i.
M_i	Mapping from digits to bases used at position i. $M_i(d)$ indicates the base for digit d to base at position i.
R_i	Radix at position i. It is the size of mapping M_i.
b_i	Base at position i for S.
S_i	Encoded sequence at position i. Note that S_0 is empty and the length of S_i is i.
$S_{i,j}$	Subsequence of S from i to j (excluded). if $i = j$, $S_{i,j}$ is empty.
d_i	Digit at position i under radix R_i.
D_i	Binary data decoded at position i.
W_i	Weight at position i.
$a\|\|b$	Concatenation of a and b.
$\lfloor x \rfloor$	Floor function.
$index_{M_i}(b)$	Index of base b in M_i.
$GEN(B, seq, constraints)$	A function to generate mapping M_i from basic candidate bases B, previous sequence seq and constraints.

To satisfy constraints like limited homoplymer runs and GC content, for each position $0 \le i < L$, not every base in basic candidate bases B can be used. We denote these bases that can be used at position i as B_i. A mapping from digits to bases at position i is $M_i = \{0 \to b_0, 1 \to b_1, \ldots, k-1 \to b_{k-1}\}, k = |B_i|, b_k \in B_i$, the order of bases should be deterministic, here we simply adopt the alphabetical order. For example, if $B_i = \{A, C, G\}$, then M_i would be $\{0 \to A, 1 \to C, 2 \to G\}$. If there is just one element in B_i, which means radix at this position is 1, it will not store any information, but may stop repeated bases as a placeholder, or adjust GC content.

The encoding algorithm includes two steps: i) generate the mapping M_i at position i, and ii) represent the value of binary data D at position i in Big-Endian using M_i. In every loop of encoding, D is updated by the result divided by $|M_i|$. When D equals zero, the algorithm stops and outputs the encoded sequence S_i. Algorithm 1 is the pseudo code for MRC encoding.

Algorithm 1 Encoding Algorithm

Input D, B, $constraints$
Output S_i

$i \leftarrow 0$
$S_i \leftarrow empty$
repeat
  $M_i \leftarrow GEN(B, S_i, constraints)$
  $R_i \leftarrow |M_i|$
  $d_i \leftarrow D \mod R_i$
  $S_{i+1} \leftarrow S_i || M_i(d_i)$
  $D \leftarrow \lfloor D / R_i \rfloor$
  $i \leftarrow i + 1$
until $D = 0$

The generation of M_i varies depending on the bases we use. Next, we will discuss the implementation of the GEN function of different bases.

1) Natural Bases: Naturally occurring DNA consists of four types of bases: A, C, G, T. GEN function is described as follows:

Repeated bases: Suppose that the maximum number of occurrences of the repeated base $b \in B$ we allowed to be n. For previous sequence $S = b_0 \dots b_{i-1}, i \geq n$, if $\forall b_k \in S_{i-n,i}, i-n \leq k < i$ and $b_k = b$, then base b cannot appear again thus $B_i = B \setminus \{b\}$. Example is shown as Fig. 3(a).

Special fragment: Suppose that the avoided fragment of length l is $f = b_0 \dots b_{l-1}$. For previous sequence $S = b_0 \dots b_{i-1}, i \geq l-1$, if $S_{i-l+1,i} = f_{0,l-1}$, then base b_{l-1} cannot appear thus $B_i = B \setminus \{b_{l-1}\}$. Example is shown as Fig. 3(b).

GC content: Assume that the GC content should be limited in $[C_{min}, C_{max}]$, C_i is GC content for current position. If $C_i < C_{min}$, then we should increase the GC content thus $B_i = \{C, G\}$. If $C_i > C_{max}$, then we should reduce the GC content thus $B_i = \{A, T\}$. Example is shown as Fig. 3(c).

2) Degenerate/Composite Bases: Since the sequence composed of degenerate/composite bases is still composed by natural bases after synthesis, we should make these synthesized sequences satisfy constraints above. We denote that a degenerate/composite base of m candidate natural bases as $db = [b_0, \dots, b_{m-1}]$. For degenerate/composite bases, the implementation of GEN function is a little different:

Repeated bases: Suppose that the maximum number of occurrences of the repeated natural base b we allowed to be n. For previous sequence $S = db_0 \dots db_{i-1}, i \geq n$ and its subsequence $S_{i-l+1,i} = db_{i-l+1} \dots db_{i-1}$, $CP = db_{i-l+1} \times \dots \times db_{i-1}$ is the Cartesian product. If $\exists cp \in CP, \forall b_k \in cp$ and $b_k = b$, then degenerate/composite bases contained natural base b cannot appear again thus $B_i = B \setminus \{dp | b \in dp\}$.

Special fragment: Suppose the avoided fragment of length l is $f = b_0 \dots b_{l-1}$. For previous sequence $S_i = db_0 \dots db_{i-1}, i \geq n$ and its subsequence $S_{i-l+1,i} = db_{i-l+1} \dots db_{i-1}$, $CP = db_{i-l+1} \times \dots \times db_{i-1}$ is the Cartesian product. If $\exists cp \in CP, cp = f_{0,l-1}$, then degenerate/composite bases contained natural base b_{l-1} cannot appear thus $B_i = B \setminus \{dp | b_{l-1} \in dp\}$.

GC content: The entire sequence needs to be scanned for calculating GC content and it is impractical to exhaust all possible synthesized sequences since it grows exponentially. Here we use a probabilistic GC content for reducing calculations. We denote that the GC contribution rate of degenerate/composite bases is taken as probability of G or C after synthesis. Assume that the GC content should be limited in $[C_{min}, C_{max}]$, because the GC content of a sequence is now a mathematical expectation, which conforms to the normal distribution. Assuming the offset from expectation is ρ, then the actual GC content range should be $[\frac{C_{min}+C_{max}}{2} \times (1-\rho), \frac{C_{min}+C_{max}}{2} \times (1+\rho)]$. Let C_i be the probabilistic GC content. If $C_i < \frac{C_{min}+C_{max}}{2} \times (1-\rho)$, we have $B_i = \{dp | C \in dp \text{ or } G \in dp\}$. If $C_i > \frac{C_{min}+C_{max}}{2} \times (1+\rho)$, we have $B_i = B \setminus \{dp | C \in dp \text{ or } G \in dp\}$.

We should satisfy all the constraints above. In other words, suppose that there are m constraints $\{C_0, \dots, C_{m-1}\}$ to satisfy, we will get a $B_i(C_j), 0 \leq j < m$ under constraint C_j for each constraint, the final bases set is:

$$B_i = B_i(C_0) \bigcap \cdots \bigcap B_i(C_{m-1}) \tag{9}$$

C. Decoding

Decoding is the reverse operation to convert a sequence $S = b_0 \dots b_{L-1}, b_i \in B$ of length L, where B is a set of bases, to a n bits binary data $D = d_{n-1} \dots d_0, d_i \in \{0, 1\}$. The key point of decoding is that we need to reconstruct the same mapping M_i and radix R_i used in encoding at position $0 \leq i < L$. From Equation (8), we know that digits at different positions have different weights, we denote that $W_i = R_0 \times \cdots \times R_{i-1}$ as weight at position i.

First, we generate the same mapping M_i used in encoding at position i by decoded subsequence $S_{0,i}$. Then we reconstruct data D_i from W_i and the index of b_i in M_i. In every loop of decoding, W_i is updated by previous W_{i-1} and current radix R_i. When we have decoded all the bases in S, the algorithm stops and outputs D_i as final decoding result. Algorithm 2 is the pseudo code for MRC decoding.

Algorithm 2 Decoding Algorithm

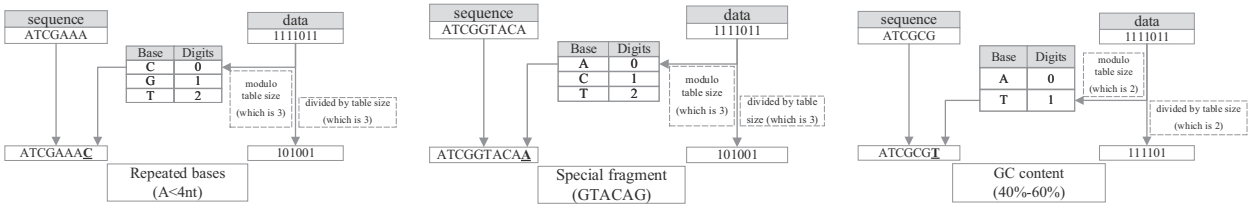
Input S, B, $constraints$
Output D_i

$i \leftarrow 0$
$D_i \leftarrow 0$
$W_i \leftarrow 1$
while $i \neq L$ **do**
  $M_i \leftarrow GEN(B, S_{0,i}, constraints)$
  $R_i \leftarrow |M_i|$
  $idx \leftarrow index_{M_i}(b_i)$
  $D_{i+1} \leftarrow D_i + W_i \times idx$
  $W_{i+1} \leftarrow W_i \times R_i$
  $i \leftarrow i + 1$
end while

It is easy to know the complexity of encoding/decoding algorithm is between $O(\log_2 D)$ and $O(\log_{|B|} D)$.

D. Special handling

1) Exception: It may happen that B_i is an empty set. If so, we will take the majority of $B_i(C_j)$ as the final bases B_i to make the size of B_i greater than 1. Let $t_0 \geq t_1 \geq \cdots \geq t_{i-1}, 0 \leq i < |B|$ be occurrence numbers of candidate bases in candidate bases sets $\{B_i(C_0), \dots, B_i(C_{m-1})\}$, We will take j bases as B_i, where $0 < j \leq |B|$ and $t_{j-1} > t_j$. In this way, a small part of synthesized sequences may be against constraints. But it will have little impact on decoding, since these synthesized sequences are inherently redundant.



(a) Repeated bases (b) Special fragment (c) GC content

Figure 3. *GEN* function: (a) Repeated A should be less than 4 and there are already three A in previous sequence, so A should be removed from the mapping table. (b) Fragment GTACAG cannot appear. Since GTACA is the last fragment in previous sequence, the last base of GTACAG (which is G) should be removed from the mapping table. (c) GC content is too high (66.7%), so G and C are removed from the mapping table to reduce GC content.

2) Padding: The length of encoded sequences produced by MRC is not fixed, and we can't predict its exact length. In general, for convenience, we may pad the entire DNA sequences to the same length L via a padding scheme. Padding scheme of MRC is to add a boundary to the data. To achieve a higher information density, we choose 2^N as the boundary.

Encoding with padding: Assuming the maximum bits of data to be encoded is N, we add 2^N to data D before encoding as a boundary(in other words, set the N bit of D to 1), than fill padding data in the end.

Decoding with padding: When decoding, if $D_i \geq 2^N$, which means we have encountered the boundary, output $D_i - 2^N$ as result of decoding.

The padding data can be some random data, but it can also be used to store some additional information to improve utilization, such as file name, file attributes, etc.

IV. Performance Evalution

A. Simulation Settings

In order to evaluate the performance and encoding information density of MRC, we implement MRC and test with some data sets. To make a fair comparison with other methods [4, 7, 8, 2, 11, 12], we set the same or similar constraints as follows:

- Repeated C/G/A/T cannot exceed 3nt.
- GC content should be limited in 40% – 60%.
- 10 avoided fragments.

Two 20nt primers are concatenated at the front and the end of encoded DNA sequence.

B. Natural Bases

We encoded 10,000 random binary data of 300 bits into DNA sequences composed of natural bases, the distribution of encoding information density d_e for these 10,000 sequences is as Fig. 4(a). Compared with other encoding method, the average d_e of MRC is about 1.98 bits/nt, which is extremely close to the theoretical value $\log_2 4 = 2$ bits/nt.

According to common sense, it can be speculated that the more the data bits, the closer the average encoding information density is to the theoretical value. We encoded random binary data with length from 1 bits to 1,000 bits, and the experimental results are shown as Fig. 4(b). starting from about 135 bits, the average encoding information density of MRC reached 1.98 bits/nt, which is much closer to the upper boundary than other methods.

C. Degenerate/Composite Bases

Since we need to consider all the sequences after synthesis, we use a probabilistic way to calculate the GC content (III-B2). In the experiment, we set ρ to 0.2.

Same as natural bases, we encoded 10,000 random binary data of 300 bits into DNA sequences composed of degenerate bases, the encoding information density distribution is as Fig. 4(d). The average encoding information density of MRC is about 3.11 bits/nt, which is extremely close to the theoretical value $\log_2 15 \approx 3.9$ bits/nt too. And starting from about 16 bits, the average information density reached 3.0 bits/nt (shown as Fig. 4(e)).

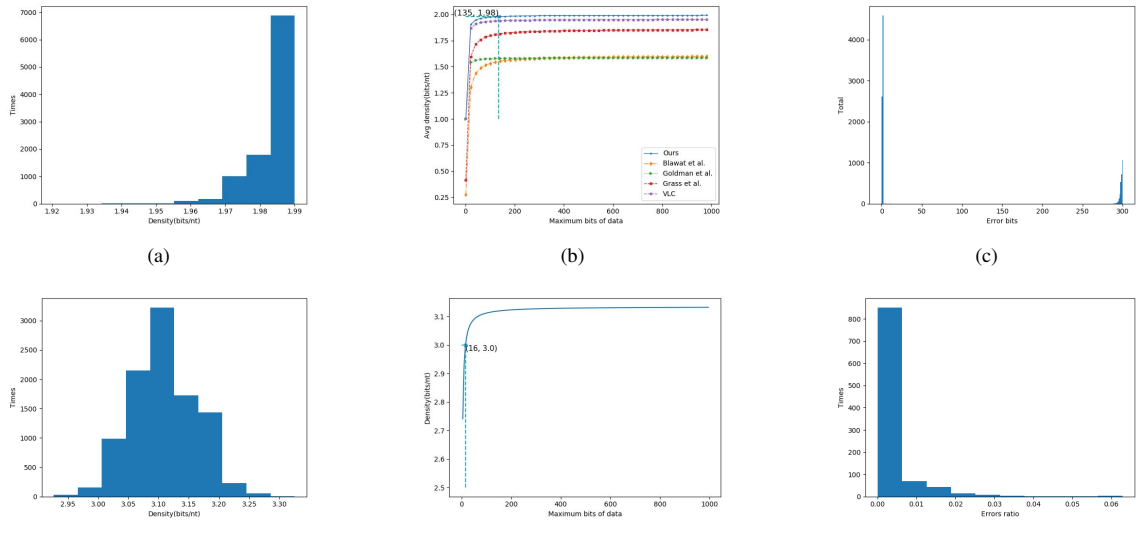
For 1,000 encoded sequences composed of degenerate bases, we simulated DNA synthesis for 10,000 times, and then verified these synthesized DNA sequences to see if they satisfied all the constraints or not. We found that very small part of the synthesized sequences were against constraints (as Fig. 4(f)), which means we have almost satisfied all the constraints in case of degenerate bases.

D. Error propagation

MRC does not stop error propagation, so additional error correction codes are needed to ensure the correctness of decoding. We encoded 10,000 random binary data of 300 bits into DNA sequences composed of natural bases, and introduced one modification error at first base. Then we decoded these incorrect sequences and compared decoded data with correct data to see how many bits were influenced by one error. From the experimental result (as Fig. 4(c)), we can see that one error generally affects the next few bits, but not all of them. About 90% of errors will only affect 1–4 bits, while for other 10%, almost all bits was influenced. In general, MRC also has some tolerance of error propagation.

V. Conclusion

In this paper, we proposed a binary-to-DNA encoding method MRC that can effectively prevent long homopolymer runs, keep GC content within a limited range, avoid some DNA fragments and satisfy other constraints in DNA-based storage, while achieving a high encoding information density of 1.98 bits/nt as well. MRC is very easy to implement



(a) (b) (c)

(d) (e) (f)

Figure 4. Evalution of MRC: (a) The encoding information density distribution of natural bases for 10,000 data. (b) The average encoding information density distribution of different bits for natural bases (from 1 bits to 1,000 bits). (c) Error propagation distribution of one error. (d) The encoding information density distribution of degenerate bases for 10,000 data. (e) The average encoding information density distribution of different bits for degenerate bases (from 1 bits to 1,000 bits). (f) Distribution of errors ratio for synthesized sequences.

in practice and make extensions. Futhermore, MRC can produce not only variable-length oligos, but also fixed-length oligos via padding. Compared with other methods, MRC not only has high encoding information density, but also can be applied to degenerate/composite bases schemes. MRC is demostrated as a powerful DNA-storage encoding method suitable for a variety of constraints.

References

[1] Leon Anavy, Inbal Vaknin, Orna Atar, Roee Amit, and Zohar Yakhini. “Data storage in DNA with fewer synthesis cycles using composite DNA letters”. In: *Nat Biotechnol* 37.10 (Oct. 2019), pp. 1229–1236.

[2] Meinolf Blawat, Klaus Gaedke, Ingo Hütter, Xiao-Ming Chen, Brian Turczyk, Samuel Inverso, Benjamin W. Pruitt, and George M. Church. “Forward Error Correction for DNA Data Storage”. In: *Procedia Computer Science* 80 (2016), pp. 1011–1022.

[3] Yeongjae Choi, Taehoon Ryu, Amos C. Lee, Hansol Choi, Hansaem Lee, Jaejun Park, Suk-Heung Song, Seojoo Kim, Hyeli Kim, Wook Park, and Sunghoon Kwon. “High information capacity DNA-based data storage with augmented encoding characters using degenerate bases”. In: *Sci Rep* 9.1 (Dec. 2019), p. 6582.

[4] G. M. Church, Y. Gao, and S. Kosuri. “Next-Generation Digital Information Storage in DNA”. In: *Science* 337.6102 (Sept. 28, 2012), pp. 1628–1628.

[5] Pavani Yashodha De Silva and Gamage Upeksha Ganegoda. “New trends of digital data storage in DNA”. In: *BioMed research international* 2016 (2016).

[6] Yaniv Erlich and Dina Zielinski. “DNA Fountain enables a robust and efficient storage architecture”. In: *Science* 355.6328 (Mar. 3, 2017), pp. 950–954.

[7] Nick Goldman, Paul Bertone, Siyuan Chen, Christophe Dessimoz, Emily M. LeProust, Botond Sipos, and Ewan Birney. “Towards practical, high-capacity, low-maintenance information storage in synthesized DNA”. In: *Nature* 494.7435 (Feb. 2013), pp. 77–80.

[8] Robert N. Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J. Stark. “Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes”. In: *Angew. Chem. Int. Ed.* 54.8 (Feb. 16, 2015), pp. 2552–2555.

[9] Zhi Ping, Dongzhao Ma, Xiaoluo Huang, Shihong Chen, Longying Liu, Fei Guo, Sha Joe Zhu, and Yue Shen. “Carbon-based archiving: current progress and future prospects of DNA-based data storage”. In: *GigaScience* 8.6 (June 1, 2019), giz075.

[10] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. “Characterizing and measuring bias in sequence data”. In: *Genome biology* 14.5 (2013), R51.

[11] Yixin Wang, Md Noor-A-Rahim, Jingyun Zhang, Erry Gunawan, Yong Liang Guan, and Chueh Loo Poh. “High capacity DNA data storage with variable-length Oligonucleotides using repeat accumulate code and hybrid mapping”. In: *J Biol Eng* 13.1 (Dec. 2019), p. 89.

[12] Yixin Wang, Md. Noor-A-Rahim, Erry Gunawan, Yong Liang Guan, and Chueh Loo Poh. “Construction of Bio-Constrained Code for DNA Data Storage”. In: *IEEE Commun. Lett.* 23.6 (June 2019), pp. 963–966.