IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 December 2023, accepted 8 January 2024, date of publication 12 January 2024, date of current version 22 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3353305

RESEARCH ARTICLE

Adaptable DNA Storage Coding: An Efficient Framework for Homopolymer Constraint Transitions

YUNFEI GAO AND ALBERT NO, (Member, IEEE)
Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, South Korea

Corresponding author: Albert No (albertno@hongik.ac.kr)

This work was supported by the Research Program through the National Research Foundation of Korea under Grant NRF-2022M3C1A3081366.

ABSTRACT Many DNA storage codes take into account homopolymer and GC-content constraints. Still, these codes often need to meet additional practical database requirements, such as error correction and data queries, necessitating considerable financial and time investment in their training or design. As DNA storage technologies, including sequencing and synthesis, continue to evolve rapidly, these codes may need to be retrained or redesigned to adapt to new constraints. In this study, we aim to design a method for adapting an existing DNA storage code to satisfy a new constraint, specifically concerning homopolymer variations. We present a simple and effective framework known as Transfer Coding, which directly maps DNA sequences from an original homopolymer constraint h_1 to a new constraint h_2. This approach essentially combines the existing coding scheme with a Transfer encoder. The proposed method uses strategic base replacements to ensure compliance with constraints, achieving results close to the theoretical limit while keeping alterations to the original sequence minimal.

INDEX TERMS DNA storage, DNA-to-DNA coding, edit distance, GC contents, homopolymer constraint.

I. INTRODUCTION

DNA storage, a pioneering technique for preserving digital information, capitalizes on the inherent properties of DNA molecules [1], [2]. DNA, a sophisticated biological molecule carrying genetic data within living entities, offers the capacity to accommodate colossal volumes of data within minuscule spaces [3]. DNA storage transmutes digital data into the four chemical bases constituting DNA: adenine (A), cytosine (C), guanine (G), and thymine (T), thereby encoding the data as a sequence. This sequence can then be synthesized and preserved within a tube or a similar storage medium. With DNA serving as the storage medium, data longevity potentially extends to thousands of years [4], far surpassing the lifespan of conventional storage devices like hard drives, while simultaneously cutting costs.

DNA storage faces two biological limitations that can negatively impact data storage. The first of these limitations is known as the homopolymer constraint, which restricts the number of consecutive instances of the same nucleotide base in a DNA sequence, namely the encoded sequence can contain at most h consecutive identical bases. The second limit is the GC content constraint, which restricts the proportion of G and C bases in all bases in the encoded sequence. Specifically, the proportion must fall within the range $[0.5 - c_{GC}, 0.5 + c_{GC}]$, for some $0 < c_{GC} < 0.5$. Deviations from these constraints can incite complications in DNA sequencing and structure, thereby possibly resulting in incomplete data recovery [5]. Addressing these constraints, various DNA storage codes have been devised by researchers. The homopolymer constraints can be managed via the utilization of 3-ary Huffman codes [6]. In an alternative approach, short sequences are engineered and subsequently concatenated to form longer sequences to satisfy these constraints [7], [8], [9]. Moreover, the constraints can also be met by adopting the minimum variance Huffman tree encoding [10] or by implementing a greedy algorithm [11]. It is worth noting that all of the aforementioned methods are

The associate editor coordinating the review of this manuscript and approving it for publication was Vincenzo Conti.

focused on the encoding of binary data into DNA sequences, and they do not extend to encoding the resultant DNA data outputs generated by other methodologies.

In addition to the above constraints, for DNA storage to operate effectively as a database, it demands several practical elements. A key element is the query feature, which enables the extraction of specific data of interest. An example of this is the image-to-sequence coding method proposed by Bee et al. [12]. This technique involves transmuting images into a DNA base sequence and employing hybridization probes to align query targets with corresponding images. The authors have trained a neural network to translate the image into a sequence in such a way that similar images are likely to hybridize. Subsequently, users can retrieve images corresponding to the query image by identifying the hybridized sequence. Furthermore, for data storage to be reliable, an error-correction code is required to minimize the effect of errors. Numerous error-correcting code methods have been developed to tackle the problems of insertion, substitution, and deletion that may transpire during the sequencing process [3], [13], [14], [15], [16].

Hence, a proficient DNA storage encoding scheme ought to concurrently accommodate the classical constraints (homopolymer and GC contents) and the pragmatic necessities (error correction functionality and query handling). However, in light of rapid advancements in sequencing and synthesis technology, these classical constraints may undergo changes over time. For example, the homopolymer constraint could exhibit variability, where it can be increased to $h+1$ or decreased to $h-1$ based on the technology development. As a result, the DNA storage encoding scheme necessitates occasional updates. The implementation of these practical requirements, such as retraining the neural network (for query-database [12]) or devising error-correcting codes, incurs substantial costs in terms of both time and financial resources. This situation prompts the exploration of *adaptation*, a method that applies minimal alterations to the DNA storage encoding to satisfy new constraints while preserving features of the pre-existing encoding scheme.

In this paper, we propose a streamlined and effective conversion coding method based on intuitive mapping that allows flexible conversion of sequences under different constraints, thereby efficiently addressing these challenges. This strategy avoids the need for retraining or redesigning the encoding scheme, leading to cost savings and enhanced capacity utilization. The proposed adaptation framework adjusts a given encoding scheme to accommodate the new homopolymer constraint while minimizing the variation between input and output (in terms of edit distance). Specifically, consider the pretrained encoder f_{pre} that satisfies a homopolymer constraint h_1 as well as its own criteria (e.g., query). Our objective is to propose a new encoding scheme f_{new} that satisfies a new homopolymer constraint h_2, while maintaining pretrained features by minimizing the edit distance between $f_{pre}(b)$ and $f_{new}(b)$ for all inputs b.

Rather than designing the new encoding scheme, f_{new}, from the ground up, our principal approach is to introduce a straightforward and efficient modification of the existing encoder, f_{pre}, using Transfer Coding. This method involves a direct mapping from a DNA sequence that satisfies homopolymer constraint h_1 to another sequence that meets a different homopolymer constraint h_2. As a result, the new encoding scheme f_{new} becomes a simple concatenation of the Transfer encoder and the existing encoder, f_{pre}. With its linear complexity, our Transfer Coding scheme is extremely easy to implement and apply to actual sequences.

We have also conducted various experiments and compared the results with the approximated theoretical limit. We consider both relaxing and restricting constraints; the relaxing case involves transferring the code for a stricter homopolymer constraint (e.g., $h=3$) to a less stringent homopolymer constraint (e.g., $h=4$). Conversely, the restricting case involves transferring the code originally designed for a less stringent homopolymer constraint (e.g., $h=4$) to a stricter homopolymer constraint (e.g., $h=3$). The proposed scheme achieves a superior rate compared to a naive scheme and shows results comparable with theoretical limits.

II. MATERIALS AND METHODS

A. PROBLEM FORMULATION

Let $\mathcal{X}=\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^*=\cup_{k=1}^{\infty}\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^k$ be the set of all lengths of DNA sequences. Similarly, let $\mathcal{X}_h\subset\mathcal{X}$ be the set of all lengths of DNA sequences that satisfies the homopolymer constraint h, i.e., any sequence $X\in\mathcal{X}_h$ can have at most h consecutive identical bases.

Suppose the encoder $f_{pre}:\{0,1\}^*\to\mathcal{X}_{h_1}$ is given for some $h_1>0$, which satisfies the homopolymer constraint h_1 as well as its own requirements (e.g., for query). Our goal is to obtain the new encoding scheme $f_{new}:\{0,1\}^*\to\mathcal{X}_{h_2}$ that satisfies the new homopolymer constraint h_2. On the other hand, we also want to preserve what f_{pre} originally aims for, such as querying. In other words, we want $d(f_{pre}(b),f_{new}(b))$ to be small where $d(\cdot,\cdot)$ denotes the edit distance metric between two sequences.

We also highlight another significant metric, the rate

$$R=\frac{\mathbb{E}\left[\ell(f_{pre}(b))\right]}{\mathbb{E}\left[\ell(f_{new}(b))\right]},$$

where $\ell(\cdot)$ represents the length of the encoded DNA sequence, and $\mathbb{E}[\cdot]$ represents the expectation. Essentially, we want to maintain the length of the encoded DNA sequence as compact as possible while simultaneously satisfying the new homopolymer constraints and minimizing the edit distance.

Our approach endeavors to strike a nuanced balance between the two objectives. Firstly, it seeks to minimize modifications to the sequence content, ensuring that both the original and adjusted sequences retain their intended functionality. Secondly, it aspires to maximize the utilization

TABLE 1. Insertion map M to satisfies homopolymer constraint. Symbol $*$ indicates any base. If not specified, $M(B, \cdot)$ is identity, for example $M(\mathtt{A}, \mathtt{GC}) = \mathtt{GCG}$.

B	x_1x_2	$M(B, x_1x_2)$	B	x_1x_2	$M(B, x_1x_2)$
A	A*	CA*	G	G*	TG*
A	CA	GCA	G	AT	ATA
A	GC	GCG	G	TG	ATG
C	C*	AC*	T	T*	GT*
C	AC	TAC	T	GT	CGT
C	TA	TAT	T	CG	CGC

Algorithm 1 Homopolymer Encoding

Input: $h \geq 1$, $\ell_X \geq 1$, $B \in \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$, and DNA sequence X
Output: DNA sequence Y
$Y \leftarrow \emptyset$
while X is nonempty **do**
  if last h bases of Y are the same **then**
    $x_1x_2 \leftarrow$ pop two bases from X
    $Y \leftarrow \text{CONCAT}(Y, M(B, x_1x_2))$
  else
    $x_1 \leftarrow$ pop one base from X
    $Y \leftarrow \text{CONCAT}(Y, x_1)$
  end if
end while
return Y

of the encoding capacity, thereby employing the available resources in the most efficient manner possible.

B. ALGORITHM

Our core proposition involves introducing a simple and effective method to modify the given encoder, f_{pre}, into a new encoder without necessitating a design from scratch. More specifically, we propose Transfer Coding that directly maps from $\mathcal{X}_{h_1}$ to $\mathcal{X}_{h_2}$, allowing the new encoding scheme to be a straightforward composition. For the Transfer encoder $f_{h_1 \to h_2} : \mathcal{X}_{h_1} \to \mathcal{X}_{h_2}$, the new encoding scheme is given by

$$f_{new} = f_{h_1 \to h_2} \circ f_{pre}, \tag{1}$$

where $f \circ g$ represents the function composition of f and g.

1) HOMOPOLYMER CODING

Before introducing Transfer Coding, which converts sequences from one homopolymer constraint (h_1) to another (h_2), we first focus on a simpler scenario that transforms any sequence to one that abides by a homopolymer constraint h. Recall that $\mathcal{X}_h$ is a set of DNA sequences of any length that satisfy the homopolymer constraint h. We then establish the encoder $f_h : \mathcal{X} \to \mathcal{X}_h$ that maps any length DNA sequence to a sequence that meets the homopolymer constraint h. Essentially, this process can be viewed as a Transfer from homopolymer constraint $\mathcal{X}_\infty$ to $\mathcal{X}_h$. As in the Transfer Coding approach, we aim for f_h to preserve the original structure of the input DNA sequence, meaning the edit distance between x and $f_h(x)$ should be minimal for any given DNA sequence $x \in \mathcal{X}$. Note that f_h will also be serving as a building block for our broader Transfer Coding scheme.

Algorithm 2 Homopolymer Decoding

Input: $h \geq 1$, $\ell_Y \geq 1$, B, and DNA sequence Y
Output: DNA sequence X
$X \leftarrow \emptyset$
while Y is nonempty **do**
  if last h bases of Y are the same **then**
    $y_1y_2y_3 \leftarrow$ pop three bases from Y
    $X \leftarrow \text{CONCAT}(X, M^{-1}(B, y_1y_2y_3))$
  else
    $y_1 \leftarrow$ pop one base from Y
    $X \leftarrow \text{CONCAT}(X, y_1)$
  end if
end while
return X

To ensure homopolymer constraints are met while minimally altering the sequence, we propose a table-based insertion scheme. Simply put, if the input DNA sequence contains a homopolymer sequence exceeding h in length, we insert a different base following the h-th base. For instance, if the input contains $h+1$ consecutive A's, we would insert a base C. The choice of C is due to the balance of GC content.

However, we must satisfy decodability, or the ability to recover the original sequence from the encoded sequence, and prevent collisions, i.e., $f_h(x_1) \neq f_h(x_2)$ for $x_1 \neq x_2$. The above insertion technique might lead to such conflicts. To illustrate, in the case of $h = 3$, both AAAA and AAACA map to AAACA, making it impossible to decode AAACA uniquely. To resolve this, we introduce additional rules for sequences that could potentially conflict when decoded. More precisely, we modify AAACA to AAAGCA and AAAGC to AAAGCG to differentiate between them. This modification is also captured in Table 1, where the symbol '*' denotes any nucleotide base. More specifically, if the preceding h bases are A and the next two bases are GC, we replace GC with GCG. We denote this mapping, as defined by Table 1, as M. All other scenarios, such as h consecutive C's, can be similarly defined and are also included in Table 1.

It is worth noting that if an input sequence concludes with h consecutive identical bases, we may not be able to read two bases afterward. In these scenarios, we interpret the any-base-symbol '*' as indicating an end-of-sequence (eos). As an illustration, if a sequence ends with AAAA where $h = 3$, we replace the last A with CA, interpreting A* as a concatenation of A and eos symbol.

We then propose a scheme called Homopolymer Coding, that reads the bases of the input DNA sequence sequentially and records the number of consecutive occurrences of the same base. If this number equals the homopolymer constraint h, then the above rules (based on mapping M) are followed

TABLE 2. Replacement map M_r to obtain the invertible sequence. If not specified, $M_r(B, \cdot)$ is identity, for example $M_r(\mathtt{A}, \mathtt{CCC}) = \mathtt{CCC}$.

B	$x_1x_2x_3$	$M_r(B, x_1x_2x_3)$	B	$x_1x_2x_3$	$M_r(B, x_1x_2x_3)$
A	GCG	GCAG	A	$\mathtt{GCX}_\mathtt{G}$	$\mathtt{GCGX}_\mathtt{G}$
C	TAT	TACT	C	$\mathtt{TAX}_\mathtt{T}$	$\mathtt{TATX}_\mathtt{T}$
G	ATA	ATGA	G	$\mathtt{ATX}_\mathtt{A}$	$\mathtt{ATAX}_\mathtt{A}$
T	CGC	CGTC	T	$\mathtt{CGX}_\mathtt{C}$	$\mathtt{CGCX}_\mathtt{C}$
B	x_1x_2	$M_r(x)$	B	x_1x_2	$M_r(x)$
A	CA	GCA	C	AC	TAC
G	TG	ATG	T	GT	CGT

Algorithm 3 Replacement Encoding

Input: $h \geq 1$, $\ell_X \geq 1$, B, and DNA sequence X
Output: DNA sequence Y
$Y \leftarrow \emptyset$
while X is nonempty **do**
  if last h bases of Y are the same **then**
    if len$(X) = 2$ **then**
      $x_1x_2 \leftarrow$ pop two bases from X
      $Y \leftarrow$ CONCAT$(Y, M_r(B, x_1x_2))$
    else
      $x_1x_2x_3 \leftarrow$ pop three bases from X
      $Y \leftarrow$ CONCAT$(Y, M_r(B, x_1x_2x_3))$
    end if
  else
    $x_1 \leftarrow$ pop one base from X
    $Y \leftarrow$ CONCAT(Y, x_1)
  end if
end while
return Y

when reading the next bases. Otherwise, the bases are output directly. This scheme inserts 'dummy' bases to satisfy homopolymer constraints, i.e., striving to keep the number of insertions minimal, thereby maintaining a small edit distance between the input and output sequences. The complete encoding process is detailed in Algorithm 1.

Decoder g_h serves as the left inverse of f_h, i.e., $g_h(f_h(x)) = x$ for all $x \in \mathcal{X}$. Mirroring the encoder, the decoder g_h sequentially decodes the sequence y, where $y = f_h(x)$. It starts by checking and counting the number of consecutive identical bases. If the last h bases are identical ($Y_{i-h} = \cdots = Y_{i-1} = B$ at step i), it then reads the three subsequent bases ($Y_iY_{i+1}Y_{i+2}$) and locates the corresponding inverse mapping base in Table 1. This decoding process is outlined in Algorithm 2.

2) TRANSFER CODING

We now present a Transfer Coding scheme. The proposed scheme includes an encoder, denoted as $f_{h_1 \to h_2}$, which modifies a sequence that satisfies a homopolymer constraint of h_1 to one that satisfies with a homopolymer constraint of h_2. We aim to maintain a minimal edit distance between x and $f_{h_1 \to h_2}(x)$ for any given $x \in \mathcal{X}_{h_1}$, while also modifying the input sequence minimally.

TABLE 3. Rate of the Homopolymer encoder, evaluated across varying sequence lengths (100, 200, 500, and 1000) and differing homopolymer constraints (3, 4, 5). TL denotes the theoretical limits [3] and ℓ_i denotes the input sequence length of i.

h	TL [3]	ℓ_{100}	ℓ_{200}	ℓ_{500}	ℓ_{1000}
i.i.d. input					
3	0.9915	0.9830	0.9827	0.9826	0.9825
4	0.9979	0.9958	0.9957	0.9957	0.9956
5	0.9994	0.9990	0.9989	0.9989	0.9989
Markov input					
3	-	0.8643	0.8628	0.8621	0.8618
4	-	0.9137	0.9124	0.9117	0.9115
5	-	0.9435	0.9424	0.9418	0.9416

TABLE 4. Normalized edit distance between the input and the output of the Homopolymer encoder, evaluated across varying sequence lengths (100, 200, 500, and 1000) and differing homopolymer constraints (3, 4, 5). Recall that ℓ_i denotes the input sequence length of i.

h	ℓ_{100}	ℓ_{200}	ℓ_{500}	ℓ_{1000}
i.i.d. input				
3	0.01732	0.01758	0.01777	0.01781
4	0.00425	0.00431	0.00437	0.00439
5	0.00104	0.00108	0.00109	0.00110
Markov input				
3	0.15704	0.15898	0.16002	0.16042
4	0.09445	0.09596	0.09686	0.09711
5	0.05990	0.06117	0.06180	0.06204

It may be tempting to think that we can directly apply f_{h_2} to the output of f_{pre} to satisfy the stricter condition of h_2 if $h_2 < h_1$ (i.e., $f_{new} = f_{h_2} \circ f_{pre}$). However, this approach does not effectively utilize the information that the output of f_{pre} already satisfies the homopolymer constraint h_1. Likewise, if $h_2 > h_1$, we could theoretically maintain the same encoding strategy with $f_{new} = f_{pre}$, since f_{pre} already complies with the stricter homopolymer constraint h_1, and therefore, automatically fulfill the homopolymer constraint of h_2. Nonetheless, this approach would fail to make full use of the available sequence options in the output space.

Our core approach is to construct

$$f_{h_1 \to h_2}(x) = f_{h_2}(g_{h_1}(x)). \tag{2}$$

This entails converting $x \in \mathcal{X}_{h_1}$ to a general sequence in $\mathcal{X}$ without having any homopolymer constraints using decoder g_{h_1}, then applying Homopolymer encoding f_{h_2} to satisfy homopolymer constraint h_2. However, encoder f_{h_1} is not surjective, and there are instances where $g_{h_1}(x)$ is not well-defined. For instance, Table 1 demonstrates cases where a redundant base is introduced to meet the homopolymer constraint, leaving certain codewords like AAAGCC unused, as AAAGC is always converted to AAAGCG. Hence, combinations like AAAGCC and AAAGCT are non-invertible.

For sequences $x \in \mathcal{X}_{h_1}$ that cannot be decoded, we suggest an additional encoding r, ensuring that $r(x)$ is decodable, i.e., $g_{h_1}(r(x))$ is well-defined. We refer to r as 'replacement,' which is based on the mapping shown in Table 2. We replace both base combinations that cannot be decoded and those that could potentially cause conflicts. For instance, when $h = 3$, the replacement r modifies AAAGCG to AAAGCAG, and AAAGCX$_G$ to AAAGCGX$_G$, with X$_G$ representing any base other than G. This strategy allows us to minimally alter the sequence while still meeting our objectives. Like the mapping in Table 1, we interpret X$_G$ as an end-of-sequence (eos) symbol when it occurs at the end of a sequence.

The formal encoding process involves reading the sequence sequentially and counting consecutive occurrences of the same base in the sequence. When the last h_1 bases are identical, we read the next three bases and replace them if they are specified in Table 2. The entire replacement process is detailed in Algorithm 3.

Finally, our proposed Transfer encoding can be expressed as:

$$f_{h_1 \to h_2}(x) = f_{h_2}(g_{h_1}(r_{h_1}(x))), \tag{3}$$

for $x \in \mathcal{X}_{h_1}$. Conversely, the decoding step is the reverse of the encoding.

$$g_{h_2 \to h_1}(x) = r_{h_1}^{-1}(f_{h_1}(g_{h_2}(x))), \tag{4}$$

for $x \in \mathcal{X}_{h_2}$

III. ENCODING RATES

In this section, we examine the theoretically achievable rates of both Homopolymer encoding and Transfer encoding, using results from the theory of binary to DNA encoding.

A. HOMOPOLYMER ENCODING

When it comes to encoding binary sequences into DNA sequences, Erlich et al. [3] propose a theoretical rate limit considering homopolymer constraints of h. The maximum achievable rate, $R_B(h)$, is given by

$$R_B(h) \approx 2 - \frac{3\log_2 e}{4^{h+1}} (bits/nt), \tag{5}$$

where e is Euler number. In the case of Homopolymer encoding where we map a DNA sequence to a sequence with a homopolymer constraint h, we consider the unconstrained DNA sequence as providing 2 bits per base. Consequently, the theoretically achievable rate becomes

$$R_H(h) = \frac{\mathbb{E}[\ell(x)]}{\mathbb{E}[\ell(y)]} = \frac{R_B(h)}{2} \approx 1 - \frac{3\log_2 e}{2 \times 4^{h+1}}. \tag{6}$$

We implicitly assume that the input unconstrained sequence x results from binary to DNA encoding with an optimal coding ratio of 2 bits.

B. TRANSFER ENCODING

For Transfer encodings, in order to compare the optimal coding ratio, for $y = f_{h_1 \to h_2}(x)$, we need to compute the rate as the ratio between the expected sequence lengths of y and x, i.e., $R = \mathbb{E}[\ell(x)]/\mathbb{E}[\ell(y)]$. Assuming that both x and y are outcomes of optimal encoding from binary to $\mathcal{X}_{h_1}$ and $\mathcal{X}_{h_2}$, respectively, the theoretical limit of the rate would be

$$R_T(h_1 \to h_2) = \frac{\mathbb{E}[\ell(x)]}{\mathbb{E}[\ell(y)]} = \frac{2/\mathbb{E}[\ell(y)]}{2/\mathbb{E}[\ell(x)]} = \frac{R_H(h_2)}{R_H(h_1)}. \tag{7}$$

For example, $R_B(2)$ is 1.93237, $R_B(3)$ is 1.98309, and $R_B(4)$ is 1.99577 can be obtained through equation 5, so $R_T(2 \to 3) = R_H(3)/R_H(2) \approx 1.0262$. In the subsequent section on experiments, we will offer the rate of the proposed scheme and compare it to these theoretical limits.

TABLE 5. Rate of Transfer Coding evaluated across differing sequence lengths (100, 200, 400, and 1000) and varying constraint transitions (4 → 3, 3 → 2, and 4 → 2). 'TL' represents theoretical limits [3] and ℓ_i denotes an input sequence of length i. The rate of the Naive approach (which directly applies f_h to the input sequence) is also provided for comparison.

ℓ_i	$h_1 \to h_2$	TL	Rate (Ours)	Rate (Naive)
i.i.d. input				
100	$4 \to 3$	0.9936	0.9839	0.9831
	$3 \to 2$	0.9744	0.9370	0.9347
	$4 \to 2$	0.9682	0.9345	0.9335
200	$4 \to 3$	0.9936	0.9838	0.9829
	$3 \to 2$	0.9744	0.9364	0.9339
	$4 \to 2$	0.9682	0.9339	0.9330
400	$4 \to 3$	0.9936	0.9837	0.9828
	$3 \to 2$	0.9744	0.9362	0.9337
	$4 \to 2$	0.9682	0.9335	0.9326
1000	$4 \to 3$	0.9936	0.9836	0.9827
	$3 \to 2$	0.9744	0.9360	0.9335
	$4 \to 2$	0.9682	0.9334	0.9324
Markov input				
100	$4 \to 3$	-	0.8831	0.8803
	$3 \to 2$	-	0.8194	0.8158
	$4 \to 2$	-	0.8330	0.8300
200	$4 \to 3$	-	0.8821	0.8792
	$3 \to 2$	-	0.8184	0.8148
	$4 \to 2$	-	0.8323	0.8292
400	$4 \to 3$	-	0.8816	0.8787
	$3 \to 2$	-	0.8180	0.8143
	$4 \to 2$	-	0.8320	0.8288
1000	$4 \to 3$	-	0.8813	0.8784
	$3 \to 2$	-	0.8178	0.8140
	$4 \to 2$	-	0.8318	0.8286

IV. EXPERIMENTS

In this section, we report the experimental results from testing our proposed encoding scheme, employing simulated data for the evaluation process. To gauge the performance of our Homopolymer encoding, we utilize DNA sequences originating from an independent and identically distributed (i.i.d.) uniform distribution (each base has a 1/4 probability) as the input. Conversely, Transfer encoding $f_{h_1 \to h_2}$ focuses on transitioning between DNA sequences under different

homopolymer constraints, necessitating the generation of DNA sequences that fulfill these homopolymer constraints. To fulfill this requirement, we generate DNA sequences in a sequential manner, utilizing the i.i.d. uniform distribution (each base has a 1/4 probability), and if there are h_1 identical bases B in a sequence, the subsequent base is sampled from the i.i.d. uniform distribution excluding base B (i.e., probability of 1/3 excluding B).

To further substantiate our findings, we conducted an additional experiment using a Markov distribution for input sequence generation. The initial base was chosen with equal likelihood from among the four possible nucleobases. For subsequent bases, the probability of the same base reoccurring was set at 0.7, whereas each of the remaining three bases had an equal chance of being chosen, with a probability of 0.1. To ensure the generated input DNA sequences adhered to specified homopolymer constraints, we implemented logic such that if adding another identical base would violate the constraint, a different base was randomly selected from the remaining three, each with an equal likelihood. Conversely, in states where the homopolymer constraint was not at risk of being violated, the same base was generated again with a probability of 0.7, while the remaining bases each had a 0.1 probability of being selected.

We randomly create 100,000 DNA sequences of lengths 100, 200, 400, and 1000, ensuring that they satisfy homopolymer constraints $h_1 \in \{2, 3, 4\}$ respectively, and then encode them into sequences that comply with a homopolymer constraint of $h_2 \in \{2, 3, 4\}$ using Transfer encoding. We analyze the experimental results separately, segmenting them into two categories: cases with stricter constraints ($h_1 > h_2$) and those with relaxed constraints ($h_1 < h_2$). The source code of our program is available for review on https://github.com/gyfbianhuanyun/DNA2DNA_Codec_for_Homopolymer_constraints. In this experiment, featuring a DNA sequence length of 100 and a total of 100,000 sequences, the program's runtime for function $f_{3\to4}$ was recorded as 65.78 seconds. In contrast, the runtime for $f_{4\to3}$ was 62.31 seconds. It is noteworthy that when using Homopolymer encoding exclusively, the execution time is 51.59 seconds for $f_{4\to3}$, while there is no additional execution time for $f_{3\to4}$.

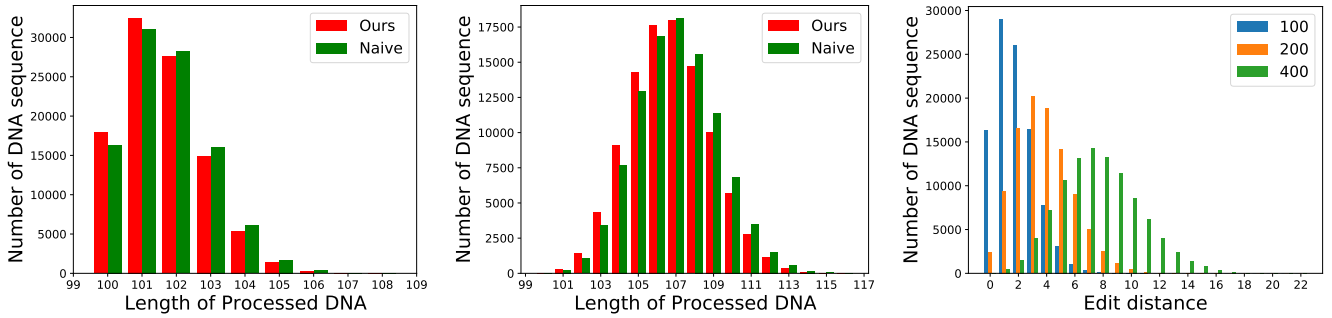
A. HOMOPOLYMER ENCODING

Before presenting the results of Transfer encoding, we first illustrate the performance of our Homopolymer encoding. In this case, we administer our method to input sequences x, considering various homopolymer constraints and DNA sequence lengths. We apply the Homopolymer encoder f_h to 100,000 randomly generated DNA sequences of assorted lengths, and the outcomes are presented in Table 3 and Table 4. As evidenced in Table 3, our algorithm stays remarkably close to the theoretical limit, irrespective of whether h is small or large. Table 4 exhibits the ratio of edit distance to input sequence length between the input and output sequences of the Homopolymer encoder. It can be observed that, while the change in the sequence length's edit distance is up to 1.7% in the i.i.d experiment, in most instances it remains constant (especially when $h = 4$ and $h = 5$). Under identical homopolymer conditions, the ratio remains essentially the same. In addition, we calculated the proportion of GC content before and after encoding. In the i.i.d. experiment using homopolymer encoding with $h = 3$ and DNA sequence length of 100, the average proportion of GC content before the experiment was 50.039%, and the proportion after encoding was 50.037%, which remained basically unchanged.

To highlight the superiority of our method in preserving edit distance, we conducted a comparison with existing DNA encoding techniques. In our comparative experiments, we initially used an i.i.d. process to generate a dataset of 10,000 DNA sequences, each 100 bases long. These DNA sequences were then converted into binary format, assigning 2 bits per base, which is a standard practice in this field. This binary data was then encoded into DNA sequences using different algorithms as documented in previous papers. The final step involved comparing the encoded DNA sequences with the original sequences, focusing on the ratio and

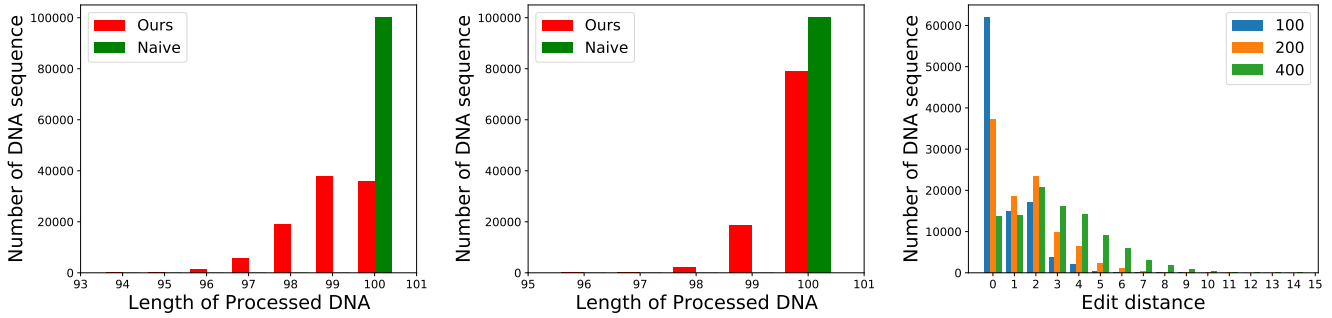
TABLE 6. Rate of Transfer Coding evaluated across differing sequence lengths (100, 200, 400, and 1000) and varying constraint transitions ($2 \to 3$, $3 \to 4$, and $2 \to 4$). 'TL' represents theoretical limits [3] and ℓ_i denotes an input sequence of length i. The rate of the Naive approach (does not apply any additional coding), which is 1, is also provided for comparison.

ℓ_i	$h_1 \to h_2$	TL	Rate (Ours)	Rate (Naive)
i.i.d. input				
100	$2 \to 3$	1.0262	1.0100	1.0
	$3 \to 4$	1.0063	1.0023	1.0
	$2 \to 4$	1.0328	1.0153	1.0
200	$2 \to 3$	1.0262	1.0102	1.0
	$3 \to 4$	1.0063	1.0024	1.0
	$2 \to 4$	1.0328	1.0157	1.0
400	$2 \to 3$	1.0262	1.0103	1.0
	$3 \to 4$	1.0063	1.0025	1.0
	$2 \to 4$	1.0328	1.0158	1.0
1000	$2 \to 3$	1.0262	1.0104	1.0
	$3 \to 4$	1.0063	1.0024	1.0
	$2 \to 4$	1.0328	1.0159	1.0
Markov input				
100	$2 \to 3$	-	1.0039	1.0
	$3 \to 4$	-	1.0020	1.0
	$2 \to 4$	-	1.0126	1.0
200	$2 \to 3$	-	1.0039	1.0
	$3 \to 4$	-	1.0020	1.0
	$2 \to 4$	-	1.0127	1.0
400	$2 \to 3$	-	1.0039	1.0
	$3 \to 4$	-	1.0021	1.0
	$2 \to 4$	-	1.0128	1.0
1000	$2 \to 3$	-	1.0040	1.0
	$3 \to 4$	-	1.0021	1.0
	$2 \to 4$	-	1.0129	1.0



(a) $f_{4\to3}$ (b) $f_{3\to2}$ (c) Strictness $f_{4\to3}$

FIGURE 1. Distribution of length and edit Distance for Transfer Coding in a stricter constraint case when the input sequence length is 100. (a, b) display the length distribution of output sequences produced by the proposed Transfer Coding (in red) and the Naive Coding (in green) for transitions 4 → 3 and 3 → 2, respectively. (c) demonstrates the edit distance between the input and output sequences of the Transfer encoder in the 4 → 3 case, given input sequence lengths of 100, 200, and 400.



(a) $f_{2\to3}$ (b) $f_{3\to4}$ (c) Relaxation $f_{3\to4}$

FIGURE 2. Distribution of length and edit Distance for Transfer Coding in a relaxing constraint case when the input sequence length is 100. (a, b) display the length distribution of output sequences produced by the proposed Transfer Coding (in red) and the Naive Coding (in green) for transitions 2 → 3 and 3 → 4, respectively. (c) demonstrates the edit distance between the input and output sequences of the Transfer encoder in the 3 → 4 case, given input sequence lengths of 100, 200, and 400.

TABLE 7. Comparative analysis of edit distance preservation capabilities. Ed indicates the normalized edit distance. Our Result (Ours (*h*)) demonstrates results with constraint *h*.

	Goldman [6]	Ours (1)	Song [7]	Wang [8]	Ours (3)
Rate	0.7877	0.7363	0.9091	0.9259	0.9830
Ed	0.6288	0.3582	0.5831	0.5696	0.0173

normalized edit distance as key metrics. It is crucial to note that Goldman et al. [6] uses a homopolymer setting with $h = 1$, while the methods by Song et al. [7] and Wang et al. [8] use a homopolymer setting with $h = 3$. This specific setting aligns with the object settings as mentioned in the paper. The comparisons, detailed in Table 7, demonstrate that our encoding method results in fewer alterations to the original DNA data than the others. This can be attributed to the tendency of other techniques to completely reprocess the binary data, resulting in a more significant deviation from the original DNA sequence. Overall, our innovative encoding strategy more effectively maintains the integrity of the original DNA data compared to other recent methods.

B. THE STRICTER CONSTRAINT

Firstly, we consider the scenario of stricter constraints, where Transfer encoding is implemented on a DNA sequence with a higher homopolymer constraint h_1, while modifying the encoder so that the output meets the more rigorous constraint $h_2 < h_1$. In this case, the straightforward approach would be to directly apply the encoder f_{h_2}, which is designed to meet a homopolymer constraint of h_2. However, this naive approach achieves sub-optimality as it fails to utilize the information that the input has a homopolymer sequence with a length h_1 at most. In this section, we compare our proposed Transfer encoding against this naive approach.

Simulation experiments are carried out on random DNA sequences with lengths of 100, 200, 400, and 1000, fulfilling the homopolymer constraint $h_1 \in \{2, 3, 4\}$. The objective was to transfigure these into DNA sequences complying with $h_2 \in \{2, 3, 4\}$. The results are displayed in Table 5. In the scenario of transfer to stricter constraints, the rate falls below 1. The rate of our proposed scheme demonstrates comparable results with the theoretical limits and also exhibits sufficiently small edit distances. The impact analysis of the proposed encoding is illustrated in Figure 1. When comparing the outcomes from Naive coding (using only f_{h_2}), our Transfer encoding method excels in achieving a higher rate and shorter sequence length, enabling the representation of the same data using fewer bases.

Additionally, we assessed the results of the edit distance, as portrayed in Figure 1(c). In the figure, we selected the performance of $f_{4\to3}$ under sequences of different lengths, where the edit distance is small compared to the input sequence length.

C. RELAXING THE CONSTRAINT

Then, we take into consideration the scenario of a more relaxed homopolymer constraint, wherein the original sequence complies with a stricter homopolymer constraint. In such a scenario, the sequence can be directly utilized without encoding (with the rate of 1). This is a baseline for comparison with our results. Note that this simplistic approach (utilizing without encoding) is strictly suboptimal as it does not completely exploit the available sequence combinations in the output space. For comparison purposes, simulations were conducted on random DNA sequences with lengths of 100, 200, 400, and 1000.

Table 6 illustrates that our results closely align with the theoretical limit. It is worth noting that the rate values exceed 1 due to the relaxed constraint, indicating a smaller number of bases are needed to represent the same amount of data. In Figure 2, we explore the output sequence lengths of our proposed method, $f_{h_1\to h_2}$. We also provide results of Naive encoder using only f_{h_2} where all sequence lengths remained constant. Our Transfer encoding method showcased superior performance by representing the same data volume with fewer bases. Additionally, we present the edit distance outcomes in Figure 2(c). Our method affords several benefits, including shorter sequence length and an elevated rate. The results highlight that the edit distance is minimal relative to the original sequence length.

V. CONCLUSION

We have presented a novel, efficient method for DNA-to-DNA encoding. This approach is grounded in a mapping methodology that significantly curtails the number of base insertions and substitutions required. Despite its linear complexity, our method outperforms conventional encoding strategies. It facilitates the swift conversion of DNA sequences across disparate homopolymer constraints, offering a viable and efficient solution for the storage of digital data via DNA molecules.

REFERENCES

[1] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, Sep. 2012.

[2] Y. Dong, F. Sun, Z. Ping, Q. Ouyang, and L. Qian, "DNA storage: Research landscape and future prospects," *Nat. Sci. Rev.*, vol. 7, no. 6, pp. 1092–1107, Jun. 2020.

[3] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, Mar. 2017.

[4] 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," *Angew. Chem. Int. Ed.*, vol. 54, no. 8, pp. 2552–2555, Feb. 2015.

[5] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 1, pp. 1–6, Jul. 2017.

[6] 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, Feb. 2013.

[7] W. Song, K. Cai, M. Zhang, and C. Yuen, "Codes with run-length and GC-content constraints for DNA-based data storage," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 2004–2007, Oct. 2018.

[8] Y. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, and C. L. Poh, "Construction of bio-constrained code for DNA data storage," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 963–966, Jun. 2019.

[9] K. A. S. Immink and K. Cai, "Properties and constructions of constrained codes for DNA-based data storage," *IEEE Access*, vol. 8, pp. 49523–49531, 2020.

[10] P. Mishra, C. Bhaya, A. K. Pal, and A. K. Singh, "Compressed DNA coding using minimum variance Huffman tree," *IEEE Commun. Lett.*, vol. 24, no. 8, pp. 1602–1606, Aug. 2020.

[11] S.-J. Park, Y. Lee, and J.-S. No, "Iterative coding scheme satisfying GC balance and run-length constraints for DNA storage with robustness to error propagation," *J. Commun. Netw.*, vol. 24, no. 3, pp. 283–291, Jun. 2022.

[12] C. Bee, Y.-J. Chen, M. Queen, D. Ward, X. Liu, L. Organick, G. Seelig, K. Strauss, and L. Ceze, "Molecular-level similarity search brings computing to DNA data storage," *Nature Commun.*, vol. 12, no. 1, p. 4764, Aug. 2021.

[13] M. Blawat, K. Gaedke, I. Hütter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for DNA data storage," *Proc. Comput. Sci.*, vol. 80, pp. 1011–1022, Jan. 2016.

[14] S. Chandak, K. Tatwawadi, B. Lau, J. Mardia, M. Kubit, J. Neu, P. Griffin, M. Wootters, T. Weissman, and H. Ji, "Improved read/write cost tradeoff in DNA-based data storage using LDPC codes," in *Proc. IEEE Allerton*, Sep. 2019, pp. 147–156.

[15] W. H. Press, J. A. Hawkins, S. K. Jones, J. M. Schaub, and I. J. Finkelstein, "HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 31, pp. 18489–18496, Aug. 2020.

[16] J. Jeong, S.-J. Park, J.-W. Kim, J.-S. No, H. H. Jeon, J. W. Lee, A. No, S. Kim, and H. Park, "Cooperative sequence clustering and decoding for DNA storage system with fountain codes," *Bioinformatics*, vol. 37, no. 19, pp. 3136–3143, Oct. 2021.

YUNFEI GAO is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, Hongik University, Seoul, South Korea. His research interests include deep learning, DNA storage, and computational biology.

ALBERT NO (Member, IEEE) received the B.S. degree in electrical engineering and mathematics from Seoul National University, Seoul, South Korea, in 2009, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2012 and 2015, respectively. From 2015 to 2017, he was a Data Scientist with Roche. In 2017, he joined Hongik University, Seoul, where he is currently an Assistant Professor of electronic and electrical engineering. His research interests include learning theory, lossy compression, and bioinformatics.