Error-Correcting Codes for Labeled DNA Sequences

Dganit Hanania[1], and **Eitan Yaakobi**[1]
[1]Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel
Email: {dganit, yaakobi}@cs.technion.ac.il

***Abstract*—Labeling of DNA molecules is a fundamental technique for DNA visualization and analysis. This process was mathematically modeled in [1], where the received sequence indicates the positions of the used labels. In this work, we develop error correcting codes for labeled DNA sequences, establishing bounds and constructing explicit systematic encoders for single substitution, insertion, and eletion errors. We focus on two cases: (1) using the complete set of ength-two labels and (2) using the minimal set of length-two labels hat ensures the recovery of DNA sequences from their labeling for almost' all DNA sequences.**

I. Introduction

Labeling of DNA molecules using fluorescent markers is a ey technique in molecular biology and medicine, enabling the isualization and analysis of DNA at the molecular level [2]–[4]. t is widely used in genomics and microbiology for studying gene xpression, DNA-protein interactions, and structural variations. echniques such as Fluorescence in situ Hybridization (FISH) [2], RISPR [3], [5], and Methyltransferases [6] allow targeting of NA sequences. Labeling can be applied at the per-base level, hich is essential for sequencing technologies [7] and epigenomic tudies [8], or at specific target sequences for applications in pecies identification [9] and optical mapping [10], [11]. This pproach also extends to proteins and RNA, supporting molecular nalysis [12], [13] and regulatory studies.

The labeling process was first mathematically modeled in [1], 14] as follows. Let $\boldsymbol{x} \in \{A, C, G, T\}^n$ be a DNA sequence nd let $\boldsymbol{\alpha} \in \{A, C, G, T\}^\ell$ be a label of length ℓ. The labeling equence of $\boldsymbol{x}$ indicates the position where the label appears in $\boldsymbol{x}$. or example, for $\boldsymbol{x} = ACGTATAGACAC$ and the label $\boldsymbol{\alpha}_1 = C$, the labeling sequence of $\boldsymbol{x}$ is $L_{\boldsymbol{\alpha}_1}(\boldsymbol{x}) = 100000001010$. his model was extended to accommodate multiple labels. For xample, for $\boldsymbol{\alpha}_2 = T$, $L_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2}(\boldsymbol{x}) = 100202001010$. In [14], the uthors defined labeling capacity as the maximum information ate achievable using specific patterns as labels on a DNA olecule. They also studied the minimum number of labels required to reconstruct the original DNA sequence from its labeling sequence. For labels of length two, it was shown that ten labels are both sufficient and necessary over the DNA alphabet. The authors of [15] extended these results to labels of lengths greater than two. However, the analysis in those papers assumed a perfect labeling and reading process, free of errors.

In contrast to the ideal assumptions made in previous studies, real-world labeling processes are subject to various errors. These errors include missed labels (where some labels are undetected due to low efficiency), off-target labeling (where labels bind to unintended sequences), deletion errors (where certain symbols are unreadable by the sequencer), and synchronization errors (where labels are misaligned, shifting left or right in the output).

In this work, we aim to reconstruct the original labeling sequence from a received sequence that may contain errors. We define such error-correcting codes as *error labeling codes* and we focus on cases where the correct labeling sequence uniquely determines the original DNA sequence. Our study examines errors in labeled DNA sequences, focusing on two key cases: (1) using a minimal number of length-two labels to achieve maximal capacity and (2) utilizing the complete set of length-two labels.

We specifically address substitution, deletion, and insertion errors, providing both theoretical bounds and explicit constructions for error-correcting codes. Our results extend beyond DNA sequences and are applicable to general alphabets.

When considering the full set of labels, we draw connections to the well-studied symbol-pair read channel model [16]–[19] for length-two labels and the b-symbol read channel [20]–[22] for longer labels, offering an upper bound and a construction for single deletion or insertion labeling code. For the minimal label set case, we demonstrate the existence of codes that correct a single substitution, as well as those that handle a single deletion or insertion, and present explicit systematic encoding methods.

The rest of this paper is organized as follows. Section II formally defines the labeling channel, error labeling codes, and provides key definitions and preliminaries. In Section III, we present an upper bound for single insertion or deletion labeling codes and provide explicit constructions for both using all labels and a minimal set. In Section IV, we establish a lower bound on the maximal size of a single substitution labeling code and introduce an explicit encoder for the minimal label set case. Some proofs are omitted due to space limitations.

II. Definitions and Preliminaries

Let Σ_q denote the q-ary alphabet $\{0, 1, \ldots, q-1\}$. For $q = 4$, we mostly refer to the DNA alphabet, that is, $\Sigma_4 = \{A, C, G, T\}$. For a positive integer n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For a sequence $\boldsymbol{x} = (x_1, \ldots, x_n) \in \Sigma_q^n$, and $1 \leq i \leq n - k + 1$, let $\boldsymbol{x}_{[i;k]} = (x_i, \ldots, x_{i+k-1})$. A *label* $\boldsymbol{\alpha} \in \Sigma_q^\ell$ is a sequence of (relatively short) length ℓ over Σ_q. We use the *labeling model* as described in [14].

Definition 1. Let $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k$ be k labels of lengths $\ell_1, \ldots, \ell_k$, respectively. Assume no label is a prefix of another label. Denote by $\mathcal{A}$ the set $\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k\}$, where $\boldsymbol{\alpha}_1 \leq \cdots \leq \boldsymbol{\alpha}_\ell$ are ordered lexicographically.

- The $\mathcal{A}$***−labeling sequence*** of $\boldsymbol{x} = (x_1, \ldots, x_n) \in \Sigma_q^n$ is the sequence $L_{\mathcal{A}}(\boldsymbol{x}) = (c_1, \ldots, c_n) \in \Sigma_{k+1}^n$, where $c_i = j$ if $\boldsymbol{x}_{[i;\ell_j]} = \boldsymbol{\alpha}_j$ and $i \leq n - \ell_j + 1$, and $c_i = 0$ if no such j exists.
- A sequence $\boldsymbol{u} \in \Sigma_{k+1}^n$ is termed a ***valid*** $\mathcal{A}$***-labeling sequence*** if there exists an $\boldsymbol{x} \in \Sigma_q^n$ such that $\boldsymbol{u} = L_{\mathcal{A}}(\boldsymbol{x})$.

The research was Funded by the European Union (ERC, DNAStorage, 101045114 and EIC, DiDAX 101115134). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

- The ***labeling capacity*** of $\mathcal{A}$ is

$$\mathsf{cap}(\mathcal{A}) \triangleq \limsup_{n\to\infty} \frac{\log_2(|\{L_{\mathcal{A}}(\boldsymbol{x}) : \boldsymbol{x} \in \Sigma_q^n\}|)}{n}.$$

Example 1. For the set of labels $\mathcal{A} = \{A, CC\}$ and $\boldsymbol{x} = TAGCCAACCCG$, it holds that $L_{\mathcal{A}} = 01020112200$.

In [14], the labeling capacity was determined for almost all cases involving a single label, as well as for some cases involving multiple labels. Another problem explored was determining the minimal number of labels of the same length required to achieve the full capacity. Specifically, in the context of DNA sequences, the question can be framed as: how many labels of a given length ℓ are needed to ensure that, given a labeling sequence, the original DNA sequence can (almost always) be reconstructed. For $\ell = 1$, it can be verified that $q-1$ labels are required. For $\ell = 2$, the results for any q were given in [14] by establishing a connection to a graph property where, for any k, there exists at most one path of length k between any two vertices in a graph with q vertices. Such graphs were termed *path-unique graphs*. For $\ell > 2$, the analysis was further developed in [15], using a similar connection to de Bruijn graphs. The results for $\ell = 2$ are provided in the following theorem.

Theorem 1. ([14]) The minimal number of labels of length two over Σ_q required to achieve the full labeling capacity is $\phi(q) := q^2 - n(q)$, where

$$n(q) = \begin{cases} \frac{(q+1)^2}{4} & q \text{ is odd} \\ \frac{q(q+2)}{4} & q \text{ is even.} \end{cases}$$

Specifically, for the DNA case, when $q = 4$, ten labels of length two are sufficient and necessary to gain the maximum capacity. An example of such set of labels is

$$\mathcal{S} = \{AC, CA, GA, GC, GG, GT, TA, TC, TG, TT\}.$$

In this work, we aim to analyze the use of a set of labels, considering the possibility of errors in the labeling sequence. Our goal is to recover the original labeling sequence from its erroneous version.

Definition 2. Let $\underline{\boldsymbol{e}} = (e_1, e_2, e_3) \in \mathbb{N}^3$ and let $\mathcal{A}$ be a set of labels over Σ_q. For $\boldsymbol{x} \in \Sigma_q^n$, its *$\underline{\boldsymbol{e}}$-error $\mathcal{A}$-labeling ball*, denoted by $\mathcal{B}L_{\mathcal{A}}(\boldsymbol{x}, \underline{\boldsymbol{e}})$, is the set of all possible $\mathcal{A}$-labeling sequences after introducing at most e_1 substitution errors, e_2 insertions and e_3 deletions to $L_{\mathcal{A}}(\boldsymbol{x})$.

A code $\mathcal{C} \subseteq \Sigma_q^n$ is called an *$\underline{\boldsymbol{e}}$-error $\mathcal{A}$-labeling code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$, $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$, it holds that

$$\mathcal{B}L_{\mathcal{A}}(\boldsymbol{x}_1, \underline{\boldsymbol{e}}) \cap \mathcal{B}L_{\mathcal{A}}(\boldsymbol{x}_2, \underline{\boldsymbol{e}}) = \emptyset.$$

In the case where $e_2 = e_3 = 0$, the code is referred to as an e_1-substitution $\mathcal{A}$-labeling code. Similarly, for $e_1 = e_3 = 0$, it is called an e_2-insertion $\mathcal{A}$-labeling code, and for $e_1 = e_2 = 0$, it is referred to as an e_3-deletion $\mathcal{A}$-labeling code.

Our objective in this work is to construct codes of maximal size for a given $\mathcal{A}$ and $\underline{\boldsymbol{e}}$. Observe that if the labeling sequences belong to an $\underline{\boldsymbol{e}}$-error correcting code, then the original sequences corresponding to these labeling sequences form an $\underline{\boldsymbol{e}}$-error $\mathcal{A}$-labeling code. This observation is formalized in the following theorem.

Theorem 2. Let $\mathcal{A}$ be a set of labels over Σ_q. Let $\mathcal{C}' \subseteq \Sigma_{|\mathcal{A}|+1}^n$ be an $\underline{\boldsymbol{e}}$-error-correcting code. Then,

$$\mathcal{C} = \{\boldsymbol{x} \in \Sigma_q^n : L_{\mathcal{A}}(\boldsymbol{x}) \in \mathcal{C}'\}$$

is an $\underline{\boldsymbol{e}}$-error $\mathcal{A}$-labeling code.

While this theorem provides a useful characterization, it is important to note that not every sequence over $\Sigma_{|\mathcal{A}|+1}$ is a valid labeling sequence. This adds complexity to the task of finding explicit codes. In this paper, we address this challenge by constructing explicit codes and deriving bounds on their cardinality.

In this paper, we focus on cases where $\phi(q)$ or more labels of length two are used, which allows sequence reconstruction of all sequences when no labeling errors occur. In this case, correcting errors in the labeling sequence enables recovery of the original sequence. In particular, we provide systematic constructions and bounds for single deletion/insertion $\mathcal{A}$-labeling codes and single substitution $\mathcal{A}$-labeling codes.

Remark 1. In [14], it was shown that full capacity can be achieved using $\phi(q)$ labels. However, ambiguity may arise during recovery of the original sequence if the first or last symbols in the labeling sequence are zero. In such cases, the start or end of the original sequence may not be uniquely determined. To address this issue, we explicitly fix the symbols before the first and after the last position in the sequence, for example, by setting $x_0 = x_{n+1} = 0$. While alternative methods could potentially reduce redundancy, we adopt this approach for its simplicity.

Remark 2. In general, using $|\mathcal{A}|$ labels results in labeling sequences over $\Sigma_{|\mathcal{A}|+1}$. However, when all labels have the same length ℓ, i.e., $|\mathcal{A}| = q^\ell$, this is effectively equivalent to using $q^\ell - 1$ labels. In this specific case, the labeling sequences are constructed over the alphabet Σ_{q^ℓ} rather than $\Sigma_{q^\ell+1}$.

The scenario of utilizing all labels of length two corresponds to the well-studied symbol-pair read channel model [16]–[19], which can be described as follows.

Definition 3. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \Sigma_q^n$ be a sequence of length n over Σ_q. The *symbol-pair read sequence of $\boldsymbol{x}$* is

$$\pi(\boldsymbol{x}) = ((x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n), (x_n, x_1)) \in \Sigma_{q^2}^n.$$

This paradigm closely resembles our framework, with the key distinction being the cyclic reading in the symbol-pair model, where the pair (x_n, x_1) is included. In contrast, our approach does not assume cyclic reading. Instead, we assume that the symbols before the first and after the last position in the sequence are known, allowing us to complete the labeling sequence in a manner consistent with the cyclic case. Specifically, given a sequence $\boldsymbol{x}' = x_0\boldsymbol{x}x_{n+1}$, where $x_0, x_{n+1} \in \Sigma_q$ are fixed and known, its labeling sequence is

$$L_{\mathcal{A}}(\boldsymbol{x}') = ((x_0, x_1), \ldots, (x_{n-1}, x_n), (x_n, x_{n+1})) \in \Sigma_{q^2}^{n+1}.$$

The final term (x_{n+1}, x_0) can be uniquely determined and added to the labeling sequence.

A generalization of the symbol-pair model has also been studied; it involves reading b symbols together and is referred to as the b-symbol read channel [20]–[22]. Previous work primarily focused on correcting substitution errors in these models. In [19], the problem of synchronization errors was studied, and, in particular, correcting a single deletion in the symbol-pair sequence was

addressed. However, their solution was restricted to the binary case and did not provide a general bound on the size of the corresponding codes. In this paper, we establish a general bound on the code size for the single-deletion case. Additionally, we establish the existence of such codes by extending results from the binary case in [23].

To derive this bound and construct the codes, we introduce the notion of the derivative of a sequence, defined as follows.

Definition 4. Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \Sigma_q^n$. The *derivative* of $\boldsymbol{x}$, denoted by $d(\boldsymbol{x})$, is defined as

$$d(\boldsymbol{x}) = (x_1, x_2 - x_1, \ldots, x_n - x_{n-1}) \in \Sigma_q^n.$$

where all operations are modulo q.

III. Single Deletion/Insertion Labeling Codes

In this section, we discuss single insertion or deletion $\mathcal{A}$-labeling codes for two scenarios: (1) the set of all labels of length two, and (2) a minimal set of labels of length two that achieves full capacity.

It is well known [24] that a code $\mathcal{C}$ is a t-deletion error-correcting code if and only if it is a t-insertion error-correcting code, and if and only if it is a (t_1, t_2)-error-correcting code for deletions and insertions, respectively, for any $t_1 + t_2 \leq t$. This property can be also extended for labeling codes.

Claim 1. Any code $\mathcal{C}$ is a t-deletion $\mathcal{A}$-labeling code if and only if it is a t-insertion $\mathcal{A}$-labeling code, and if and only if it is a t_1-deletion and t_2-insertion $\mathcal{A}$-labeling code, for any $t_1 + t_2 \leq t$.

Therefore, we will focus on single-deletion labeling codes, noting that these codes are also capable of correcting a single insertion as well.

A. Using All Labels of Length Two

In this section, we first present a construction for single-deletion labeling codes. Then, we derive an upper bound on the size of such codes. Specifically, we focus on labeling sequences obtained using all of the labels of length two. It can be verified that determining an upper bound on the code size for the case of all labels will directly yield an upper bound for the case involving $\phi(q)$ labels.

As mentioned earlier, the scenario of using all labels of length two corresponds to the symbol-pair read channel model. To construct and bound the single deletion case, we first analyze how a single deletion affects the labeling sequence. One part of the following observation was proved in [19] for the binary pair-symbol channel, and the proof for the other part and the non-binary case is similar. Before presenting the lemma, we define a zero-deletion error-correcting code as a code capable of correcting the deletion of a zero from its sequences.

Lemma 1. Let $\mathcal{A}$ be the set of all labels of length two, i.e., $\mathcal{A} = \Sigma_q^2$. Let $x_0, x_{n+1} \in \Sigma_q$, $\mathcal{C}' \subseteq \Sigma_q^{n+1}$ and let

$$\mathcal{C} = \left\{ \boldsymbol{x} \in \Sigma_q^n : d(\boldsymbol{x}')_{[2;n+1]} \in \mathcal{C}', \boldsymbol{x}' = x_0 \boldsymbol{x} x_{n+1} \right\}.$$

It holds that $\mathcal{C}'$ is a zero-deletion error correcting code if and only if $\mathcal{C}$ is a single-deletion $\mathcal{A}$-labeling code.

Note that if the first or last label in the labeling sequence is deleted, the deletion can be immediately corrected since x_0 and x_{n+1} are assumed to be known.

Next, we present a single-deletion $\mathcal{A}$-labeling code constructed based on Lemma 1. Note that any zero-deletion error-correcting code can be used to construct such a single-deletion $\mathcal{A}$-labeling code. The authors of [25] characterized codes that address our problem but did not provide explicit constructions for such codes. The authors of [23] proposed a code capable of correcting a single insertion of a zero in the binary case. Since any code that can correct a single insertion of a zero can also correct a single deletion of a zero [26], this code satisfies our requirements. Building on their construction, we extend this method to the non-binary case.

As defined in [23], let A_w^n be the set of binary sequences of length m with Hamming weight w. Moreover, define

$$S_{w,a}^{m,w+1} = \left\{ (s_1, \ldots, s_m) | \sum_{i=1}^{m} i \cdot s_i \equiv a \pmod{w+1} \right\}.$$

The following lemma was proved in [23].

Lemma 2. Each subset $S_{w,a}^{m,w+1}$ is a single zero-insertion correcting code. Moreover, there exists a such that

$$|S_{w,a}^{m,w+1}| \geq \frac{1}{w+1} \binom{m}{w}.$$

We will use these definitions and lemma in order to construct a single deletion $\mathcal{A}$-labeling code, when $\mathcal{A} = \Sigma_q^2$.

Theorem 3. There exists a single deletion $\mathcal{A}$-labeling code $\mathcal{C}$ such that,

$$|\mathcal{C}| \geq \frac{q^{n+1}}{(q-1)(n+2)}.$$

Proof. Define the following set of sequences of length m over Σ_q,

$$T_{w,a}^{m,w+1} = \left\{ (t_1, \ldots, t_m) : (\lceil \frac{t_1}{q-1} \rceil, \ldots, \lceil \frac{t_m}{q-1} \rceil) \in S_{w,a}^{m,w+1} \right\}.$$

From Lemma 2, this is a single zero-insertion correcting code over Σ_q. Since $|T_{w,a}^{m,w+1}| = (q-1)^w \cdot |S_{w,a}^{m,w+1}|$, there exists a such that

$$|T_{w,a}^{m,w+1}| \geq \frac{(q-1)^w}{w+1} \binom{m}{w}.$$

Now, similarly to the proof in [23], since two non-binary codewords of different Hamming weights cannot result in the same sequence after at most one zero is inserted, we can look at the union over different weights of $T_{w,a^*}^{m,w+1}$, when $a^* = \text{argmax}(|T_{w,a}^{m,w+1}|)$,

$$|\mathcal{C}'| = |\bigcup_{w=0}^{m} T_{w,a^*}^{m,w+1}| \geq \sum_{w=0}^{m} \frac{(q-1)^w}{w+1} \binom{m}{w} = \frac{1}{q-1} \cdot \frac{q^{m+1}}{m+1}.$$

The received set is a single zero insertion (or deletion) correcting code over Σ_q for sequences of length m. From Lemma 1, for $m = n+1$ and any $x_0, x_{n+1} \in \Sigma_q$,

$$\mathcal{C} = \left\{ \boldsymbol{x} \in \Sigma_q^n : d(\boldsymbol{x}')_{[2;m]} \in \mathcal{C}', \boldsymbol{x}' = x_0 \boldsymbol{x} x_{n+1} \right\}.$$

is a single deletion $\mathcal{A}$-labeling code. Note that for $\boldsymbol{x} \in \mathcal{C}$, we have $\sum_{i=2}^{n+2} d(x_0 \boldsymbol{x} x_{n+1})_i = x_{n+1} - x_0$. Given x_0, by the pigeonhole principle, there exists $x_{n+1} \in \Sigma_q$ such that

$$|\mathcal{C}| \geq \frac{1}{q-1} \cdot \frac{q^{n+2}}{n+2} \cdot \frac{1}{q} = \frac{1}{q-1} \cdot \frac{q^{n+1}}{n+2}.$$

□

Remark 3. The explicit cardinality of $S_{w,a}^{m,w+1}$ is being computed in [23]. Since $|T_{w,a}^{m,w+1}| = (q-1)^w \cdot |S_{w,a}^{m,w+1}|$, the cardinality of $T_{w,a}^{m,w+1}$ may be computed as well.

Next, we will give an upper bound on a single deletion $\mathcal{A}$-labeling code size. In order to do so, we will use the method of the generalized sphere packing bound, as described in [27], [28]. To discuss our results, some background is given.

We construct a hypergraph $\mathcal{H}(X, \mathcal{E})$ for which the vertices represent sequences of length $m-1$ and the hyperedges are the zero-deletion balls around any sequence of length m. More formally, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{q^{m-1}}\} = \Sigma_q^{m-1}$ and $\mathcal{E} = \{E_1, \ldots, E_{q^m}\} = \{B_1^d(\boldsymbol{x}) : \boldsymbol{x} \in \Sigma_q^m\}$, when $B_1^d(\boldsymbol{x})$ is the zero-deletion ball of $\boldsymbol{x}$. A *transversal* $T \subseteq X$ is a set of vertices that intersects all of the hyperedges in $\mathcal{H}$. Our goal is to find a minimal size of such a transversal. Let A be the $q^{m-1} \times q^m$ incidence matrix of $\mathcal{H}$, that is, $A(i,j) = 1$ if $x_i \in E_j$. A transversal can be described as a binary vector $\boldsymbol{w} \in \Sigma_2^{q^{m-1}}$ that satisfies $A^T \cdot \boldsymbol{w} \geq 1$. A vector that holds this inequality and its components are values in $\mathbb{R}^+$ is called a *fractional transversal*. Our goal is finding such a fractional transversal, since the size of any single zero-deletion correcting code $\mathcal{C}$, is bounded by

$$|\mathcal{C}| \leq \sum_{\boldsymbol{x} \in \Sigma_q^{m-1}} w_{\boldsymbol{x}}.$$

Theorem 4. Let $\mathcal{C} \subseteq \Sigma_q^m$ be a zero-deletion error-correcting code. For $q = 2$, it holds that $|\mathcal{C}| \leq \frac{2^{m+2}}{m-2}$. For $q > 2$, $|\mathcal{C}|$ is bounded from above by:

$$\sum_{z=1}^{\lfloor \frac{m}{2} \rfloor} \sum_{i=0}^{m-2z} \binom{i+z-1}{z-1} \binom{m-z-i}{z} \frac{(q-1)^{m-1-z-i}}{z}.$$

Proof. For any $\boldsymbol{x} \in \Sigma_q^m$, denote by $z(\boldsymbol{x})$ the number of zero runs in $\boldsymbol{x}$. Note that $|B_1^d(\boldsymbol{x})| = z(\boldsymbol{x})$. Consequently, a property similar to monotonicity holds, meaning that for any $\boldsymbol{y} \in B_1^d(\boldsymbol{x})$, it follows that $z(\boldsymbol{y}) \leq z(\boldsymbol{x})$. For this reason, choosing $w_{\boldsymbol{x}} = \frac{1}{z(\boldsymbol{x})}$ defines a valid fractional transversal [27].

When $q = 2$, there are $\binom{m+1}{2z}$ sequences of length m over Σ_2 with exactly z runs of zeroes. Thus,

$$|\mathcal{C}| \leq \sum_{x \in \Sigma_2^{m-1}} w_x = \sum_{x \in \Sigma_2^{m-1}} \frac{1}{z(\boldsymbol{x})} = \sum_{z=1}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{2z} \frac{1}{z}.$$

From [29], [30], it holds that

$$\sum_{z=1}^{\lfloor \frac{m}{2} \rfloor} \binom{m}{2z} \frac{1}{z} \leq 2 \sum_{z=1}^{m} \binom{m}{z} \frac{1}{z} \leq 2 \sum_{j=1}^{m} \frac{2^j - 1}{j} \leq \frac{2^{m+2}}{m-2}.$$

For the non-binary case, the calculation becomes more complex. In this case, there are

$$\sum_{i=0}^{n-(2z-1)} \binom{i+z-1}{z-1} \binom{n-(2z-1)-i+z}{z} (q-1)^{n-z-i}$$

sequences of length $n = m-1$ over Σ_q with exactly z runs of zeroes. To explain this, observe that there are z zeroes and $z-1$ non-zero symbols that must appear in such sequences. From the remaining $n-(2z-1)$ symbols, we allocate i zeroes into z runs, and the remainder are divided into $z+1$ non-zero segments. Each non-zero symbol can take any of $q-1$ values. □

The following corollary can be derived from Lemma 1 and Theorem 4 by setting $m = n+1$.

Corollary 1. Let if $\mathcal{C} \subseteq \Sigma_q^n$ be a single-deletion $\mathcal{A}$-labeling code. For $q = 2$, it holds that $|\mathcal{C}| \leq \frac{2^{n+3}}{n-1}$. For $q > 2$, $|\mathcal{C}|$ is bounded from above by:

$$\sum_{z=1}^{\lfloor \frac{n+1}{2} \rfloor} \sum_{i=0}^{n+1-2z} \binom{i+z-1}{z-1} \binom{n+1-z-i}{z} \frac{(q-1)^{n-z-i}}{z}.$$

We computed the difference between the base-q logarithms of the code size given in Theorem 3 and the upper bound established in Theorem 4. This difference, which reflects the redundancy gap between the construction and the bound, stabilizes around 1 as n increases. These observations indicate that the redundancy of the construction in Theorem 3 differs from the upper bound by at most a constant additive gap, which appears to converge to 1.

B. Using Minimal set of Labels of Length Two

In the previous section, we discussed the scenario where all available labels are used. Now, we turn our attention to a more intricate case: using the minimal number of labels required to achieve full capacity. This scenario is more challenging because deletions or insertions within the labeling sequence can manifest as various types of errors in the original sequence. To address this complexity, we aim to design error-correcting codes for the labeling sequences rather than for the original sequences.

Some of the results in this section will be derived by drawing a connection to known systematic single insertion or deletion error correcting codes. The construction of Varshamov-Tenengolts [31] can correct a single insertion or deletion in a binary sequence, and is defined as follows.

Definition 5. For $n > 0, a \in \Sigma_{n+1}$, let

$$\mathrm{VT}_a(n) \triangleq \left\{ \boldsymbol{x} \in \{0,1\}^n : \sum_{i=1}^{n} i x_i \equiv a \pmod{n+1} \right\}.$$

Tenengolts presented in [32] a non-binary single insertion or deletion correcting code. In order to present this construction, first we need to introduce the definition of a *signature vector*.

Definition 6. Let $\boldsymbol{x} \in \Sigma_q^n$ be a sequence over Σ_q, when $q > 2$. The *signature vector* of $\boldsymbol{x}$ is a binary sequence $s(\boldsymbol{x}) \in \Sigma_2^{n-1}$, where $s(\boldsymbol{x})_i = 1$ for $x_{i+1} \geq x_i$ and $s(\boldsymbol{x})_i = 0$ otherwise for $i \in [n-1]$.

Definition 7. For $q, n > 0, a \in \Sigma_n$ and $b \in \Sigma_q$, let

$$\mathrm{T}_{a,b}(n;q) \triangleq \left\{ \boldsymbol{x} \in \Sigma_q^n : s(\boldsymbol{x}) \in \mathrm{VT}_a(n-1), \sum_{i=1}^{n} x_i \equiv b \bmod q \right\}.$$

Let $q > 0$ and let $\mathcal{A}$ be a set of $\phi(q)$ labels of length two that achieves full capacity. The following theorem establishes the existence of a single deletion $\mathcal{A}$-labeling code of length n with a size of at least $\frac{q^n}{(\phi(q)+1)n}$.

Theorem 5. There exist $a \in \Sigma_n, b \in \Sigma_{\phi(q)+1}$ such that

$$\mathcal{C} = \{\boldsymbol{x} \in \Sigma_q^n : L_{\mathcal{A}}(\boldsymbol{x}) \in T_{a',b'}(n; \phi(q)+1)\}$$

is a single deletion $\mathcal{A}$-labeling code with redundancy of at most $\log_2(n) + \log_2(\phi(q)+1)$ bits.

Proof. Let $X \subseteq \Sigma_q^n$ be the set of sequences over Σ_q of length n and let $L_{\mathcal{A}}(X)$ be the set of valid labeling sequences. Denote $p = \phi(q) + 1$. For any labeling sequence $\boldsymbol{y} \in L_{\mathcal{A}}(X)$, there exists $a \in \Sigma_n, b \in \Sigma_p$ such that $\boldsymbol{y} \in T_{a,b}(n;p)$. Since there are pn options to choose a and b, from the pigeonhole principle, there exists $a' \in \Sigma_n, b' \in \Sigma_p$ for which

$$|T_{a',b'}(n;p)| \geq \frac{|L_{\mathcal{A}}(X)|}{pn} \geq \frac{q^n}{pn} = \frac{q^n}{q^{\log_q(pn)}}.$$

Since $T_{a',b'}(n;p)$ corrects a single deletion or insertion error, the proof follows from Theorem 2. □

Next, we will employ Tenengolts' construction [32] to design an explicit encoder. His approach involves appending the syndromes (as redundancy) to the data symbols and using two symbols to separate these parts. However, in our case, directly appending redundancy to the end of the sequence may not yield a valid labeling sequence. Hence, a more sophisticated construction is necessary. The construction presented here is demonstrated using DNA sequences with the label set $\mathcal{A} = \{AC, CA, GA, GC, GG, GT, TA, TC, TG, TT\}$. However, it can be adapted to other minimal label sets that achieve maximal labeling capacity and to different alphabet sizes q.

Let $\boldsymbol{x} \in \{A, C, G, T\}^k$ and let $\boldsymbol{y} = L_{\mathcal{A}}(\boldsymbol{x}) \in \Sigma_{11}^k$ be its labeling sequence. According to Tenengolts' construction, let $s(\boldsymbol{y})$ be the binary signature vector of $\boldsymbol{y}$. The term $a_{11\to4}$ represents the conversion of the value a from base 11 to base 4. For example, the number 7 in base 11 is equivalent to 13 in base 4, so $7_{11\to4} = 13$.

Construction 1. The systematic encoder E_1 is defined as follows:

$$E_1 : \{A, C, G, T\}^k \to \{A, C, G, T\}^n,$$
$$E_1(\boldsymbol{x}) = (\boldsymbol{x}, s, s, \beta(L_{\mathcal{A}}(\boldsymbol{x}))_{11\to4}, \gamma(L_{\mathcal{A}}(\boldsymbol{x}))_{11\to4}),$$

when
$$s = \begin{cases} T \text{ if } x_k = A, C, G \\ G \text{ if } x_k = T \end{cases},$$

$$\beta(\boldsymbol{y}) \triangleq \sum_{i=1}^{k} y_i \pmod{11}, \gamma(\boldsymbol{y}) \triangleq \sum_{i=1}^{k-1} i \cdot s(\boldsymbol{y})_i \pmod{k}.$$

Theorem 6. The code $\mathcal{C} = \{E_1(\boldsymbol{x}) : \boldsymbol{x} \in \Sigma_q^k\}$ is a single deletion $\mathcal{A}$-labeling code, with redundancy of $r = \lceil \log_2(k) \rceil + 8 < \lceil \log_2(n) \rceil + 8$ bits. Furthermore, there exist efficient encoding and decoding algorithms that can correct a single deletion or insertion error, both running in linear time.

Proof. In order to prove the theorem, we will provide a decoder. Let $\boldsymbol{y}' \in \Sigma_{11}^{n'}$ be the received sequence. If $n' = n$, then no deletion or insertion has occurred.

If $n' = n - 1$, a deletion occurred. In this case, if $y'_k \neq 5, 10$ (i.e., the kth symbol is not the label corresponding to GG or TT), then the data part of the sequence remains error-free. We return the DNA sequence corresponding to the first k symbols in $\boldsymbol{y}'$ as their labeling sequence. Otherwise, the deletion has occurred in the data part. As a result, we can recover the redundancy portion of the DNA sequence from the labeling sequence $\boldsymbol{y}'_{[k;n-k+1]}$. Let this DNA segment be denoted by (s, s, β, γ). We will convert β and γ to base 11. Since there is no error in this portion, this process allows us to determine $\beta(\boldsymbol{y})$ and $\gamma(\boldsymbol{y})$, where $\boldsymbol{y}$ is the original labeling sequence without deletion. Using these values, we can correct the deletion in $\boldsymbol{y}$, according to the Tenengolts' algorithm. Afterwards, we can reconstruct the original DNA sequence from the labeling sequence.

If $n' = n+1$, an insertion has occurred. In this case if $y'_{k+1} = 5, 10$, then the data part of the sequence is error-free. Otherwise, the insertion has occurred in the data part. Similarly to the case of deletion, we can recover the syndromes and correct the error in the data part.

The overall time complexity follows from the complexity of Tenengolts' algorithm, combined with the fact that all individual operations involved in encoding and decoding have either constant $O(1)$ or linear $O(n)$ complexity. □

IV. Single Substitution Labeling Codes

Substitution errors in the symbol-pair read channel model have been extensively studied [16]–[18], so we do not revisit the case of using all labels in this section. Rather, we focus on single substitution $\mathcal{A}$-labeling codes for a minimal set of length-two labels that achieve full capacity. First, we prove the existence of a code that corrects a single substitution error with redundancy of at most $\log_2(\phi(q)n+1)+\log_2(\phi(q)+1)$. In the DNA case, where $q = 4$, we aim to use a code whose codewords have labeling sequences that belong to the Hamming code over GF($\phi(4) + 1$) = GF(11). Since it is not guaranteed that there are enough valid labeling sequences in the Hamming code, we will use one of the Hamming code cosets. Note that the following results apply not only to the DNA case but also to any alphabet size q.

Let $\mathcal{A} \subseteq \Sigma_q^2$ be a minimal set of labels required for achieving full capacity, i.e., $|\mathcal{A}| = \phi(q)$. Denote the Hamming code over $GF(t)$ by $\mathcal{H}_t$ and the coset that is received by adding to any codeword the word $\boldsymbol{y} \in \Sigma_t^n$ by $\mathcal{H}_t + \boldsymbol{y}$. The proof of the following theorem is similar to the proof of Theorem 5.

Theorem 7. Let $p \geq \phi(q) + 1$ be the smallest integer such that there exists a field of size p. There exists $\boldsymbol{y} \in \Sigma_p^n$ such that

$$\mathcal{C} = \{\boldsymbol{x} \in \Sigma_q^n : L_{\mathcal{A}}(\boldsymbol{x}) \in \mathcal{H}_p + \boldsymbol{y}\}$$

is a single substitution $\mathcal{A}$-labeling code with redundancy of at most $\log_2((p-1)n + 1) + \log_2(p)$ bits.

Next, we will present an explicit systematic encoder for DNA sequences. Ideally, we aim to select DNA sequence, determine its labeling sequence, add redundancy using a Hamming code, and convert it back into DNA sequence. However, the challenge lies in the fact that the received sequence may not necessarily be a valid labeling sequence. Therefore, our construction will be designed with greater care. The construction is presented for the DNA case but can be adapted to construct codes for different alphabets in a similar manner. Let $\boldsymbol{x} \in \{A, C, G, T\}^k$ and let $\boldsymbol{y} = L_{\mathcal{A}}(\boldsymbol{x}) \in \Sigma_{11}^k$ be its labeling sequence.

Construction 2. The systematic encoder E_2 is defined as follows:

$$E_2 : \{A, C, G, T\}^k \to \{A, C, G, T\}^n,$$
$$E_2(\boldsymbol{x}) = (\boldsymbol{x}, par(\boldsymbol{y}), G, red(\boldsymbol{y})_{11\to4}),$$

when $par(\boldsymbol{y})$ is the label $\alpha_t \in \mathcal{A}$ such that $t = \sum_{i=1}^{k} y_i \pmod{11}$ and $red(\boldsymbol{y})$ is the redundancy of $\boldsymbol{y}$ according to p-ary Hamming code when $p = 11$.

Theorem 8. The code $\mathcal{C} = \{E_2(\boldsymbol{x}) : \boldsymbol{x} \in \Sigma_q^k\}$ is a single substitution $\mathcal{A}$-labeling code, with redundancy of $r = \lceil \log(k + 1) \rceil + 6 < \lceil \log(n + 1) \rceil + 6$. Furthermore, there exist efficient encoding and decoding algorithms, both running in linear time.

REFERENCES

[1] D. Hanania, D. Bar-Lev, Y. Nogin, Y. Shechtman, and E. Yaakobi, "On the capacity of DNA labeling," in *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2023, pp. 567–572.

[2] A. Moter and U. B. Göbel, "Fluorescence in situ hybridization (FISH) for direct visualization of microorganisms," *Journal of microbiological methods*, vol. 41, no. 2, pp. 85–112, 2000.

[3] B. Chen, W. Zou, H. Xu, Y. Liang, and B. Huang, "Efficient labeling and imaging of protein-coding genes in living cells using CRISPR-tag," *Nature communications*, vol. 9, no. 1, p. 5065, 2018.

[4] D. Gruszka, J. Jeffet, S. Margalit, Y. Michaeli, and Y. Ebenstein, "Single-molecule optical genome mapping in nanochannels: Multidisciplinarity at the nanoscale," *Essays in Biochemistry*, vol. 65, no. 1, pp. 51–66, 2021.

[5] H. Ma, A. Naseri, P. Reyes-Gutierrez, S. A. Wolfe, S. Zhang, and T. Pederson, "Multicolor crispr labeling of chromosomal loci in human cells," *Proceedings of the National Academy of Sciences*, vol. 112, no. 10, pp. 3002–3007, 2015.

[6] J. Deen, C. Vranken, V. Leen, R. K. Neely, K. P. Janssen, and J. Hofkens, "Methyltransferase-directed labeling of biomolecules and its applications," *Angewandte Chemie International Edition*, vol. 56, no. 19, pp. 5182–5200, 2017.

[7] B. Canard and R. S. Sarfati, "DNA polymerase fluorescent substrates with reversible 3'-tags," *Gene*, vol. 148, no. 1, pp. 1–6, 1994.

[8] M. F. Fraga and M. Esteller, "DNA methylation: A profile of methods and applications," *BioTechniques*, vol. 33, no. 3, pp. 632–649, 2002.

[9] H. Frickmann, A. E. Zautner, A. Moter, J. Kikhney, R. M. Hagen, H. Stender, and S. Poppert, "Fluorescence in situ hybridization (fish) in the microbiological diagnostic routine laboratory: a review," *Critical Reviews in Microbiology*, vol. 43, no. 3, pp. 263–293, 2017, pMID: 28129707.

[10] M. Levy-Sakin and Y. Ebenstein, "Beyond sequencing: Optical mapping of DNA in the age of nanotechnology and nanoscopy," *Current opinion in biotechnology*, vol. 24, no. 4, pp. 690–698, 2013.

[11] V. Müller and F. Westerlund, "Optical DNA mapping in nanofluidic devices: Principles and applications," *Lab on a Chip*, vol. 17, no. 4, pp. 579–590, 2017.

[12] S. Ohayon, A. Girsault, M. Nasser, S. Shen-Orr, and A. Meller, "Simulation of single-protein nanopore sensing shows feasibility for whole-proteome identification," *PLoS computational biology*, vol. 15, no. 5, p. e1007067, 2019.

[13] J. A. Alfaro, P. Bohländer, M. Dai, M. Filius, C. J. Howard, X. F. Van Kooten, S. Ohayon, A. Pomorski, S. Schmid, A. Aksimentiev *et al.*, "The emerging landscape of single-molecule protein sequencing technologies," *Nature methods*, vol. 18, no. 6, pp. 604–617, 2021.

[14] D. Hanania, D. Bar-Lev, Y. Nogin, Y. Shechtman, and E. Yaakobi, "On the capacity of DNA labeling," *arXiv preprint arXiv:2305.07992*, 2024.

[15] C. Hofmeister, A. Gruica, D. Hanania, R. Bitar, and E. Yaakobi, "Achieving DNA labeling capacity with minimum labels through extremal de bruijn subgraphs," in *2024 IEEE International Symposium on Information Theory (ISIT)*, 2024, pp. 452–457.

[16] Y. Cassuto and M. Blaum, "Codes for symbol-pair read channels," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 8011–8020, 2011.

[17] H. Q. Dinh, B. T. Nguyen, A. K. Singh, and S. Sriboonchitta, "On the symbol-pair distance of repeated-root constacyclic codes of prime power lengths," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2417–2430, 2017.

[18] O. Elishco, R. Gabrys, and E. Yaakobi, "Bounds and constructions of codes over symbol-pair read channels," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1385–1395, 2019.

[19] Y. M. Chee and V. K. Vu, "Codes correcting synchronization errors for symbol-pair read channels," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 746–750.

[20] E. Yaakobi, J. Bruck, and P. H. Siegel, "Constructions and decoding of cyclic codes over b-symbol read channels," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1541–1551, 2016.

[21] B. Ding, T. Zhang, and G. Ge, "Maximum distance separable codes for b-symbol read channels," *Finite Fields and Their Applications*, vol. 49, pp. 180–197, 2018.

[22] A. Sharma and T. Sidana, "On b-symbol distances of repeated-root constacyclic codes," *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7848–7867, 2019.

[23] L. Dolecek and V. Anantharam, "Repetition error correcting sets: Explicit constructions and prefixing methods," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 2120–2146, 2010.

[24] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Proceedings of the Soviet physics doklady*, 1966.

[25] S. Jain, F. F. Hassanzadeh, M. Schwartz, and J. Bruck, "Duplication-correcting codes for data storage in the DNA of living organisms," *IEEE Transactions on Information Theory*, vol. 63, no. 8, pp. 4996–5010, 2017.

[26] L. G. Tallini, N. Alqwaifly, and B. Bose, "On deletion/insertion of zeros and asymmetric error control codes," in *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2019, pp. 2384–2388.

[27] A. Fazeli, A. Vardy, and E. Yaakobi, "Generalized sphere packing bound," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2313–2334, 2015.

[28] A. A. Kulkarni and N. Kiyavash, "Nonasymptotic upper bounds for deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5115–5130, 2013.

[29] I. Smagloy, L. Welter, A. Wachter-Zeh, and E. Yaakobi, "Single-deletion single-substitution correcting codes," *IEEE Transactions on Information Theory*, 2023.

[30] R. Gabrys, E. Yaakobi, and L. Dolecek, "Correcting grain-errors in magnetic media," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2256–2272, 2015.

[31] R. Varshamov and G. Tenengolts, "Codes which correct single asymmetric errors (in russian)," *Automatika i Telemkhanika*, vol. 161, no. 3, pp. 288–292, 1965.

[32] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion (corresp.)," *IEEE Transactions on Information Theory*, vol. 30, no. 5, pp. 766–769, 1984.