

Correcting a Single Deletion in Reads from a Nanopore Sequencer

Anisha Banerjee*, Yonatan Yehezkeally*, Antonia Wachter-Zeh*, and Eitan Yaakobi†

*Institute for Communications Engineering, Technical University of Munich (TUM), Munich, Germany

†Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel

Email: anisha.banerjee@tum.de, yonatan.yehezkeally@tum.de, antonia.wachter-zeh@tum.de, yaakobi@cs.technion.ac.il

Abstract—Owing to its several merits over other DNA sequencing technologies, nanopore sequencers hold an immense potential to revolutionize the efficiency of DNA storage systems. However, their higher error rates necessitate further research to devise practical and efficient coding schemes that would allow accurate retrieval of the data stored. Our work takes a step in this direction by adopting a simplified model of the nanopore sequencer inspired by Mao *et al.*, which incorporates some of its physical aspects. This channel model can be viewed as a sliding window of length ℓ that passes over the incoming input sequence and produces the Hamming weight of the enclosed ℓ bits, while shifting by one position at each time step. The resulting $(\ell+1)$ -ary vector, referred to as the ℓ -read vector, is susceptible to deletion errors due to imperfections inherent in the sequencing process. We establish that at least $\log n - \ell$ bits of redundancy are needed to correct a single deletion. An error-correcting code that is optimal up to an additive constant, is also proposed. Furthermore, we find that for $\ell \geq 2$, reconstruction from two distinct noisy ℓ -read vectors can be accomplished without any redundancy, and provide a suitable reconstruction algorithm to this effect.

I. INTRODUCTION

Our ever-increasing data storage requirements have prompted extensive research into DNA storage, as it promises high density and unmatched longevity. While there continue to be significant efforts to improve upon existing synthesis and sequencing technologies, nanopore sequencing holds particular appeal due to better portability, ability to read longer DNA strands, and real-time analysis [1]–[3]. The sequencing operation involves the transmigration of a DNA fragment through a microscopic pore in a lipid membrane, across which a voltage difference exists. The nucleotides in the pore at a given time instant, influence the variations in the ionic current, which are measured and fed to a basecaller that predicts the nucleotides in the examined DNA strand. Despite its strengths, certain physical aspects of the nanopore sequencer lead to various distortions in the final readout. For instance, the variations

This material is based upon work supported by the National Science Foundation under Grant No. CCF 2212437. This work has also received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (Grant agreement No. 801434). It was also funded by the European Union (ERC, DNASStorage, 865630). Additionally, this project was funded by the European Union (DiDAX, 101115134). Views and opinions expressed are however those of the author(s) 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 work of Yonatan Yehezkeally was supported by the Alexander von Humboldt Foundation under a Carl Friedrich von Siemens Post-Doctoral Research Fellowship.

in the measured current are governed by multiple nucleotides instead of just one due to the depth of the pore, thus hinting at the presence of intersymbol interference (ISI). Moreover, the DNA strand often passes through the pore unevenly, i.e., a few nucleotides may be skipped, or some backtracking may occur. This naturally implies deletions and duplications in the final readout, respectively.

Prior work in this area was largely aimed at either developing faithful mathematical models for the sequencer or designing error-correcting codes that incorporate such models to correct errors in the readouts efficiently. For instance, the authors of [4] introduced a channel model that incorporates ISI, deletions, and measurement noise. Upper bounds on channel capacity were also established. The work in [5] adopted a more deterministic model, devised an algorithm to compute the capacity of the same, and also suggested efficient coding schemes. A finite-state semi-Markov channel (FSMC)-based model, introduced more recently in [6], encapsulates the effects of ISI, duplications, and noisy measurements that affect the final sequencing output. Another promising line of work aims to design codes such that the current reading of the constituent DNA sequences, as produced by a nanopore sequencer, can be decoded accurately with high probability despite sample duplications and amplitude noise [7]–[9]. In [10], a specific model inspired by [4], [5] was considered, and an optimal single-substitution-correcting code was presented.

This work endeavors to extend [10] by designing efficient deletion-correcting codes for nanopore sequencers. To this end, we use the channel model employed in [10]. This model also resembles the transverse-read channel [11], [12], which is relevant to racetrack memories. More specifically, nanopore sequencing is interpreted as a concatenation of three channels, as illustrated in Fig. 1. The ISI component, parameterized by ℓ , signifies how the measured current depends on the ℓ consecutive nucleotides in the pore at any given time. This stage may be viewed as a sliding window of length ℓ passing over an input sequence and shifting by a single position after each time step, producing a sequence of ℓ -mers, i.e., strings of ℓ symbols. Subsequently, a discrete memoryless channel converts each of the ℓ -mers into a discrete voltage level based on a deterministic function, in our analysis the Hamming weight. In the end, the deletion channel accounts for the effect of skipping forward by corrupting the sequence of discrete voltage levels with deletions.

We now state the problem more formally. For an input

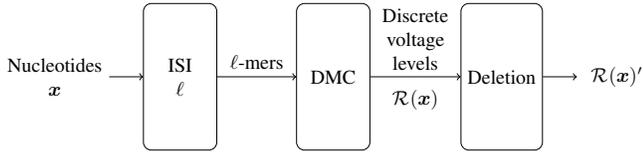


Figure 1. Simplified model of a nanopore sequencer

$\mathbf{x} \in \Sigma_2^n$, let $\mathcal{R}_\ell(\mathbf{x})$ represent the deletion-free channel output (Definition 1). Thus, we are interested in codes that correct t deletions in $\mathcal{R}_\ell(\mathbf{x})$ as opposed to \mathbf{x} itself, to guarantee the unique recovery of \mathbf{x} despite ISI, followed by at most t deletions.

To summarize the main contributions of this work, we establish a lower bound on the redundancy required by a code that corrects a single deletion in ℓ -read vectors, and suggest an instantiation of the same whose redundancy is optimal up to an additive constant. Since nanopore sequencers tend to produce multiple erroneous reads for each input strand, we also examine how leveraging this feature might help achieve a lower redundancy requirement. To this end, we find that for any $\mathbf{x} \in \Sigma_2^n$ and $\ell \geq 2$, two distinct noisy channel outputs that arise from the same input \mathbf{x} , suffice to recover \mathbf{x} uniquely. A suitable reconstruction algorithm is also stated.

II. PRELIMINARIES

A. Notations and Terminology

In the following, we let Σ_q indicate the q -ary alphabet $\{0, 1, \dots, q-1\}$. Additionally, $[n]$ is used to denote the set $\{1, 2, \dots, n\}$. Element-wise modulo operation on a vector, say $\mathbf{y} \in \Sigma_q^n$, is represented as

$$\mathbf{y} \bmod a \triangleq (y_1 \bmod a, y_2 \bmod a, \dots, y_n \bmod a).$$

For any vector $\mathbf{x} = (x_1, \dots, x_n)$, we refer to its substring $(x_i, x_{i+1}, \dots, x_j)$ as \mathbf{x}_i^j . The Hamming weight of a vector \mathbf{x} is denoted by $\text{wt}(\mathbf{x})$, while the number of runs in \mathbf{x} , which are of length greater than or equal to some $a \geq 1$, is represented as $\rho_{\geq a}(\mathbf{x})$. We designate the number of all runs, i.e., when $a = 1$, by $\rho(\mathbf{x})$. We also extensively use the Hamming distance, which is defined for any two vectors $\mathbf{x}, \mathbf{y} \in \Sigma_q^n$ as

$$d_H(\mathbf{x}, \mathbf{y}) = |\{i : i \in [n], x_i \neq y_i\}|.$$

We focus on the case of $q = 2$ and in this framework, the channel output is defined as follows.

Definition 1 The ℓ -read vector of any $\mathbf{x} \in \Sigma_2^n$ is of length $n + \ell - 1$ over $\Sigma_{\ell+1}$, and is denoted by

$$\mathcal{R}_\ell(\mathbf{x}) \triangleq (\text{wt}(\mathbf{x}_{2-\ell}^1), \text{wt}(\mathbf{x}_{3-\ell}^2), \dots, \text{wt}(\mathbf{x}_n^{n+\ell-1})),$$

where for ease of notation we simply let $x_i = 0$ for any $i \notin [n]$ (i.e., when the above definition includes indices which are either negative or greater than n).

Additionally, the i -th element of $\mathcal{R}_\ell(\mathbf{x})$ is denoted by $\mathcal{R}_\ell(\mathbf{x})_i$; that is, $\mathcal{R}_\ell(\mathbf{x})_i = \text{wt}(\mathbf{x}_{i-\ell+1}^i)$. When clear from the context, ℓ will be removed from the preceding notations.

Example 2 The 3-read vector of $\mathbf{x} = (1, 0, 1, 1, 0, 0)$ is given by $\mathcal{R}(\mathbf{x}) = (1, 1, 2, 2, 2, 1, 0, 0)$. Its third element is $\mathcal{R}(\mathbf{x})_3 = 2$.

A similar model was investigated in [11], wherein the output sequence, termed as the transverse-read vector, is a substring of the ℓ -read vector as defined here, for certain parameter choices. The information limit of this transverse-read channel was computed for various parameters, and several coding schemes enabling exact recovery were presented. Some error-correcting codes for the case of $\ell = 2$ were also suggested.

Next, to facilitate our analysis, some key properties of ℓ -read vectors are stated below.

Proposition 3 ([13]) P1 For any $\ell \geq 1$ and $\mathbf{x} \in \Sigma_2^n$, it holds that $\sum_{i=1}^{n+\ell-1} \mathcal{R}(\mathbf{x})_i = \ell \cdot \text{wt}(\mathbf{x})$, where the sum is performed over integers.

P2 For any $\ell \geq 1$ and $\mathbf{x} \in \Sigma_2^n$, it holds that $|\mathcal{R}(\mathbf{x})_{j+1} - \mathcal{R}(\mathbf{x})_j| \leq 1$ for all $j \in [n + \ell - 2]$.

P3 For any $\mathbf{x} \in \Sigma_2^n$, \mathbf{x} can be uniquely and efficiently determined from the first or last n elements of $\mathcal{R}(\mathbf{x}) \bmod 2$.

The preceding definitions can be extended to the non-binary alphabet by replacing the notion of Hamming weights with compositions, as done in [10].

B. Error Model

To suitably define what constitutes an error-correcting construction in our framework, we first let $D(\mathbf{u})$ refer to the set of all vectors of length $n-1$, that can be obtained by deleting one symbol from $\mathbf{u} \in \Sigma^n$, for any alphabet Σ . Naturally, we are interested in $D(\mathcal{R}(\mathbf{x}))$ for some $\mathbf{x} \in \Sigma_2^n$. A code that corrects a single deletion in $\mathcal{R}(\mathbf{x})$ can thus be defined as follows.

Definition 4 For $n \geq \ell$, a code $\mathcal{C} \subseteq \Sigma_2^n$ is said to be a *single-deletion ℓ -read code* if for any two distinct $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, it holds that $D(\mathcal{R}_\ell(\mathbf{x})) \cap D(\mathcal{R}_\ell(\mathbf{y})) = \emptyset$.

III. CORRECTING A SINGLE DELETION

A useful consequence of the inherent characteristics of ℓ -read vectors, summarized in Proposition 3, is that certain deletions can be corrected immediately without any redundancy, as shown in the next lemma.

Lemma 5 Let \mathcal{R}' arise from a single deletion on $\mathcal{R}(\mathbf{x})$ for some $\mathbf{x} \in \Sigma_2^n$. If there exists some $i \in [n + \ell - 3]$ such that $|\mathcal{R}'_{i+1} - \mathcal{R}'_i| > 1$, then $\mathcal{R}(\mathbf{x})$ (and thereby also \mathbf{x}) can be readily recovered.

Proof: From P1, we infer that the existence of an $i \in [n + \ell - 3]$ such that $|\mathcal{R}'_{i+1} - \mathcal{R}'_i| > 1$ unambiguously reveals the error location. Note that $|\mathcal{R}'_{i+1} - \mathcal{R}'_i| \leq 2$. Assuming $\mathcal{R}'_{i+1} - \mathcal{R}'_i = 2$, we observe that the only \mathbf{x} for which $\mathcal{R}' \in D(\mathcal{R}(\mathbf{x}))$, bears the following ℓ -read vector.

$$\mathcal{R}(\mathbf{x}) = (\mathcal{R}'_1, \dots, \mathcal{R}'_i, \mathcal{R}'_i + 1, \mathcal{R}'_{i+1}, \dots, \mathcal{R}'_{n+\ell-2}).$$

For the case of $\mathcal{R}'_{i+1} - \mathcal{R}'_i = -2$, the argument works similarly. Once $\mathcal{R}(\mathbf{x})$ is known, \mathbf{x} can also uniquely recovered as suggested by P3. ■

A. Upper Bound on the Size of Codes

This section establishes an upper bound on the cardinality of a single-deletion ℓ -read code. We do so by limiting our focus to a subset of deletion patterns on an ℓ -read vector, say $\mathcal{R}(\mathbf{x})$, which bear an intriguing connection with sticky deletions [14]–[17] on its respective binary vector, \mathbf{x} . We consider a specific variant of a sticky deletion, defined below.

Definition 6 An r -sticky deletion, for $r \geq 1$, is a deletion in a run of length at least r , in \mathbf{x} .

The error ball of a single r -sticky deletion for any $\mathbf{u} \in \Sigma_2^n$ is represented by

$$\text{DS}(\mathbf{u}; r) \triangleq \left\{ (u_1 u_2 \dots u_{i-1} u_{i+1} \dots u_n) : i \in [n - r + 1], \right. \\ \left. u_i = \dots = u_{i+r-1} \right\}.$$

Naturally, $\text{DS}(\mathbf{u}; 1) = \text{D}(\mathbf{u})$ and $|\text{DS}(\mathbf{u}; r)| = \rho_{\geq r}(\mathbf{u})$. As will be established shortly, such sticky deletions translate to specific deletion events on the respective ℓ -read vectors, which we define formally as follows.

Definition 7 A k -restricted deletion is a deletion that only deletes a symbol if it equals 0 or k .

The error ball of a k -restricted deletion for any $\mathbf{u} \in \Sigma_q^n$ can be expressed as

$$\text{DR}(\mathbf{u}; k) \triangleq \left\{ (u_1 u_2 \dots u_{i-1} u_{i+1} \dots u_n) : i \in [n - \ell + 1], \right. \\ \left. u_i \in \{0, k\} \right\}.$$

The upcoming lemma explains the link between ℓ -sticky deletions in binary vectors and ℓ -restricted deletions in their respective ℓ -read vectors.

Lemma 8 For any $\mathbf{x} \in \Sigma_2^n$ with $\rho_{\geq \ell}(0^{\ell-1} \mathbf{x} 0^{\ell-1}) \geq 1$, it holds that

$$\text{DR}(\mathcal{R}(\mathbf{x}); \ell) = \{ \mathcal{R}(\mathbf{y}) : 0^{\ell-1} \mathbf{y} 0^{\ell-1} \in \text{DS}(0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell) \}.$$

Proof: Consider an $\mathcal{R}' \in \text{DR}(\mathcal{R}(\mathbf{x}); \ell)$, that arises from a deletion in $\mathcal{R}(\mathbf{x})$ at index i , and assume that $\mathcal{R}(\mathbf{x})_i = 0$, or equivalently, $\mathbf{x}_{i-\ell+1}^i = 0^\ell$. For the case of $\mathcal{R}(\mathbf{x})_i = \ell$, the proof follows similarly.

If $\ell + 1 \leq i \leq n$, we note that for $\mathbf{y} = (\mathbf{x}_1^{i-\ell} 0^{\ell-1} \mathbf{x}_{i+1}^n) \in \text{DS}(\mathbf{x}; \ell)$, we get $\mathcal{R}' = \mathcal{R}(\mathbf{y})$, since $\mathbf{x}_1^{i-1} = \mathbf{y}_1^{i-1}$ ensures that $\mathcal{R}(\mathbf{y})_j = \mathcal{R}(\mathbf{x})_j$ for $j \in [i-1]$, while for $i \leq j \leq n + \ell - 2$, we have $\mathcal{R}(\mathbf{y})_j = \text{wt}(\mathbf{y}_{j-\ell+1}^j) = \text{wt}(\mathbf{x}_{j-\ell+2}^{j+1}) = \mathcal{R}(\mathbf{x})_{j+1}$.

Next, consider $i \in [\ell]$. Since $\mathbf{x}_1^i = 0^i$, it follows that for $\mathbf{y} = (0^{i-1} \mathbf{x}_{i+1}^n)$, $\mathcal{R}' = \mathcal{R}(\mathbf{y})$. Observe that $(0^{\ell-1} \mathbf{y}) \in \text{DS}(0^{\ell-1} \mathbf{x}; \ell)$.

Finally when $i \geq n+1$, we can similarly argue that for $\mathbf{y} = (\mathbf{x}_1^{i-\ell} 0^{\ell-(i-n)-1})$, which upholds $(\mathbf{y} 0^{\ell-1}) \in \text{DS}(\mathbf{x} 0^{\ell-1}; \ell)$, $\mathcal{R}' = \mathcal{R}(\mathbf{y})$. In all of the aforementioned cases $0^{\ell-1} \mathbf{y} 0^{\ell-1} \in \text{DS}(0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell)$ and since $\mathcal{R}(\mathbf{y})$ is fixed, \mathbf{y} is clearly unique.

To prove the other direction, assume some $\mathbf{x} \in \Sigma_2^n$ and $\mathbf{y} \in \Sigma_2^{n-1}$ such that $\hat{\mathbf{y}} = 0^{\ell-1} \mathbf{y} 0^{\ell-1} \in \text{DS}(\hat{\mathbf{x}} = 0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell)$. In particular, let the run (of length $\geq \ell$) in $\hat{\mathbf{x}}$ that suffers the ℓ -sticky deletion start at index j , i.e., $\hat{\mathbf{x}}_j^{j+\ell-1} = a^\ell$ for some $a \in \Sigma_2$ and $\hat{\mathbf{y}} = \hat{\mathbf{x}}_1^{j-1} \hat{\mathbf{x}}_{j+1}^{n+2\ell-1}$. If $j = 1$, then $\hat{\mathbf{x}}_1^\ell = \mathbf{x}_{-\ell+2}^1$,

$\mathcal{R}(\mathbf{x})_1 = 0$ and $\hat{\mathbf{y}} = 0^{\ell-1} \mathbf{y} 0^{\ell-1} = 0^{\ell-1} \mathbf{x}_2^n 0^{\ell-1}$ directly imply that for all $p \in [n + \ell - 2]$, $\mathcal{R}(\mathbf{y})_p = \mathcal{R}(\mathbf{x})_{p+1}$ and thereby $\mathcal{R}(\mathbf{y}) \in \text{DR}(\mathcal{R}(\mathbf{x}); \ell)$.

For the remaining cases, we have $j \in \{\ell, \dots, n + \ell - 1\}$. Note that $\mathbf{y}_1^{j-\ell} = \hat{\mathbf{y}}_1^{j-1} = \hat{\mathbf{x}}_1^{j-1} = \mathbf{x}_1^{j-\ell}$ in combination with $\mathbf{y}_{j-\ell+1}^{j-1} = \mathbf{x}_{j-\ell+1}^{j-1} = a^{\ell-1}$ establishes that for all $p \in [j-1]$, $\mathcal{R}(\mathbf{y})_p = \mathcal{R}(\mathbf{x})_p$. Evidently, $\mathcal{R}(\mathbf{x})_j = \ell \cdot a \in \{0, \ell\}$. Next, from $\mathbf{y}_{j-\ell+1}^{n-1} 0^{\ell-1} = \hat{\mathbf{y}}_j^{n+2\ell-3} = \hat{\mathbf{x}}_{j+1}^{n+2\ell-2} = \mathbf{x}_{j-\ell+2}^n 0^{\ell-1}$, observe that for all $p \in \{j, \dots, n + \ell - 2\}$, $\mathcal{R}(\mathbf{y})_p = \mathcal{R}(\mathbf{x})_{p+1}$. Thus $\mathcal{R}(\mathbf{y}) \in \text{DR}(\mathcal{R}(\mathbf{x}); \ell)$ and the statement follows. ■

Example 9 Recall $\mathbf{x} = (1, 0, 1, 1, 0, 0)$ from Example 2, with the 3-read vector $\mathcal{R}(\mathbf{x}) = (1, 1, 2, 2, 2, 1, 0, 0)$. Consider $\mathcal{R}' = (1, 1, 2, 2, 2, 1, 0)$ that arises from deleting a 0 in $\mathcal{R}(\mathbf{x})$, i.e., $\mathcal{R}' \in \text{DR}(\mathcal{R}(\mathbf{x}); \ell)$ where $\ell = 3$. Observe that for $\mathbf{y} = (1, 0, 1, 1, 0)$, we have $0^{\ell-1} \mathbf{y} 0^{\ell-1} \in \text{DS}(0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell)$ and $\mathcal{R}(\mathbf{y}) = \mathcal{R}'$.

Thus for any $\mathbf{x} \in \Sigma_2^n$ with $\rho_{\geq \ell}(0^{\ell-1} \mathbf{x} 0^{\ell-1}) \geq 1$, it holds that $|\text{DR}(\mathcal{R}(\mathbf{x}); \ell)| = |\text{DS}(0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell)| = \rho_{\geq \ell}(0^{\ell-1} \mathbf{x} 0^{\ell-1}) \geq \rho_{\geq \ell}(\mathbf{x})$. The following corollary summarizes our strategy for bounding the size of any single-deletion-correcting read code.

Corollary 10 Any single-deletion ℓ -read code is also a single- ℓ -sticky-deletion-correcting code.

Proof: Based on Lemma 8, $\{ \mathcal{R}(\mathbf{y}) : \mathbf{y} \in \text{DS}(\mathbf{x}; \ell) \} \subseteq \{ \mathcal{R}(\mathbf{y}) : 0^{\ell-1} \mathbf{y} 0^{\ell-1} \in \text{DS}(0^{\ell-1} \mathbf{x} 0^{\ell-1}; \ell) \} = \text{DR}(\mathcal{R}(\mathbf{x}); \ell) \subseteq \text{D}(\mathcal{R}(\mathbf{x}))$ for any $\mathbf{x} \in \Sigma_2^n$. Then, for any two distinct codewords \mathbf{c}, \mathbf{c}' of a single-deletion ℓ -read code we necessarily have $\{ \mathcal{R}(\mathbf{y}) : \mathbf{y} \in \text{DS}(\mathbf{c}; \ell) \} \cap \{ \mathcal{R}(\mathbf{y}) : \mathbf{y} \in \text{DS}(\mathbf{c}'; \ell) \} = \emptyset$, which in particular implies $\text{DS}(\mathbf{c}; \ell) \cap \text{DS}(\mathbf{c}'; \ell) = \emptyset$, i.e., the code is also single- ℓ -sticky-deletion-correcting. ■

Consequently, the cardinality of a single-deletion ℓ -read code is bounded from above by the size of the largest code that corrects a single ℓ -sticky deletion.

Now to establish an upper bound on the cardinality of a code that corrects a single ℓ -sticky deletion. We first note that for a randomly chosen $\mathbf{x} \in \Sigma_2^n$, the expected value of $\rho_{\geq a}(\mathbf{x})$ is given by $2^{-a}(n - a + 2)$. Also note that a change in any x_i may increase or decrease $\rho_{\geq a}(\mathbf{x})$ by at most 1. It then follows from McDiarmid's inequality [18] that for any $\epsilon > 0$

$$2^{-n} \left| \left\{ \mathbf{x} \in \Sigma_2^n : \rho_{\geq a}(\mathbf{x}) < \frac{n - a + 2}{2^a} - \epsilon n \right\} \right| \leq \exp(-2\epsilon^2 n).$$

By choosing $\epsilon = 2^{-a-1}$, we obtain

$$\left| \left\{ \mathbf{x} \in \Sigma_2^n : \rho_{\geq a}(\mathbf{x}) < \frac{n - 2a + 4}{2^{a+1}} \right\} \right| \leq 2^n \exp\left(-\frac{n}{2^{2a+1}}\right).$$

In the following analysis, assume that $\ell \geq 2$. We proceed to apply the generalized sphere-packing bound [19] to obtain an upper bound on the cardinality of the largest length- n single ℓ -sticky-deletion-correcting code, which we denote by $A(n, \ell)$.

Theorem 11 [19] For an error channel that outputs any $\mathbf{y} \in B(\mathbf{x})$ given an input $\mathbf{x} \in \mathcal{X}$, construct a hypergraph $\mathcal{H}(Y, \mathcal{E})$, whereby the vertex set Y comprises all possible channel outputs while the hyperedge set $\mathcal{E} = \{B(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$. Assume an assignment of weights $w_{\mathbf{y}}$ for each $\mathbf{y} \in Y$,

satisfying $w_{\mathbf{y}} \geq 0$ for all $\mathbf{y} \in Y$, and $\sum_{\mathbf{y} \in E} w_{\mathbf{y}} \geq 1$ for all $E \in \mathcal{E}$. Then, the size of any error-correcting code for this channel is upper bounded by $\sum_{\mathbf{y} \in Y} w_{\mathbf{y}}$.

The hypergraph for our single ℓ -sticky deletion channel, say $\mathcal{H}(Y, \mathcal{E})$, constitutes the vertex set $Y = \Sigma_2^{n-1}$ and the hyperedge set $\mathcal{E} = \{\text{DS}(\mathbf{x}; \ell) : \mathbf{x} \in \Sigma_2^n, \rho_{\geq \ell}(\mathbf{x}) \geq 1\}$. The restriction on $\rho_{\geq \ell}(\mathbf{x})$ simply serves to exclude the empty set. We choose the following weight assignment for the vertices in Y . For each $\mathbf{y} \in Y$,

$$w_{\mathbf{y}} \triangleq \begin{cases} \frac{1}{\rho_{\geq \ell}(\mathbf{y})} & \text{if } \rho_{\geq \ell}(\mathbf{y}) \geq 1, \\ 1 & \text{if } \rho_{\geq \ell}(\mathbf{y}) = 0. \end{cases}$$

While this clearly fulfills the positivity criterion of Theorem 11, observe that for all $\mathbf{x} \in \Sigma_2^n$ that fulfill either $\rho_{\geq \ell}(\mathbf{x}) \geq 2$ or $\rho_{\geq \ell}(\mathbf{x}) = \rho_{\geq \ell+1}(\mathbf{x}) = 1$, it holds that $\rho_{\geq \ell}(\mathbf{y}) \geq 1$ for each $\mathbf{y} \in \text{DS}(\mathbf{x}; \ell)$, since a single ℓ -sticky deletion only shortens a single run by one. Thus $\sum_{\mathbf{y} \in \text{DS}(\mathbf{x}; \ell)} w_{\mathbf{y}} = \sum_{\mathbf{y} \in \text{DS}(\mathbf{x}; \ell)} \frac{1}{\rho_{\geq \ell}(\mathbf{y})} \geq \sum_{\mathbf{y} \in \text{DS}(\mathbf{x}; \ell)} \frac{1}{\rho_{\geq \ell}(\mathbf{x})} = 1$. Similarly, when $\rho_{\geq \ell}(\mathbf{x}) = 1$ while $\rho_{\geq \ell+1}(\mathbf{x}) = 0$ (i.e., \mathbf{x} has a unique longest run, of length ℓ), it follows that $\text{DS}(\mathbf{x}; \ell) = \{\mathbf{y}\}$ wherein $\rho_{\geq \ell}(\mathbf{y}) = 0$, leading us to $\sum_{\mathbf{y} \in \text{DS}(\mathbf{x}; \ell)} w_{\mathbf{y}} = 1$.

We now derive an upper bound on the cardinality $A(n, \ell)$ of the largest length- n single ℓ -sticky-deletion-correcting code.

$$\begin{aligned} A(n, \ell) &\leq \sum_{\mathbf{y} \in Y} w_{\mathbf{y}} = |\{\mathbf{y} \in \Sigma_2^{n-1} : \rho_{\geq \ell}(\mathbf{y}) = 0\}| + \\ &\quad \sum_{i=1}^{\lfloor \frac{n-1}{\ell} \rfloor} \frac{1}{i} |\{\mathbf{y} \in \Sigma_2^{n-1} : \rho_{\geq \ell}(\mathbf{y}) = i\}| \\ &< \left| \left\{ \mathbf{y} \in \Sigma_2^{n-1} : \rho_{\geq \ell}(\mathbf{y}) < \frac{n-2\ell+3}{2^{\ell+1}} \right\} \right| + \\ &\quad \frac{1}{\lfloor \frac{n-2\ell+3}{2^{\ell+1}} \rfloor} \sum_{i=\lceil \frac{n-2\ell+3}{2^{\ell+1}} \rceil}^{\lfloor \frac{n-1}{\ell} \rfloor} |\{\mathbf{y} \in \Sigma_2^{n-1} : \rho_{\geq \ell}(\mathbf{y}) = i\}| \\ &\leq 2^{n-1} \exp\left(-\frac{n-1}{2^{2\ell+1}}\right) + \frac{2^{n+\ell}}{n-2\ell+3} \\ &\leq \frac{2^{n+\ell-1}}{n} \left(2^{-\ell} n \exp\left(-\frac{n-1}{2^{2\ell+1}}\right) + \frac{2n}{n-2\ell} \right). \end{aligned}$$

We merge this with Corollary 10 to get the following bound.

Theorem 12 *The redundancy of a single-deletion ℓ -read code is bounded from below by*

$$\log n - \ell + 1 - \log \left(\frac{2}{1-2\ell/n} + 2^{-\ell} n \exp\left(-\frac{n-1}{2^{2\ell+1}}\right) \right).$$

Remark: As mentioned earlier, for any $\mathbf{x} \in \Sigma_2^n$, the transverse-read vector as defined in [11], is a substring of $\mathcal{R}_{\ell}(\mathbf{x})$ for certain choices of parameters. Consequently in these cases, Lemma 8 can be suitably modified to help establish a similar redundancy bound for the transverse-read channel.

B. Single-Deletion ℓ -Read Codes

It is implied by P3 that protecting the first n entries of the ℓ -read vector taken modulo 2 suffices to exactly recover the corresponding length- n binary vector. This naturally leads us to the following single-deletion ℓ -read code.

Construction A

$$\mathcal{C}(n, \ell, a) = \{\mathbf{x} \in \Sigma_2^n : \sum_{i=1}^n i(\mathcal{R}(\mathbf{x})_i \bmod 2) = a \pmod{n+1}\},$$

where $a \in \Sigma_{n+1}$. \square

Note that for every $\mathbf{u} \in \Sigma_2^n$, there exists $\mathbf{x} \in \Sigma_2^n$ such that $\mathbf{u} = (\mathcal{R}(\mathbf{x})_1, \dots, \mathcal{R}(\mathbf{x})_n) \bmod 2$. Thus by the pigeonhole principle, there exists $a \in \Sigma_{n+1}$ for which the redundancy required by the preceding construction is $\log_2(n+1)$ bits, implying that Construction A is optimal up to a constant.

The accuracy of this construction is demonstrated below.

Theorem 13 *For all $a \in \Sigma_{n+1}$, the code $\mathcal{C}(n, \ell, a)$ is a single-deletion ℓ -read code.*

Proof: Consider a vector \mathcal{R}' resulting from a single deletion on the ℓ -read vector of some $\mathbf{x} \in \mathcal{C}(n, \ell, a)$. Also, due to Lemma 5, we only consider the case when no pair of consecutive elements in \mathcal{R}' have an absolute difference exceeding 1, as the deletion is immediately correctable otherwise.

Now consider a truncation of \mathcal{R}' , given by $\tilde{\mathcal{R}}' = (\mathcal{R}'_1, \dots, \mathcal{R}'_{n-1})$. Due to the VT constraint in Construction A and the fact that $\tilde{\mathcal{R}}' \bmod 2 \in \text{D}((\mathcal{R}(\mathbf{x})_1, \dots, \mathcal{R}(\mathbf{x})_n) \bmod 2)$, one can uniquely recover $(\mathcal{R}(\mathbf{x})_1, \dots, \mathcal{R}(\mathbf{x})_n) \bmod 2$, which suffices for the recovery of $\mathcal{R}(\mathbf{x})$ and thereby \mathbf{x} , as indicated by P3. \blacksquare

IV. MULTIPLE READS

DNA synthesis technologies typically generate multiple copies of each strand, while PCR amplification during sequencing tends to boost the number of copies even further, albeit introducing errors in the process [20]–[23]. Investigating how the availability of multiple noisy versions at the receiver might ease the reconstruction process, is thus a relevant and intriguing problem [24]–[31].

In one of the earliest works on reconstruction from multiple noisy sequences, Levenshtein [25, Corollary 1] established that for any alphabet Σ and two distinct vectors $\mathbf{u}, \mathbf{v} \in \Sigma^n$, it holds that $|\text{D}(\mathbf{u}) \cap \text{D}(\mathbf{v})| \leq 2$. Now given any two distinct binary vectors $\mathbf{x}, \mathbf{y} \in \Sigma_2^n$, we may replace \mathbf{u} and \mathbf{v} with $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ respectively, and thereby conclude that three noisy versions of an ℓ -read vector are sufficient to uniquely determine the channel input. To examine if and how the intrinsic characteristics of ℓ -read vectors might allow for more efficient reconstruction strategies, we endeavor to assess in this section, how the availability of two distinct noisy ℓ -read vectors might lower the redundancy required to correct a single deletion. Equivalently, we are interested in the conditions under which two distinct binary vectors $\mathbf{x}, \mathbf{y} \in \Sigma_2^n$ have ℓ -read vectors such that $|\text{D}(\mathcal{R}(\mathbf{x})) \cap \text{D}(\mathcal{R}(\mathbf{y}))| = 2$. To accomplish this, we employ the following proposition from [32].

Definition 14 [32] Two words \mathbf{u} and \mathbf{v} of length n are said to be *confusable* if there exist subwords \mathbf{a} , \mathbf{b} and \mathbf{c} such that

- $\mathbf{u} = \mathbf{acb}$ and $\mathbf{v} = \mathbf{a\bar{c}b}$ with $|\mathbf{c}| = |\bar{\mathbf{c}}| \geq 2$;
- $\{\mathbf{c}, \bar{\mathbf{c}}\} = \{\alpha\beta\alpha\beta \dots \alpha\beta, \beta\alpha\beta\alpha \dots \beta\alpha\}$,

for some $\alpha, \beta \in \Sigma_q$.

Remark: Such vectors are named Type-A confusable in [32].

Proposition 15 [32] For any two distinct words $\mathbf{u}, \mathbf{v} \in \Sigma_q^n$, it holds that if $d_H(\mathbf{u}, \mathbf{v}) \geq 2$, we have $|\mathcal{D}(\mathbf{u}) \cap \mathcal{D}(\mathbf{v})| = 2$ if and only if \mathbf{u} and \mathbf{v} are confusable.

Thus, we seek to ascertain when and how two binary vectors might possess confusable ℓ -read vectors.

Lemma 16 When $\ell \geq 2$, there exist no two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_2^n$ such that $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ are confusable.

Proof: We prove this by contradiction, i.e., we proceed by assuming the existence of two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_2^n$ such that their respective read vectors are confusable.

Let i refer to the first index where $\mathcal{R}(\mathbf{x})$ and $\mathcal{R}(\mathbf{y})$ disagree, implying that $x_1^{i-1} = y_1^{i-1}$. Now assume w.l.o.g. that $(x_i, y_i) = (0, 1)$, causing $\mathcal{R}(\mathbf{y})_i = \mathcal{R}(\mathbf{x})_i + 1$. For simplicity of exposition, we let $\mathcal{R}(\mathbf{x})_i$ and $\mathcal{R}(\mathbf{y})_i$ be denoted by α and β respectively, where $\beta = \alpha + 1$. By virtue of the confusability of the read vectors, we know that for some $m > 0$, $(\mathcal{R}(\mathbf{x})_i, \dots, \mathcal{R}(\mathbf{x})_{i+2m-1}) = (\alpha\beta)^m$ while $(\mathcal{R}(\mathbf{y})_i, \dots, \mathcal{R}(\mathbf{y})_{i+2m-1}) = (\beta\alpha)^m$. Now the fact that $\mathcal{R}(\mathbf{x})_{i+1} - \mathcal{R}(\mathbf{x})_i = \mathcal{R}(\mathbf{y})_i - \mathcal{R}(\mathbf{y})_{i+1} = 1$ necessitates $(x_{i+1}, x_{i-\ell+1}) = (y_{i-\ell+1}, y_{i+1}) = (1, 0)$. However for $\ell \geq 2$, the requirement $x_{i-\ell+1} \neq y_{i-\ell+1}$ contradicts $x_1^{i-1} = y_1^{i-1}$. Thus, \mathbf{x} and \mathbf{y} do not exist. ■

For $\ell \geq 2$, the above lemma implies the following outcome on the redundancy needed to uniquely recover a binary vector from two distinct erroneous copies of its ℓ -read vector.

Lemma 17 When $\ell \geq 2$, for any two distinct $\mathbf{x}, \mathbf{y} \in \Sigma_2^n$, $|\mathcal{D}(\mathcal{R}(\mathbf{x})) \cap \mathcal{D}(\mathcal{R}(\mathbf{y}))| \leq 1$.

Proof: We know from [25, Corollary 1] that $|\mathcal{D}(\mathcal{R}(\mathbf{x})) \cap \mathcal{D}(\mathcal{R}(\mathbf{y}))| \leq 2$ since distinct binary vectors have distinct ℓ read vectors, as suggested by P3. Additionally [13, Lemma 1] asserts that for $\ell \geq 2$, it holds that $d_H(\mathcal{R}(\mathbf{x}), \mathcal{R}(\mathbf{y})) \geq 2$ for distinct \mathbf{x} and \mathbf{y} . Upon combining these facts with Proposition 15 and Lemma 16, we arrive at the statement of the lemma. ■

We can thus infer the following on the redundancy required for reconstruction with two noisy read vectors.

Corollary 18 For any $\ell \geq 2$, $\mathbf{x} \in \Sigma_2^n$ and given any two distinct noisy read vectors $\mathcal{R}', \mathcal{R}'' \in \mathcal{D}(\mathcal{R}(\mathbf{x}))$, $\mathcal{R}(\mathbf{x})$ can be uniquely recovered.

One possible method to accomplish reconstruction with two corrupted ℓ -read vectors is outlined in Algorithm 1 and its correctness is proved in the next lemma.

Lemma 19 For any $\ell \geq 2$ and $\mathbf{x} \in \Sigma_2^n$ such that $\mathcal{D}(\mathcal{R}(\mathbf{x})) \geq 2$, given any two distinct vectors in $\mathcal{D}(\mathcal{R}(\mathbf{x}))$, Algorithm 1 returns $\mathcal{R}(\mathbf{x})$.

Proof: Let the two noisy reads be denoted by \mathcal{R}' and \mathcal{R}'' respectively. By virtue of Lemma 5, we deem it sufficient to study the case wherein neither of these vectors has a pair of consecutive elements with an absolute difference exceeding 1.

Algorithm 1: Reconstruct

Input: n, ℓ , set $\{\mathcal{R}', \mathcal{R}''\} \subseteq \mathcal{D}(\mathcal{R}(\mathbf{x}))$ for some $\mathbf{x} \in \Sigma_2^n$

Output: $\mathcal{R}(\mathbf{x})$

init

Let i and j be the first and last indices at which \mathcal{R}' and \mathcal{R}'' disagree.

$\widehat{\mathcal{R}}(\mathbf{x}) \leftarrow (\mathcal{R}'_1, \dots, \mathcal{R}'_{i-1}, \mathcal{R}''_i, \mathcal{R}'_i, \dots, \mathcal{R}'_{n+\ell-2});$

$\widetilde{\mathcal{R}}(\mathbf{x}) \leftarrow (\mathcal{R}'_1, \dots, \mathcal{R}'_j, \mathcal{R}''_{j+1}, \mathcal{R}'_{j+1}, \dots, \mathcal{R}'_{n+\ell-2}).$

if $\widehat{\mathcal{R}}(\mathbf{x})$ is the ℓ -read vector¹ of any vector in Σ_2^n **then**

└ $\mathcal{R}(\mathbf{x}) \leftarrow \widehat{\mathcal{R}}(\mathbf{x}).$

else

└ $\mathcal{R}(\mathbf{x}) \leftarrow \widetilde{\mathcal{R}}(\mathbf{x}).$

Let i and j denote the first and last indices at which \mathcal{R}' and \mathcal{R}'' disagree. Of course, $i = j$ when $d_H(\mathcal{R}', \mathcal{R}'') = 1$. Also assume that \mathcal{R}' and \mathcal{R}'' arise from a deletion on $\mathcal{R}(\mathbf{x})$ at indices a and b respectively. Evidently, $\{a, b\} = \{i, j + 1\}$. Now consider the vectors

$\widehat{\mathcal{R}}(\mathbf{x}) = (\mathcal{R}'_1, \dots, \mathcal{R}'_{i-1}, \mathcal{R}''_i, \mathcal{R}'_i, \dots, \mathcal{R}'_{n+\ell-2}),$

$\widetilde{\mathcal{R}}(\mathbf{x}) = (\mathcal{R}'_1, \dots, \mathcal{R}'_j, \mathcal{R}''_{j+1}, \mathcal{R}'_{j+1}, \dots, \mathcal{R}'_{n+\ell-2}).$

These are clearly distinct and depending on whether $a = i$ or $a = j + 1$, $\mathcal{R}(\mathbf{x})$ is either equal to $\widehat{\mathcal{R}}(\mathbf{x})$ or $\widetilde{\mathcal{R}}(\mathbf{x})$. Next, observe that $\widehat{\mathcal{R}}(\mathbf{x})$ and $\widetilde{\mathcal{R}}(\mathbf{x})$ cannot be legitimate read vectors according to [13, Proposition 1], simultaneously, as Lemma 17 would be otherwise contradicted. Thus, Algorithm 1 chooses one of $\widehat{\mathcal{R}}(\mathbf{x})$ and $\widetilde{\mathcal{R}}(\mathbf{x})$, as advised by [13, Proposition 1]. ■

Remark: Prior work [14], [26], [33] states that for the standard single deletion channel, i.e., $\ell = 1$, the required redundancy decreases gracefully from $\log_2 n + O(1)$ to $\log_2 \log_2 n - O(1)$, given one and two distinct erroneous copies of an ℓ -read vector, respectively. It is thus somewhat surprising to learn that when $\ell \geq 2$, the minimal redundancy cost remains the same for one received sequence, while for two noisy copies, it instantly drops to 0. This behavior can perhaps be attributed to the close connection between certain deletions in the read vector and sticky deletions in the original sequence, established in Lemma 8, as it is known that two distinct erroneous reads are sufficient for reconstruction from a single sticky deletion (an equivalent result for sticky insertions appears, e.g., in [31]).

V. CONCLUSION

This work investigates how the inherent redundancy imbued by the physical aspects of a nanopore sequencer into its reads might help achieve more efficient deletion-correcting codes. To this end, we found that for the simplified model of nanopore sequencing adopted here, the minimal redundancy required to correct a single deletion reduces remarkably when the receiver is provided two distinct erroneous received sequences, as opposed to just one. This raises further questions regarding the case of multiple deletions and more received sequences, which we plan to explore in future work.

¹For more details, and a description of an efficient verification procedure, we refer the reader to [13].

REFERENCES

- [1] D. Deamer, M. Akeson, and D. Branton, “Three decades of nanopore sequencing,” *Nature Biotechnology*, vol. 34, no. 5, pp. 518–524, May 2016.
- [2] A. H. Laszlo, I. M. Derrington, B. C. Ross, *et al.*, “Decoding long nanopore sequencing reads of natural DNA,” *Nature Biotechnology*, vol. 32, no. 8, pp. 829–833, Aug. 2014.
- [3] J. J. Kasianowicz, E. Brandin, D. Branton, and D. W. Deamer, “Characterization of individual polynucleotide molecules using a membrane channel,” *Proc. Natl. Acad. Sci.*, vol. 93, no. 24, pp. 13 770–13 773, Nov. 1996.
- [4] W. Mao, S. N. Diggavi, and S. Kannan, “Models and information-theoretic bounds for nanopore sequencing,” *IEEE Trans. Inf. Theory*, vol. 64, no. 4, pp. 3216–3236, Apr. 2018.
- [5] R. Hulett, S. Chandak, and M. Wootters, “On coding for an abstracted nanopore channel for DNA storage,” in *IEEE Intl. Symp. on Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 2465–2470.
- [6] B. McBain, E. Viterbo, and J. Saunderson, “Finite-state semi-markov channels for nanopore sequencing,” in *IEEE Intl. Symp. Inf. Theory (ISIT)*, Espoo, Finland, Jun. 2022, pp. 216–221.
- [7] A. Vidal, V. B. Wijekoon, and E. Viterbo, “Concatenated Nanopore DNA Codes,” *IEEE Tran. on NanoBioscience*, vol. 23, no. 2, pp. 310–318, Apr. 2024.
- [8] A. Vidal, V. B. Wijekoon, and E. Viterbo, “Error bounds for decoding piecewise constant nanopore signals in dna storage,” in *ICC 2023 - IEEE Intl. Conf. Comm.*, May 2023, pp. 4452–4457.
- [9] A. Vidal, V. B. Wijekoon, and E. Viterbo, “Union bound for generalized duplication channels with dtw decoding,” in *2023 IEEE Intl. Symp. Inf. Theory (ISIT)*, Jun. 2023, pp. 358–363.
- [10] A. Banerjee, Y. Yehezkeally, A. Wachter-Zeh, and E. Yaakobi, “Error-correcting codes for nanopore sequencing,” *IEEE Tran. Inf. Theory Early Access*, 2024.
- [11] Y. M. Chee, A. Vardy, V. K. Vu, and E. Yaakobi, “Transverse-read-codes for domain wall memories,” *IEEE Journal on Selected Areas in Inf. Theory*, vol. 4, pp. 784–793, 2023.
- [12] O. Yerushalmi, T. Etzion, and E. Yaakobi, “The capacity of the weighted read channel,” in *Proc. IEEE Intl. Symp. Inf. Theory (ISIT)*, Accepted Apr 2024, (arXiv preprint arXiv:2401.15368).
- [13] A. Banerjee, Y. Yehezkeally, A. Wachter-Zeh, and E. Yaakobi, “Error-correcting codes for nanopore sequencing,” in *IEEE Intl. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan: IEEE, Jun. 2023, pp. 364–369.
- [14] V. Levenshtein, “Binary codes capable of correcting spurious insertions and deletions of ones,” *Prob. Inf. Trans.*, vol. 1, pp. 8–17, 1965.
- [15] 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, Jan. 2010.
- [16] H. Mahdaviifar and A. Vardy, “Asymptotically optimal sticky-insertion-correcting codes with efficient encoding and decoding,” in *2017 IEEE Intl. Symp. Inf. Theory (ISIT)*, Aachen, Germany: IEEE, Jun. 2017, pp. 2683–2687.
- [17] S. Wang, V. K. Vu, and V. Y. F. Tan, “Codes for correcting t limited-magnitude sticky deletions,” in *2023 IEEE Intl. Symp. Inf. Theory (ISIT)*, Jun. 2023, pp. 1148–1153.
- [18] J. L. Doob, “Regularity properties of certain families of chance variables,” *Transactions of the American Mathematical Society*, vol. 47, no. 3, pp. 455–486, 1940.
- [19] A. Fazeli, A. Vardy, and E. Yaakobi, “Generalized sphere packing bound,” *IEEE Tran. Inf. Theory*, vol. 61, no. 5, pp. 2313–2334, May 2015.
- [20] G. M. Church, Y. Gao, and S. Kosuri, “Next-Generation Digital Information Storage in DNA,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012.
- [21] N. Goldman, P. Bertone, S. Chen, *et al.*, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.
- [22] L. Organick, S. D. Ang, Y.-J. Chen, *et al.*, “Random access in large-scale DNA data storage,” *Nature Biotechnology*, vol. 36, no. 3, pp. 242–248, Mar. 2018.
- [23] S. M. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, “DNA-Based Storage: Trends and methods,” *IEEE Tran. on Molecular, Biological and Multi-Scale Comm.*, vol. 1, no. 3, pp. 230–248, Sep. 2015.
- [24] V. Levenshtein, “Efficient reconstruction of sequences,” *IEEE Tran. Inf. Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.
- [25] V. I. Levenshtein, “Efficient reconstruction of sequences from their subsequences or supersequences,” *Journal of Combinatorial Theory, Series A*, vol. 93, no. 2, pp. 310–332, Feb. 2001.
- [26] J. Chrisnata, H. M. Kiah, and E. Yaakobi, “Correcting deletions with multiple reads,” *IEEE Trans. Inf. Theory*, vol. 68, no. 11, pp. 7141–7158, Nov. 2022.
- [27] M. Abu-Sini and E. Yaakobi, “On Levenshtein’s reconstruction problem under insertions, deletions, and substitutions,” *IEEE Tran. Inf. Theory*, vol. 67, no. 11, pp. 7132–7158, Nov. 2021.
- [28] T. Batu, S. Kannan, S. Khanna, and A. McGregor, “Reconstructing strings from random traces,” in *Proc. of the Fifteenth Annual ACM-SIAM Symp. on Discrete Algorithms*, ser. SODA ’04, USA: Society for Industrial and Applied Mathematics, Jan. 2004, pp. 910–918.
- [29] V. L. Phuoc Pham, K. Goyal, and H. M. Kiah, “Sequence reconstruction problem for deletion channels: A complete asymptotic solution,” in *2022 IEEE Intl. Symp. Inf. Theory (ISIT)*, Jun. 2022, pp. 992–997.
- [30] R. Gabrys and E. Yaakobi, “Sequence reconstruction over the deletion channel,” *IEEE Tran. Inf. Theory*, vol. 64, no. 4, pp. 2924–2931, Apr. 2018.
- [31] Y. Yehezkeally and M. Schwartz, “Reconstruction codes for DNA sequences with uniform tandem-duplication errors,” *IEEE Tran. Inf. Theory*, vol. 66, no. 5, pp. 2658–2668, May 2020.
- [32] K. Cai, H. M. Kiah, T. T. Nguyen, and E. Yaakobi, “Coding for sequence reconstruction for single edits,” *IEEE Tran. Inf. Theory*, vol. 68, no. 1, pp. 66–79, Jan. 2022.
- [33] N. J. A. Sloane, “On Single-Deletion-Correcting Codes,” *arXiv:math/0207197*, Jul. 2002. arXiv: math/0207197.