DNA-Correcting Codes: End-to-end Correction in DNA Storage Systems

Avital Boruchovsky*, **Daniella Bar-Lev***, and **Eitan Yaakobi***
*Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel
Email: {avital.bor ,daniellalev, yaakobi}@cs.technion.ac.il

***Abstract*—This paper introduces a new solution to DNA storage that integrates all three steps of retrieval, namely clustering, reconstruction, and error correction. *DNA-correcting codes* are presented as a unique solution to the problem of ensuring that the output of the storage system is unique for any valid set of input strands. To this end, we introduce a novel distance metric to capture the unique behavior of the DNA storage system and provide necessary and sufficient conditions for DNA-correcting codes. The paper also includes several upper bounds and constructions of DNA-correcting codes.**

I. Introduction

The first two experiments that showed the potential of using synthetic DNA as a means for a large-scale information storage system were done in [6] and [8]. Since then, together with the developments in synthesis and sequencing technologies, more research groups showed the potential of in vitro DNA storage; see e.g. [1]–[4], [7], [12], [19], [20].

A typical DNA storage system consists of three components: (1) synthesizing the strands that contain the encoded data. In current technologists, each strand has millions of copies, and the length of these strands is usually limited to 250-300 nucleotides; (2) a storage container that stores the synthetic DNA strands; (3) a DNA sequencer that reads the strands, the output sequences from the sequencing machine are called *reads*. This novel technology has several properties that are fundamentally different from its digital counterparts, while the most prominent one is that the erroneous copies are stored in an unordered manner in the storage container (see e.g. [11]). The most common solution to overcome this challenge is to use indices that are stored as part of the strand. Each strand is prefixed with some nucleotides that indicate the strand's location, with respect to all other strands, these indices are usually protected using an *error-correcting code* (ECC) [2], [3], [10], [12], [19]. The retrieval of the input information is usually done by the following three steps. The first step is to partition all the reads into *clusters* such that the reads at each cluster are all copies of the same information strand. The second step is applying a *reconstruction algorithm* on every cluster to retrieve an approximation of the original input strands. In the last step, an ECC is used in order to correct the remaining errors and to retrieve the user's information.

While previous works tackled each of these steps independently (see e.g. [1], [2], [3], [12], [18], [19]), this work aims to tackle all of them *together*. This is accomplished by limiting the stored messages in the DNA storage system, such that for any two input messages, the sets of all the possible outputs will be mutually disjoint, we call this family of codes *DNA-correcting codes*. Our point of departure is the recent work [17] of *clustering-correcting codes* that proposed codes for successful clustering. However, their results have been established under the assumption that the correct reads in every cluster satisfy some dominance property. Furthermore, the codes in [17] do not aim to recover the input data, but only to achieve a successful clustering. On the contrary, our suggested codes also guarantee the recovery of the input data, while eliminating the dominance assumption. Similar to [17], it is assumed that every information strand consists of an *index-field* and a *data-field*.

The rest of the paper is organized as follows. Section II presents the definitions and the problem statement. In Section III, we consider the case where the data-field is error-free. In addition, we present the DNA-distance metric, which is used in order to prove necessary and sufficient conditions for DNA-correcting codes. In section IV, we study codes over the index-field. Using these codes we present constructions for DNA-correcting codes and bounds on the size of DNA-correcting codes. Lastly, several generalizations and open problems are discussed in Section V. Due to space limitations, some of the proofs do not appear here, but they can all be viewed in the extended version of the paper [5].

II. Definitions, Problem Statement, and Related Works

The following notations will be used in this paper. For a positive integer n, the set $\{0, 1, \ldots, n-1\}$ is denoted by $[n]$ and $\{0,1\}^n$ is the set of all length-n binary vectors. For two vectors $\boldsymbol{x}, \boldsymbol{y}$, of the same length, the Hamming distance between them is the number of coordinates in which they differ and is denoted by $d_H(\boldsymbol{x}, \boldsymbol{y})$. For two sets of vectors of the same size Z, Y, let $BI(Z, Y)$ be the space of all bijective functions (matchings) from Z to Y and for a matching $\pi \in BI(Z, Y)$, let $w_H(\pi)$ denote the maximal Hamming distance between any two matched vectors, i.e., $w_H(\pi) = \max_{\boldsymbol{z} \in Z} \{d_H(\boldsymbol{z}, \pi(\boldsymbol{z}))\}$. We assume a binary alphabet in the paper as the generalization to higher alphabets will be immediate and all logs are taken according to base 2.

Assume that a set of M length-L strands are stored in a DNA-based storage system. We will assume that $M = 2^{\beta L}$ for some $0 < \beta < 1$, and for simplicity it is assumed that βL is an integer. Every stored length-L strand $\boldsymbol{s}$ is of the form $\boldsymbol{s} = (ind, \boldsymbol{u})$, where ind is the length-ℓ index-field of the strand (which represents the relative position of this strand in relation to all other strands) and $\boldsymbol{u}$ is the length-$(L-\ell)$ data-field of the strand. Different strands are required to have a different index-field, as otherwise, it will not be possible to determine the order of the strands. The length of

The research was funded by the European Union (ERC, DNAStorage, 865630). 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. This work was also supported in part by NSF Grant CCF 2212437.

978-1-6654-7554-9/23/$31.00 ©2023 IEEE

the index-field of all the strands is the same and since all indices are different it holds that $\ell \geq \log(M) = \beta L$.

For M, L, and ℓ, the space of all possible messages that can be stored in the DNA storage system is:

$$\mathcal{X}_{M,L,\ell} = \big\{\{(ind_1, \boldsymbol{u}_1), (ind_2, \boldsymbol{u}_2), \ldots, (ind_M, \boldsymbol{u}_M)\} | \\ \forall i : ind_i \in \{0,1\}^{\ell}, \boldsymbol{u}_i \in \{0,1\}^{L-\ell}, \\ \forall i \neq j : ind_i \neq ind_j\big\}. \quad (1)$$

Note that $|\mathcal{X}_{M,L,\ell}| = \binom{2^\ell}{M} 2^{(L-\ell)M}$ since there are $\binom{2^\ell}{M}$ options to choose the different set of index-fields and then $2^{(L-\ell)M}$ more options to choose the data-field for every index. Under this setup, a code $\mathcal{C}$ will be a subset of $\mathcal{X}_{M,L,\ell}$.

When a set $Z = \{(ind_1, \boldsymbol{u}_1), \ldots, (ind_M, \boldsymbol{u}_M)\}$ is synthesized, each of its strands has a large number of noisy copies, and during the sequencing process a subset of these copies is read, while the number of copies depends on the biological process and the technologies that are being used. Throughout this paper, it is assumed that the number of copies for each strand is exactly K, and so, the sequencer's output is a set of MK reads, where every output read is a noisy copy of one of the input strands. It is also assumed that the noise is of substitution type and in Section V we explain how most of the results hold for edit errors as well when changing the Hamming distance to edit. Let τ denote the maximal relative fraction of incorrect copies that every input strand can have and by e_i, e_d the largest number of errors that can occur at the index, data-field of each strand, respectively. Formally, the DNA storage channel is modeled as follows.

Definition 1. A DNA-based storage system is called a $(\tau, e_i, e_d)_K$-*DNA storage system* if it satisfies the following properties: (1) Every input strand $(ind, \boldsymbol{u})$ has exactly K output copies, (2) at most $\lfloor \tau K \rfloor$ of these copies are erroneous, and (3) if $(ind', \boldsymbol{u}')$ is a noisy copy of $(ind, \boldsymbol{u})$ then $d_H(ind, ind') \leq e_i$ and $d_H(\boldsymbol{u}, \boldsymbol{u}') \leq e_d$.

For a set $Z \in \mathcal{X}_{M,L,\ell}$, let $B^K_{(\tau,e_i,e_d)}(Z)$ be the set of all possible MK reads one can get from Z after it passes through a $(\tau, e_i, e_d)_K$-DNA storage system (i.e., every element in $B^K_{(\tau,e_i,e_d)}(Z)$ is a multiset of MK reads). Under this setup, a code $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ is called a $(\tau, e_i, e_d)_K$-*DNA-correcting code* if for every two codewords $Z, Z' \in \mathcal{C}$ such that $Z \neq Z'$, it holds that $B^K_{(\tau,e_i,e_d)}(Z) \cap B^K_{(\tau,e_i,e_d)}(Z') = \emptyset$, i.e., the sets of possible outputs for all codewords are mutually disjoint when the parameters are τ, e_i, e_d, and K. The redundancy of such a code is defined by $r(\mathcal{C}) = \log_2(|\mathcal{X}_{M,L,\ell}|) - \log_2(|\mathcal{C}|)$.

Let $A_{M,L,\ell}(\tau, e_i, e_d, K)$ denote the size of a largest $(\tau, e_i, e_d)_K$-DNA-correcting code given the parameters $M, L, \ell, \tau, e_i, e_d$, and K. The goal of this work is to find necessary and sufficient conditions for a code to be a DNA-correcting code and to study the value of $A_{M,L,\ell}(\tau, e_i, e_d, K)$ for different values.

A. Related Work

Previous studies on information retrieval in DNA storage systems have typically tackled the problem by addressing the three steps (i.e., clustering, reconstruction, and error correction) individually, utilizing a combination of ECC and algorithmic methods. In most works, the clustering step was performed by protecting each of the indices with an ECC and then using the decoder of this code to correct the indices and cluster the reads [2], [3], [10], [12], [19]. Consequently, this process results in a fixed set of indices to the code. Other works used algorithmic methods which are usually time-consuming or not accurate enough in clustering [13], [14]. The reconstruction task is commonly studied independently, and it is usually assumed that the clustering step was successful [9], [15], [18]. Additionally, in most previous works, an ECC is applied on the data and is used for correcting errors on the reconstructed strands, see e.g. [3], [12], [19]. Another approach, which is the most related to ours, appears in [17], where the authors studied the clustering problem from a coding theory perspective, however, their work only tackles the first step in the retrieval process of the data, i.e., the clustering step. Our approach in this work considers the indices together as a set, this may result in different indices sets which are used for different information messages. The key advantage of this work with respect to previous studies is that we present a novel approach for error-correcting codes in DNA storage systems that encapsulate all the information retrieval steps together into a single code.

III. Error Free Data-Field

We start by studying the case where the data part is free of errors, i.e., $e_d = 0$. For a set $Z = \{(ind_1, \boldsymbol{u}_1), \ldots, (ind_M, \boldsymbol{u}_M)\} \in \mathcal{X}_{M,L,\ell}$, let $S(Z)$ denote *the data-field set* of Z which is defined by $S(Z) = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M\}$ and $MS(Z)$ denotes *the data-field multiset* of Z, $MS(Z) = \{\!\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M\}\!\}$. We use the notation of $MS(\mathcal{X}_{M,L,\ell})$ to denote the set of all possible data-field multisets of the elements in $\mathcal{X}_{M,L,\ell}$.

For a code $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ and a data-field multiset $U \in MS(\mathcal{X}_{M,L,\ell})$, let $\mathcal{C}_U \subseteq \mathcal{C}$ be the set of all codewords $Z \in \mathcal{C}$ for which $MS(Z) = U$. The next claim presents a necessary and sufficient condition for DNA-correcting codes for $e_d = 0$.

Claim 1. A code $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ is a $(\tau, e_i, e_d = 0)_K$-DNA-correcting code if and only if for every data-field multiset $U \in MS(\mathcal{X}_{M,L,\ell})$, it holds that $\mathcal{C}_U$ is a $(\tau, e_i, 0)_K$-DNA-correcting code.

Proof: If $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ is a $(\tau, e_i, 0)_K$-DNA-correcting code, then every subset of it is a $(\tau, e_i, 0)_K$-DNA-correcting code as well. On the other hand, if $Z_1, Z_2 \in \mathcal{C}$ such that $MS(Z_1) \neq MS(Z_2)$ then $B^K_{(\tau,e_i,0)}(Z_1) \cap B^K_{(\tau,e_i,0)}(Z_2) = \emptyset$ since the data-field is free of errors, and for $Z_1, Z_2 \in \mathcal{C}$ such that $MS(Z_1) = MS(Z_2)$ we have that $B^K_{(\tau,e_i,0)}(Z_1) \cap B^K_{(\tau,e_i,0)}(Z_2) = \emptyset$, since $\mathcal{C}_{MS(Z_1)}$ is a $(\tau, e_i, e_d)_K$-DNA-correcting code. ∎

For a data-field multiset $U \in MS(\mathcal{X}_{M,L,\ell})$, let $A_{M,L,\ell}(\tau, e_i, e_d, K)_U$ denote the largest size of a $(\tau, e_i, e_d)_K$-DNA-correcting code whose all codewords are with a data-field multiset U. The next corollary follows immediately from Claim 1.

Corollary 1. It holds that

$$A_{M,L,\ell}(\tau, e_i, 0, K) = \sum_{U \in MS(\mathcal{X}_{M,L,\ell})} A_{M,L,\ell}(\tau, e_i, 0, K)_U.$$

The last corollary implies that for $e_d = 0$, in order to find the largest DNA-correcting code it is sufficient to find the

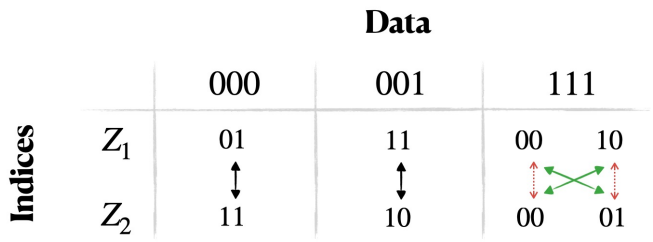

Figure 1: All possible matchings between $I(\boldsymbol{u}, Z_1)$ and $I(\boldsymbol{u}, Z_2)$ for every date field $\boldsymbol{u} \in S(z_1)$.

largest DNA-correcting code for every data-field multiset U. To this end, we define the DNA-distance, a metric on $\mathcal{X}_{M,L,\ell}$, which will be useful for determining what conditions a $(\tau, e_i, 0)_K$-DNA-correcting code $\mathcal{C}_U$ must hold.

A. The DNA-Distance

For $Z = \{(ind_1, \boldsymbol{u}_1), \ldots, (ind_M, \boldsymbol{u}_M)\} \in \mathcal{X}_{M,L,\ell}$ and $\boldsymbol{u} \in S(Z)$, let $I(\boldsymbol{u}, Z)$ be the set of all indices of $\boldsymbol{u}$ in Z, that is, $I(\boldsymbol{u}, Z) = \{ind_i \mid \boldsymbol{u}_i = \boldsymbol{u}\}$. For $Z_1, Z_2 \in \mathcal{X}_{M,L,\ell}$, their *DNA-distance* is defined as

$$\mathcal{D}(Z_1, Z_2) = \begin{cases} \infty, & \text{if } MS(Z_1) \neq MS(Z_2), \\ \max\limits_{\boldsymbol{u} \in S(Z_1)} \min\limits_{\pi \in BI(I(\boldsymbol{u}, Z_1), I(\boldsymbol{u}, Z_2))} \{w_H(\pi)\}, & \text{otherwise.} \end{cases}$$

That is, if the data-field multisets are different, then the distance is infinity. Otherwise, for each data-field $\boldsymbol{u}$ we look at all the possible matchings between the two sets of indices of $\boldsymbol{u}$ in Z_1 and Z_2, and choose the matching with minimal Hamming weight. Then we take the maximal data-field $\boldsymbol{u}$ and the distance is the Hamming weight of the minimal matching for this data-field. The motivation for using the DNA-distance is Claim 1 and the observation that if the codewords Z_1 and Z_2 have different data-field multisets then they cannot share the same output. However, if their data-field multisets are the same, then we consider the Hamming distance between the index-fields of the same data-field. Given a set $Z \in \mathcal{X}_{M,L,\ell}$, we define the radius-$r$ ball[1] of Z by $B_r(Z) = \{Y \in \mathcal{X}_{M,L,\ell} \mid \mathcal{D}(Z, Y) \leq r\}$.

Example 1. Consider the following two words in $\mathcal{X}_{4,5,2}$

$$Z_1 = \{(00, 111), (01, 000), (10, 111), (11, 001)\},$$
$$Z_2 = \{(00, 111), (01, 111), (10, 001), (11, 000)\}.$$

Both words have the same data-field multiset and thus the DNA-distance between them is not infinity. In Fig. 1, for every $\boldsymbol{u} \in S(Z_1)$, we show all possible matchings between $I(\boldsymbol{u}, Z_1)$ and $I(\boldsymbol{u}, Z_2)$. The data-fields 000 and 001 have only one index in both Z_1 and Z_2. Thus, there is only one matching in both cases and the weight of each matching is one. On the other hand, the data-field 111 has two indices in Z_1 and Z_2, and thus there are two optional matchings, corresponding to the dashed red one and the solid green. The weight of the red matching is 2 since $\max\{d_H(00, 00), d_H(10, 01)\} = 2$ and the weight of the green matching is 1 since $\max\{d_H(00, 01), d_H(10, 00)\} = 1$. Hence, $\mathcal{D}(Z_1, Z_2) = 1$.

As will be seen later, the DNA-distance $\mathcal{D}$ will be essential in order to determine if a code is a DNA-correcting code. First, we give the following two important results regarding the distance function $\mathcal{D}$.

Lemma 1. The DNA-distance $\mathcal{D}$ is a metric on $\mathcal{X}_{M,L,\ell}$.

Claim 2. The metric $\mathcal{D}$ is not a graphic metric[2].

Even though the DNA-distance is not a graphic metric, it still satisfies several properties that hold trivially for such ones. In particular, using the metric $\mathcal{D}$ it is possible to derive necessary and sufficient conditions for a code to be a DNA-correcting code, which are shown in the next subsection.

B. Necessary and Sufficient Conditions for DNA-Correcting Codes

For a code $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$, the DNA-distance of $\mathcal{C}$ is defined by $\mathcal{D}(\mathcal{C}) \triangleq \min\limits_{Z_1 \neq Z_2 \in \mathcal{C}} \mathcal{D}(Z_1, Z_2)$. Next, we draw connections between DNA-correcting codes and their DNA-distance. These connections will depend upon the value of τ. First, the case $\tau = 1$ is considered. In the proof of Theorem 2, we use Hall's marriage theorem, which is stated next.

Theorem 1 (*Hall, 1935*)**.** For a finite bipartite graph $G = (L \cup R, E)$, there is an L-perfect matching if and only if for every subset $Y \subseteq L$ it holds that $|Y| \leq |N_G(Y)|$, where $N_G(Y)$ is the set of all vertices that are adjacent to at least one element of Y.

Theorem 2. A code $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ is a $(1, e_i, 0)_K$-DNA-correcting code if and only if $\mathcal{D}(\mathcal{C}) > 2e_i$.

Proof: From Claim 1 it is sufficient to show that the claim holds for every $\mathcal{C}_U \subseteq \mathcal{C}$. Let $U = \{\!\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M\}\!\} \in MS(\mathcal{X}_{M,L,\ell})$ and assume that $\mathcal{C}_U \subseteq \mathcal{X}_{M,L,\ell}$ is a $(1, e_i, 0)$-DNA-correcting code. Assume to the contrary that there are two codewords $Z_1, Z_2 \in \mathcal{C}_U$ such that $\mathcal{D}(Z_1, Z_2) \leq 2e_i$. It will be shown that there exists $W \in B^K_{1,e_i,0}(Z_1) \cap B^K_{1,e_i,0}(Z_2)$. For every data-field $\boldsymbol{u}_i \in S(Z_1)$, there exists $\pi_i \in BI\left((I(\boldsymbol{u}_i, Z_1), I(\boldsymbol{u}_i, Z_2))\right)$ such that $w_H(\pi_i) \leq 2e_i$. Thus, for every index $i_j \in I(\boldsymbol{u}_i, Z_1)$ there exists $r_{i_j} \in \{0,1\}^\ell$ with $d_H(i_j, r_{i_j}) \leq e_i$ and $d_H(\pi_i(i_j), r_{i_j}) \leq e_i$ (since the Hamming metric is graphic). The word W is built in the following way. For every index i_j there are K copies of the form (r_{i_j}, u_i), i.e., we move all the copies of each strand in both codewords to a word in the middle. It is easy to verify that $W \in B^K_{1,e_i,0}(Z_1) \cap B^K_{1,e_i,0}(Z_2)$, which is a contradiction since $\mathcal{C}$ is a $(1, e_i, 0)_K$-DNA-correcting code.

For the opposite direction, let $Z_1, Z_2 \in \mathcal{C}_U$ such that $Z_1 \neq Z_2$. Our goal is to show that $B^K_{(1,e_i,0)}(Z_1) \cap B^K_{(1,e_i,0)}(Z_2) = \emptyset$. From the assumption that $\mathcal{D}(Z_1, Z_2) > 2e_i$ and the definition of $\mathcal{D}$, we have that there exists a data-field $\boldsymbol{u} \in S(Z_1)$ such that there is no $\pi \in BI\left((I(\boldsymbol{u}, Z_1), I(\boldsymbol{u}, Z_2))\right)$ with $w_H(\pi) \leq 2e_i$. Equivalently, if we construct a bipartite graph $G = (L \cup R, E)$ where $L = I(\boldsymbol{u}, Z_1), R = I(\boldsymbol{u}, Z_2)$ and

[1] We use the terminology of a ball since $\mathcal{D}$ is a metric, as shown in Lemma 1.

[2] A metric $\mathcal{D} : X \times X \to \mathbb{N}$ is graphic if the graph $G = (V, E)$ with $V = X$ and edges connect between any two words of distance one, satisfies the following property: for $x_1, x_2 \in X$ it holds that $\mathcal{D}(x_1, x_2) = t$ if and only if the length of the shortest path between x_1 and x_2 in G is t as well.

$E = \{(i,j)|i \in L, j \in R, d_H(i,j) \leq 2e_i\}$ then from Hall's marriage theorem there is a subset $Y \subseteq L$ such that $|Y| > |N_G(Y)|$.

We say that a read $s = (ind, \boldsymbol{u})$ is in the e_i *area of* Y, if its index-field is at distance at most e_i from at least one of the indices in Y. Consider a general output word $W_1 \in B^K_{(1,e_i,0)}(Z_1)$, the number of reads in W_1 that are in the e_i area of Y is at-least $K \cdot |Y|$. On the other hand, for every $W_2 \in B^K_{(1,e_i,0)}(Z_2)$, the number of reads in W_2 that are in the e_i area of Y is at most $K \cdot |N_G(Y)| < K \cdot |Y|$. Thus $B^K_{(1,e_i,0)}(Z_1) \cap B^K_{(1,e_i,0)}(Z_2) = \emptyset$. ∎

Next, we study the case of $\tau < 1$ such that $\frac{K}{2} \leq \lfloor \tau K \rfloor$ and present a similar necessary condition for this case.

Lemma 2. For $\tau < 1$ such that $\frac{K}{2} \leq \lfloor \tau K \rfloor$ and $U \in MS(\mathcal{X}_{M,L,\ell})$, if $\mathcal{C}_U$ is a $(\tau, e_i, 0)_K$-DNA-correcting code then $\mathcal{D}(\mathcal{C}_U) > e_i$.

The opposite direction of Lemma 2 does not hold in general, however, it holds if one assures that all data-fields in the stored sets are different. Let $\overline{\mathcal{X}}_{M,L,\ell}$ denote all such sets, i.e., $\overline{\mathcal{X}}_{M,L,\ell} = \{Z \in \mathcal{X}_{M,L,\ell} | \ |S(Z)| = M\}$. Note that the size of $\overline{\mathcal{X}}_{M,L,\ell}$ is $\binom{2^\ell}{M}\binom{2^{L-\ell}}{M}M!$, and that $\overline{\mathcal{X}}_{M,L,\ell} \neq \emptyset$ if and only if $L - \ell \geq \log_2(M)$. Although restricting to only sets in $\overline{\mathcal{X}}_{M,L,\ell}$ might reduce the number of information bits that is possible to store in the DNA storage system, it is verified in the next lemma, using the results from [17], that for practical values of β there is only a single-bit reduction.

Lemma 3. For $\beta < \frac{1}{2}\left(1 - \frac{\ell}{L}\right)$ it holds that $r(\overline{\mathcal{X}}_{M,L,\ell}) < 1$. Furthermore, for $\beta \leq \frac{\ln(2)(L-\ell)+2\ell}{L(\ln(2)+2\ell)}$ there exists an efficient construction of $\overline{\mathcal{X}}_{M,L,\ell}$ that uses a single redundancy bit.

The notation $MS(\overline{\mathcal{X}}_{M,L,\ell})$ is used to denote the set of all possible data-field multisets of elements in $\overline{\mathcal{X}}_{M,L,\ell}$, which are in essence sets. The next lemma presents a sufficient condition for such sets.

Lemma 4. For $\tau < 1$ such that $\frac{K}{2} \leq \lfloor \tau K \rfloor$ and $U \in MS(\overline{\mathcal{X}}_{M,L,\ell})$ if $\mathcal{D}(\mathcal{C}_U) > e_i$ then $\mathcal{C}_U$ is a $(\tau, e_i, 0)_K$-DNA-correcting code.

The next corollary summarizes this discussion.

Corollary 2. For $\tau < 1$ such that $\frac{K}{2} \leq \lfloor \tau K \rfloor$ and $U \in MS(\overline{\mathcal{X}}_{M,L,\ell})$, $\mathcal{C}_U$ is a $(\tau, e_i, 0)_K$-DNA-correcting code if and only if $\mathcal{D}(\mathcal{C}_U) > e_i$.

We continue to study the case of τ such that $\lfloor \tau K \rfloor < \frac{K}{2}$ in the next two lemmas.

Lemma 5. For τ such that $\lfloor \tau K \rfloor < \frac{K}{2}$, it holds that for every e_i, $\overline{\mathcal{X}}_{M,L,\ell}$ is a $(\tau, e_i, 0)_K$-DNA-correcting code.

Lemma 6. For τ such that $\lfloor \tau K \rfloor < \frac{K}{2}$, if $\mathcal{C} \subseteq \mathcal{X}_{M,L,\ell}$ is a code with $\mathcal{D}(\mathcal{C}) > e_i$ then $\mathcal{C}$ is a $(\tau, e_i, 0)_K$-DNA-correcting code.

C. Codes for a Fixed Data-Field Set

So far in the paper we focused on properties and conditions of DNA-correcting codes that guarantee successful decoding of the data. In particular, Corollary 1 showed that it is enough to construct codes for every data-field multiset $U \in MS(\mathcal{X}_{M,L,\ell})$ independently, while the conditions with respect to the DNA-distance were established in Theorem 2, and Lemmas 2, 4, 5, and 6. These conditions depend upon the value of τ and whether U is a set/multiset. In Lemma 3, it was shown that for all practical values of β, restricting to using only sets in $\overline{\mathcal{X}}_{M,L,\ell}$ imposes only a single bit of redundancy and therefore, the rest of the paper provides DNA-correcting codes for $\overline{\mathcal{X}}_{M,L,\ell}$.

Note that for $U, U' \in MS(\overline{\mathcal{X}}_{M,L,\ell})$ it holds that $A_{M,L,\ell}(\tau, e_i, 0, K)_U = A_{M,L,\ell}(\tau, e_i, 0, K)_{U'}$ and thus for the rest of the paper we fix $U = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_M\} \in MS(\overline{\mathcal{X}}_{M,L,\ell})$ and our goal is to find DNA-correcting codes for U with a given DNA-distance. Note that for $Z = \{(ind_1, \boldsymbol{u}_1), \ldots, (ind_M, \boldsymbol{u}_M)\}, Z' = \{(ind'_1, \boldsymbol{u}_1), \ldots, (ind'_M, \boldsymbol{u}_M)\}$, it holds that

$$\mathcal{D}(Z_1, Z_2) = \max_{1 \leq i \leq M} d_H(ind_i, ind'_i).$$

Thus, we focus on studying the next family of codes for the index-fields.

Definition 2. Let $I(\ell, M) = \{(ind_1, \ldots, ind_M) | \forall i : ind_i \in \{0,1\}^\ell, \forall i \neq j : ind_i \neq ind_j\}$. For every two codewords $\boldsymbol{c} = (c_1, \ldots, c_M), \boldsymbol{c}' = (c'_1, \ldots, c'_M) \in I(\ell, M)$, their *index-distance* is defined by $\mathcal{D}_I(\boldsymbol{c}, \boldsymbol{c}') \triangleq \max_{1 \leq i \leq M} d_H(c_i, c'_i)$ and for a code $\mathcal{C} \subseteq I(\ell, M)$, its *index-distance* is defined by $\mathcal{D}_I(\mathcal{C}) \triangleq \min_{\boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}} \mathcal{D}_I(\boldsymbol{c}, \boldsymbol{c}')$. A code $\mathcal{C} \subseteq I(\ell, M)$ will be called an (ℓ, M, d) *index-correcting code* if $\mathcal{D}_I(\mathcal{C}) \geq d$. We denote by $F(\ell, M, d)$ the size of a maximal (ℓ, M, d) index-correcting code.

Example 2. The rows of the following matrix form a $(2, 4, 2)$ index-correcting code, while each row corresponds to a codeword,

$$P = \begin{pmatrix} 00 & 01 & 11 & 10 \\ 00 & 11 & 10 & 01 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 00 & 10 \\ 11 & 00 & 10 & 01 \\ 11 & 10 & 01 & 00 \end{pmatrix}. \tag{2}$$

One can verify, that for every two different rows i, i', there exists a column j such that $d_H(P(i,j), P(i',j)) = 2$.

The motivation for studying this family of codes comes from the following observation which results from Theorem 2, Corollary 2, and Lemma 5.

Observation 1. For $U \in MS(\overline{\mathcal{X}}_{M,L,\ell})$, it holds that

$$A_{M,L,\ell}(\tau, e_i, 0, K)_U = \begin{cases} F(\ell, M, 2e_i + 1), & \tau = 1 \\ F(\ell, M, e_i + 1), & \frac{K}{2} \leq \lfloor \tau K \rfloor < K \\ \binom{2^\ell}{M} M!, & \lfloor \tau K \rfloor < \frac{K}{2} \end{cases}$$

Note that the study of index-correcting codes and in particular the value of $F(\ell, M, d)$ is interesting on its own and can be useful for other problems, independently of the problem of designing codes for DNA storage. The next section is dedicated to a careful investigation of these codes.

IV. Index-Correcting Codes

We start by studying the special case of $\ell = \log(M)$.

A. $\ell = \log(M)$

In this case, every possible codeword in $I(\log(M), M)$ is a permutation over $\{0,1\}^{\log(M)}$, and for $f, g \in I(\log(M), M)$, their index-distance is equivalent to the ℓ_∞ distance over the Hamming distance of the indices, i.e., $\mathcal{D}_I(f, g) =$

$\max_{i\in\{1,2,\ldots,M\}} d_H(f(i),g(i))$. For $f \in I(\log(M),M)$, let $B_r(f)$ be the ball of radius r centered at f in $I(\log(M),M)$, i.e., $B_r(f) = \{g \in I(\log(M),M) | \mathcal{D}_I(f,g) \leq r\}$. In this case, it holds that $\mathcal{D}_I$ is right invariant, i.e., for $f,g,p \in I(\log(M),M)$ it holds that $\mathcal{D}_I(f,g) = \mathcal{D}_I(f\circ p, g\circ p)$, and thus the size of the balls is the same. Let $B_{r,M}$ denote the size of the balls of radius r in $I(\log(M),M)$. An important matrix with respect to $B_{r,M}$ is the matrix $A_{r,M} = (a_{i,j})$ of size $M \times M$ which is defined by $a_{i,j} = \mathbb{I}_{d_H(i,j)\leq r}$ where $i,j \in \{0,1\}^{\log(M)}$. Let $per(A)$ denote the permanent of a square matrix A. Then the next lemma, which follows in a similar way to the one presented in [16], holds.

Lemma 7. It holds that $B_{r,M} = per(A_{r,M})$.

Next, two bounds on $F(\log(M),M,d)$ are presented. Lemma 8 uses the sphere packing bound with a known bound on the permanent of a matrix, while Lemma 9 uses a method that is similar to the proof of the Singleton bound.

Lemma 8. Let $r(d,M) = \sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{\log(M)}{i}$ then $F(\log(M),M,d) \leq \frac{M!}{(r(d,M)!)^{\frac{M}{r(d,M)}}}$.

Lemma 9. $F(\log(M),M,d) \leq \frac{M!}{(2^{d-1}!)^{2^{\log(M)-d+1}}}$.

Note, that the code in Example 2 achieves the bound in Lemma 9, and hence this bound can be tight in some cases.

Next, we present a construction by building a matrix whose rows form an $(\log(M),M,d)$ index-correcting code. Such a matrix whose rows form an (ℓ,M,d) index-correcting code will be called an (ℓ,M,d)-*matrix*. The construction uses codes over $\{0,1\}^{\log(M)}$ with Hamming distance d and afterward an example for small values of d is presented.

Construction 1. Let $\mathcal{C} \subseteq \{0,1\}^{\log(M)}$ be a maximal linear code with Hamming distance d. Denote by A the size of $\mathcal{C}$ and note that the $\frac{M}{A}$ cosets of $\mathcal{C}$ form a partition of $\{0,1\}^{\log(M)}$. Denote the cosets of $\mathcal{C}$ by $\mathcal{C}_1 = \mathcal{C}, \mathcal{C}_2, \ldots, \mathcal{C}_{\frac{M}{A}}$. We start by building a matrix that consists of $(A!)^{\frac{M}{A}}$ rows, where the first A entries of every row are permutations over the first coset, the second A entries are permutations over the second coset, and so on. Since the entries of every column belong to the same coset, the distance between different rows is at least d. Next, we take every coset $\mathcal{C}_i$ for $2 \leq i \leq \frac{M}{A}$ and remove from it all words that are at distance at most d from the zero vector $\mathbf{0}$, and denote by $\mathcal{C}_i'$ the achieved codes. Then, for every $c' \in \mathcal{C}_i'$, we can look at all the rows in the matrix where c' and $\mathbf{0}$ are fixed to the first entry of their coset (note that there are $(A-1)!^2(A!)^{\frac{M}{A}-2}$ such rows) and add the same row where we replace the entry of c' with $\mathbf{0}$. Since we do so for every i we have $(\frac{M}{A}-1)\cdot|\mathcal{C}'|\cdot\frac{1}{A^2}(A!)^{\frac{M}{A}}$ more rows.

Example 3. We apply Construction 1 to the case of $d=2$ and $d=3$. For $d=2$ we have that the maximal linear code $\mathcal{C}$ is the parity with $|\mathcal{C}| = \frac{M}{2}$, and that $|\mathcal{C}'| = \frac{M}{2} - \log(M)$. Thus in this case we get a $(\log(M),M,2)$ index-correcting code with size of $(\frac{M}{2}!)^2 + (\frac{M}{2} - \log(M))(\frac{M-2}{2}!)^2$. For $d=3$ and $\log(M) = \ell = 2^m - 1$ we have that the maximal linear code $\mathcal{C}$ is the binary Hamming code with $|\mathcal{C}| = 2^{2^m-1-m}$, and that there are 2^m cosets (including the code itself). In addition, every coset $\mathcal{C}_i \neq \mathcal{C}$ has one word of weight 1 and $\frac{2^m-2}{2}$ words of weight 2. Thus we have a $(2^m-1, 2^{2^m-1}, 3)$ index-correcting code with size of $(\frac{M}{\log(M)+1}!)^{\log(M)+1}(1+g(M))$, where $g(M) = \Theta(\frac{\log^2(M)}{M})$. For more detailed analyses, see the extended version [5].

B. $\ell > \log(M)$

In this case, the set of possible indices is larger than the number of strands. We show how to construct an (ℓ',M,d) index-correcting code from an (ℓ,M,d) index-correcting code for $\ell < \ell'$.

Lemma 10. For $\ell' = \ell + \lceil \frac{d}{2} \rceil$ it holds that $F(\ell',M,d) \geq F(\ell,M,d)\cdot 2^M$

Proof: We show an iterative construction of a matrix P' using a maximal (ℓ,M,d)-matrix P with $F(\ell,M,d)$ rows and the proof that the matrix P' is a legal (ℓ',M,d)-matrix with $F(\ell,M,d)\cdot 2^M$ rows appears in the extended version of the paper [5]. The construction is described next.

1) Obtain a matrix P_0' by adding $\lceil \frac{d}{2} \rceil$ bits of 0 at the end of every entry in P.
2) For $j = 1,2,\ldots,M$: Denote the matrix obtained after the j'th step by P_j'. For every row i of P_{j-1}', add a similar row $b_j(i)$ which differs from the i-th row only in the j-th column. The difference is that in $b_j(i)$, the first and last $\lceil \frac{d}{2} \rceil$ bits of the j-th entry, are the transpose of the corresponding entry in row i. ■

Example 4. The next matrix is the matrix P_1' which is obtained from the matrix in Example 2.

$$P_1' = \begin{pmatrix} 000 & 010 & 110 & 100 \\ 000 & 110 & 100 & 010 \\ 000 & 100 & 010 & 110 \\ 110 & 010 & 000 & 100 \\ 110 & 000 & 100 & 010 \\ 110 & 100 & 010 & 000 \\ 101 & 010 & 110 & 100 \\ 101 & 110 & 100 & 010 \\ 101 & 100 & 010 & 110 \\ 011 & 010 & 000 & 100 \\ 011 & 000 & 100 & 010 \\ 011 & 100 & 010 & 000 \end{pmatrix} \tag{3}$$

V. Generalizations and Future Work

Note, that in all the proofs of the necessary and sufficient conditions, only the fact that the Hamming metric is graphic was used. Thus all the results for $e_d = 0$ hold also when replacing every instance of the Hamming distance with the edit distance. The case of $e_d > 0$ is more complicated since Claim 1 does not hold for $e_d > 0$. Nonetheless, if one wishes to construct a $(\tau,e_i,e_d)_K$-DNA-correcting code $\mathcal{C}_U$ for $U \in MS(\mathcal{X}_{M,L,\ell})$ with DNA distance larger than $2e_d+1$, then the constructions in Section IV will work. For future work, we plan to continue studying the value of $F(\ell,M,d)$, especially for $\ell > \log(M)$ which is an interesting and important question, as well as to study the case of $e_d > 0$, and in particular, find necessary and sufficient conditions for this case.

VI. Acknowledgments

The authors would like to thank Tuvi Etzion and Moshe Schwartz for helpful discussions which contributed to this work.

REFERENCES

[1] L. Anavy, I. Vaknin, O.Atar, R. Amit and Z. Yakhini, "Data storage in DNA with fewer synthesis cycles using composite DNA letters". *Nat. Biotechnol*. 37, 2019.

[2] D. Bar-Lev, I. Orr, O. Sabary, T. Etzion, and E. Yaakobi, "Deep DNA storage: Scalable and robust DNA storage via coding theory and deep learning," *arXiv preprint arXiv:2109.00031*, 2021.

[3] M. Blawat, K. Gaedke, I. Hutter, X.-M. Chen, B. Turczyk, S. Inverso, B.W. Pruitt, and G.M. Church, "Forward error correction for DNA data storage," *Int. Conf. on Computational Science*, vol. 80, pp. 1011–1022, 2016.

[4] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system", *Proc. of the Twenty-First Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 637–649, Atlanta, GA, Apr. 2016.

[5] A. Boruchovsky, D. Bar-Lev and E. Yaakobi, "DNA-Correcting Codes: End-to-end Correction in DNA Storage Systems", *arXiv preprint arXiv:2304.10391*, 2023.

[6] 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.

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

[8] N. Goldman, P. Bertone, S. Chen, C. Dessimoz, E.M. LeProust, B. Sipos, and E. Birney, "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.

[9] P. Gopalan, S. Yekhanin, S. Dumas Ang, N. Jojic, M. Racz, K. Strauss, and L. Ceze, "Trace reconstruction from noisy polynucleotide sequencer reads," 2018, US Patent application : US 2018 / 0211001 A1.

[10] 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. Chemie - Int*, Ed. 54, 2015.

[11] R. Heckel, G. Mikutis, and R.N. Grass, "A characterization of the DNA data storage channel", *Nature*, 2018.

[12] L. Organick et al. "Random access in large-scale DNA data storage," *Nature Biotechnology*, 2018.

[13] G. Qu, Z. Yan, and H. Wu, "Clover: tree structure-based efficient DNA clustering for DNA-based data storage", *Briefings in Bioinformatics*, vol 23, issue 5, 2022.

[14] C. Rashtchian, K. Makarychev, M. Racz, S. Ang, D. Jevdjic, S. Yekhanin, L. Ceze, and K. Strauss, "Clustering billions of reads for DNA data storage", *Advances in Neural Information Processing Systems*, vol 30, 2017.

[15] O. Sabary, A. Yucovich, G. Shapira, and E. Yaakobi, "Reconstruction Algorithms for DNA-Storage Systems". *bioRxiv https://doi.org/10.1101/2020.09.16.300186* 2020.

[16] M. Schwartz and P.O. Vontobel, "Improved lower bounds on the size of balls over permutations with the infinity metric", *IEEE Transactions on Information Theory*, vol. 63, no. 10, pp. 6227–6239, Oct. 2017.

[17] T. Shinkar, E. Yaakobi, A. Lenz, and A. Wachter-Zeh, "Clustering correcting codes", *IEEE Transactions on Information Theory*, vol. 68, no. 3, pp. 1560–1580 March 2022.

[18] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, and S. Yekhanin, "Trellis BMA: Coded trace reconstruction on IDS channels for DNA storage", *IEEE International Symposium on Information Theory (ISIT)*, 2021.

[19] S. M H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Reports*, vol. 7, no.1, pp. 5011, 2017.

[20] S. M. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Nature Scientific Reports*, vol. 5, no. 14138, Aug. 2015