Covering All Bases: The Next Inning in DNA Sequencing Efficiency

Hadas Abraham[†], **Ryan Gabrys**[‡], and **Eitan Yaakobi**[†]
[†]The Henry and Marilyn Faculty of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel.
[‡]Calit2, University of California, San Diego.
Emails: hadasabraham@campus.technion.ac.il, yaakobi@cs.technion.ac.il, rgabrys@eng.ucsd.edu

***Abstract*—DNA emerges as a promising medium for the exponential growth of digital data due to its density and durability. This study extends recent research by addressing the *coverage depth problem* in practical scenarios, exploring optimal error-correcting code pairings with DNA storage systems to minimize coverage depth. Conducted within random access settings, the study provides theoretical analyses and experimental simulations to examine the expectation and probability distribution of samples needed for files recovery. Structured into sections covering definitions, analyses, lower bounds, and comparative evaluations of coding schemes, the paper unveils insights into effective coding schemes for optimizing DNA storage systems.**

I. Introduction

The rapid growth of digital data, projected to reach 180 zettabytes by 2025, is causing a data storage crisis, with demand surpassing supply [1]. Existing storage technologies face challenges meeting big data demands. In response, DNA emerges as a promising medium due to its density and durability. The DNA storage process involves *synthesis*, creating artificial DNA strands encoding user information with limitations leading to short strands and multiple noisy copies [2], storage by a *storage container* and *sequencing*, a key component [3], [4], [5], [6], translates DNA into digital sequences. Despite the potential of DNA storage, current DNA sequencers face challenges such as slow throughput and high costs compared to alternatives [7], [8], [9]. Coverage depth, the ratio of sequenced reads to designed strands, impact system latency and costs, highlighting the need for optimization [10], [4].

We extend recent research addressing the *coverage depth problem* [11] by generalizing it to a more practical scenario. Specifically, we consider a container storing m files, each composed of k information strands. These strands are encoded into mn strands using some coding scheme, and the objective is to recover a files out of the total m. Our focus is on investigating the required coverage depth, considering factors such as the DNA storage channel and the error-correcting code. Additionally, we aim to explore the optimal pairing of an error-correcting code with a given DNA storage system to minimize coverage depth. This investigation is conducted within the framework of random access settings, where the user seeks to retrieve only a fraction of the stored information. In this context, we conduct both theoretical and experimental analyses to examine the expectation and probability distribution of the number of samples needed to fully recover the specified a files.

The DNA coverage depth problem is akin to well-known problems such as the coupon collector's, dixie cup, and urn problems, where the objective is to collect all types of coupons or objects [12], [13], [14], [15]. In our context, the "coupons" represent copies of synthesized strands, and the aim is to read at least one copy of each information strand. For example, if n coupons are drawn uniformly at random with repetition, it is well known that the expected number of draws needed to obtain at least one copy of every strand is approximately $n \log n$. However, in this work, we consider the setting where one is allowed to employ the use of a code in order to reduce the number of draws necessary to recover a given subset of information, and it is required to read a specific set of strands that constitute a file.

The paper is structured as follows. In Section II, we provide definitions and articulate the problem statement, focusing on the coverage depth problem in our more practical settings. We also discuss some relevant prior results on this matter. In Section III, we address the scenario where the user aims to retrieve a single file ($a = 1$ out of m). We conduct analyses for three coding schemes: the local MDS scheme, which employs an $[n, k]$ MDS code for each of the m files; the global MDS scheme, employing an $[mn, mk]$ systematic MDS code on the combined strands of the m files; and the partial MDS scheme (PMDS), specifically analyzed for the case of $m = 2$ files. We present the expected value of samples required to recover a file and explore the expected limit as n approaches infinity for both local and global schemes. In Section IV, we establish two lower bounds on the expected number of samples needed for file recovery. Section V includes a comparative analysis of the coding schemes. We prove that, in terms of expectation, the local scheme surpasses the global one. Then, a simulation is conducted, providing insights crucial for determining the optimal coding scheme. While the local scheme demonstrates superior expectations, analysis of probability distribution and variance suggests that the global and PMDS schemes may be more favorable options. Finally, in Section VI, we present results for the case where we aim to recover $a \geq 1$ files and extend our lower bound to this case. Due to the lack of space, we omit some of the proofs in the paper and they appear in the long version of this paper [16].

II. Definitions, Problem Statement, Related Work

A. Definitions

For a positive integer n, $[n]$ denotes the set $\{1, \ldots, n\}$ and H_n denotes the n-th Harmonic number. We consider a DNA-based storage system in which the data is stored as a codeword, described by a vector of length-ℓ sequences or strands over the alphabet $\Sigma = \{A, C, G, T\}$, so the set of all length-ℓ vectors over Σ is denoted by Σ^ℓ. Often, an outer error-correcting code is employed to protect the data across these length-ℓ sequences. In the setting studied in this paper, it is assumed that these strands represent some m files and so the input is represented by a vector of m files $\mathcal{U} = (U_1, U_2, \ldots, U_m)$, where each file consists of k length-ℓ information strands $U_i = (\boldsymbol{u}_{i,1}, \boldsymbol{u}_{i,2}, \ldots, \boldsymbol{u}_{i,k}) \in (\Sigma^\ell)^k$ for $i \in [m]$. The mk information strands are then encoded to mn encoded strands

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 CCF2212437.

using some linear $[mn, mk]$ code $\mathcal{C}$ over Σ^ℓ (typically Σ^ℓ is embedded into a field of size 4^ℓ). The resulting encoded vector is denoted by $\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{mn})$ which represents the input vector to the DNA storage system. Note that the files can be encoded either seprately or all together.

The DNA storage channel, denoted by S, initially produces numerous noisy copies for each strand in $\mathcal{X}$. These noisy copies undergo amplification using PCR, and a *sample* of M strands is then sequenced [17]. The output of the sequencing process is a multiset $\mathcal{Y}_M = \{\!\{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_M\}\!\}$, which consists of *reads* $\boldsymbol{y}_j$ for $j \in [M]$, each being a noisy version of some $\boldsymbol{x}_i$, $i \in [mn]$. The model assumes that the index $i \in [mn]$ such that $\boldsymbol{y}_j$ is a noisy copy of $\boldsymbol{x}_i$ is known. The number of reads in $\mathcal{Y}_M$ corresponding to the i-th strand $\boldsymbol{x}_i, i \in [mn]$, depends on a categorical probability distribution $\boldsymbol{p} = (p_1, \ldots, p_{mn})$, where for $i \in [mn]$, p_i is the probability to sample a read of $\boldsymbol{x}_i$. However, for simplicity, it is assumed in this work that $\boldsymbol{p}$ is the uniform distribution and we further assume that there is no noise in the reading process so every read in an error-free copy of some $\boldsymbol{x}_i$. Since we consider the noiseless scenario, there is no need to apply clustering or reconstruction algorithm as well as an error-correcting code to correct the errors during reading. However, we do apply an error-correcting code in order to reduce the required number of reads in order to decode the information. For a more detailed description of this model which include noise we refer the reader to [11].

B. Problem Statement

The main goal of this paper is to explore the necessary sample size for the retrieval of some requested a files by the user out of m from $\mathcal{U}$. Successful decoding of a file U_i for $i \in [m]$ is defined as sampling enough encoded strands from $\mathcal{X}$ that are sufficient to decode all the k information strands of U_i. Note that since the strands in $\mathcal{U}$ are encoded using an error-correcting code $\mathcal{C}$, it is not necessary to sample *all* the k information strands from U_i but any set of encoded strands from $\mathcal{X}$ that allows to decode them. We also note that the main difference in the model studied in this paper and the one from [11] is that the latter work does not assume the partition of the data into files and considers it as one file. Then, the goal is to either decode the entire file or one information strand.

Mathematically speaking, assume $\mathcal{C}$ is the code which is used to encode the m files and let $F \subseteq [m]$ be the set of files that are requested by the user. Let $\nu_{(m,F)}(\mathcal{C})$ be the random variable that governs the number of reads that should be sampled for successful decoding of the a files in F. The problems studied in this paper are formally defined as follows.

Problem 1. Given an $[mn, mk]$ code $\mathcal{C}$, $F \subseteq [m], |F| = a$. Find the following values:

1) The expectation value $\mathbb{E}[\nu_{(m,F)}(\mathcal{C})]$ and the probability distribution $\Pr[\nu_{(m,F)}(\mathcal{C}) > r]$ for any $r \in \mathbb{N}$.
2) The maximal expected number of samples to retrieve any a files, i.e.,

$$T^{\mathcal{C}}_{\max}(a) \triangleq \max_{|F|=a, F\subseteq[m]} \mathbb{E}[\nu_{(m,F)}(\mathcal{C})].$$

Problem 2. For given values of n, k, m, a find:

1) An $[mn, mk]$ code $\mathcal{C}$, that is optimal with respect to minimizing $T^{\mathcal{C}}_{\max}(a)$.
2) The minimum value of $T^{\mathcal{C}}_{\max}(a)$ over all possible $[mn, mk]$ codes $\mathcal{C}$. That is, find the value $T(n, k; m, a) \triangleq \min_{\mathcal{C}}\{T^{\mathcal{C}}_{\max}(a)\}$.

In order to address Problem 1, we consider in Section III three coding schemes and analyze their maximal expected number of samples. These results, in particular, provide an upper bound on the value of $T(n, k; m, a)$ from Problem 2, while lower bounds on this value are given in Section IV.

C. Previous Results

Two special cases of Problem 2 have been investigated in [11]. Specifically, in case there is only one file, i.e., $m = 1$, which implies that $a = 1$ the value of $T(n, k; 1, 1)$ has been fully solved and it was shown that $T(n, k; 1, 1) = n(H_n - H_{n-k})$, which is achieved by any $[n, k]$ MDS code. Similarly, it is easily deduced that $T(n, k; m, m) = mn(H_{mn} - H_{mn-mk})$, which is achieved by any $[mn, mk]$ MDS code. On the other hand, if $k = 1$ and $m > 1$ then we achieve the random access version of the problem in [11], which was studied mainly for $a = 1$. However, the value of $T(n, 1; m, 1)$ is still far from being solved. A lower bound states that for all n, $T(n, 1; m, 1) \geq n = \frac{n(n-m)}{m}(H_n - H_{n-m})$, while several code constructions verify that $T(n, 1; m, 1) < m$. For example, it was shown that there exists n large enough such that $T(n, 1; 2, 1) < 0.91 \cdot 2$ and $T(n, 1; 3, 1) < 0.89 \cdot 3$ and if m is a multiple of 4 then $T(n = 2m, 1; m, 1) < 0.95 \cdot m$. Based on the results [11], it is simple to deduce that $T(k, k; m, a) = mkH_{ak}$, and thus for the rest of the paper we assume that $n > k$. Several more results on this value and related problems have been studied lately in [18], [19].

III. Random Access Expectation for a Single File ($a = 1$)

This section studies the problem of optimizing the sample size for random access queries, where the user wishes to retrieve 1 file. Three coding schemes will be analyzed.

A. The Local MDS Scheme

In this coding scheme, denoted as $\mathcal{C}_1$, we employ an MDS code on each of the m files separately and store each file in n strands. Note that in this coding scheme, in order to decode any file, it is necessary and sufficient to retrieve any k out of its n encoded strands.

Our main goal in this section is to determine the expected number of samples for recovering any of the m files while applying the coding scheme $\mathcal{C}_1$. To analyze the performance of the coding scheme $\mathcal{C}_1$, we let $t(h, i)$ be the random variable that denotes the number of samples required to progress from drawing i to $i + 1$ different strands out of the pool of n encoded strands for the h-th file. Note that $t(h, i)$ follows a geometric distribution $t(h, i) \sim \text{Geo}\left(\frac{n-i}{mn}\right)$, where $\frac{n-i}{mn}$ is the probability of drawing the $(i+1)$-st strand, and consequently, the expected value, $\mathbb{E}[t(h, i)] = \frac{mn}{n-i}$. Furthermore, let $T(h, b)$ be the random variable representing the number of samples needed to progress from drawing 0 to b different strands of the h-th file. Hence, by definition $T(h, b) = \sum_{i=0}^{b-1} t(h, i)$ and thus

$$\begin{aligned}\mathbb{E}[T(h, b)] &= \mathbb{E}\left[\sum_{i=0}^{b-1} t(h, i)\right] = \sum_{i=0}^{b-1} \mathbb{E}[t(h, i)] \\ &= \sum_{i=0}^{b-1} \frac{mn}{n-i} = mn(H_n - H_{n-b}). \end{aligned} \tag{1}$$

Theorem 1. For any $1 \leq k \leq n$ and $m \geq 1$, it holds that

$$T(n, k; m, 1) \leq T^{\mathcal{C}_1}_{\max}(1) = mn(H_n - H_{n-k}).$$

Proof. Assume without loss of generality that $F = \{1\}$. Note that by applying $b = k$ in (1) it holds that $\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_1)] = \mathbb{E}[T(1, k)] = mn(H_n - H_{n-k})$. This also implies that

$$T(n, k; m, 1) \leq T^{\mathcal{C}_1}_{\max}(1) = mn(H_n - H_{n-k}). \qquad \square$$

Corollary 1. For fixed m, $0 < R < 1$, $R = \frac{k}{n}$ for n large enough it holds that $T^{\mathcal{C}_1}_{\max}(1) = mn \log\left(\frac{1}{1-R}\right)$. Furthermore, for any fixed m and k, it holds that,

$$\liminf_{n\to\infty} T(n,k;m,1) \leq \liminf_{n\to\infty} T^{\mathcal{C}_1}_{\max}(1) = mk.$$

B. The Global MDS Scheme

In this coding scheme, denoted as $\mathcal{C}_2$, we employ a systematic MDS code on the combined strands of the m files. Hence, we store the mk information strands into mn encoded strands. In order to decode any of the m files, it is necessary and sufficient to either retrieve all the systematic k strands of the file, or any mk out of mn strands. The latter option decodes all m files and in particular the required file.

Our main goal is to find the expected number of samples to recover any of the m files as defined in Problem 1 while applying the coding scheme $\mathcal{C}_2$. Let $t(i)$ be the random variable that denotes the number of samples required to progress from drawing i to $i+1$ different strands out of the pool of mn encoded strands. Note that $t(i)$ follows a geometric distribution $t(i) \sim \text{Geo}\left(\frac{mn-i}{mn}\right)$, where $\frac{mn-i}{mn}$ is the probability of drawing the $(i+1)$-st strand, and thus $\mathbb{E}\left[t(i)\right] = \frac{mn}{mn-i}$. Furthermore, let $T(b)$ be the random variable representing the number of samples needed to progress from drawing 0 to b different strands. Hence, by definition $T(b) = \sum_{i=0}^{b-1} t(i)$ and

$$\mathbb{E}[T(b)] = \sum_{i=0}^{b-1} \mathbb{E}\left[t(i)\right] = \sum_{i=0}^{b-1} \frac{mn}{mn-i} = mn(H_{mn} - H_{mn-b}).$$

In order to analyze the collection process we will represent it as a discrete-time Markov chain. Let X_b be a random variable that represents the state of the collection process **after** drawing b different strands. Indeed, the collection process satisfies the Markovian property, i.e., for a collection of b states say $s_0, s_1, \ldots, s_{b-1}$ $\Pr\left(X_b = s_b | X_0 = s_0, \ldots, X_{b-1} = s_{b-1}\right) = \Pr\left(X_b = s_b | X_{b-1} = s_{b-1}\right)$. For the setups under consideration, the states will be all compositions of different types of collected strands which will depend in general on the underlying coding scheme itself. We denote $\widehat{s}_i$ to be the sum of collected strands at state s_i. Moreover, we let M denote the transition matrix of this Markov chain where $M_{s_j,s_i} = \Pr(X_b = s_i | X_{b-1} = s_j)$ for two states s_i and s_j. Define $M^{(n)}_{s_j,s_i} = \Pr(X_n = s_i | X_0 = s_j)$ which is the probability of collecting the additional strands from s_j to the composition of collected strands in s_i (i.e., $\widehat{s}_i - \widehat{s}_j = n$ strands)[1]. Also, define the n-step transition probability matrix $M^{(n)} \triangleq (M^{(n)}_{s_j,s_i})$. Note that $M^{(n)}_{s_j,s_i} = \sum_{s_y} M^{(n-1)}_{s_j,s_y} M_{s_y,s_i}$, where for $\widehat{s}_i - \widehat{s}_j \neq n$ then, $M^{(n)}_{s_j,s_i} = 0$. For shorthand, we refer to the initial state as s_0 (i.e., collect nothing from the pool of strands).

Our aim is to compute the expected hitting times for the absorbing states which will depend in general on the underlying coding scheme itself. This is established in the next theorem.

Theorem 2. For any $1 \leq k \leq n$ and $m \geq 1$ it holds that $T(n,k;m,1) \leq T^{\mathcal{C}_2}_{\max}(1)$ and

$$T^{\mathcal{C}_2}_{\max}(1) = mn\left(H_{mn} - H_{mn-mk}\right) - \frac{mn}{\binom{mn}{k}}\left(\sum_{j=0}^{mk-k-1} \binom{k-1+j}{k-1} H_{mn-(k+j)} - H_{mn-mk}\binom{mk-1}{k}\right).$$

Proof sketch. Assume without loss of generality that $F = \{1\}$.

- **States definition:** The set of states is $\mathcal{S}_1 = \{(i,j) \mid 0 \leq i \leq k, 0 \leq j \leq mn-k\}$, where i is the number of strands drawn from the k systematic strands of the first file, and j is the number of strands that were drawn from the other $mn-k$ strands.
- **Transition matrix:** The valid transitions in M (i.e., $M_{(i_1,j_1),(i_2,j_2)} \neq 0$) and their values are

$$M_{(i,j),(i+1,j)} = \frac{k-i}{mn-(i+j)}, M_{(i,j),(i,j+1)} = \frac{mn-k-j}{mn-(i+j)}.$$

The next claim provides a closed formula for $M^{(i+j)}_{(0,0),(i,j)}$ which holds for the non-absorbing states.

Claim 1. $M^{(i+j)}_{(0,0),(i,j)} = \frac{\binom{i+j}{i}\binom{mn-(i+j)}{k-i}}{\binom{mn}{k}}$.

Proof sketch. At state (i,j), we have $\binom{k}{i}$ options to choose i systematic strands and $\binom{mn-k}{j}$ options to choose j strands from the rest of the pool, considering all possibilities for drawing a total of $(i+j)$ strands out of the mn strands, which is $\binom{mn}{i+j}$. The algebraically derived formula is presented in the long version [16]. □

- **Absorbing states:** These are the states that allow the recovery of the first file, so the drawing process ends. In coding scheme $\mathcal{C}_2$, the absorbing states are those where we either drew the k systematic strands of the first file or any mk different strands from the pool of mn strands. We denote Θ_1, Θ_2 as the set of absorbing states corresponding to the first, second option, respectively. That is

$$\Theta_1 \triangleq \{(k,j) \mid 0 \leq j \leq mk-k-1\},$$
$$\Theta_2 \triangleq \{(i,mk-i) \mid 0 \leq i \leq k\}.$$

Note that given $(k,j) \in \Theta_1$, since $(k,j-1)$ is also an absorbing state, we have that

$$M^{(k+j)}_{(0,0),(k,j)} = M^{(k-1+j)}_{(0,0),(k-1,j)} \cdot M_{(k-1,j),(k,j)},$$

which follows directly from the definition of Θ_1.

- **The expectation:** In order to calculate $\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)]$, we let Y be the random variable representing in which absorbing state the collection process ends. The expectation $\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)]$ is conditioned on Y. Hence,

$$\begin{aligned}
&\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)] = \mathbb{E}_Y[\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)|Y]] \\
&= \sum_{\theta\in\Theta_1\cup\Theta_2} \Pr(Y=\theta)\cdot\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)|Y=\theta] \\
&= \sum_{\theta\in\Theta_1\cup\Theta_2} M^{(\widehat{\theta})}_{(0,0),(\theta)} \cdot \mathbb{E}[T(\widehat{\theta})] = mn\left(H_{mn} - H_{mn-mk}\right) \\
&- \frac{mn}{\binom{mn}{k}}\left(\sum_{j=0}^{mk-k-1}\binom{k-1+j}{k-1}H_{mn-(k+j)} - H_{mn-mk}\binom{mk-1}{k}\right).
\end{aligned} \tag{2}$$

The full proof of (2) can be found in [16]. Since the code is symmetric it implies that $\mathbb{E}[\nu_{(m,\{1\})}(\mathcal{C}_2)] = T^{\mathcal{C}_2}_{\max}(1)$. □

Corollary 2. For any fixed m and $k \geq 2$, it holds that

$$\liminf_{n\to\infty} T(n,k;m,1) \leq \liminf_{n\to\infty} T^{\mathcal{C}_2}_{\max}(1) = mk.$$

C. The Partial MDS Scheme

In this section, we consider the case where the underlying retrieval code, denoted $\mathcal{C}_3$, is a partial-MDS (PMDS) code and we apply to $m = 2$ files. We briefly review the definition of a $[r;s]$ code before proceeding.

Definition 1. Let C be a linear $[m_P n_P, m_P(n_P - r_P) - 2s]$ code over a field such that if codewords are taken row-wise as $m_P \times n_P$ arrays, each row belongs to an $[n_P, n_P - r_P, r_P + 1]$

[1] At state s_j, we have two potential transitions: either remaining in the current state by drawing a previously collected strand or collect a new one and progressing to a new state. The variables M, and $M^{(n)}$ are analyzed and defined as the conditional probabilities of transitioning to a new state given new strands were drawn.

MDS code. Given $\sigma_1, \ldots, \sigma_t$ such that for $j \in [t]$, $\sigma_j \geq 1$, we say that C is an $(r_P; \sigma_1, \ldots, \sigma_t)$-erasure correcting code if, for any $1 \leq i_1 < \cdots < i_t \leq m$, C can correct up to $\sigma_j + r_P$ erasures in row i_j of an array in C. We say that C is an $(r_P; 2s)$ PMDS code if, for every $(\sigma_1, \ldots, \sigma_t)$ where $\sum_{j=1}^{t} \sigma_j = 2s$, C is an $(r_P; \sigma_1, \ldots, \sigma_t)$-erasure correcting code.

Constructions of $(r; s)$ PMDS codes have been shown to exist for all r and s provided large enough field sizes [20]. For the purposes of our problem, we assume that $m_P = 2$ and that the information dimension of each of the two files we encode is $k = \frac{1}{2}(2(n_P - r_P) - 2s) = n_P - r_P - s$ and where $n = n_P$ so that we can interpret for instance the information for the first file being contained in a systematic code that appears in the first row and the information for the second file appearing as the second row in our codeword according to the previous definition. Then according to Definition 1, we can recover file 1 in the following ways:

1) File 1 can be recovered by collecting the k systematic strands for File 1 which appear in the first row.
2) File 1 can be recovered by collecting $n_P - r_P = k + s$ strands out of the n strands in the first row.
3) File 1 can be recovered by collecting $2k$ distinct strands whereby at least $n_P - r_P - 2s = k - s$ and at most $n_P - r_P = k + s$ originate from the first row.

Since the code is symmetric, the ways for recovering file 2 mirror those for recovering file 1. By using the same notations of $t(i), T(b)$ and the Markov chain properties as mentioned in Section III-B, the next theorem is proved similarly.

Theorem 3. For any $1 \leq k \leq n$ and $m = 2$, $T(n, k; 2, 1) \leq T^{\mathcal{C}_3}_{\max}(1)$ and:

$$T^{\mathcal{C}_3}_{\max}(1) = 2n \sum_{j=0}^{s} \sum_{h=0}^{k-j} \frac{k\binom{n-k}{j}\binom{n}{h}}{\binom{2n}{k-1+j+h}} \cdot \frac{H_{2n} - H_{2n-(k+j+h)}}{2n-(k-1+j+h)}$$

$$+2n \sum_{i=1}^{k-1} \sum_{h=0}^{k-s} \frac{\binom{k}{i-1}\binom{n-k}{k+s-i}\binom{n}{h}}{\binom{2n}{k+s-1+h}} \cdot \frac{(k-i+1)(H_{2n}-H_{2n-(k+s+h)})}{2n-(k+s-1+h)}$$

$$+2n \sum_{i=0}^{k-1} \sum_{h=0}^{k-s} \frac{\binom{k}{i}\binom{n-k}{k+s-i-1}\binom{n}{h}}{\binom{2n}{k+s-1+h}} \cdot \frac{(n-2k-s+i+1)(H_{2n}-H_{2n-(k+s+h)})}{2n-(k+s-1+h)}$$

$$+2n \sum_{i=1}^{k-s} \sum_{h=k+s+1}^{n} \frac{\binom{k}{i-1}\binom{n-k}{k-s-i}\binom{n}{h}}{\binom{2n}{k-s-1+h}} \cdot \frac{(k-i+1))(H_{2n}-H_{2n-(k-s+h)})}{2n-(k-s-1+h)}$$

$$+2n \sum_{i=0}^{k-s} \sum_{h=k+s+1}^{n} \frac{\binom{k}{i}\binom{n-k}{k-s-i-1}\binom{n}{h}}{\binom{2n}{k-s-1+h}} \cdot \frac{(n-2k+s+i+1)(H_{2n}-H_{2n-(k-s+h)})}{2n-(k-s-1+h)}$$

$$+2n \sum_{i=0}^{k-1} \sum_{j=\max(0,k-s-i)}^{k+s-i-1} \frac{\binom{k}{i}\binom{n-k}{j}\binom{n}{2k-(i+j)}}{\binom{2n}{2k}} (H_{2n} - H_{2n-2k}).$$

IV. Lower Bounds

In this section, we present two lower bounds on the value of $T(n, k; m, 1)$. The first bound does not depend on n, while the second presents an improvement considering it.

Lemma 1. For any n, k, m it holds $T(n, k; m, 1) \geq \frac{k(m+1)}{2}$.

In order to consider the effect of n on the value of $T(n, k; m, 1)$, we obtain a tighter lower bound on $T(n, k; m, 1)$ compared with Lemma 1.

Theorem 4. For any n, k, m it holds that

$$T(n, k; m, 1) \geq mnH_{mn} - n\sum_{i=1}^{m} H_{mn-ki} \geq \frac{k(m+1)}{2}.$$

The asymptotic behavior of this bound is given in the next corollary.

Corollary 3. For fixed $m, k, R = \frac{k}{n}$ it hold that $\lim_{n\to\infty} T(n, k; m, 1) \geq \lim_{n\to\infty}(mnH_{mn} - n\sum_{i=1}^{m} H_{mn-ki}) = \frac{k(m+1)}{2}$. Also, for fixed $0 < R < 1$, $\lim_{n\to\infty} \frac{T(n,k;m,1)}{n} \geq \lim_{n\to\infty} mH_{mn} - \sum_{i=1}^{m} H_{mn-ki} = \frac{R(m+1)}{2}$.

V. Comparisons and Evaluations

In this section, we will conduct a comparative analysis of the coding schemes introduced in Section III, focusing on their expected retrieval time, variance, and probability distribution. Our evaluation will commence with a comparison of expected retrieval times. Then, we will present and discuss the simulation results of the three coding schemes. Finally, we will conclude which of the schemes is superior. This knowledge proves pivotal for the optimization of DNA storage systems, as our objective is to minimize the number of samples required for file recovery.

First, we note that for two files ($m = 2$) the first coding scheme is superior of the second one in terms of the expectation for the number of reads. This is proved in the next lemma.

Lemma 2. For any $1 \leq k \leq n, m = 2$. We have that, $T^{\mathcal{C}_2}_{\max}(1) \geq T^{\mathcal{C}_1}_{\max}(1)$.

For each $\mathcal{C}_i, i \in [3]$, we conducted a simulation comprising 10 million experiments with parameters $n = 35$, $m = 2$, $k = 20$, and for $\mathcal{C}_3$, $s = 2$; see Fig. 1. We assessed the values of $T^{\mathcal{C}_i}_{\max}(1)$. Furthermore, we assess the probability distribution of the schemes, considering them to be normally distributed based on prior research [21], [22] that has demonstrated this tendency using the Central Limit Theorem. To utilize this distribution, the long version of the paper [16] provides a methodology for computing the variance. Consequently, employing the normal distribution, we determine the minimum sample size required to ensure confidence levels of 90%, 95%, and 99% for each coding scheme. Although $T^{\mathcal{C}_1}_{\max}(1)$ might have the smallest

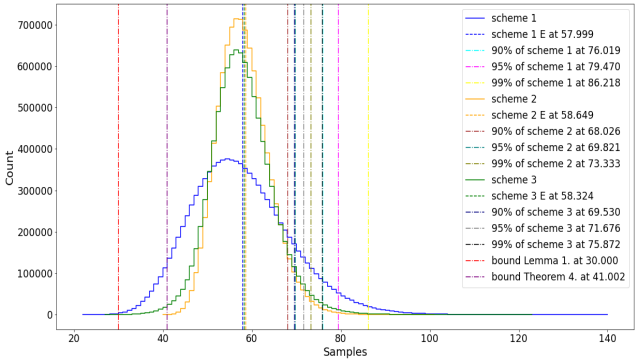

Fig. 1: Illustrates the distribution of necessary sample sizes for file recovery and confidence levels of 90%, 95%, and 99% across 3 coding schemes. And the lower bounds specified in Theorem 4 and in Lemma 1.

value, $T^{\mathcal{C}_2}_{\max}(1)$ and $T^{\mathcal{C}_3}_{\max}(1)$ demonstrate greater stability. Notably, the number of samples ensuring a 95% successful file recovery significantly differs from $T^{\mathcal{C}_1}_{\max}(1)$; however, it closely aligns with $T^{\mathcal{C}_2}_{\max}(1)$ and $T^{\mathcal{C}_3}_{\max}(1)$. While $\mathcal{C}_3$ mirrors $\mathcal{C}_2$ in terms of expectation with minor discrepancies, $\mathcal{C}_2$ exhibits superior stability, albeit not as pronounced as $\mathcal{C}_1$. The average values of the expected number of samples for file recovery across all three coding schemes exhibited negligible disparities compared to their respective expected value as analyzed in Section III all registering at 0.00. Similarly, for the minimum sample sizes required to attain confidence levels of 90%, 95%, and 99%, differences between the suggested normal distribution and

experimental simulations were minor. Full simulation results are detailed in the extended version [16]. Considering this comprehensive analysis, the coding scheme $\mathcal{C}_2$ emerges as a preferable choice. In essence, while expected values are crucial, factors like stability also wield significant influence. Thus, identifying the optimal scheme demands meticulous analysis and consideration of various factors, pivotal for achieving the overarching goal of minimization.

VI. Random Access Expectation for Multiple Files

In this section, we extend the results in the paper to randomly accessing multiple files, i.e., $a > 1$. We will analyze the expectation results of the first two coding schemes and show how to extend the bound from Theorem 4. For both analyses, let us use the same notations of $t(i), T(b)$ and the Markov chain properties as mentioned in Section III-B. We start with the local MDS scheme. We let $F \subseteq [m], |F| = a$ be the set of requested files. Assume without loss of generality that $F = \{1, \ldots, a\}$ Then, the Markov chain is described as follows.

- **States definition:** The set of states is: $\mathcal{S}_3 = \{\boldsymbol{f} = (f_1, \ldots, f_a, f_{a+1}) \mid \forall i \in F, f_i \leq n, f_{a+1} \leq mn - an\}$. where for each i in F, f_i is the number of strands drawn from the n encoded strands of file i and f_{a+1} is the number of strands that were drawn from the other files (i.e., the other $mn - an$ strands). Given state $\boldsymbol{f}$, let $(\boldsymbol{f}_{-i}, b)$ denote a new state where only the i-th value of $\boldsymbol{f}$ changes to b, that is, $(\boldsymbol{f}_{-i}, b) \triangleq (f_1, f_2, \ldots, b, \ldots, f_{a+1})$.
- **Transition matrix:** The valid transitions in M (i.e., $M_{\boldsymbol{f},\boldsymbol{y}} \neq 0$) and their values are:

$$M_{\boldsymbol{f},(\boldsymbol{f}_{-i},f_i+1)} = \frac{n - f_i}{mn - \widehat{\boldsymbol{f}}}, 1 \leq i \leq a,$$

$$M_{\boldsymbol{f},(\boldsymbol{f}_{-(a+1)},f_{a+1}+1)} = \frac{n \cdot (m-a) - f_{a+1}}{mn - \widehat{\boldsymbol{f}}}, i = a+1.$$

The next claim provides a closed formula for $M^{\widehat{\boldsymbol{f}}}_{s_0,\boldsymbol{f}}$,which holds for the non-absorbing states.

Claim 2. $M^{\widehat{\boldsymbol{f}}}_{s_0,\boldsymbol{f}} = \frac{\binom{n}{f_1}\binom{n}{f_2}\binom{n}{f_3}\cdots\binom{n}{f_a}\binom{n(m-a)}{f_{a+1}}}{\binom{mn}{\widehat{\boldsymbol{f}}}}$

Proof. At state $\boldsymbol{f}$, we have $\binom{n}{f_i}$ options to choose f_i encoded strands of file i for each i in F and $\binom{n(m-a)}{f_{a+1}}$ options to choose f_{a+1} strands from the rest of the strands in the pool, considering all possibilities for drawing a total of $\widehat{\boldsymbol{f}}$ strands out of the mn strands, which is $\binom{mn}{\widehat{\boldsymbol{f}}}$. □

- **Absorbing states:** These are the states that allow to recover the files in F, so the drawing process ends. For $\mathcal{C}_1$, the absorbing states are those where we drew the k-th strand from file $j \in F$, which is last to be recovered, i.e., we already read at least k strands from the other files in F and j is the last one. Denote Θ_1 as the set of absorbing states. Our approach involves investigating the transient states prior to absorption since those states determine a specific absorbing state. Denote $\mathcal{G}_1$ as the set of states reachable from a non-absorbing state to an absorbing one.

  $\mathcal{G}_1 \triangleq \{\boldsymbol{f} \mid \exists j \in F \forall i \in F \setminus \{j\} (f_i \geq k \text{ and } f_j = k-1)\}$.

  For $g \in \mathcal{G}_1$ with j as the last file to be recovered, it is possible to reach exactly 1 absorbing state $\theta = (g_{-j}, k) \in \Theta_1$. Thus the probability of reaching θ from g is: $M^{(\widehat{g})}_{s_0,g} \cdot M_{g,\theta}$, which follows from the definition of $\mathcal{G}_1$.

Theorem 5. For any $1 \leq k \leq n$ and $1 \leq a \leq m$, it holds that

$$T(n,k;m,a) \leq T^{\mathcal{C}_1}_{\max}(a) = \sum_{j=1}^{a} \sum_{g \in \mathcal{G}_1, g_j = k-1} \frac{\binom{n}{g_1}\binom{n}{g_2}\cdots\binom{n}{g_j}\cdots\binom{n}{g_a}\binom{n(m-a)}{g_{a+1}}}{\binom{mn}{\widehat{g}}} \cdot M_{g,(g_{-j},k)} \cdot mn(H_{mn} - H_{mn-(\widehat{g}+1)}).$$

Proof. Assume without loss of generality that $F = \{1, \ldots, a\}$. We wish to find $\mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)]$. We let Y be the random variable representing in which absorbing state the collection process ends. The expectation $\mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)]$ is conditioned on Y. Hence,

$$\begin{aligned}
&\mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)] = \mathbb{E}_Y[\mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)|Y]] \\
&= \sum_{\theta \in \Theta_1} \Pr(Y = \theta) \cdot \mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)|Y = \theta] \\
&=^{(*)} \sum_{j=1}^{a} \sum_{g \in \mathcal{G}_1, g_j = k-1} M^{(\widehat{g})}_{s_0,g} \cdot M_{g,g_{-j}(k)} \cdot \mathbb{E}[T(\widehat{g}+1)] \\
&= \sum_{j=1}^{a} \sum_{g \in \mathcal{G}_1, g_j = k-1} \frac{\binom{n}{g_1}\binom{n}{g_2}\cdots\binom{n}{g_j}\cdots\binom{n}{g_a}\binom{n(m-a)}{g_{a+1}}}{\binom{mn}{\widehat{g}}} \\
&\cdot M_{g,(g_{-j},k)} \cdot mn(H_{mn} - H_{mn-(\widehat{g}+1)}),
\end{aligned}$$

where $(*)$ follows from the definitions of $\mathcal{G}_1$ so we iterate over all absorbing states $\sum_{\theta \in \Theta_1} \Pr(Y = \theta) = \sum_{j=1}^{a} \sum_{g \in \mathcal{G}_1, g_j = k-1} M^{(\widehat{g})}_{s_0,g} \cdot M_{g,(g_{-j},k)}$. Since the code is symmetric it implies that $\mathbb{E}[\nu_{(m,F)}(\mathcal{C}_1)] = T^{\mathcal{C}_1}_{\max}(a)$. □

The next theorem states the extension result of the global MDS scheme for accessing multiple files.

Theorem 6. For any $1 \leq k \leq n$ and $1 \leq a \leq m$, it holds that

$$T(n,k;m,a) \leq T^{\mathcal{C}_2}_{\max}(a) = mn\,(H_{mn} - H_{mn-mk}) - \frac{mn}{\binom{mn}{ak}}\left(\sum_{j=0}^{mk-ak-1}\binom{ak-1+j}{ak-1}H_{mn-(ak+j)} - H_{mn-mk}\binom{mk-1}{ak}\right).$$

Lastly, we present our lower bound of $T(n,k;m,a)$.

Theorem 7. Let $\mathcal{C}$ be an $[mn, mk]$ code. It holds that

$$T(n,k;m,a) \geq \frac{mn}{\binom{m}{a}} \sum_{i=a-1}^{m-1} \binom{i-1}{a-1}(m-i)(H_{mn-ki} - H_{mn-k(i+1)}).$$

VII. Conclusion And Future Work

This paper investigates the random access coverage depth problem in practical scenarios, focusing on storing m files and retrieving portions of them. By analyzing the maximal expected number of samples required for file recovery and $T(n,k;m,a)$, the study sheds light on the structural attributes of various coding schemes that impact random access expectations and probability distributions. While the findings represent significant progress in this domain, several intriguing avenues for future research remain unexplored. In our future research we will extend our analysis to encompass the more general setup of comparing across all 3 schemes when $a, m \geq 2$, and plan to find the exact value of $T(n,k;m,a)$ and study the probability distribution additionally, attention will be directed towards addressing challenges related to the noisy channel of DNA storage, specifically concerning Problem 1 and Problem 2.

VIII. Acknowledgment

The authors wish to thank Zohar Nagel for her progress on initial results on the problems studied in the paper. They also thank Ido Feldman for helpful discussions.

REFERENCES

[1] J. Rydning, "Worldwide IDC global datasphere forecast, 2022–2026: Enterprise organizations driving most of the data growth," tech. rep., Technical Report, 2022.

[2] E. M. LeProust, B. J. Peck, K. Spirin, H. B. McCuen, B. Moore, E. Namsaraev, and M. H. Caruthers, "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process," *Nucleic acids research*, vol. 38, no. 8, pp. 2522–2540, 2010.

[3] L. Anavy, I. Vaknin, O. Atar, R. Amit, and Z. Yakhini, "Data storage in DNA with fewer synthesis cycles using composite DNA letters," *Nature biotechnology*, vol. 37, no. 10, pp. 1229–1236, 2019.

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

[5] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, *et al.*, "Random access in large-scale DNA data storage," *Nature biotechnology*, vol. 36, no. 3, pp. 242–248, 2018.

[6] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific reports*, vol. 7, no. 1, p. 5011, 2017.

[7] I. Shomorony, R. Heckel, *et al.*, "Information-theoretic foundations of DNA data storage," *Foundations and Trends® in Communications and Information Theory*, vol. 19, no. 1, pp. 1–106, 2022.

[8] S. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 3, pp. 230–248, 2015.

[9] D. D. S. Alliance, "Preserving our digital legacy: an introduction to DNA data storage," 2021.

[10] S. Chandak, K. Tatwawadi, B. Lau, J. Mardia, M. Kubit, J. Neu, P. Griffin, M. Wootters, T. Weissman, and H. Ji, "Improved read/write cost tradeoff in DNA-based data storage using LDPC codes," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 147–156, IEEE, 2019.

[11] D. Bar-Lev, O. Sabary, R. Gabrys, and E. Yaakobi, "Cover your bases: How to minimize the sequencing coverage in DNA storage systems," *arXiv preprint arXiv:2305.05656*, 2023.

[12] P. Erdős and A. Rényi, "On a classical problem of probability theory," *Magyar Tud. Akad. Mat. Kutató Int. Közl*, vol. 6, no. 1, pp. 215–220, 1961.

[13] W. Feller, *An introduction to probability theory and its applications, Volume 2*, vol. 81. John Wiley & Sons, 1991.

[14] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.

[15] D. J. Newman, "The double dixie cup problem," *The American Mathematical Monthly*, vol. 67, no. 1, pp. 58–61, 1960.

[16] H. Abraham, R. Gabrys, and E. Yaakobi, "Covering all bases: The next inning in DNA sequencing efficiency," *arXiv preprint*, 2024.

[17] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific reports*, vol. 9, no. 1, p. 9663, 2019.

[18] A. Gruica, D. Bar-Lev, A. Ravagnani, and E. Yaakobi, "Reducing coverage depth in DNA storage: A combinatorial perspective on random access efficiency," *arXiv preprint*, 2024.

[19] I. Preuss, B. Galili, Z. Yakhini, and L. Anavy, "Sequencing coverage analysis for combinatorial DNA-based storage systems," *bioRxiv*, pp. 2024–01, 2024.

[20] R. Gabrys, E. Yaakobi, M. Blaum, and P. H. Siegel, "Constructions of partial MDS codes over small fields," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3692–3701, 2018.

[21] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[22] C. J. Geyer, "Practical markov chain monte carlo," *Statistical Science*, vol. 7, no. 4, pp. 473–483, 1992.