Sequencing coverage analysis for combinatorial DNA-based storage systems

Inbal Preuss, Ben Galili, Zohar Yakhini, and Leon Anavy

***Abstract*—This study introduces a novel model for analyzing and determining the required sequencing coverage in DNA-based data storage, focusing on combinatorial DNA encoding. We explore the application of the coupon collector model for combinatorial-letter reconstruction, post-sequencing, which ensure efficient data retrieval and error reduction. We use a Markov Chain model to compute the probability of error-free reconstruction. We develop theoretical bounds on the decoding probability and use empirical simulations to validate these bounds. The work contributes to the understanding of sequencing coverage in DNA-based data storage, offering insights into decoding complexity, error correction, and sequence reconstruction. We provide a Python package that takes the code design and other message parameters as input, and then computes the required read coverage to guarantee reconstruction at a given desired confidence.**

I. INTRODUCTION

THE growing volume of the world's digital data and the limitations of existing storage technologies motivate the need for new and innovative storage solutions [1].

DNA-based data storage (or DNA-based storage) emerges as a viable solution, offering unmatched density and durability. This novel approach involves the synthesis, storage, and sequencing of DNA molecules to encode, store and retrieve information. However, challenges such as short, error-prone strands and limitations in current synthesis technologies still remain [2] [3] [4] [5] [6] [7] [8].

While DNA-based storage stands as a promising technology, and the cost of DNA sequencing has been decreasing, it remains significantly more expensive than reading from established archival storage solutions [9] [10] [11] [12]. In the context of DNA sequencing costs and throughput, recent work [13] defined the DNA coverage depth problem, which considers the expected sample size, to guarantee successful decoding of the information. A related concept was suggested by Chandak et al. [14], who explored the balance of writing and reading costs in DNA-based data storage, studying the LDPC-based coding schemes.

Combinatorial DNA encoding is a recently introduced encoding scheme, which uses a set of clearly distinguishable DNA shortmers to construct large combinatorial alphabets, where each letter is encoded by a subset of shortmers [15]. The nature of these combinatorial alphabets minimizes mix-up errors, while also ensuring the robustness of the system. combinatorial shortmer encoding is an extension of other composite coding schemes, such as [2] [16] [17].

This work presents the first model for analyzing the sequencing coverage depth problem under combinatorial DNA encoding. In this work, we define and study a new model to compute the required coverage depth. While the model presented in [13] assumes 1D encoding applied on the strands, our model considers 2D (inner-outer) MDS codes. Each sequence is encoded using the inner-code, to protect against symbol errors, while the outer-code adds redundancy to a block of sequences, protecting against sequence-level errors. This allows for a more thorough and detailed analysis of the required sequencing coverage when using the inner-outer code approach, a widely used coding technique in DNA-based storage [18] [19] [3] [2].

We first address the question of reconstructing a single combinatorial letter by utilizing a reduction of this problem to the well-known coupon collector's problem. This provides a framework for determining the required number of reads to ensure that at least one copy of every member shortmer in the combinatorial letter is observed [13] [20] [21] [22] [23]. For this purpose, we present a Markov Chain approach to calculate decoding probabilities and provide computer code. We also generalize this model by considering a threshold for the minimum number of copies of each shortmer required for letter reconstruction.

Next, we analyze the decoding probabilities of full-length combinatorial sequences that constitute a single complete message encoded using combinatorial DNA. We provide bounds on the decoding probability given the number of analyzed reads, and present an operational algorithm for determining the required coverage of reads. We explore our coverage depth model on different design parameters and compare the results to simulation experiments of combinatorial DNA reading.

Lastly, we provide computer code implementing our coverage model that, given a sequence and message design, outputs the read coverage required for recovering the data with a user defined confidence level. This work combines theoretical progress represented by studying the coverage depth problem for combinatorial DNA-based storage, and also the practical

December 31, 2023

This work was supported in part by European Union's Horizon Europe Research and Innovation Programme under Grant 101115134.

Inbal Preuss is with the School of Computer Science, Reichman University, Herzliya, 4610101, Israel. (e-mail: inbal.preuss@post.runi.ac.il).

Ben Galili is with the Faculty of Computer Science, Technion, Haifa, 3200003, Israel, and the School of Computer Science, Reichman University, Herzliya, 4610101, Israel (e-mail: benga@campus.technion.ac.il).

Zohar Yakhini is with the School of Computer Science, Reichman University, Herzliya, 4610101, Israel, and the Faculty of Computer Science, Technion, Haifa, 3200003, Israel. (e-mail: zohar.yakhini@runi.ac.il).

Leon Anavy is with the School of Computer Science, Reichman University, Herzliya, 4610101, Israel. (e-mail: leon.anavy@post.runi.ac.il).

Code related to this article are available online, at [https://github.com/InbalPreuss/combinatorial_sequencing_coverage].

No animal or human subjects were involved in this work.

aspect supporting the design and implementation of such systems.

II. Results

The decoding complexity is analyzed here by breaking the process down into its basic components. First, the decoding probability of a single combinatorial letter is analyzed, considering various design parameters and decoding approaches. Next, this paper addresses the decoding of a single combinatorial letter, while considering the use of error correction codes with varying redundancy levels. Finally, the decoding of a complete combinatorial DNA message is analyzed, considering a general 2D error correction MDS code (i.e., a code that protects against sequence dropouts as well as errors on each sequence).

A. *Reconstruction of a single combinatorial letter*

Let Ω be a set of valid k-mers used for a combinatorial DNA-based data storage system. Consider a binomial combinatorial alphabet Σ with $|\Sigma| \leq \binom{N}{K}$ letters where each letter $\sigma \in \Sigma$ consists of a subset of size K of k-mers from Ω. This subset is referred to as the member k-mers of σ. Let R be the number of analyzed reads of a given combinatorial letter. We define a decoding algorithm in which we accumulate reads until we observe at least t copies of K unique k-mers from Ω. These K k-mers are refrred to as the inferred member k-mers, and are used to reconstruct a combinatorial letter σ' (See Algorithm 1).

Algorithm 1: Decoding Algorithm for a Single Combinatorial Letter

Input: A set of R reads, a list of N k-mers from the set Ω, a robustness threshold t
Output: A set of K inferred k-mers or FALSE if decoding fails

define the binomial combinatorial alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_{|\Sigma|}\}$ with $|\Sigma| \leq \binom{N}{K}$;
initialize a counter for each k-mer in Ω;
while *there are reads to analyze in* R **do**
    extract next read r from R;
    increment the counter for the k-mer observed in the read;
    if *the counters for* K *k-mers are larger or equal to* t **then**
        return *these* K *k-mers as the inferred member k-mers of* σ';
return *FALSE*;

To analyze the probability of decoding a single combinatorial letter, we first assume that each read uniformly draws one of the K member k-mers. Let $T(K,t)$ be a random variable representing the number of reads analyzed until the decoding algorithm successfully stops. Let $\pi(K,t)(R)$ be the probability of stopping with a successful inference after at most R_{single} reads.

$$\pi(K,t)(R) = P(T(K,t) \leq R) \quad (1)$$

For $t = 1$ the random variable $T(K,t)$ represents the classical coupon collector model [24] and we get (See Appendix B:

$$\pi(K,t=1)(R) = \sum_{i=0}^{K}(-1)^i \binom{K}{i}\left(1 - \left(\frac{i}{K}\right)\right)^R \quad (2)$$

$$E\big(T(K,1)\big) = K \cdot H_K \quad (3)$$

where $H_K = \sum_{i=1}^{K} \frac{1}{i}$ is the K[th] harmonic number.

For $t > 1$ we can obtain [23]:

$$E\big(T(K,t)\big) = K(\ln(K) + (t-1)\ln(ln(K)) + O(1)) \quad (4)$$

To calculate $\pi(K,t)(R)$ for $t > 1$, we use a Markov Chain (MC) formulation. Each state in the MC represents the status of the member k-mers in σ, in terms of the number of times each has been seen. Specifically, a state is represented by a vector:

$$\big(v(0), \ldots, v(t)\big); v(i) \in \{0, \ldots, K\} \quad (5)$$

for $0 \leq j < t, v(j)$ indicates the number of member k-mers seen exactly j times, while $v(t)$ indicates the number of member k-mers seen t times or more.

Clearly, this vector satisfies:

$$\sum_{i=0}^{t} v(i) = K \quad (6)$$

$$\sum_{i=0}^{t} i * v(i) \leq R \quad (7)$$

and when $v(t) = 0$ the inequality in (7) hold as equality $\sum_{i=0}^{t} i * v(i) = R$. We also note that since there are $t+1$ values in the vector $(v(0), v(1), \ldots, v(t))$, there are a total of $N = \binom{K+t}{t}$ possible solutions to the equation, representing N states.

For example, considering $K = 10$ member k-mers and a threshold $t = 2$. The following states can be defined:

(10,0,0): All 10 k-mers have not been seen yet. This is the case before we start analyzing the reads.

(8,2,0): After analyzing two reads, two unique k-mers have been observed exactly once while the remaining 8 k-mers have not been observed yet.

(7,2,1): After analyzing at least four reads, two unique k-mers have been observed exactly once, one k-mer has been observed 2 times or more and the remaining 7 k-mers have not been observed yet.

We define the following transition matrix A where each transition is defined by the observation of one read.

$$A\big[\big(v(0), \ldots, v(i), v(i+1) \ldots, v(t)\big)\big]\big[\big(v(0), \ldots, v(i) - 1, v(i+1) + 1 \ldots, v(t)\big)\big] = \frac{v(i)}{K} \quad (8)$$

This represents observing one of the $v(i)$ k-mers that were observed $i < t$ times.

And,

$$A[(v(0), \ldots, v(i), \ldots, v(t))][(v(0), \ldots, v(i), \ldots, v(t))] = \frac{v(t)}{K} \quad (9)$$

This represents observing one of the $v(t)$ k-mers that were observed at least t times.

For example, the first two transitions are:

$$P\big(s_0 = (\mathbf{10}, 0,0), \mathrm{s}_1 = (9,1,0)\big) = \frac{v(0)}{K} = 1$$

$$P\big(\mathrm{s}_1 = (\mathbf{9}, 1,0), \mathrm{s}_2 = (8,2,0)\big) = \frac{v(0)}{K} = \frac{9}{10}$$

This happens when one out of the 9 yet unseen k-mers is drawn.

$$P\big(\mathrm{s}_1 = (9, \mathbf{1}, 0), \mathrm{s}_2 = (9,0,1)\big) = \frac{v(1)}{K} = \frac{1}{10}$$

This happens only when the first observed k-mer is observed again.

To get $\pi(K,t)(R)$ we set the initial state to be

$$s_0 = (v(0) = \mathrm{K}, v(1) = 0, \dots, v(t) = 0) \quad (10)$$

That is $P_0 = (P(s_0) = 1,0, \dots ,0)$ is the state distribution vector.

We derive the distribution vector over the states after R steps,

$$P_R = P_0 A^R \quad (11)$$

Let $s_f = (0,0, \dots, v(t) = K)$ be the desired state in which all K k-mers have been observed at least t times.

$$\pi(K,t)(R) = P_R\big(s_f\big) \quad (12)$$

Fig. 1 and Appendix A. demonstrate the state distribution vector for several values of R using $K = 5$ member k-mers and a threshold of $t = 1$. Clearly, after analyzing the first read, a single k-mer is observed once while the other four have not been observed yet. With $R = 5$, the probability of having seen all unique coupons reached $\pi(5,1)(5) = \prod_{i=1}^{5} \frac{i}{5} = 0.038$. At $R = 15$, this probability significantly increased to $\pi(5,1)(15) = 0.829$. Finally, at $R = 30$, the probability of observing all coupons was $\pi(5,1)(30) = 0.994$.

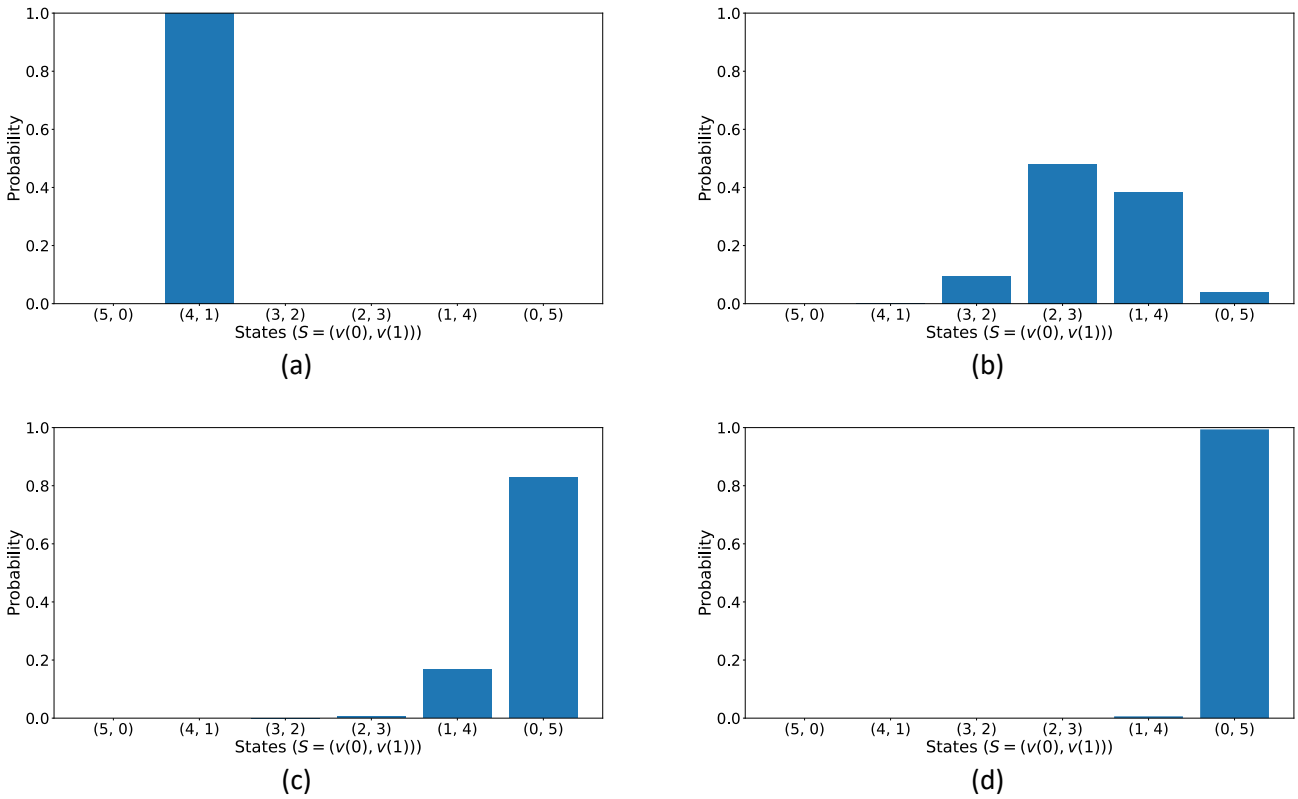

Fig. 1. Evolution of probability in the coupon collector model. (a) The probability distribution across the 6 states (X-axis) after observing $R = 1$ reads. (b-d) Similar to (a) with $R = 5, 15, 30$ respectivley. Calculated for $K = 5$, $t = 1$, and no errors, $\epsilon = 0$.

This algorithm ignores possible synthesis and sequencing error as it assumes that all observed k-mers come from the set of K valid k-mers. Introducing an error probability ϵ of observing an invalid k-mer requires a modified transition matrix B:

$$B[(v(0), \dots, v(i), v(i+1) \dots, v(t))][(v(0), \dots, v(i) - 1, v(i+1) + 1 \dots, v(t))] = (1-\epsilon)\frac{v(i)}{K} \quad (13)$$

This represents observing one of the $v(i)$ (valid) member k-mers that were observed $i < t$ times.

And,

$$B[(v(0), \dots, v(i), \dots, v(t))][(v(0), \dots, v(i), \dots, v(t))] = \frac{v(t)}{K}(1-\epsilon) + \epsilon \quad (14)$$

This represents observing one of the $v(t)$ k-mers that were observed at least t times, or observing an invalid k-mer.

For example, the first two transitions are:

$$P\big(s_0 = (\mathbf{10}, 0,0), \mathrm{s}_1 = (9,1,0)\big) = (1-\epsilon)\frac{v(0)}{K} = 1 - \epsilon$$

$$P\big(s_0 = (\mathbf{10}, 0,0), \mathrm{s}_1 = (10,0,0)\big) = (1-\epsilon)\frac{v(2)}{K} + \epsilon = \epsilon$$

$$P\big(\mathrm{s}_1 = (\mathbf{9}, 1,0), \mathrm{s}_2 = (8,2,0)\big) = (1-\epsilon)\frac{v(0)}{K} = (1-\epsilon)\frac{9}{10}$$

$$P\big(\mathrm{s}_1=(9,\mathbf{1},0),\mathrm{s}_2=(9{,}0{,}1)\big)=(1-\epsilon)\frac{v(1)}{K}=(1-\epsilon)\frac{1}{10}$$

$$P\big(\mathrm{s}_1=(9,\mathbf{1},0),\mathrm{s}_2=(9{,}1{,}0)\big)=(1-\epsilon)\frac{v(2)}{K}+\epsilon=\epsilon$$

Fig. 2 depicts the decoding probabilities for varying number of analyzed reads using different values for the threshold t. The calculated probabilities are compared to a simulation experiment. As expected, as t increases, more reads are required to reconstruct a combinatorial letter. Notably, when R reaches 100 or more, the probability effectively becomes 1, indicating full data recovery. This represents the balance between threshold level required for achieving precise combinatorial reconstruction and the read depth complexity.

Note that throughout this section, we ignored the possibility of an error that results in k-mer mix-up (i.e., the output of the decoding algorithm is different from the original combinatorial letter, $\sigma' \neq \sigma$). This is due to the assumptions that the design parameters render this error type very unlikely. We further discuss this issue in Section IV. Discussion.

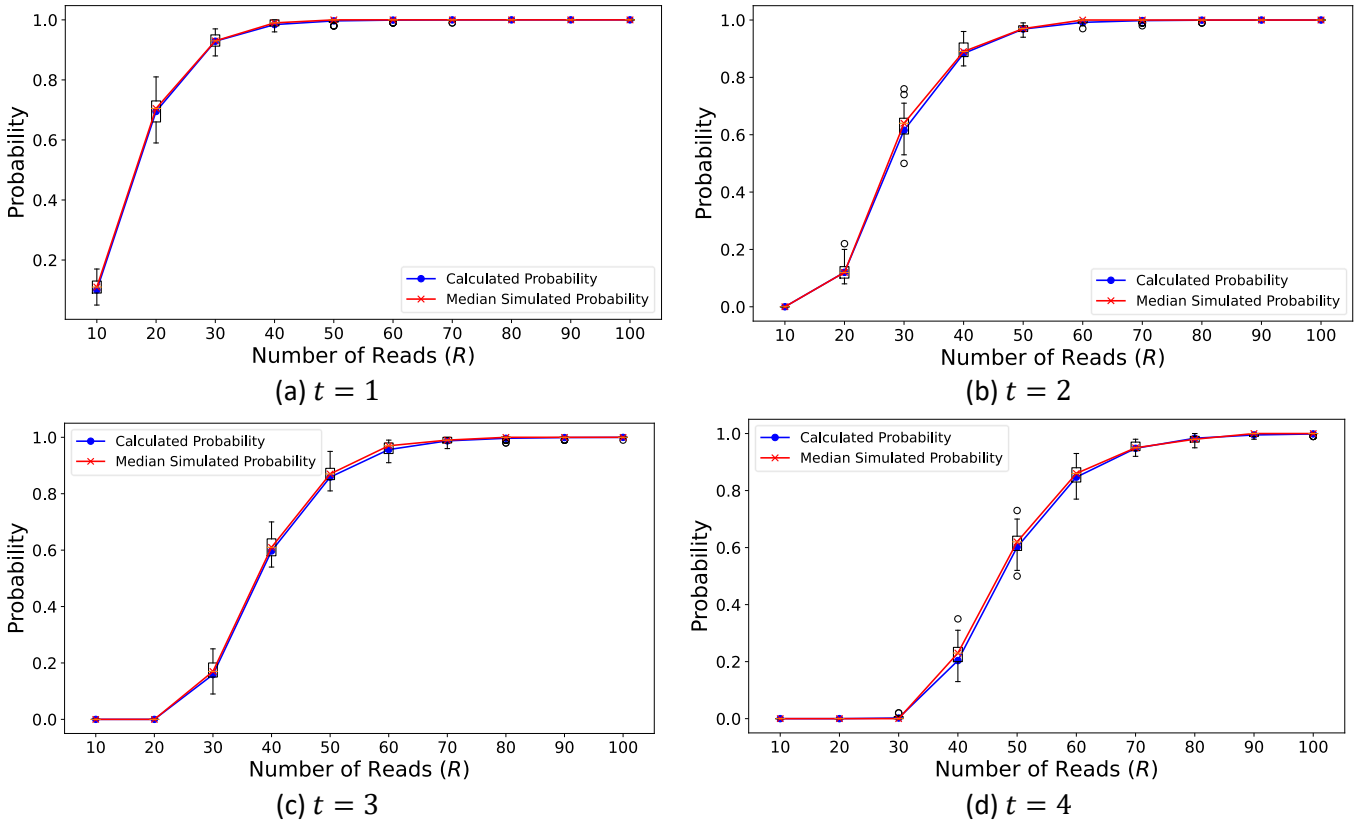

Fig. 2. Decoding probability for varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analyses were conducted for $K = 7$ and $\epsilon = 0.01$. (a) results for $t = 1$, the blue line corresponds to the calculated probability based on the MC model while the red line represents the median of 50 simulation runs, where each simulation calculates the success rate of 100 uniform drawing of R reads across K member k-mers. The simulation results are also presented as boxplots. (b-d). Like (a) with $t = 2, 3$ and 4 respectivley.

B. Reconstruction of a combinatorial sequence

Let $s = \sigma^{(1)}\sigma^{(2)} \ldots \sigma^{(m)}$ be a sequence of length m over the same binomial alphabet defined in the previous section. Assuming the use of a proper MDS error correction code, we say that decoding only $b \leq m$ letters is sufficient for decoding the complete sequence. Let R be the number of analyzed reads, fixing K and t, we denote $\pi(K,t)(R)$ as $\pi(R)$. Let W be a random variable representing the number of letters in s that were decoded. Assuming independence between the letters in s we get

$$W \sim Binom(m, \pi(R)) \tag{15}$$

We are interested in the probability of decoding the sequence s, P_{single}:

$$P_{single}(R,m,b) = P(W \geq b) = \sum_{i=b}^{m}\binom{m}{i}\pi(R)^i(1-\pi(R))^{m-i} \tag{16}$$

We can approximate this probability using the normal estimation (based on Central Limit Theorem).

$$W \dot{\sim} N\big(m\pi(R), m\pi(R)(1-\pi(R))\big) \tag{17}$$

$$P(W \geq b) = 1 - P(W < b) = 1 - \Phi\left(\frac{b-m\pi(R)}{\sqrt{m\pi(R)(1-\pi(R))}}\right) \tag{18}$$

Where Φ is the CDF of the standard normal distribution.

Fig. 3 presents the decoding probabilities of a combinatorial sequence with length $m = 100$, examining how the number of analyzed reads (R) affects the accuracy of sequence reconstruction across various redundancy levels ($b = 100, 95, 90, 85$) keeping other parameters constant ($K = 7$, $t = 4$). We observe that the probability of successful reconstruction varies significantly with different redundancy levels. Notably,

higher redundancy levels (lower b values) enable accurate reconstruction using fewer reads. These results also align with the results obtained from the Normal Approximation (Not shown). The results demonstrate the role of sequence level redundancy in affecting the likelihood of accurate reconstruction, making it an important tunable parameter in the overall design.

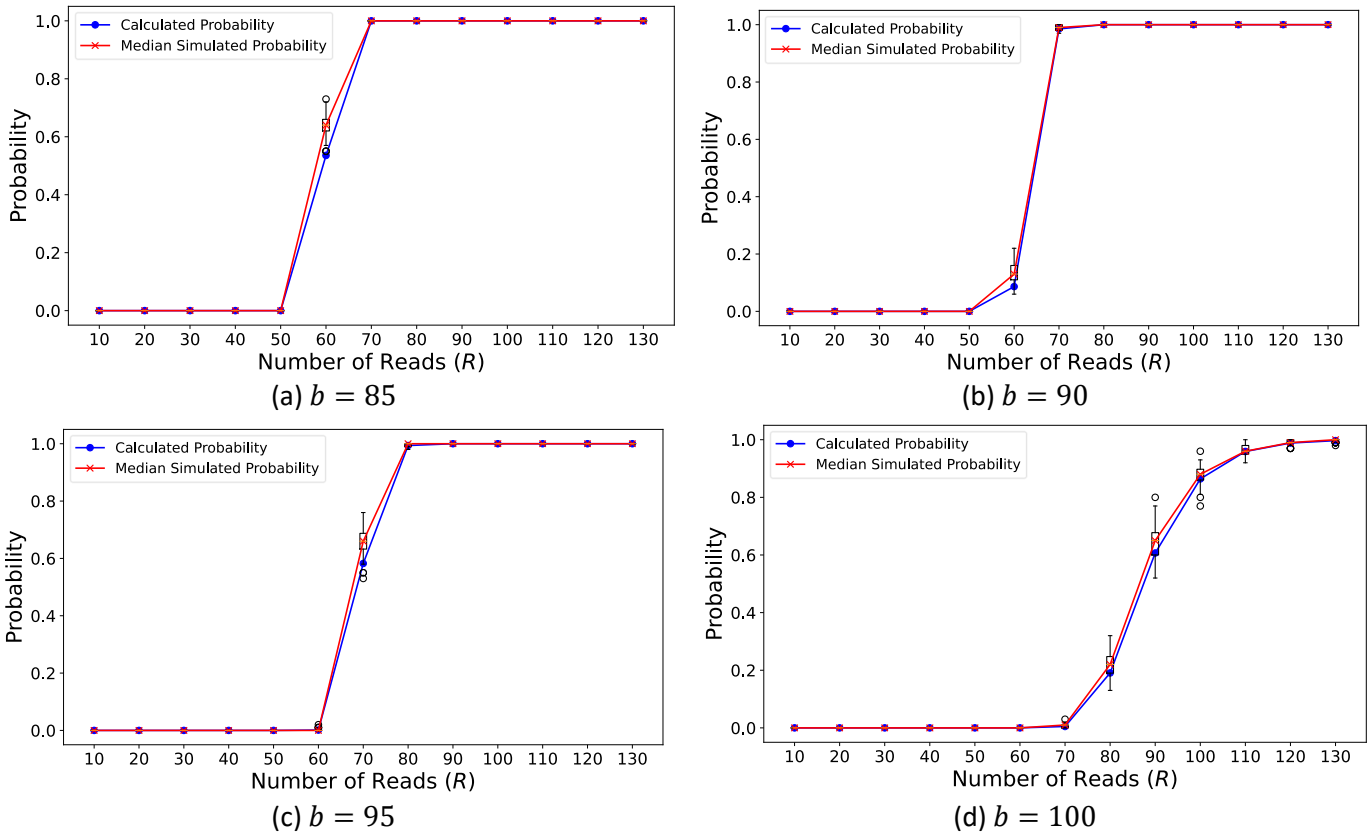

Fig. 3. Decoding probability of a complete combinatorial sequence with varying redundancy levels. Results shown for a sequence of length $m = 100$, with $K = 7$, and requiring $t = 4$. (a) Calculated decoding probability (blue line) as a function of the number of analyzed reads for redundancy level of $b = 85$. Median results from 50 simulation runs are presents (red line) with boxplots representing the distribution of the simulation results. Each simulation run represents 100 uniformly drawn sets of R reads, each comprising m letters drawn from $K = 7$ member k-mers. (b-d) Like (a) with $b = 90, 95$, and 100, respectively. All analyses incorporate an error rate of $\epsilon = 0.01$.

C. Reconstructing a complete combinatorial message

Let $M = \{s_i\}_i^l$ be a complete combinatorial message encoded using a binomial alphabet like in the previous sections. The message is encoded using l combinatorial sequences and, assuming proper MDS error correction code, $a \leq l$ of which are sufficient for the decoding of M.

Let R_{all} be the total number of analyzed reads over all sequences. We are interested in the probability of decoding at least a of l sequence using R_{all} reads, $P_{all}(R_{all}, l, a)$.

Fig. 4 presents an overview of the decoding process and the analysis steps for a complete combinatorial message.
First, the R_{all} reads are distributed between the l sequences, using, for example, the barcodes. Then, the decoding probability of each of the l sequences is determined using the derivation from the previous section. The decoding probability of a single letter is analyzed using the coupon collector's model. We now formally define each of these steps and analyze the decoding probability $P_{all}(R_{all}, l, a)$ or simply P_{all}.

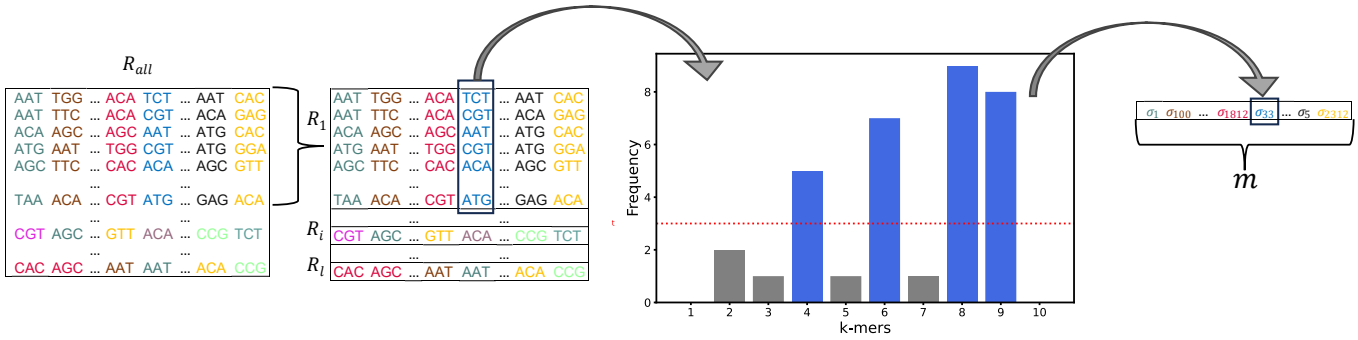

Fig. 4. Reconstructing a complete combinatorial message. (a) R_{all} reads are distributed between l sequences and at least a sequences need to be decoded (b) The decoding probability of each of the letters is analyzed using the coupon collector's model (Blue bins indicate the members k-mer) (c) Each sequence requires b of the m combinatorial letters to be decoded.

The distribution of the R_{all} reads across the l sequences is modeled using a multinomial distribution

$$(R_1, \dots, R_l) \sim Multinom\left(R_{all}, \left(\frac{1}{l}, \dots, \frac{1}{l}\right)\right) \quad (19)$$

Given a specific distribution of the R reads $(r_1, \dots, r_l)$, to successfully decode the message we need to decode at least a of the sequences.

$$P_{all}(r_1, \dots, r_l) = P\left(\sum_{i=1}^{l} I_i \geq a\right) \quad (20)$$

Where I_i is an indicator of decoding sequence s_i using r_i reads.

Using the law of total probability and setting $P(R_1 = r_1, \dots, R_l = r_l) = P(r_1, \dots, r_l)$:

$$P_{all} = \sum_{\substack{(r_1,\dots,r_l) \\ \sum r_i = R_{all}}} P(r_1, \dots, r_l) P_{all}(r_1, \dots, r_l) \quad (21)$$

Calculating P_{all} directly becomes infeasible even for small values of R_{all}, l and a. We therefore bound this probability.

First, we note that since for every sequence s_i we have

$$P(I_i) = P_{single}(r_i, m, b) \geq P_{single}(r_{min}, m, b)$$

Where $r_{min} = \min_{j=1,\dots,l} r_i$ and $\pi_{r_{min}}$ is obtained by using r_{min} in the coupon collector's model. If we plug this back to (20) we can define a new binomial random variable X that represents the number of sequences decoded:

$$X \sim Binom\left(l, P_{single}(r_{min}, m, b)\right) \quad (22)$$

And,

$$P_{all}(r_1, \dots, r_l) \geq P(X \geq a) \quad (23)$$

Yielding a lower bound on $P_{all}(R_{all}, l, a)$

$$P_{all} \geq P(X \geq a) \sum_{\substack{(r_1,\dots,r_l) \\ \sum r_i = R_{all}}} P(r_1, \dots, r_l) \quad (24)$$

In the multinomial distribution for $(R_1, \dots, R_l)$, many possible read distributions are very unlikely. We can further bound P_{all} by setting a constant value ρ and only considering read distributions for which $\min_{j=1,\dots,l}(r_j) \geq \rho$. Let X_ρ be a random variable representing the number of sequences decoded when the decoding probability of each sequence is calculated using ρ reads. That is, $X_\rho \sim Binom\left(l, P_{single}(\rho, m, b)\right)$.

We therefore have

$$P_{all} \geq P(X_\rho \geq a) \sum_{\substack{(r_1,\dots,r_l) \\ \sum r_i = R_{all} \\ \min_j r_j \geq T}} P(r_1, \dots, r_l) \quad (25)$$

Given a small $\delta > 0$, we check whether R_{all} reads are sufficient to decode the message with $1 - \delta$ confidence level.

$$P_{all} \geq 1 - \delta$$

This can be achieved by choosing ρ such that

(a) $P(X_\rho \geq a) \geq \sqrt{1 - \delta}$

And,

(b) $$\sum_{\substack{(r_1,\dots,r_l) \\ \sum r_i = R_{all} \\ \min_j r_j \geq \rho}} P(r_1, \dots, r_l) \geq \sqrt{(1 - \delta)} \quad (26)$$

Since X_ρ has a binomial distribution, we can find ρ for which condition (a) holds. For condition (b), we use Sanov's Theorem on the multinomial distribution as follows. For more on Sanov's Theorem and the behavior of multinomials, see [25].

Sanov's theorem bounds the probability that the distribution of the reads into barcodes significantly deviates from the expected uniform ($1/l$ for each) distribution, particularly where at least one sequence gets fewer than ρ reads. Fig. 5 demonstrates this using a simulation of 100,000 instances each drawn from the multinomial distribution with $p = \left(\frac{1}{50}, \dots, \frac{1}{50}\right)$ and $n = 4500$ or $n = 5000$. The plots show the distribution of the minimal values obtained. Clearly, increasing R_{all} reduces the probability of the minimal value to be below a fixed threshold ρ. Decreasing the threshold ρ yields a similar effect.

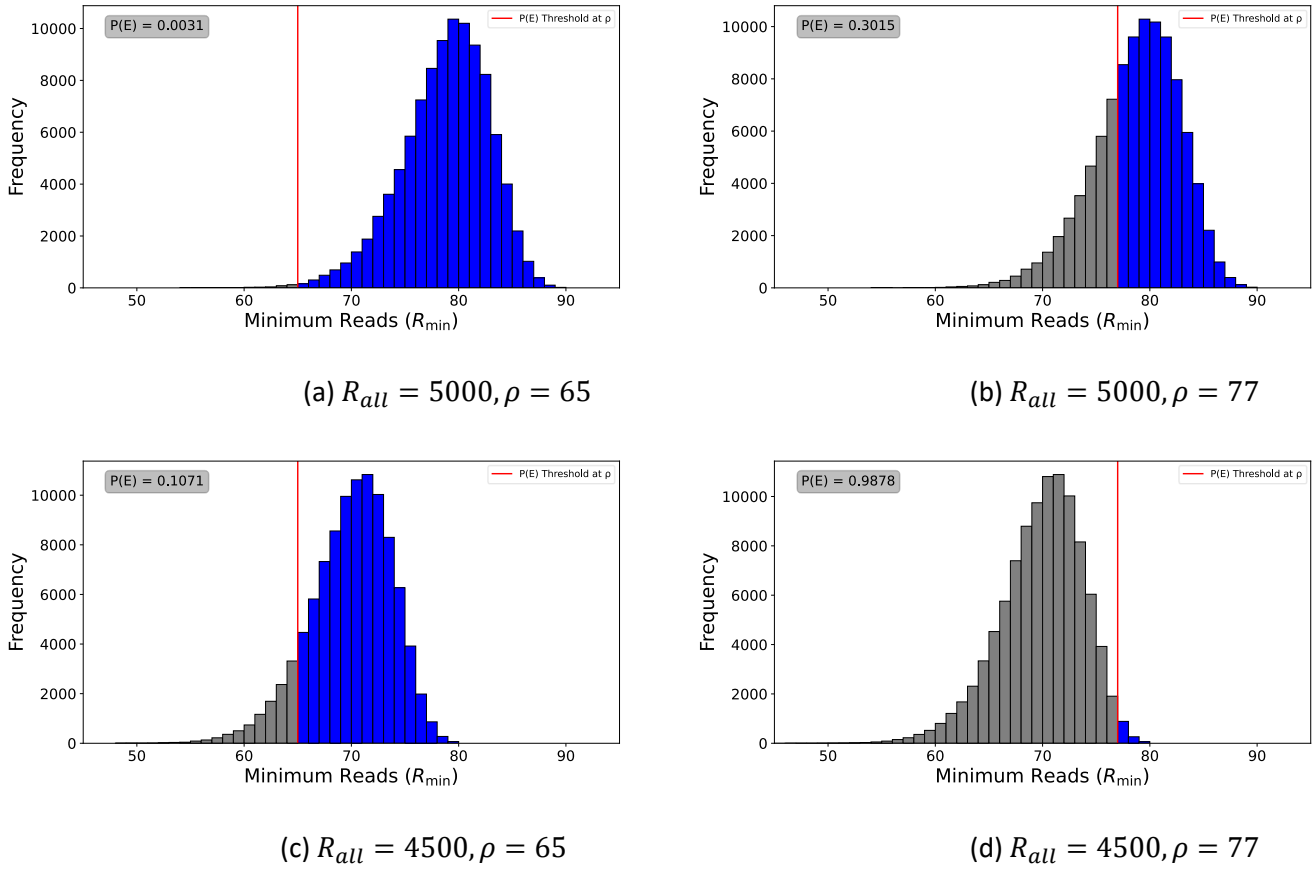

Fig. 5. The minimum value of a multinomial distribution $Y = \min_j(X_j)$ where $(X_1, \dots, X_l) \sim Multinom\left(R_{all}, \left(\frac{1}{l}, \dots, \frac{1}{l}\right)\right)$. (a) A histogram of the values of Y attained in 100,000 instances with $l = 50$ and $R_{all} = 5000$. The red line represents $\rho = 65$. The gray box show the probability $P\big(E(\rho)\big) = P(Y < \rho)$. (b-d) Like (a) for $(R_{all}, \rho) = (5000, 77), (4500, 65), (4500, 77)$.

Let $U = \left(\frac{1}{l}, \dots, \frac{1}{l}\right)$ be the expected uniform distribution equivalent to the expected read distribution for $(R_1, \dots, R_l)$.

Let $E(\rho)$ be the set of probability vectors equivalent to read distributions $(r_1, \dots, r_l)$ for which $\sum r_j = R_{all}$, $\min_{j=1,\dots,l}(r_j) < \rho$:

$$E(\rho) = \left\{P = (p_1, \dots, p_l) | \sum p_i = 1; \min_i(p_i) < \frac{\rho}{R_{all}}\right\} \quad (27)$$

We define $\zeta(\rho) = min_{P \in E(T)} D(P \parallel U)$ where $D(P||U)$ is the Kullback-Leibler divergence:

$$D(P \parallel U) = \sum_{i=1}^{l} p_i \log\left(\frac{p_i}{q_i}\right) \quad (28)$$

Let $P^* = \arg\min_{P \in E(\rho)} D(P||U)$ the closest element to Q in $E(\rho)$ in terms of the KL-divergence. That is $\zeta(\rho) = D(P^* \parallel U)$

Next we show that P^* is the distribution of reads in which $\rho - 1$ reads are assigned to one sequence and the remaining $R_{all} - \rho + 1$ reads are uniformly distributed over the remaining $l - 1$ sequences.

Lemma:

$$\text{Let } U = \left(\frac{1}{l}, \dots, \frac{1}{l}\right), \text{ Let } \alpha < \frac{1}{l} \quad (29)$$

Let $P^* = \left(\alpha, \frac{1-\alpha}{l-1}, \dots, \frac{1-\alpha}{l-1}\right)$, then

$$\forall P = (p_1, \dots, p_l), s.t. \exists i; p_i < \alpha$$

We have

$$D(P||U) \geq D(P^*||U) \quad (30)$$

The proof for this lemma is found in the appendix C. For intuition, this is simply the result of the symmetric nature of the KL-divergence function and of Q.

Sanov's Theorem [25] provides a bound on the probability of observing any distribution within $E(\rho)$.

$$P\big(E(\rho)\big) \leq (R_{all} + 1)^l 2^{-R_{all}\zeta(\rho)} \quad (31)$$

Where,

$$\zeta(\rho) = D(P^*||U) = \Sigma_{i=1}^{l} p_i^* \log\left(\frac{p_i^*}{q_i}\right) = \alpha \log(\alpha l) + (1 - 1)\left(\frac{1-\alpha}{l-1}\right) \log \frac{\frac{1-\alpha}{l-1}}{\frac{1}{l}} = \alpha \log(\alpha l) + (1-\alpha) \log\left(\frac{l(1-\alpha)}{l-1}\right) \quad (32)$$

This bound implies that the likelihood of observing a significantly non-uniform distribution of reads decreases exponentially as the total number of reads R_{all} increases.

We recall that

$$\sum_{\substack{(r_1,\dots,r_l) \\ \sum r_i = R_{all} \\ \min_j r_j \geq \rho}} P(r_1, \dots, r_l) = 1 - P\big(E(\rho)\big) \quad (33)$$

And so we get

$$P_{all} \geq P\big(X_\rho \geq a\big)\Big(1 - P\big(E(\rho)\big)\Big) \quad (34)$$

$$P_{all} \geq P\big(X_\rho \geq a\big)\big(1 - (R_{all} + 1)^l 2^{-R_{all}\zeta(\rho)}\big) \quad (35)$$

This gives us an operational algorithm for checking if R_{all} reads are sufficient to ensure successful decoding with confidence $1-\delta$, as specified in Algorithm 2.

Algorithm 2: Finding required sequencing depth R_{all} for a complete message

Data: Design parameters.
Input: δ (Acceptable failure probability)
Output: A value for R_{all} ensuring decoding with probability $1-\delta$
Initialize ρ to find threshold where $P(X_\rho \geq a) \geq \sqrt{1-\delta}$;
for *incrementing values of* ρ **do**
  if $P(X_\rho \geq a) \geq \sqrt{1-\delta}$ **then**
    Break loop and use found value of ρ;
  end
end
Set $R_{\text{all}} = \rho \times l$;
for *incrementing values of* R_{all} **do**
  Calculate probability $P(E)$ for current R_{all};
  if $1-P(E) \geq \sqrt{1-\delta}$ **then**
    Break loop and finalize value of R_{all};
  end
end

Fig. 6a demonstrates the approach by presenting the probability of successful message decoding $P(X_\rho \geq a)$ and the probability of considering "enough" of the read distribution $\big(1-P(E)\big)$ for a fixed number of overall reads $R_{all} = 3000$ as a function of the threshold ρ. Clearly, $P\big(X_\rho > a\big)$ increases as ρ increases since each sequence s_i is decoded using more reads. On the other hand, as was demonstrated in Fig. 5. increasing ρ decreases $1-P\big(E(\rho)\big)$ since less read distributions with $\min_j r_j \geq \rho$ are expected.

We note that the bound achieved by using Sanov's theorem is not tight and therefore present an alternative approach for finding ρ using empirical simulations. Fig. 6b presents the probability $\Big(1-P\big(E(\rho)\big)\Big)$ calculated like in Fig. 5 by 100,000 instances of simulating the multinomial distribution with $R_{all} = 1000$. Clearly, this method yields a tighter bound on the decoding probability while also requires analyzing less read overall.

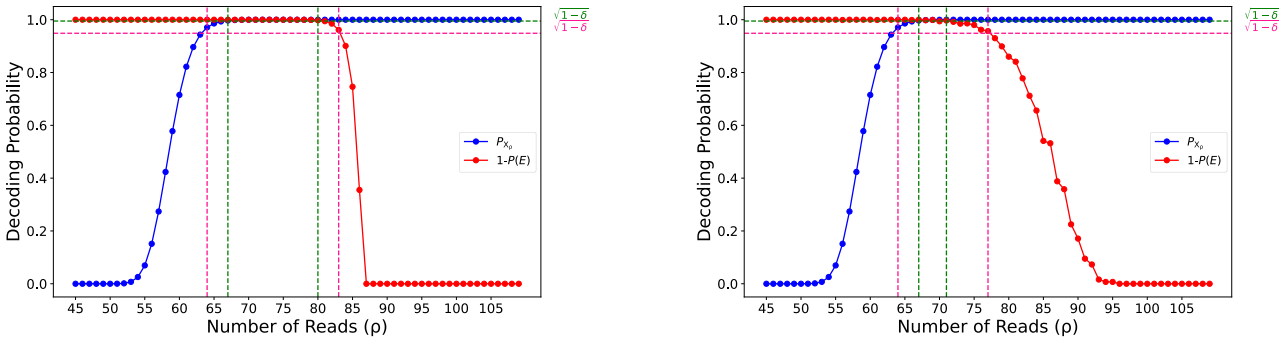

Fig. 6. Bounding the decoding probability. (a) Overall decoding probability $P(X_\rho \geq a)$ (blue line) and the Sanov bound on the probability of obtaining a read distribution across the sequences with $\min_j r_j \leq \rho$, $1-P\big(E(\rho)\big)$ (red line) as functions of the threshold T for a fixed number of analyzed reads $R_{all} = 3000$. The threshold $\sqrt{1-\delta}$ on the probability is marked with dotted lines for $\delta = 0.1$ (pink dotted lines) and $\delta = 0.2$ (green dotted lines), Setting ρ to any value between these lines ensures decoding with $1-\delta$ confidence. All values are calculated for $K = 7, t = 4, \epsilon = 0.01, R = 110, m = 10, b = 8, l = 10, a = 8$. (b) Like (a) with $P\big(E(\rho)\big)$ calculated using simulations instead of the Sanov bound and where the total number of reads $R_{all} = 1000$.

D. A tool for determining the required sequencing coverage

We have developed a tool designed to calculate the necessary sequencing coverage for DNA-based data storage systems.

Parameters, Input, and Output

The tool gets as parameters the sequence design and coding schemes and computes the required sequencing coverage for a desired confidence level. Specifically:

Design parameters:

- K – Total number of unique k-mers in each position.

- t – The required threshold on the number of observed occurrences of each of the k-mer
- m – sequence length
- b – the number of letters required to be successfully decoded in each sequence
- l – The number of sequences in the message
- a – The number of sequences required to be successfully decoded
- ϵ – Error probability of observing an invalid k-mer

Input:

- δ – acceptable failure rate

Output:

- R_{all} – required sequencing coverage

Description of tool run

Fig. 7 presents a high-level description of the tool workflow. Given the design parameters $K, t, \varepsilon, m, b, l$, and a, the tool finds a threshold ρ and a total number of reads R_{all} for which conditions (a) and (b) hold for the input confidence level $1-\delta$ (Part C). First, ρ is found such the decoding of at least a sequences is ensured, $P\left(X_\rho \geq a\right) \geq \sqrt{1-\delta}$ (Part C). This calculation requires the probability to decode a single sequence, $P_{single}(\rho, m, b)$ (Part B) which uses the reconstruction probability of a single combinatorial letter $\pi(K,t)(\rho)$ (Part A).

Once ρ is determined, the algorithm searches for the required number of overall reads R_{all} that ensures $1-P\left(E(\rho)\right) \geq \sqrt{1-\delta}$. This can be achieved using either using the bound from Sanov's Theorem or using the empirical estimation of $P\left(E(\rho)\right)$. When a value for R_{all} that satisfies the condition is found then the tool run exits outputting R_{all} to the user

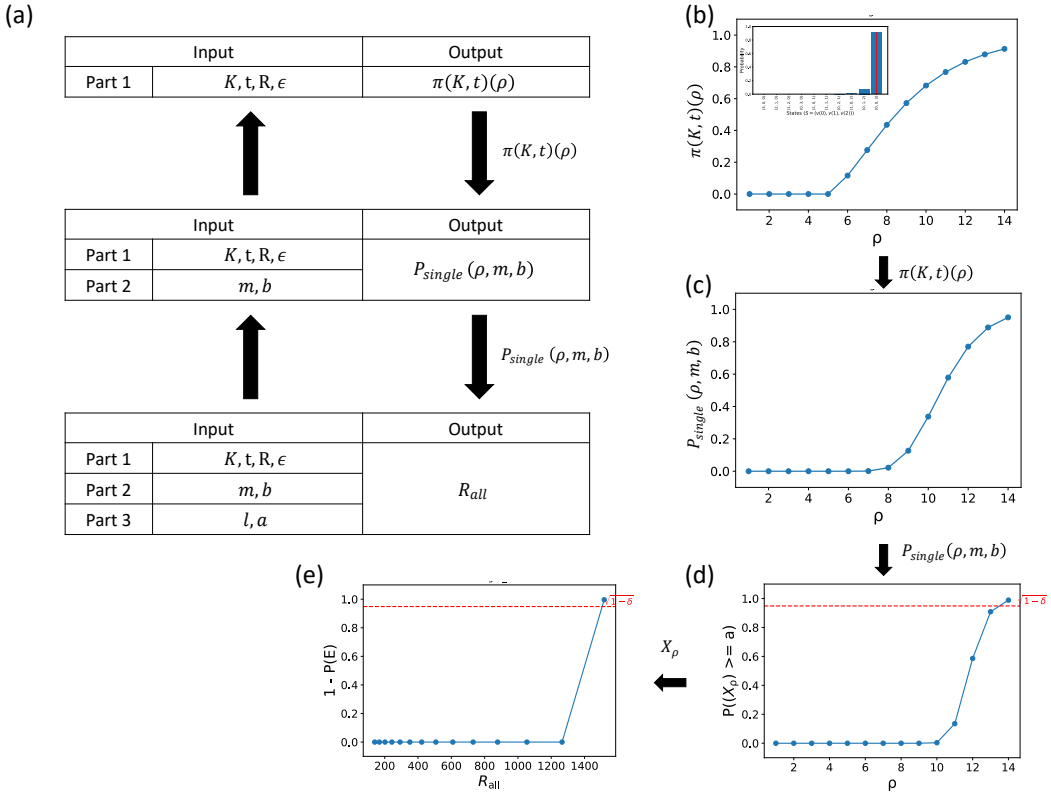

Fig. 7. Complexity calculation tool workflow. (a) Overview of the tool's run including internal dependencies, input parameters and outputs for each part. (b) Reconstruction probabilities of a single combinatorial position, $\pi(K,t)(\rho)$, calculated using the coupon collector's model (inset, like in Fig. 1) as a function of the threshold ρ. (c) Decoding probability for a full-length combinatorial sequence, $P_{single}(\rho, m, b)$, calculated using the binomial model with the probabilities from (a) as input. Plotted as function of the threshold ρ. (d) Finding ρ. Full message decoding probability, $P\left(X_\rho \geq a\right)$, calculated using the binomial model for X_ρ obtained from (c). Plotted as function of the threshold ρ. The target confidence level $\sqrt{1-\delta}$ is presented in the dotted red line. (e) Finding R_{all} given the selected ρ. The probability of considering enough read distributions (across the l sequences), $1-P\left(E(\rho)\right)$, based on either theoretical bound or the empirical calculation. Plotted as a function of R_{all}. The target confidence level $\sqrt{1-\delta}$ is presented in the dotted red line.

Example runs

To demonstrate the tool's functionality, we used it to determine the required sequencing coverage for different sets of design parameters, similar to those used in [15], and for various confidence levels. These results are presented in Table 1. Clearly, increasing the desired confidence level (smaller values for δ) requires increasing the sequencing converge. Scaling up the system's capacity by taking l to be 10 times larger results in a proportional increase in the R_{all}. Increasing the redundancy level (lower value for a) reduces the number of required reads to analyzed. We note that the different design parameters influence both the threshold ρ and the sequencing

coverage R_{all}. While R_{all} is affected by all the design parameters, ρ is primarily affected by m and b. These findings underscore the importance of carefully selecting system parameters to optimize the efficiency and reliability of DNA-based data storage systems. Future work may explore the boundaries of these parameters to further enhance system performance.

m	b	l	a	δ	ρ	R_{all}
100	80	100	80	0.1	60	8,868
				0.01	60	9,778
				0.001	60	10,267
			90	0.1	61	9,017
				0.01	61	9,942
				0.001	61	10,440
		1000	800	0.1	59	96,111
				0.01	60	102,625
				0.001	60	107,757
			900	0.1	60	97,738
				0.01	61	104,336
				0.001	61	109,553
100	90	100	80	0.1	67	9,903
				0.01	67	10,399
				0.001	68	11,080
			90	0.1	68	10,049
				0.01	69	10,709
				0.001	69	11,245
		1000	800	0.1	67	103,944
				0.01	67	114,600
				0.001	67	120,330
			900	0.1	68	105,494
				0.01	68	110,769
				0.001	68	122,124

Table 1. Required sequencing coverage for different design parameters and confidence levels.

IV. Discussion

Our study presents a novel model for analyzing coverage depth in DNA-based data storage, particularly focusing on combinatorial DNA encoding. We use the coupon collector's problem framework to model the reconstruction of combinatorial letters from sequencing data. We present a Markov Chain (MC) formulation for the calculation of the decoding probability and provide a tool for computing the probability. This solution is, however, limited in its scale due to the size of states space. Further work may be done to allow scaling up this model, either by developing more efficient computations or by developing approximation to the model.

One of the key aspects of the combinatorial approach is the strategic selection of Ω to consist easily distinguishable k-mers. This, together with the use of a threshold $t > 1$ in the reconstruction algorithm (See Algorithm 1) effectively mitigate k-mer mixup errors, as was demonstrated in [15]. We therefore chose to ignore k-mer mixup error in the model used for the reconstruction probability.

We also present a unified model for analyzing coverage depth of a complete combinatorial storage system considering an inner-outer error correction model. We present theoretical bounds on the decoding probability using Sanov's Theorem on the multinomial model for read distribution or using an empirical estimation.

We also provide a python tool for determining the sequencing depth required to achieve a desired confidence level for a system given design and encoding scheme. We demonstrate the tool's results on a selection of design parameter sets.

Future exploration in DNA data storage will significantly benefit from further understanding and optimizing coverage depth and from further improving efficient combinatorial coding. These elements are key to enhancing data storage capacity and reliability, promising exciting advancements in the field.

Appendix

A. *Evolution of Probability in the Coupon Collector Problem Video*

The coupon collector parameters that are showed in the video are: $K = 5, t = 2, R = 30$.
A. Evolution of Probability in the Coupon Collector Problem Video K=5, t=2, R=30.gif

B. *Classical coupon collector problem*

$$\pi_{(K,1)}(R) = \sum_{i=0}^{K}(-1)^i \binom{K}{i}\left(1-\left(\frac{i}{K}\right)\right)^R \quad (36)$$

$\pi(K,t)(R)$ is the probability of collecting all n unique coupons within R trials.

We will show that

$$(K,1)(R) = P(T(K,1) \le R) = \sum_{i=0}^{K}(-1)^i \binom{K}{i}\left(1-\left(\frac{k}{K}\right)\right)^R \quad (37)$$

for the coupon collector's problem, we can approach it using the principle of inclusion-exclusion. The formula calculates the probability of collecting all n unique coupons within R trials.

Let A_i be the event that the i-th coupon is not collected in R trials.

$$P(A_i) = \left(1-\frac{1}{K}\right)^R \quad (38)$$

Let $\bigcup_{i=1}^{K} A_i$ be the probability of not collecting at least one coupon in R trials.

Note that we are interested in:

$$\pi(K,1)(R) = 1 - P(\bigcup_{i=1}^{K} A_i) \quad (39)$$

$P(\bigcup_{i=1}^{K} A_i)$ is calculated using the principle of inclusion-exclusion.

$$P(\bigcup_{i=1}^{K} A_i) = \sum_{j=1}^{K}(-1)^{j-1}\left(\binom{K}{j}\right)\left(1-\frac{j}{K}\right)^R \quad (40)$$

And finally,

$$\pi(K,1)(R) = 1 - P(\bigcup_{i=1}^{K} A_i) = 1 - \sum_{j=1}^{K}(-1)^{j-1}\left(\binom{K}{j}\right)\left(1-\frac{j}{K}\right)^R = \sum_{j=0}^{K}(-1)^{j}\left(\binom{K}{j}\right)\left(1-\frac{j}{K}\right)^R \quad (41)$$

This follows from:

$$(\bigcup_{i=1}^{K} A_i) = \sum_{j=1}^{K}(-1)^{j-1}\sum_{\substack{I\subseteq\{1,\dots,K\}\\ |I|=j}} P(A_I) \quad (42)$$

Where $A_I = \bigcap_{i\in I} A_i$

For $j = 1$:

$$P(A_I) = P(A_i) = \left(1 - \frac{1}{K}\right)^R \tag{43}$$

For $j = 2$:

$$P(A_I) = P(A_m \cap A_l) = \left(1 - \frac{2}{K}\right)^R \tag{44}$$

And generally

$$P(A_I) = \left(1 - \frac{j}{K}\right)^R \tag{45}$$

And clearly:

$$\left|\left\{I;\ \begin{matrix} I \subseteq \{1,\dots,K\} \\ |I| = j \end{matrix}\right\}\right| = \binom{K}{j} \tag{46}$$

C. *Proof of the Lemma for the Sanov bound*

Lemma:

$$\text{Let } U = \left(\frac{1}{l},\dots,\frac{1}{l}\right), \text{ Let } \alpha < \frac{1}{l} \tag{47}$$

Let $P^* = \left(\alpha, \frac{1-\alpha}{l-1},\dots,\frac{1-\alpha}{l-1}\right)$, then

$$\forall P = (p_1,\dots,p_l), s.t. \exists i; p_i < \alpha$$

We have

$$D(P||U) \geq D(P^*||U) \tag{48}$$

Proof:

$$D(P||U) = \Sigma_{i=1}^{l} p_i \log(lp_i) \tag{49}$$

$$D(P^*||U) = \alpha\log(\alpha l) + (1-\alpha)\log\left(\frac{l(1-\alpha)}{l-1}\right) \tag{50}$$

We solve:

$$\min D(P) = \Sigma_{i=1}^{l} p_i \log(lp_i) \tag{51}$$

Subject to:

1. $\Sigma_{i=1}^{l} p_i = 1$ (52)
2. $p_1 \leq \alpha$ (WLOG) (53)

Therefore, the Lagrangian is:

$$\mathcal{L}(p_1, p_2, \dots, p_l, \lambda, \mu) =$$
$$\Sigma_{i=1}^{l} p_i \log(lp_i) - \lambda\left(\Sigma_{i=1}^{l} p_i - 1\right) - \mu(p_1 - \alpha) \tag{54}$$

The KKT conditions are:

1. Stationarity $\frac{\partial \mathcal{L}}{\partial p_i} = 0$:

   for $i > 1$, $\frac{\partial \mathcal{L}}{\partial p_i} = 0 \rightarrow p_i = \frac{e^{\lambda-1}}{l}$ (55)

   for $i = 1$, $\frac{\partial \mathcal{L}}{\partial p_1} = 0 \rightarrow p_1 = \frac{e^{\lambda-1+\mu}}{l}$ (56)

2. Primal feasibility:

   $\Sigma_{i=1}^{l} p_i = 1$ (57)

   $p_1 - \alpha < 0$ (58)

3. Dual feasibility:

   $\mu, \lambda \geq 0$ (59)

4. Complementary slackness:

   $\mu(p_1 - \alpha) = 0$ (60)

Expressing λ using the primal feasibility (57):

$$p_1 + (1-l)p_i = 1 \tag{61}$$

Substituting p_i and p_1 from (55)(56):

$$\lambda = log\left(\frac{l}{e^{\mu}+l-1}\right) + 1 \tag{62}$$

Expressing p_1 with λ from (62):

$$p_1 = \frac{e^{\mu}}{l+e^{\mu}-1} \tag{63}$$

Expressing μ:

1. If $p_1 \neq \alpha$, then $\mu = 0$.
2. If $p_1 = \alpha$, then μ can be non zero.

If $p_1 = \alpha$, we substitute α for p_1 in (63):

$$\alpha = \frac{e^{\mu}}{l+e^{\mu}-1} \rightarrow \mu = \log\left(\frac{\alpha(l-1)}{1-\alpha}\right) \tag{64}$$

Using the original expressions for p_i from (55), and substituting μ we expressed in (64), we get:

$$\text{for } i > 1,\ \ p_i = \frac{1}{l + \frac{\alpha(l-1)}{1-\alpha} - 1} = \frac{1-\alpha}{l-1} \tag{65}$$

and recall that $p_1 = \alpha$

If $p_1 \neq \alpha$, we substitute $\mu = 0$ for p_1 in (56):

$$p_1 = \frac{1}{l} \tag{66}$$

Using the original expressions for p_i from (55), and substituting $\mu = 0$, we get:

$$\text{for } i > 1,\ \ p_i = \frac{e^{\lambda-1}}{l} = \frac{e^{log\left(\frac{l}{e^{\mu}+l-1}\right)}}{l} = \frac{\frac{l}{e^{\mu}+l-1}}{l} = \frac{1}{l} \tag{67}$$

And we get the trivial solution $P^* = U$ which does not satisfy the condition $p_1 < \alpha$.

Therefore, we proved that $D(P||U) \geq D(P^*||U)$.

BIBLIOGRAPHY

[1] J. Rydning, "Worldwide IDC Global DataSphere Forecast, 2022–2026: Enterprise Organizations Driving Most of the Data Growth," International Data Corporation (IDC), 2022.

[2] 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. 1237, 2019.

[3] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 335, no. 6328, pp. 950-954, 2017.

[4] L. e. a. Organick, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 36, no. 3, p. 242–248, 2018.

[5] S. Yazdi, R. Gabrys and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific Reports*, vol. 7, no. 5011, 2017.

[6] E. LeProust, B. Peck, K. Spirin, H. McCuen, B. Moore, E. Namsaraev and M. Caruthers, "Synthesis of high-quality libraries of long (105mer) oligonucleotides by a nover depurination controlled process," *Nucleic Acids Research*, no. 38, pp. 2522-2540, 2019.

[7] G. Church and et al., "Next-Generation Digital Information Storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, 2012.

[8] N. Goldman and et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature* , vol. 494, p. 77–80, 2013.

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

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

[11] S. e. a. Yazdi, "DNA-based storage: Trends and methods," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 1, no. 3, pp. 230-248, 2015.

[12] W. p. b. D. D. S. Alliance, "Preserving our digital legacy: An introduction to DNA data storage," *DNA Data Storage Aliance*, 2021.

[13] 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, 2023.

[14] S. e. a. Chandak, "Improved read/write cost tradeoff in DNA-based data storage using LDPC codes," *Annual Allerton Conference on Communication, Control, and Computing*.

[15] I. Preuss, Z. Yakhini and L. Anavy, "Data storage based on combinatorial synthesis of DNA shortmers," *bioRxiv*, vol. 08, 2021.

[16] N. Roquet and et al., "DNA-based data storage via combinatorial assembly," *bioRxiv*, 2021.

[17] Y. Yan and et al., "Scaling logical density of DNA storage with enzymatically-ligated composite motifs," *Sci Rep* , vol. 13, no. 15978, 2023.

[18] M. e. a. Blawat, "Forward error correction for DNA data storage," *Procedia Computer Science*, vol. 80, pp. 1011-1022, 2016.

[19] S. e. a. Chandak, "Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, 2020.

[20] P. Erdos and A. Renyi, "On a classical problem of probability theory," *Magyar Tud. Akad. Mat. Kutato Int.*, vol. 6, no. 1-2, pp. 215-220, 1961.

[21] W. Feller, "An introduction to probability theory and its applications," *Wiley*, vol. 1, no. 2nd edition, p. 35, 1967.

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

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

[24] P. Neal, "The generalized coupon collector problem," *Journal of Applied Probability*, vol. 45, no. (3), pp. 621-629, 2008.

[25] I. Sanov, On the probability of large deviations of random variables, United States Air Force, Office of Scientific Research, 1958.

[26] Y. Yan and et al., "Scaling logical density of DNA storage with enzymatically-ligated composite motifs," *Scientific Reports*, vol. 13, no. 15978, 2023.

[27] N. Roquet and et al., "DNA-based data storage via combinatorial assembly," *bioRxiv*, 2021.