Sequencing Coverage Analysis for Combinatorial DNA-Based Storage Systems

Inbal Preuss, *Student Member, IEEE*, Ben Galili, Zohar Yakhini, *Member, IEEE*, 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 seek to characterize the distribution of the number of sequencing reads required for message reconstruction. We use a variant of the coupon collector distribution for this purpose. For any given number of observed reads, $R \in \mathbb{N}$, we use a Markov Chain representation of the process to compute the probability of error-free reconstruction. We develop theoretical bounds on the decoding probability and use empirical simulations to validate these bounds and assess tightness. This work contributes to understanding sequencing coverage in DNA-based data storage, offering insights into decoding complexity, error correction, and sequence reconstruction. We provide a Python package, with its input being the code design and other message parameters, all of which are denoted as Θ, and a desired confidence level $1-\delta$. This package computes the required read coverage, guaranteeing the message reconstruction $R = R(\delta, \Theta)$.**

***Index Terms*—DNA, DNA-based data storage, synthethic biology, computational biology.**

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 for some applications, offering unmatched density and durability. This novel approach involves the synthesis, storage, and sequencing of DNA molecules to encode, store, and retrieve information [2], [3]. However, challenges such as short, error-prone strands and limitations of current synthesis technologies still remain [4], [5], [6], [7], [8]. While DNA-based storage stands as a promising technology and the cost of DNA sequencing is 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 the 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.

In recent years, several studies suggested the use of combinatorial DNA encoding and synthesis as an approach for increasing logical density while reducing overall cost in DNA-based storage systems [15], [16], [17]. The combinatorial encoding approach uses a set of clearly distinguishable DNA shortmers to construct large combinatorial alphabets, where each letter is encoded by a subset of shortmers. This scheme is a novel approach for DNA-based data storage, offering an increase in logical density over standard DNA-based storage systems, reduced reconstruction error levels, and scalability. See Section II for more details about combinatorial DNA encoding. Combinatorial DNA encoding can be viewed as an extension of the composite DNA coding schemes (sometimes referred to as degenerate DNA encoding) [4], [18].

This work presents the first model for analyzing the sequencing coverage depth problem under combinatorial DNA encoding. The analysis is based on the general design scheme for combinatorial DNA storage systems. In brief, we assume a 2-dimensional (2D) MDS error correction scheme over a set of short combinatorial DNA sequences. Each combinatorial sequence is encoded using an inner code, to protect against symbol errors, while an outer code adds redundancy to a block of sequences, protecting against sequence-level errors (*e.g.,* sequence dropout). The inner and outer code approach is common in DNA-based storage literature [4], [5], [19], [20]. Our analysis follows the reconstruction steps associated with this approach.

We model the reconstruction of a single combinatorial letter as a variant of the coupon collector's problem [13], [21], [22], [23], [24]. We use a Markov Chain (MC) representation of the collection process to characterize this distribution. Taking into account the error correction code parameters, we continue by analyzing the read depth requirement for a single combinatorial sequence and then for the entire message. 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 various design parameters, and compare the results to Monte Carlo simulation experiments

Manuscript received 1 January 2024; revised 7 April 2024; accepted 20 May 2024. Date of publication 31 May 2024; date of current version 17 June 2024. This work was supported by the European Union (DiDAX) under Grant 101115134. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. The associate editor coordinating the review of this article and approving it for publication was E. Yaakobi. *(Corresponding author: Inbal Preuss.)*

Inbal Preuss, Zohar Yakhini, and Leon Anavy are with the School of Computer Science, Reichman University, Herzliya 4610101, Israel, and also with the Faculty of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: inbalpreuss@gmail.com; zohar.yakhini@runi.ac.il; leon.anavy@post.runi.ac.il).

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

Digital Object Identifier 10.1109/TMBMC.2024.3408053

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 aspect supporting the design and implementation of such systems.

This paper is constructed as follows. Section II includes an overview of combinatorial DNA encoding, providing important notations while defining the overall system design and the reconstruction flow. Section III defines the coverage depth problem for combinatorial DNA storage, highlighting the differences compared to the standard DNA coverage depth problem. Section IV describes the analysis of required read depth by breaking it down into three steps associated with the reconstruction process. Section IV-A presents the coupon collector's model for a single combinatorial letter. Sections IV-B and IV-C analyze the decoding of one combinatorial sequence and the complete message, respectively. Section IV-D describes the tool for determining the required coverage depth in combinatorial systems, and demonstrates it using different design parameters. Finally, Section VI discusses the broader implications of our work on DNA-based storage systems and related fields.

II. Overview of Combinatorial DNA Encoding Scheme

Utilizing combinatorial approaches for DNA synthesis and assembly was recently suggested by several studies as a way to increase logical density and reduce overall costs in DNA-based storage systems [15], [16], [17]. While different studies suggest various molecular mechanisms for generating combinatorial sets of DNA sequences, they all share several important characteristics. This section describes these common components, using notations from [15].

A. Definitions

Let Ω be a set of N DNA k-mers. The k-mers in Ω are chosen such that mix-up errors between two k-mers are negligible (*e.g.,* by setting a minimal distance d between each pair). A binomial combinatorial alphabet Σ is defined such that every letter $\sigma \in \Sigma$ represents a subset of size K of k-mers from Ω. This subset is referred to as the member k-mers of σ. This defines an alphabet of $|\Sigma| \leq \binom{N}{K}$ letters. Encoding a binary message of length B bits using this extended combinatorial alphabet is done by generating a sequence of M combinatorial letters $\sigma^1 \dots \sigma^M$ where $M = \frac{B}{\lfloor \log_2(|\Sigma|) \rfloor}$. Fig. 1a presents a schematic view of the combinatorial encoding approach.

Fig. 1b describes the physical properties of a combinatorial DNA channel. First, the combinatorial sequences are written by generating a set of DNA molecules (using combinatorial synthesis or assembly). Each of the molecules includes a sequence of positions where each position represents a single combinatorial letter. In a given position, each sequence should be one of the member k-mers of the combinatorial letter encoded in that position. Next, a sample of the generated molecules are processed and sequenced generating a set of reads that are analyzed for the reconstruction of the combinatorial message. The reconstruction of the combinatorial sequence includes three main steps, as detailed below:

1) Grouping: The reads obtained from the sequencing output are grouped according to the combinatorial sequence they represent. This is often done using a barcode sequence at the beginning of every DNA molecule.
2) Sequence reconstruction: The combinatorial sequences are reconstructed from the grouped DNA sequences.
   - Often, each sequence is reconstructed separately from the set of reads assigned to it.
   - Each sequence is treated as a set of independent combinatorial letters and therefore reconstructed one position at a time.
   - A combinatorial letter is reconstructed by identifying K unique k-members that are observed in the relevant position of the analyzed reads.
3) Message decoding: Error correction codes are used to decode the original message.

B. Error Correction and Sequencing Coverage

Standard DNA-based data storage systems incorporate error correction schemes to mitigate common errors in DNA synthesis and sequencing. Symbol-level errors, such as single-base substitutions, insertions, and deletions are the most common errors. Most studies use error correction codes and constrained coding over the DNA alphabet to overcome these errors. Sequence-levels errors are another common type of error in DNA-based storage systems. A common sequence-level error is a sequence dropout, where certain sequences are not observed in the output at all. Sequence dropout occurs mostly due to the sampling step taken before sequencing, which may result in some sequences not being chosen. The molecular biology steps used for processing the DNA molecules may also be biased in a manner not yet fully characterized, which increases the chances of sequence dropouts. To overcome sequence dropouts, a second layer of error correction is applied on the sequence level. The combination of symbol level error correction (inner code) and the sequence level error correction (outer code) is sometimes referred to as a 2-dimensional (2D) error correction scheme.

Another common approach for eliminating both symbol and sequence-level error, is using the inherent multiplicity in DNA synthesis and sequencing technologies. Increasing the sampling rate yields higher sequence coverage, reduces the chances of sequence dropouts, and helps correct symbol level errors using various consensus-based methods. This makes studying the optimal sampling rate (or sequence coverage) a promising research direction for improving DNA-based storage.

Combinatorial DNA-based data storage requires additional considerations when examining errors and designing error correction strategies. Mainly, in every position, the subset of k-mers representing the combinatorial letter must be identified correctly. To do so, at least one copy of each k-mer in the subset must be observed. It is therefore important to understand

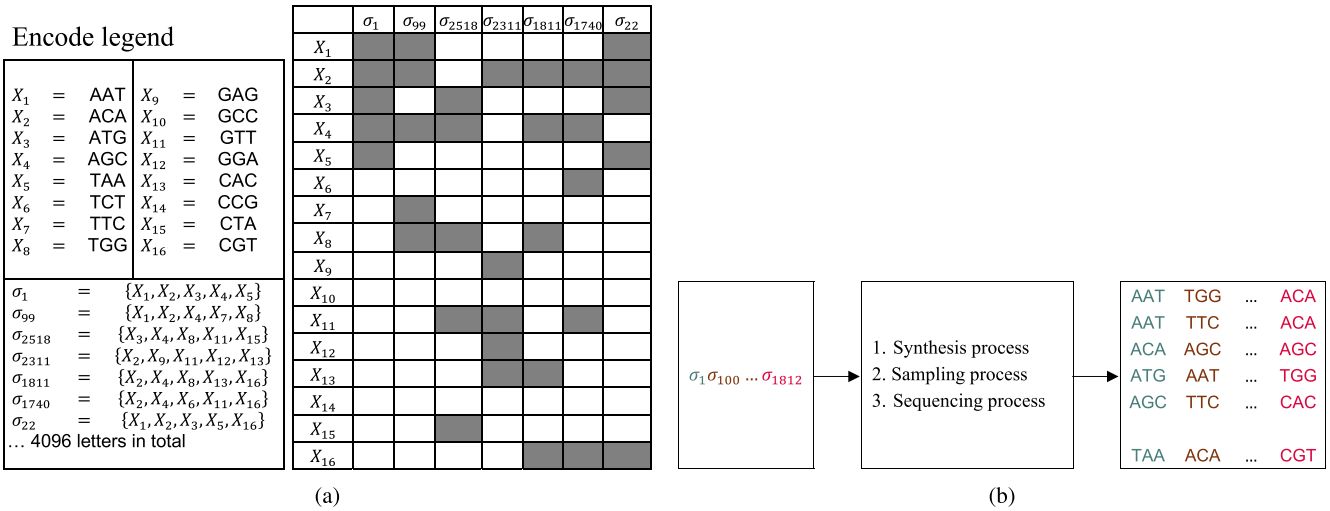


Fig. 1. Schematic view of a combinatorial alphabet process. (a) The combinatorial letters are constructed from a set of $N = 16$ k-mers, $\Omega = \{X_1, \ldots, X_{16}\}$, creating $|\Sigma| = 4096$ letters. Each letter represents a subset of $K = 5$ k-mers, as seen on the bottom left and depicted in the grayed-out cells. (b) The combinatorial sequence of letters then undergoes a process of synthesis, sampling, and sequencing that represent the combinatorial DNA storage channel.

the effect of sequence coverage on composite DNA systems, and to analyze the effect of different parameters. In order to account for possible errors in identifying the observed k-mers, a suggested approach involves observing each k-mer multiple times before inferring the combinatorial letter based on the subset of K k-mers observed most frequently. The required multiplicity is a tunable parameter (t) in combinatorial DNA systems. To demonstrate the difference between coverage analysis of standard DNA and composite DNA, consider the following simplified example.

Example 1: To recover x standard DNA sequences, we are required to observe at least one copy of each sequence, and at least $x \times 1$ reads in total. To recover x combinatorial DNA sequences with a combinatorial factor K, we need to observe at least K copies of each sequence and at least $x \times K$ reads in total. Adding the multiplicity parameter t to the combinatorial reconstruction process, makes the minimal number of reads to be analyzed $x \times K \times t$. Note that this is an unrealistic example. In reality, channel noise (*i.e.,* sampling and errors) complicates the coverage requirements in both cases.

Analysis of the sequencing coverage of combinatorial DNA systems was briefly explored in [15] mostly using simulations.

III. The Combinatorial Sequencing Coverage Problem

A. Problem Definition

In this study, we address the challenge of determining the required sequencing coverage for DNA-based data storage systems that utilize combinatorial sequences. We assume a 2D MDS error correction scheme as presented in the top panel of Fig. 2. Namely, the message is encoded using a vector of l combinatorial sequences. Each sequence is of length m. Given R_{all} analyzed reads, the decoding process includes the grouping, reconstruction, and decoding process as described in II and demonstrated in the bottom panel of Fig. 2. The error correction scheme is characterized by two parameters, b and a. Every combinatorial sequence is assumed to be correctly recovered if at least $b \leq m$ letters/positions are correctly reconstructed (inner code). The message is fully recovered if at least $a \leq l$ sequences are successfully recovered (outer code).

Problem 1 (Reconstruction of a Single Combinatorial Letter): For a given combinatorial alphabet represented by the subset size K and a given number of analyzed reads R, what is the probability of reconstructing a single combinatorial letter?

Problem 2 (Reconstruction of a Combinatorial Sequence): For a given combinatorial alphabet represented by the subset size K and given values for m and b, and a given number of analyzed reads R, what is the probability of recovering a combinatorial sequence?

Problem 3 (Reconstruction of a Complete Combinatorial Message): For a given combinatorial alphabet represented by the subset size K and given values for m, b, l, and a, and a given number of analyzed reads R_{all}, what is the probability of recovering the complete combinatorial message? Alternatively, what is the required number of reads R_{all} that guarantees a desired confidence level δ?

B. Comparison With Standard DNA Scheme

Sequencing coverage in standard DNA-based data storage systems was nicely analyzed in [13], where the main focus was on the outer code and the reconstruction of the complete message. Specifically, recovering a single sequence was considered to be a binary function of the number of copies observed for the sequence, t. In this work, we break down this problem, by:

1) Considering the inner code and modeling the probability of a combinatorial sequence to be recovered.
2) Modeling the reconstruction of each combinatorial letter (a position in the sequence) using the coupon collector's problem.

We also introduce a Markov Chain (MC) model-based calculation for the exact characterization of the coupon collector's distribution. We complete the analysis by considering the required coverage for achieving a desired decoding probability.

Algorithm 1: Decoding a Single Combinatorial Letter

Input: A set $\mathcal{R}$ 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 $\Sigma = \{\sigma_1, ..., \sigma_{|\Sigma|}\}$ with $|\Sigma| \leq \binom{N}{K}$;
initialize a counter for each k-mer in Ω;
while $|\mathcal{R}| > 0$ **do**
    extract next read r from $\mathcal{R}$;
    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';
    end
end
return *FALSE*;

Finally, we give a practical tool that can be used to design combinatorial DNA systems.

IV. 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 sequence, 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 N 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 referred to as the inferred member k-mers, and are used to reconstruct a combinatorial letter σ' (See Algorithm 1).

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. We note that the size of the k-mer set N does not play a role in this model. We also ignore invalid k-mers as we assume that the k-mers in Ω are selected such that mix-up errors are negligible (refer to Section II and [15] for details). Nonetheless, we allow detectable errors in the k-mer reading with some (low) probability ϵ, as described later in this section. 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 reads.

$$\pi_{K,t}(R) = P\big(T_{K,t} \leq R\big) \tag{1}$$

For $t = 1$, the random variable $T_{K,t}$ represents the classical coupon collector's model [25] and we get (See Appendix B):

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

$$E\big(T_{K,1}\big) = K \cdot H_K \tag{3}$$

where $H_K = \sum_{i=1}^{K} \frac{1}{i}$ is the Kth harmonic number. We note that this result for $t = 1$ is also presented in [15].

For $t > 1$, we can obtain [24]:

$$E\big(T_{K,t}\big) = K(\ln(K) + (t-1)\ln(\ln(K)) + O(1)) \tag{4}$$

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:

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

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:

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

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

And, when $v(t) = 0$, the inequality in (7) holds as equality $\sum_{i=0}^{t} i \cdot v(i) = R$. 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 $S = \binom{K+t}{t}$ possible solutions to the equation, representing the number of states.

Example 2: 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 observed yet. This is the case prior to analyzing the reads.
- (8, 2, 0): After analyzing two reads, two unique k-mers have been observed exactly once while the remaining eight 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 two times or more, and the remaining seven k-mers have not been observed yet.

The following transition matrix A is defined with dimensions $S \times S$, where each transition is defined by the observation of one read.

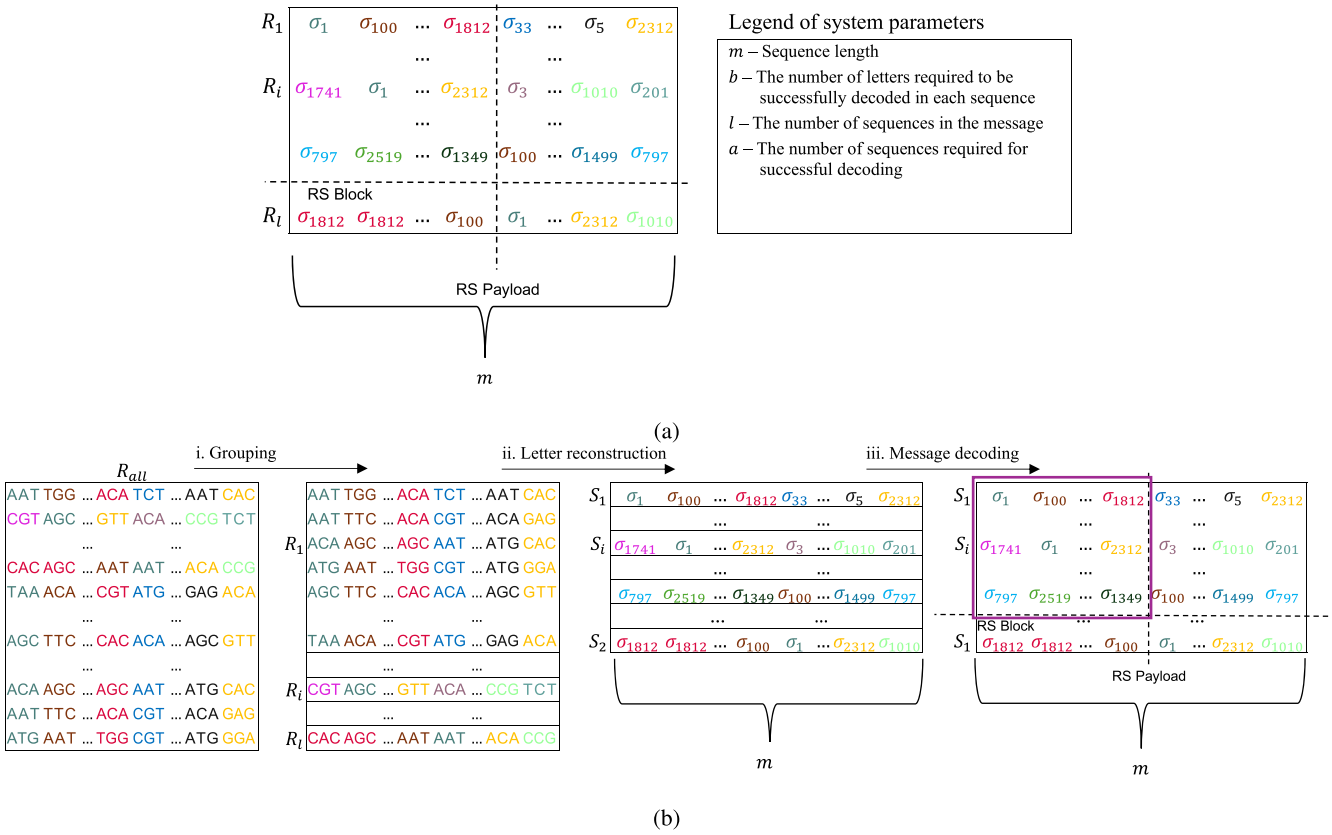


Fig. 2. Combinatorial DNA system design. (a) Error correction scheme design. (b) Message decoding and reconstruction. i. Grouping. ii. Letter reconstruction. iii. Message decoding. (The pink square is the final message.).

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

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

And:

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

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

Example 3: Considering $K = 10$ member k-mers and a threshold $t = 2$. In the first transition, the first unique k-mer must be observed:

$$P(s_0 = (\mathbf{10},0,0), s_1 = (9,1,0)) = A[(\mathbf{10},0,0)][(9,1,0)] = \frac{v(0)}{K} = 1$$

For the second transition there are two options:

- When one out of the nine yet unseen k-mers is drawn:

$$P(s_0 = (\mathbf{9},1,0), s_1 = (8,2,0)) = A[(\mathbf{9},1,0)][(8,2,0)] = \frac{v(0)}{K} = \frac{9}{10}$$

- When the same k-mer is drawn again:

$$P(s_0 = (9,\mathbf{1},0), s_1 = (9,0,1)) = A[(9,\mathbf{1},0)][(9,0,1)] = \frac{v(1)}{K} = \frac{1}{10}$$

To get to state $s_2 = (8,2,0)$, this calculation takes place:

$$P(s_0 = (10,0,0), s_2 = (8,2,0)) = \\ A[(10,0,0)][(9,1,0)] * A[(9,1,0)][(8,2,0)] = \\ 1 * \frac{9}{10} = \frac{9}{10}$$

To calculate $\pi_{K,t}(R)$, we set the initial state to be:

$$s_0 = (v(0) = K, v(1) = 0, \ldots, v(t) = 0), \tag{10}$$

where $P_0 = (P(s_0) = 1,0,\ldots,0)$ is the state distribution vector. We derive the distribution vector over the states after R steps:

$$P_R = P_0 A^R \tag{11}$$

Let $s_f = (v(0) = 0, v(1) = 0, \ldots, v(t) = K)$ be the desired state in which all K k-mers have been observed at least t times. Thus:

$$\pi_{K,t}(R) = P_R(s_f) \tag{12}$$

Fig. 3 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,

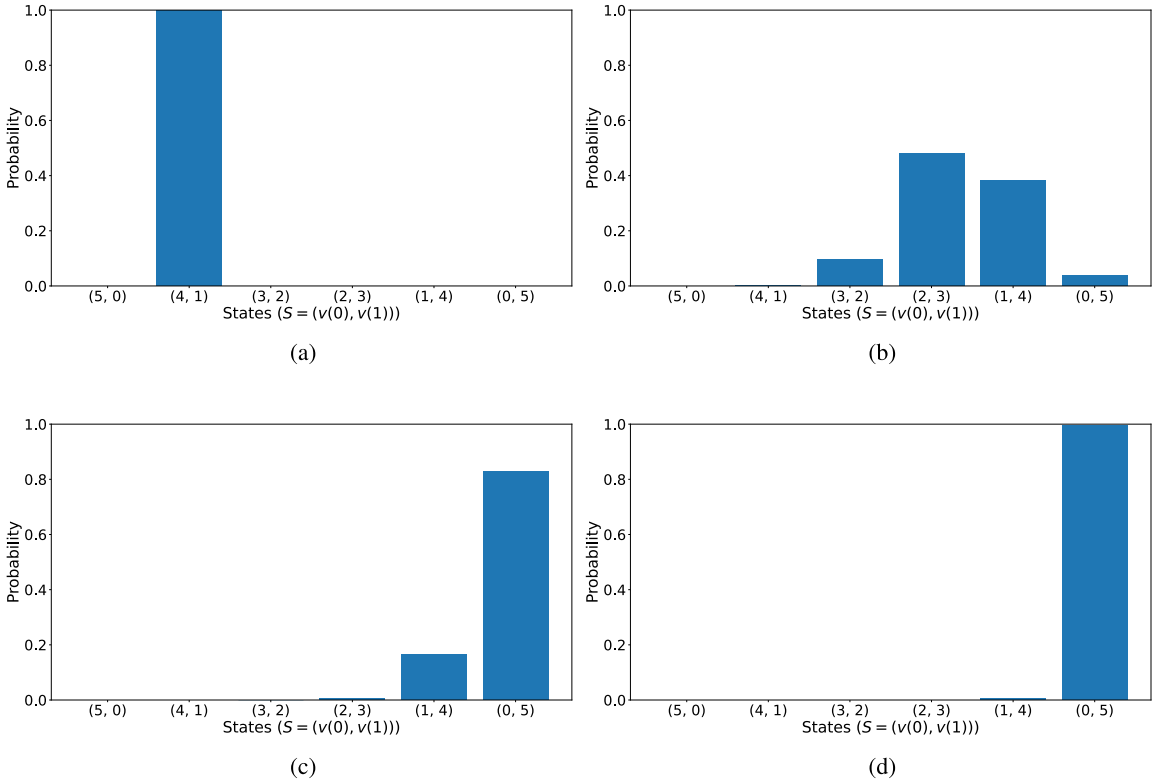


Fig. 3. Evolution of probability in the coupon collector model. (a) The probability distribution across the six states (X-axis) after observing $R = 1$ reads. (b-d) Similar to (a), with $R = 5$, 15, 30 respectively. Calculated for $K = 5$, $t = 1$, and no errors, $\epsilon = 0$.

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:

$$\pi_{5,1}(30) = 0.994$$

This algorithm ignores possible synthesis and sequencing errors 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:

$$\begin{aligned} &B[(v(0), \ldots, v(i), v(i+1), \ldots, v(t))] \\ &\quad [(v(0), \ldots, v(i) - 1, v(i+1) + 1, \ldots, v(t))] \\ &\quad = (1 - \epsilon)\frac{v(i)}{K} \end{aligned} \tag{13}$$

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

$$\begin{aligned} &B[(v(0), \ldots, v(i), \ldots, v(t))][(v(0), \ldots, v(i), \ldots, v(t))] \\ &\quad = \frac{v(t)}{K}(1 - \epsilon) + \epsilon \end{aligned} \tag{14}$$

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

Example 4: In the first transition, taking into account ϵ, there are two options:

- When one out of the ten yet unseen k-mers is drawn:

$$\begin{aligned} &P(s_0 = (\mathbf{10}, 0, 0), s_1 = (9, 1, 0)) = \\ &\quad A[(\mathbf{10}, 0, 0)][(9, 1, 0)] = (1 - \epsilon)\frac{v(0)}{K} = 1 - \epsilon \end{aligned}$$

- When an invalid k-mer is drawn:

$$\begin{aligned} &P(s_0 = (\mathbf{10}, 0, 0), s_1 = (10, 0, 0)) = \\ &\quad A[(\mathbf{10}, 0, 0)][(10, 0, 0)] = (1 - \epsilon)\frac{v(2)}{K} + \epsilon = \epsilon \end{aligned}$$

For the second transition, there are three options:

- When one out of the 9 yet unseen k-mers is drawn:

$$\begin{aligned} &P(s_0 = (\mathbf{9}, 1, 0), s_1 = (8, 2, 0)) = A[(\mathbf{9}, 1, 0)][(8, 2, 0)] \\ &\quad = (1 - \epsilon)\frac{v(0)}{K} = (1 - \epsilon) * \frac{9}{10} \end{aligned}$$

- When the same k-mer is drawn again:

$$\begin{aligned} &P(s_0 = (9, \mathbf{1}, 0), s_1 = (9, 0, 1)) = A[(9, \mathbf{1}, 0)][(9, 0, 1)] \\ &\quad = (1 - \epsilon)\frac{v(1)}{K} = (1 - \epsilon) * \frac{1}{10} \end{aligned}$$

- When an invalid k-mer is drawn:

$$\begin{aligned} &P(s_0 = (9, \mathbf{1}, 0), s_1 = (9, 1, 0)) = A[(9, \mathbf{1}, 0)][(9, 1, 0)] \\ &\quad = (1 - \epsilon) * \frac{v(2)}{K} + \epsilon = \epsilon \end{aligned}$$

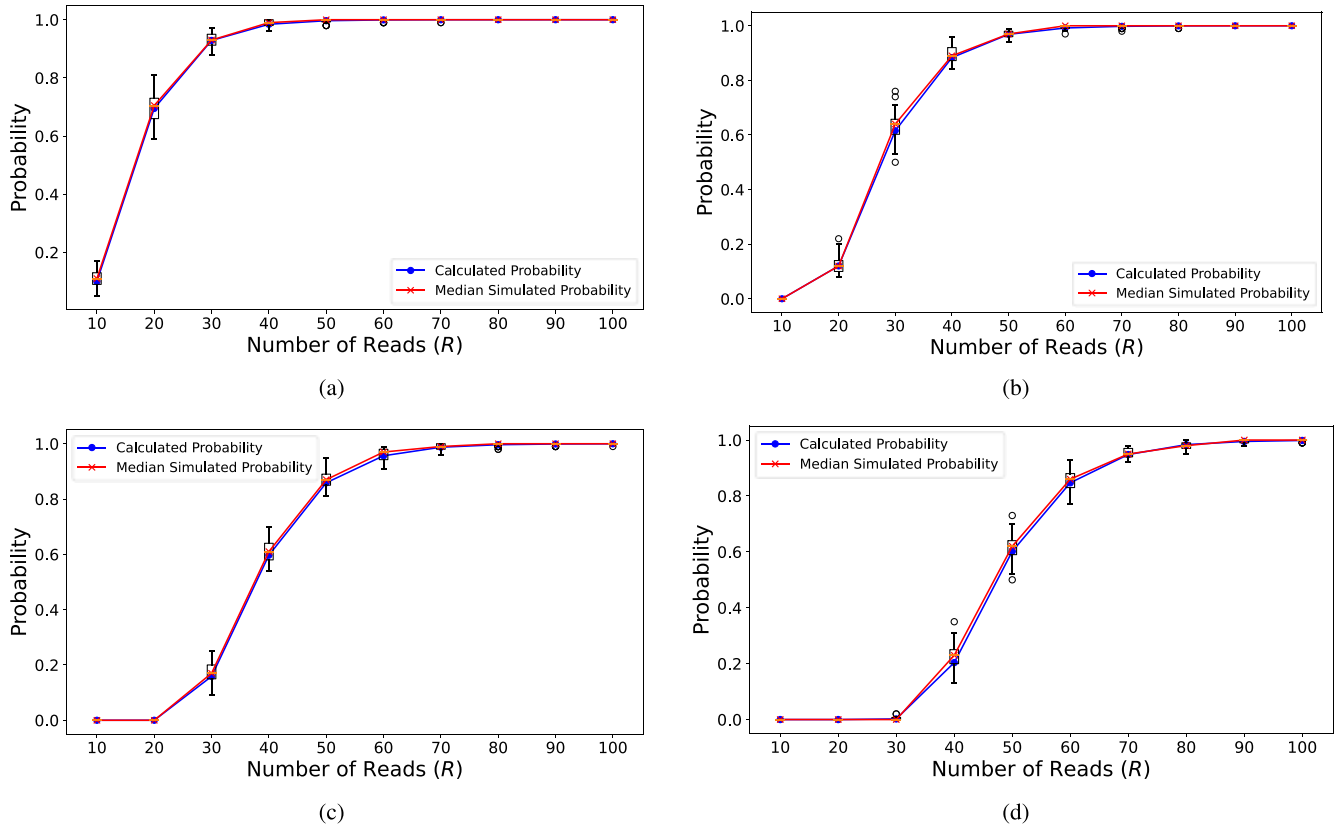


Fig. 4. Decoding probability for a varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analyses were conducted for $K = 7$ and $\epsilon = 0.01$. (a) Results for $t = 1$, the blue line corresponds to the calculated probability based on the MC model, while the red line represents the median of 50 simulation runs, where each simulation calculates the success rate of a 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 respectively.

Fig. 4 depicts the decoding probabilities for varying numbers of analyzed reads using different values for the threshold t (See Appendix E for different K and ϵ values). 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 the 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 assumption that the design parameters render this error type very unlikely (See Section II). We further discuss this issue in Section VI.

B. Reconstruction of a Combinatorial Sequence

Let $s = \sigma^{(1)}\sigma^{(2)}\dots\sigma^{(m)}$ be a sequence of length m over the same binomial alphabet defined in the previous section. Assuming the use of an MDS error correction code, we say that decoding only $b \leq m$ letters/positions 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:

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

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

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

Remark: Since there are different technological/molecular approaches for generating combinatorial DNA sequences, no specific assumption can be made regarding the dependence of different positions in the sequence. Every technology aspires to generate sequences with independent positions and we therefore chose to assume independence. From a coding point of view, assuming independence makes the solution general as no constraints are forced on the sequence.

We note that in the case of $b = m$ (*i.e.,* no error correction code), we get the result presented in [15].

We can approximate this probability using the normal estimation (based on the Central Limit theorem):

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

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

where Φ is the CDF of the standard normal distribution.

Fig. 5 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.

C. Reconstruction of a Complete Combinatorial Message

Let $M = \{s_i\}_{i=1}^{l}$ 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 an 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 sequences using R_{all} reads, $P_{\text{all}}(R_{\text{all}}, l, a)$.

Fig. 6 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 $P_{\text{all}}(R_{\text{all}}, l, a)$, or simply P_{all}.

Given a specific distribution of the R reads $(r_1, \ldots, r_l), \sum r_i = R_{all}$, to successfully decode the message at least a of the sequences must be decoded:

$$P_{\text{all}}(r_1, \ldots, r_l) = P\left(\sum_{i=1}^{l} I_i \geq a\right) \tag{19}$$

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

Assuming an unrealistic case of distributing the reads evenly over the l sequences, each sequence is represented by exactly r_{mean} reads as follows:

$$r_{\text{mean}} = \frac{R_{\text{all}}}{l} \tag{20}$$

The probability to decode each sequence is:

$$P(I_i = 1) = P_{\text{single}}(r_{\text{mean}}, m, b), \forall i \tag{21}$$

where $\pi_{r_{\text{mean}}}$ are obtained by using r_{mean} in the coupon collector's model. We can define a new binomial random variable X that represents the number of decoded sequences:

$$X \sim \text{Binom}\big(l, P_{\text{single}}(r_{\text{mean}}, m, b)\big) \tag{22}$$

And:

$$P_{\text{all}}(r_1, \ldots, r_l) \geq P(X \geq a) \tag{23}$$

However, the R_{all} reads are not evenly distributed across the l sequences and we therefore model this distribution using a multinomial distribution:

$$(R_1, \ldots, R_l) \sim \text{Multinom}\left(R_{\text{all}}, \left(\frac{1}{l}, \ldots, \frac{1}{l}\right)\right) \tag{24}$$

Remark: We note that in reality, biases in the combinatorial DNA channel may result in different distributions of the reads across the sequences. Since these biases differ between the technologies used for generating combinatorial DNA molecules, we chose to use the uniform multinomial distribution that does not require specific characterization of the channel. We also note that this distribution was shown to be optimal for some cases [13].

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

$$P_{\text{all}} = \sum_{\substack{(r_1,\ldots,r_l) \\ \sum r_i = R_{\text{all}}}} P(r_1, \ldots, r_l) P_{\text{all}}(r_1, \ldots, r_l) \tag{25}$$

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 for every sequence s_i we have:

$$P(I_i = 1) = P_{\text{single}}(r_i, m, b) \geq P_{\text{single}}(r_{\min}, m, b) \tag{26}$$

where $r_{\min} = \min_{j=1,\ldots,l} r_i$ and $\pi_{r_{\min}}$ are obtained by using $r_{\min}$ in the coupon collector's model. We therefore define X to be:

$$X \sim \text{Binom}\big(l, P_{\text{single}}(r_{\min}, m, b)\big) \tag{27}$$

And:

$$P_{\text{all}}(r_1, \ldots, r_l) \geq P(X \geq a) \tag{28}$$

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

$$P_{\text{all}} \geq \sum_{\substack{(r_1,\ldots,r_l) \\ \sum r_i = R_{\text{all}}}} P(r_1, \ldots, r_l) P(X \geq a) \tag{29}$$

In the multinomial distribution for $(R_1, \ldots, R_l)$, 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 $\min_{j=1,\ldots,l}(r_j) \geq \rho$. 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, $X_\rho \sim \text{Binom}(l, P_{\text{single}}(\rho, m, b))$.

We therefore have:

$$P_{\text{all}} \geq P\big(X_\rho \geq a\big) \sum_{\substack{(r_1,\ldots,r_l) \\ \sum r_i = R_{\text{all}} \\ \min_j r_j \geq \rho}} P(r_1, \ldots, r_l) \tag{30}$$

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

$$P_{\text{all}} \geq 1 - \delta \tag{31}$$

This can be achieved by choosing ρ such that:

1)

$$P\big(X_\rho \geq a\big) \geq \sqrt{1 - \delta} \tag{32}$$

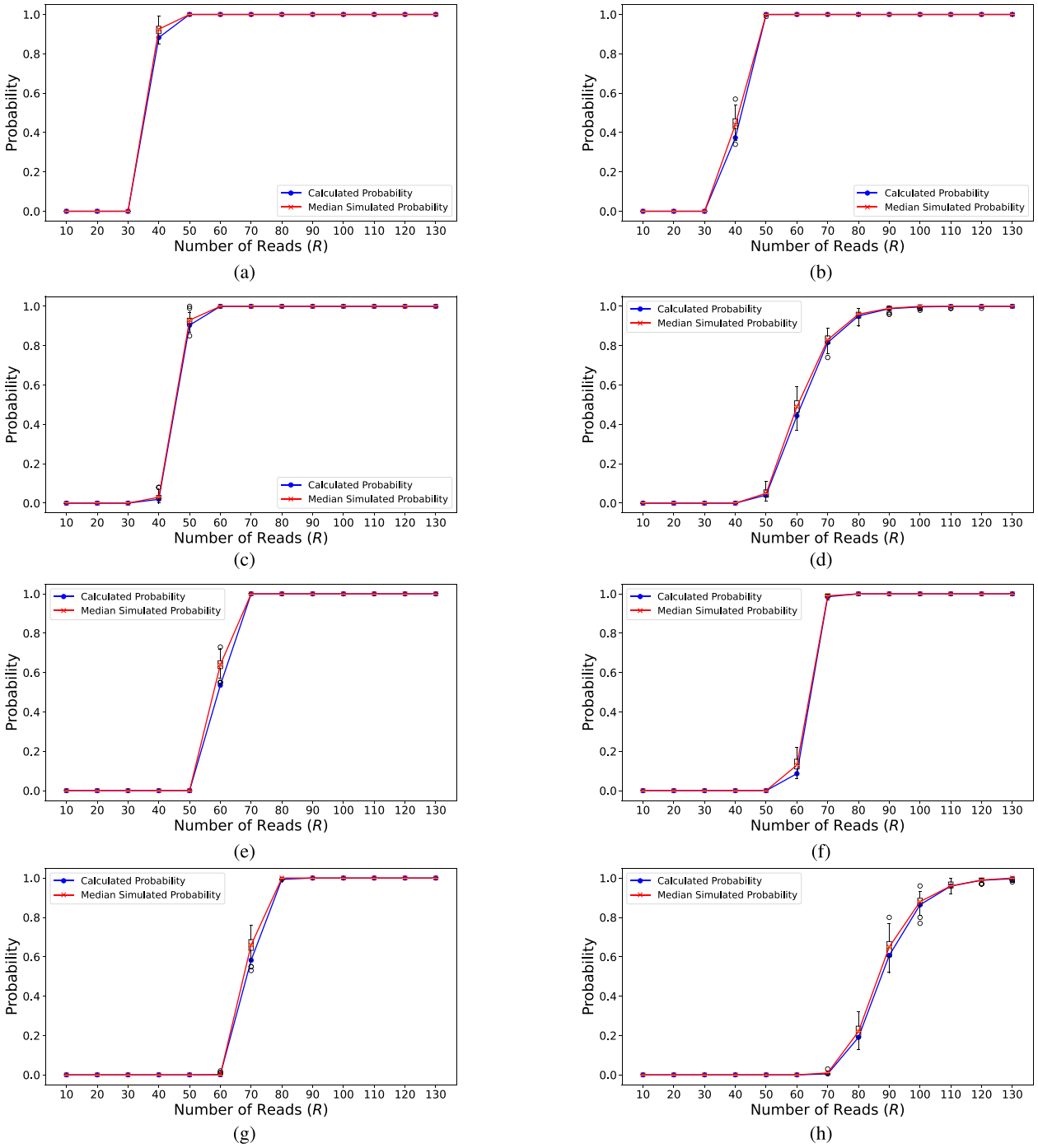


Fig. 5. Decoding probability of a complete combinatorial sequence with varying redundancy levels. Results shown for a sequence of length $m = 100$, with $K = 7$, and requiring $t = 4$. (a) Calculated decoding probability (blue line) as a function of the number of analyzed reads for redundancy level of $b = 85$. Median results from 50 simulation runs are presented (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 $\epsilon = 0.01$. (e-h) Like as (a-d), with $t = 2$.

And:

2)

$$\sum_{\substack{(r_1,\ldots,r_l)\\ \sum r_i = R_{\text{all}}\\ \min_j r_j \geq \rho}} P(r_1,\ldots,r_l) \geq \sqrt{1-\delta} \tag{33}$$

Since X_ρ has a binomial distribution, we can find ρ for which condition (32) holds. For condition (33), we use Sanov's theorem on the multinomial distribution as follows. For more on Sanov's theorem and the behavior of multinomials, see [26].

Sanov's theorem bounds the probability that the distribution of the reads into barcodes significantly deviates from the expected uniform ($(\frac{1}{l})$ for each) distribution, particularly where at least one sequence gets fewer than ρ reads. Fig. 7 demonstrates this using a simulation of 100,000 instances, each drawn from the multinomial distribution with

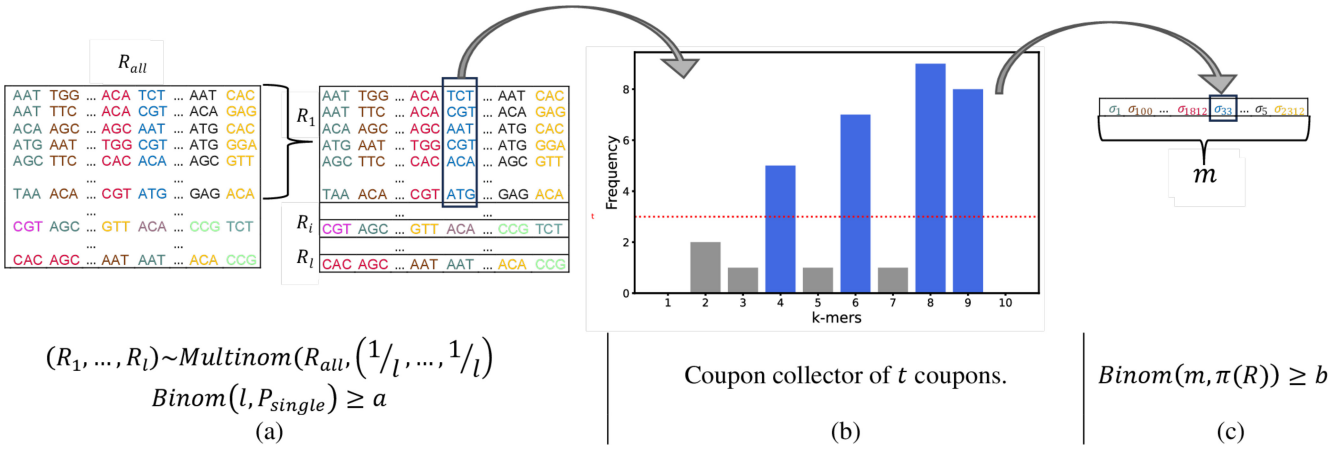


Fig. 6. Reconstructing a complete combinatorial message. (a) R_{all} reads are distributed between l sequences, and at least a sequences must be decoded. (b) The decoding probability of each of the letters is analyzed using the coupon collector's model (blue bins indicate the member k-mers). (c) Each sequence requires b of the m combinatorial letters to be decoded.

$p=(\frac{1}{50},\ldots,\frac{1}{50})$ and $R_{\text{all}}=4500$ or $R_{\text{all}}=5000$. The plots show the distribution of the minimal values obtained. Clearly, increasing R_{all} reduces the probability of the minimal value being lower than a fixed threshold ρ. Decreasing the threshold ρ yields a similar effect.

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

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

$$E(\rho)=\left\{P=(p_1,\ldots,p_l)\mid \sum p_i=1;\min_i(p_i)<\frac{\rho}{R_{\text{all}}}\right\} \tag{34}$$

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

$$D(P\|U)=\sum_{i=1}^{l}p_i\log\left(\frac{p_i}{q_i}\right) \tag{35}$$

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

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

Lemma 1:

$$\text{Let } U=\left(\frac{1}{l},\ldots,\frac{1}{l}\right),\text{Let } \alpha<\frac{1}{l} \tag{36}$$

$$\text{Let } P^*=\left(\alpha,\frac{1-\alpha}{l-1},\ldots,\frac{1-\alpha}{l-1}\right),\text{then} \tag{37}$$

$$\forall P=(p_1,\ldots,p_l),\text{s.t. }\exists i;p_i<\alpha \tag{38}$$

We have:

$$D(P\|U)\geq D(P^*\|U) \tag{39}$$

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

Sanov's theorem [26] provides a bound on the probability of observing any distribution within $E(\rho)$.

$$P(E(\rho))\leq(R_{\text{all}}+1)^l 2^{-R_{\text{all}}\zeta(\rho)} \tag{40}$$

where:

$$\begin{aligned}\zeta(\rho)&=D(P^*\|U)\\&=\sum_{i=1}^{l}p_i^*\log\left(\frac{p_i^*}{q_i}\right)\\&=\alpha\log(\alpha l)\ +(l-1)\left(\frac{1-\alpha}{l-1}\right)\log\left(\frac{\frac{1-\alpha}{l-1}}{\frac{1}{l}}\right)\\&=\alpha\log(\alpha l)\ +(1-\alpha)\log\left(\frac{l(1-\alpha)}{l-1}\right)\end{aligned} \tag{41}$$

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:

$$\sum_{\substack{(r_1,\ldots,r_l)\\ \sum r_i=R_{\text{all}}\\ \min_j r_j\geq\rho}}P(r_1,\ldots,r_l)=1-P(E(\rho)) \tag{42}$$

And so we get:

$$P_{\text{all}}\geq P\big(X_\rho\geq a\big)(1-P(E(\rho))) \tag{43}$$

$$P_{\text{all}}\geq P\big(X_\rho\geq a\big)\Big(1-(R_{\text{all}}+1)^l 2^{-R_{\text{all}}\zeta(\rho)}\Big) \tag{44}$$

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.

Fig. 8(a) 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 $(1-P(E))$ for a fixed number of overall reads $R_{\text{all}}=3000$ as a function of the threshold ρ. Clearly, $P(X_\rho>a)$ increases as ρ increases since each sequence s_i is decoded using more reads. On the other hand, as is demonstrated in

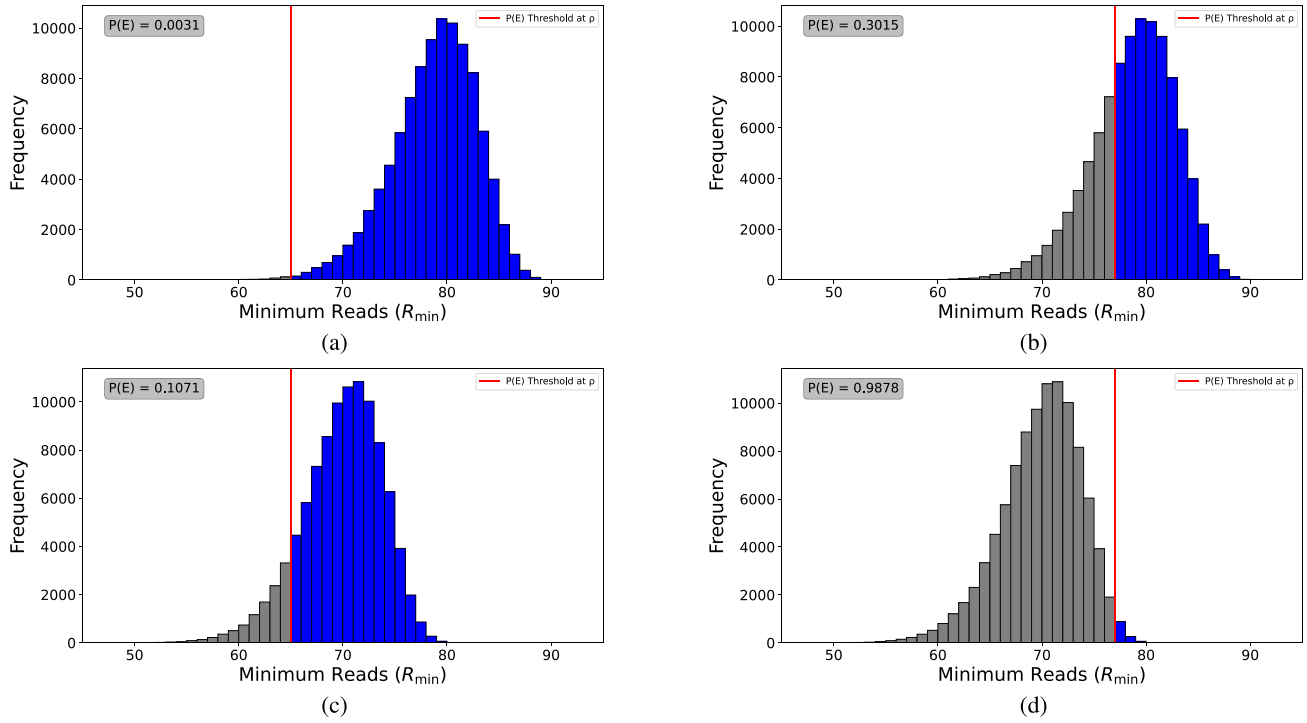


Fig. 7. The minimum value of a multinomial distribution $Y = \min_j(X_j)$ where $(X_1, \ldots, X_l) \sim \text{Multinom}(R_{\text{all}}, (\frac{1}{l}, \ldots, \frac{1}{l}))$. (a) A histogram of the values of Y attained in 100,000 instances with $l = 50$ and $R_{\text{all}} = 5000$. The red line represents $\rho = 65$. The gray box shows the probability $P(E(\rho)) = P(Y < \rho)$. (b-d) Like (a), for $(R_{\text{all}}, \rho) = (5000, 77), (4500, 65), (4500, 77)$.

Algorithm 2: Finding the 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. 5, increasing ρ decreases $1 - P(E(\rho))$, since fewer 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 presents an alternative approach for finding ρ using empirical simulations. Fig. 8(b) presents the probability $(1 - P(E(\rho)))$, calculated as shown in Fig. 7 by 100,000 instances of simulating the multinomial distribution with $R_{\text{all}} = 1000$. Clearly, this method yields a tighter bound on the decoding probability while also requiring the analysis of less reads overall. See Appendix D for details.

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.

1) Design 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 – Required threshold on the number of observed occurrences of each of the k-mers.
- m – Sequence length.
- b – Total number of letters required to be successfully decoded in each sequence.
- l – Total number of sequences in the message.
- a – Total number of sequences required for successful decoding.
- ϵ – Error probability of observing an invalid k-mer.

Input:
- δ – Acceptable failure rate.

Output:
- R_{all} – Required sequencing coverage.

2) Description of Tool Run: Fig. 9 presents a high-level description of the tool workflow. Given the design parameters

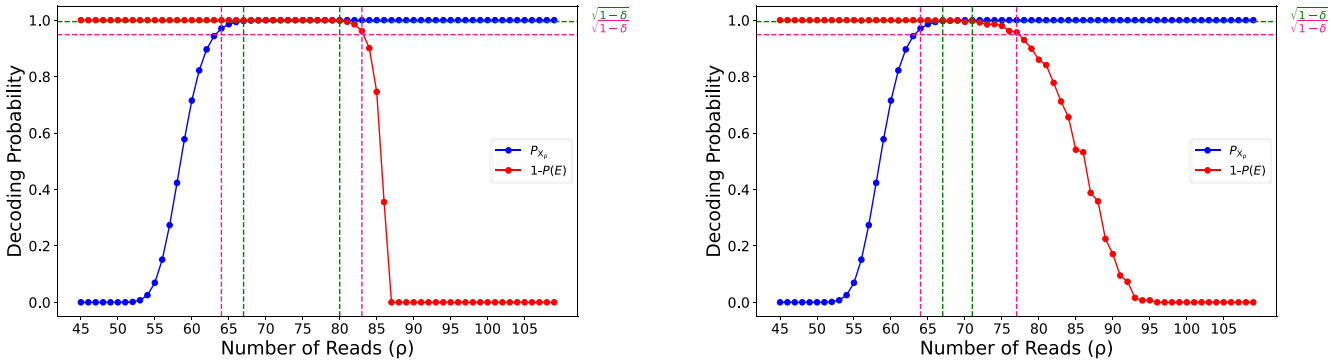


Fig. 8. Bounding the decoding probability. (a) Overall decoding probability $P(X_\rho \geq a)$ (blue line) and the Sanov's bound on the probability of obtaining a read distribution across the sequences with $\min_j(r_j) \leq \rho$, $1 - P(E(\rho))$ (red line) as functions of the threshold T for a fixed number of analyzed reads $R_{\text{all}} = 3000$. 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 $P(E(\rho))$ calculated using simulations instead of the Sanov's bound and where the total number of reads $R_{\text{all}} = 1000$.

K, t, ϵ, 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 $1-\delta$ (See Section IV-C). First, ρ is found such that the decoding of at least a sequences is ensured, $P(X_\rho \geq a) \geq \sqrt{1-\delta}$ (See Section IV-C). This calculation requires the probability to decode a single sequence, $P_{\text{single}}(\rho, m, b)$ (See Section IV-B), which uses the reconstruction probability of a single combinatorial letter $\pi_{K,t}(\rho)$ (See Section IV-A).

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

3) 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 I. Clearly, increasing the desired confidence level (smaller values for δ) requires the increasing of the sequencing coverage. Scaling up the system's capacity by taking l to be 10 times larger results in a proportional increase in R_{all}. Increasing the redundancy level (lower value for a) reduces the number of required reads to be 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.

4) Runtime Analysis: In terms of runtime complexity of evaluating R_{all} for any given set of parameters, we compared our tool to using Monte Carlo simulations. Table II presents the results for fixed values for R_{all} and ρ. Clearly an increase in a or l leads to a drastic increase in the simulation runtime, indicating that our approach is significantly faster and scales better to larger systems.

TABLE I
REQUIRED SEQUENCING COVERAGE FOR DIFFERENT SETS OF DESIGN PARAMETERS AND CONFIDENCE LEVELS

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

V. METHODS

A. Monte Carlo Simulations

1) Decoding Probability of a Single Combinatorial Letter: The decoding probability of a single combinatorial letter was calculated by simulating the distribution of R reads over K elements with uniform probability. Where applicable, an error probability ϵ was used to discard reads representing invalid k-mers. Then, a successful decoding was considered if all K k-mers where observed at least t times each. The process was repeated Q times to calculate the success rate. The median and boxplot of 50 repeats is presented. See Algorithm 3.

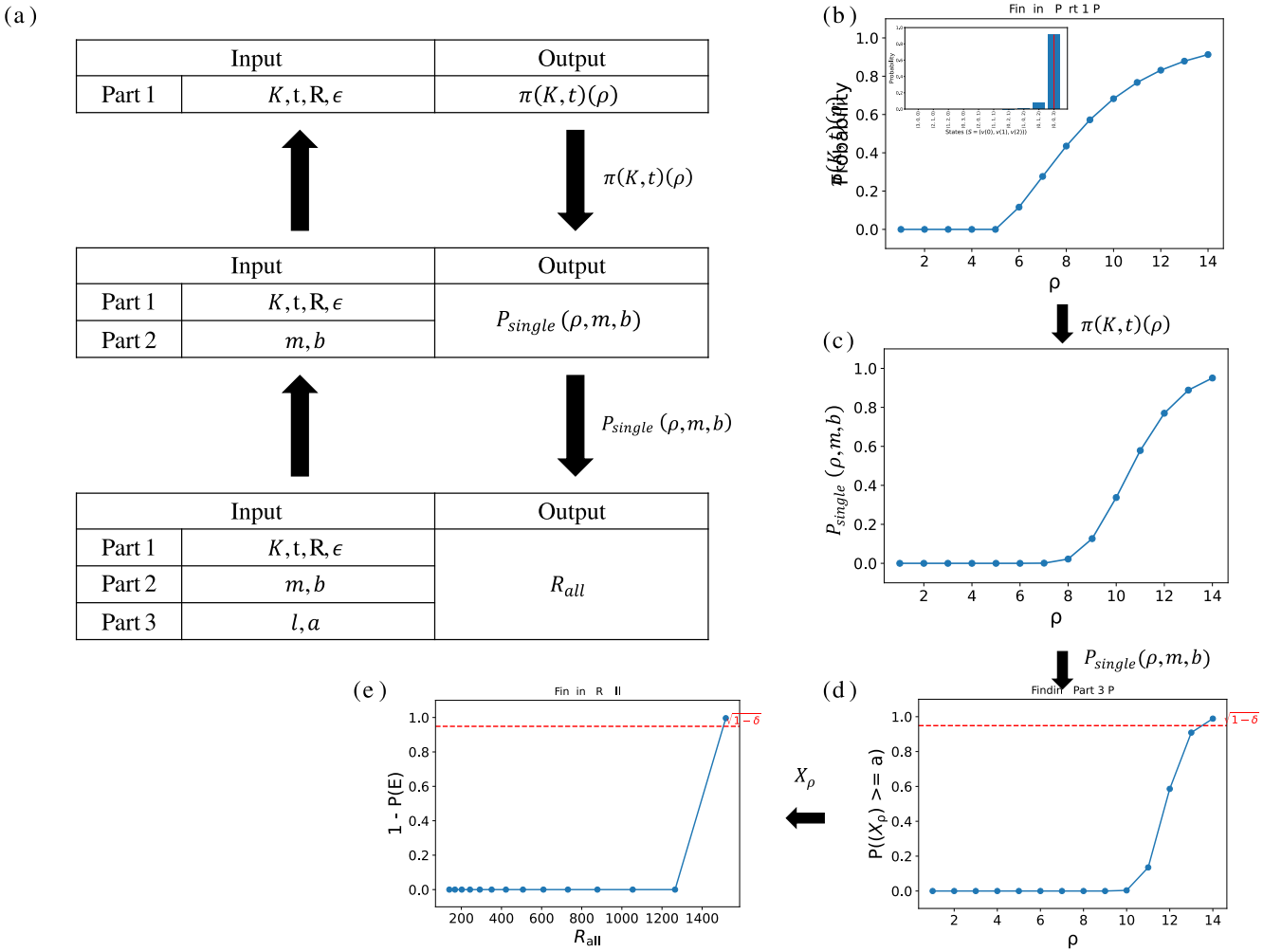


Fig. 9. Complexity calculation tool workflow. (a) Overview of the tool's run including internal dependencies, input parameters, and outputs for each part. (b) Reconstruction probabilities of a single combinatorial position, $\pi_{K,t}(\rho)$, calculated using the coupon collector's model (inset, like in Fig. 3) as a function of the threshold ρ. (c) Decoding probability for a full-length combinatorial sequence, $P_{\text{single}}(\rho, m, b)$, calculated using the binomial model with the probabilities from (a) as input. Plotted as a function of the threshold ρ. (d) Finding ρ. Full message decoding probability, $P(X_\rho \geq a)$, calculated using the binomial model for X_ρ obtained from (c). Plotted as a function of the threshold ρ. The target confidence level $\sqrt{1-\delta}$ is presented in the red dotted line. (e) Finding R_{all} given the selected ρ. The probability of considering enough read distributions (across the l sequences), $1 - P(E(\rho))$, 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 red dotted line.

TABLE II
RUNTIME COMPARISON BETWEEN MONTE CARLO SIMULATIONS (WHERE Q=100) AND A DIRECT CALCULATION WITH TWO DIFFERENT APPROACHES FOR CALCULATING $P(E)$ (SEE APPENDIX D)

R_{all}	ρ	k	t	m	b	l	a	ϵ	δ	Algorithm A runtime (sec)	Algorithm B runtime (sec)	Algorithm 5 runtime (sec)
3000	300	5	2	10	8	10	8	0.01	0.1	0.492	0.165	0.447
				100	80	10	8	0.01	0.1	0.41	0.161	4.436
				100	80	100	80	0.01	0.1	0.428	0.159	38.67
		7	3	10	8	10	8	0.01	0.1	0.916	0.706	0.454
				100	80	10	8	0.01	0.1	0.905	0.706	4.66
				100	80	100	80	0.01	0.1	0.92	0.703	39.884

2) Decoding Probability of a Combinatorial Sequence: The decoding probability of a combinatorial sequence of length m was calculated by repeating the simulation for a single letter m times. Then, a successful decoding was considered if at least b of the m letters where successfully decoded. The process was repeated Q times to calculate the success rate. The median and boxplot of 50 repeats is presented. See Algorithm 4.

3) Decoding Probability of a Complete Combinatorial Message: The decoding probability of a complete combinatorial message with l sequences was calculated by first simulating the distribution of R_{all} reads over the l sequences. Then, the simulation for a single combinatorial sequence was repeated for each sequence using the associated R_i reads. A successful decoding of the complete message was considered if at least a of the l sequences where successfully decoded.

The process was repeated Q times to calculate the success rate. The median and boxplot of 50 repeats is presented. See Algorithm 5.

B. Runtime Analysis

The simulations and calculations were performed on a personal computing system equipped with an Intel® Core™ i5-8250U CPU, which has a base clock speed of 1.60 GHz and can boost up to 1.80 GHz. The system was configured with 8.00 GB of RAM (7.84 GB usable) to facilitate the computational demands of the simulation processes. It operated under a 64-bit Windows 11 Home edition, ensuring that the software utilized for simulations could leverage the x64-based processor architecture for optimal performance.

VI. Conclusion

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 calculating the decoding probability and provide a tool for computing its probability. This solution is, however, limited in its scale due to the size of the state space. Further work can be done to allow this model to be scaled up, either by developing more efficient computation or by developing an approximation to the model.

One of the key aspects of the combinatorial approach is the strategic selection of Ω that consists of easily distinguishable k-mers. This, together with the use of a threshold $t > 1$ in the reconstruction algorithm (See Algorithm 1), effectively mitigates k-mer mix-up errors, as was demonstrated in [15]. We therefore chose to ignore k-mer mix-up errors 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-based 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's Problem Video

The coupon collector's parameters that are shown in the video, are: $K = 5$, $t = 2$, $R = 30$. See file *A. Evolution of Probability in the Coupon Collector Problem Video K=5, t=2, R=30.gif*.

Algorithm 3: Simulation: Reconstruction of a Single Combinatorial Letter

Data: K, t, R, Q Number of unique items K, threshold t, number of reads R, number of simulations Q.
Result: Success rate of achieving at least t occurrences of each item in R rounds over Q simulations.

Initialize $success\_count \leftarrow 0$
for $q \leftarrow 1$ **to** Q **do**
  Let *reads* be a sequence generated by R independent, uniformly distributed random selections from the set $\{1, 2, \ldots, K\}$ with replacement
  if *for every item* $i \in \{1, 2, \ldots, K\}$, i *appears at least* t *times in reads* **then**
    $success\_count \leftarrow success\_count + 1$
  end
end
Compute the success rate as $success\_rate \leftarrow \frac{success\_count}{Q}$
return $success\_rate$

Algorithm 4: Median Simulated Probability: Reconstruction of a Combinatorial Sequence

Data: K, t, m, k, b, R, Q. Number of k-mers *K*, threshold *t*, sequence length *m*, number of selections *n*, required successful decodings *b*, number of reads per position *R*, number of simulations *Q*.
Result: Success rate of reconstructing at least b positions from m in the sequence over Q simulations.

Initialize parameters K, t, m, n, b, R, Q
Initialize $success\_count \leftarrow 0$
for $\_ \leftarrow 1$ **to** Q **do**
  $decoded\_count \leftarrow$ Sum of successes from Algorithm 3 (Calling Algorithm 3 with $Q = 1$), for m positions
  if $decoded\_count \geq b$ **then**
    $success\_count \leftarrow success\_count + 1$
  end
end
$success\_rate \leftarrow \frac{success\_count}{Q}$
return $success\_rate$

Appendix B
Classical Coupon Collector's Problem

$$\pi_{K,1}(R) = \sum_{i=0}^{K} (-1)^i \binom{K}{i} \left(1 - \frac{i}{K}\right)^R \tag{45}$$

$\pi_{K,t}(R)$ is the probability of collecting all *n* unique coupons within *R* trials. We will show that,

$$\pi_{K,1}(R) = P\big(T_{K,1} \leq R\big) = \sum_{i=0}^{K} (-1)^i \binom{K}{i} \left(1 - \frac{k}{K}\right)^R \tag{46}$$

The coupon collector's problem can be approached using the principle of inclusion-exclusion. The formula calculates the

Algorithm 5: Median Simulated Probability: Complete Message Decoding Simulation

Data: $K, t, m, n, b, l, a, R_{\text{all}}, Q$. Number of unique k-mers *K*, threshold *t*, sequence length *m*, selections per sequence *n*, required successful decodings *b*, total number of sequences *l*, required sequences decoded *a*, total reads R_{all}, number of simulations *Q*.

Result: Success rate of decoding at least *a* sequences out of *l* in *Q* simulations.

Initialize parameters $K, t, m, n, b, l, a, R_{\text{all}}, Q$
Initialize $success\_count \leftarrow 0$
for _ $\leftarrow 1$ **to** *Q* **do**
  $reads\_distribution \leftarrow \text{multinomial}(R_{\text{all}}, [\frac{1}{l}] \times l)$
  Initialize $decoded\_sequences \leftarrow 0$
  for R_j *in* $reads\_distribution$ **do**
    if *Algorithm 4 output (with* $Q = 1) > 0$ **then**
      $decoded\_sequences \leftarrow decoded\_sequences + 1$
    end
  end
  if $decoded\_sequences \geq a$ **then**
    $success\_count \leftarrow success\_count + 1$
  end
end
$success\_rate \leftarrow \frac{success\_count}{Q}$
return $success\_rate$

Algorithm 6: Calculated probability: Reconstruction of a Single Combinatorial Letter

Data: K, t, ϵ, R. Number of unique k-mers *K*, *t* count of each k-mer, error probability ϵ, and number of reads *R*.

Result: Probability distribution over states after *R* reads.

Initialize state space $S \leftarrow \binom{K+t}{t}$ all combinations.
Initialize transition matrix $A \leftarrow 0^{(|S| \times |S|)}$
for *each state* s_i *in S* **do**
  for *each j in* $0, \ldots, t$ **do**
    if $v(j) > 0$ **then**
      $v(j) -= 1, v(j+1) += 1$
      Find index of new state s_j in *S*
      Compute probability for transitioning to a new state, $A[s_i, s_j] \leftarrow P(s_i, s_j)$, where $P(s_i, s_j) = (1 - \epsilon) \times \frac{v(j)}{K}$
    end
  end
  Compute probability for staying in the same state (without transitioning to a new state), $A[s_i, s_i] \leftarrow P(s_i, s_i)$, where $P(s_i, s_i) = (1 - \epsilon) \times \frac{v(t)}{K} + \epsilon$
end
Initialize initial vector $v_0 \leftarrow [1, 0, \ldots, 0]^T$, corresponding to the size of all combinations $\binom{K+t}{t}$
Compute result $v_R \leftarrow v_0 A^R$
Calculate $\pi_{K,t}(R)$, the total probability for target states, by summing probabilities of target states
return $\pi_{K,t}(R)$

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.

$$P(A_i) = \left(1 - \frac{1}{K}\right)^R \tag{47}$$

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:

$$\pi_{K,1}(R) = 1 - P\left(\bigcup_{i=1}^{K} A_i\right) \tag{48}$$

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

$$P\left(\bigcup_{i=1}^{K} A_i\right) = \sum_{j=1}^{K} (-1)^{j-1} \binom{K}{j} \left(1 - \frac{j}{K}\right)^R \tag{49}$$

And finally,

$$\begin{aligned} \pi_{K,1}(R) &= 1 - P\left(\bigcup_{i=1}^{K} A_i\right) \\ &= 1 - \sum_{j=1}^{K} (-1)^{j-1} \binom{K}{j} \left(1 - \frac{j}{K}\right)^R \\ &= \sum_{j=0}^{K} (-1)^{j} \binom{K}{j} \left(1 - \frac{j}{K}\right)^R \end{aligned} \tag{50}$$

This follows from:

$$\left(\bigcup_{i=1}^{K} A_i\right) = \sum_{j=1}^{K} (-1)^{j-1} \sum_{I \subseteq \{1,\ldots,K\}, |I|=j} P(A_I) \tag{51}$$

where $A_I = \bigcap_{i \in I} A_i$. For $j = 1$:

$$P(A_I) = P(A_i) = \left(1 - \frac{1}{K}\right)^R \tag{52}$$

For $j = 2$:

$$P(A_I) = P(A_m \cap A_l) = \left(1 - \frac{2}{K}\right)^R \tag{53}$$

And generally:

$$P(A_I) = \left(1 - \frac{j}{K}\right)^R \tag{54}$$

And clearly:

$$|\{I; I \subseteq \{1, \ldots, K\}, |I| = j\}| = \binom{K}{j} \tag{55}$$

APPENDIX C
PROOF OF LEMMA 1

Lemma 1:

$$\text{Let } U = \left(\frac{1}{l}, \ldots, \frac{1}{l}\right), \text{Let } \alpha < \frac{1}{l} \tag{56}$$

Algorithm 7: Calculated probability: Reconstruction of a Combinatorial Sequence

Data: $K, t, \epsilon, R, m, b, method$. Number of unique k-mers *K*, threshold *t*, error probability ϵ, number of reads *R*, total number of letters *m*, number of letters required *b*, calculation method *method*.

Result: Probability of successfully reconstructing at least *b* letters from the combinatorial sequence.

Initialize parameters $K, t, \epsilon, R, m, b, method$
Calculate probability of single letter reconstruction $\pi_{K,t}(R)$ using Algorithm 6
if $method =$ 'binomial' **then**
    Use binomial distribution to calculate probability of success
    $prob \leftarrow$ binom.sf$(b-1, m, \pi_{K,t}(R))$
else if $method =$ 'normal' **then**
    Use normal approximation to calculate probability of success
    $mean \leftarrow m \cdot \pi_{K,t}(R)$
    $std\_dev \leftarrow \sqrt{m \cdot \pi_{K,t}(R) \cdot (1 - \pi_{K,t}(R))}$
    $P_{single} \leftarrow 1-$ norm.cdf$(b-1, mean, std_dev)$
return $P_{single}, \pi_{K,t}(R)$

Algorithm 8: Calculated probability: Reconstructing a Complete Combinatorial Message

Data: $K, t, m, b, l, a, \epsilon, \text{method}, \delta, P(E(\rho))_{method}$. Total number of unique k-mers *K*, threshold *t*, sequence length *m*, number of letters required *b*, number of sequences *l*, number of sequences required to be decoded *a*, error probability ϵ, calculation method, acceptable error threshold δ.

Result: Required sequencing depth R_{all} ensuring decoding with probability $1 - \delta$.

Initialize parameters $K, t, m, b, l, a, \epsilon, \delta, \text{method}, P(E(\rho))_{method}$
$\rho \leftarrow$ `find_ρ()`
$R_{\text{all}} \leftarrow$ `calc_R_all(`ρ`)`
Function `find_ρ`:
    $\rho \leftarrow 1$
    c const
    while *True* **do**
        $P_{single} \leftarrow$ Using Algorithm 7 to calculate probability of decoding a combinatorial sequence. Using P_{single} we calculate: $P(X_\rho) \leftarrow$ Success probability of decoding at least *a* out of *l* sequences using.
        if $P(X_\rho) \geq \sqrt{1-\delta}$ **then return** ρ $\rho \leftarrow \rho + c$
    end
End Function
Function `calc_R_all(`*T*`)`:
    $R_{\text{all}} \leftarrow \rho \times l$
    while *True* **do**
        Calculate the overall error probability $P(E)$ with R_{all} and ρ
        if $1 - P(E) \geq \sqrt{1-\delta}$ **then break**
        $R_{\text{all}} \leftarrow \lceil R_{\text{all}} \times 1.05 \rceil$
    end
End Function
return R_{all}

$$\text{Let } P^* = \left(\alpha, \frac{1-\alpha}{l-1}, \ldots, \frac{1-\alpha}{l-1}\right), \text{then} \tag{57}$$

$$\forall P = (p_1, \ldots, p_l),\ s.t.\ \exists i;\ p_i < \alpha \tag{58}$$

We have:

$$D(P\|U) \geq D(P^*\|U) \tag{59}$$

Proof:

$$D(P\|U) = \sum_{i=1}^{l} p_i \log(lp_i) \tag{60}$$

$$D(P^*\|U) = \alpha \log(\alpha l)\ + (1-\alpha)\log\left(l\frac{1-\alpha}{l-1}\right) \tag{61}$$

We solve:

$$\min D(P) = \sum_{i=1}^{l} p_i \log(lp_i) \tag{62}$$

Subject to:
1)

$$\sum_{i=1}^{l} p_i = 1 \tag{63}$$

2)

$$p_1 \leq \alpha \quad \text{(WLOG)} \tag{64}$$

Therefore, the Lagrangian is:

$$L(p_1, p_2, \ldots, p_l, \lambda, \mu) = \sum_{i=1}^{l} p_i \log(lp_i) - \lambda\left(\sum_{i=1}^{l} p_i - 1\right) - \mu(p_1 - \alpha) \tag{65}$$

The KKT conditions are:

- *Stationarity* $\frac{\partial L}{\partial p_i} = 0$:

$$\text{for } i > 1,\ \frac{\partial L}{\partial p_i} = 0 \rightarrow p_i = e^{(\lambda-1)}/l \tag{66}$$

$$\text{for } i = 1,\ \frac{\partial L}{\partial p_1} = 0 \rightarrow p_1 = e^{(\lambda-1+\mu)}/l \tag{67}$$

- *Primal feasibility:*

$$\sum_{i=1}^{l} p_i = 1 \tag{68}$$

$$p_1 - \alpha < 0 \tag{69}$$

- *Dual feasibility:*

$$\mu, \lambda \geq 0 \tag{70}$$

- *Complementary slackness:*

$$\mu(p_1 - \alpha) = 0 \tag{71}$$

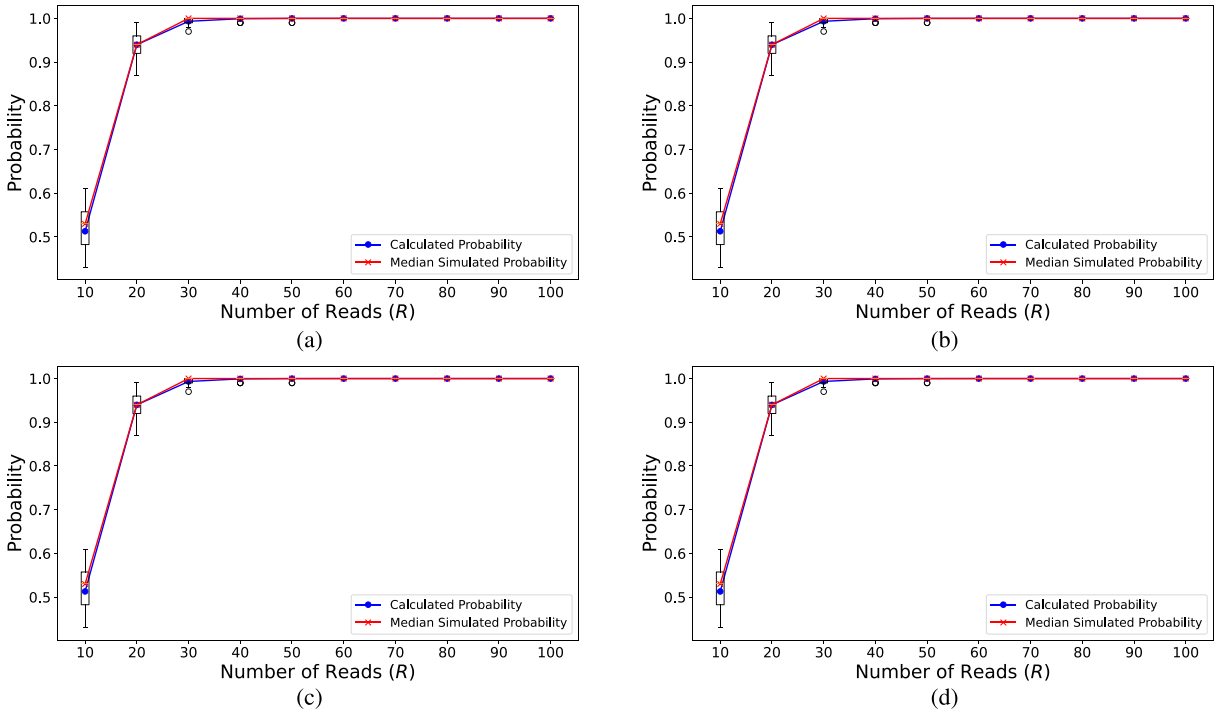


Fig. 10. Decoding probability for a varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analysis was conducted for $K = 5$ and $\epsilon = 0.01$. (a) Results for $t = 1$, the blue line corresponds to the calculated probability 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, respectively.

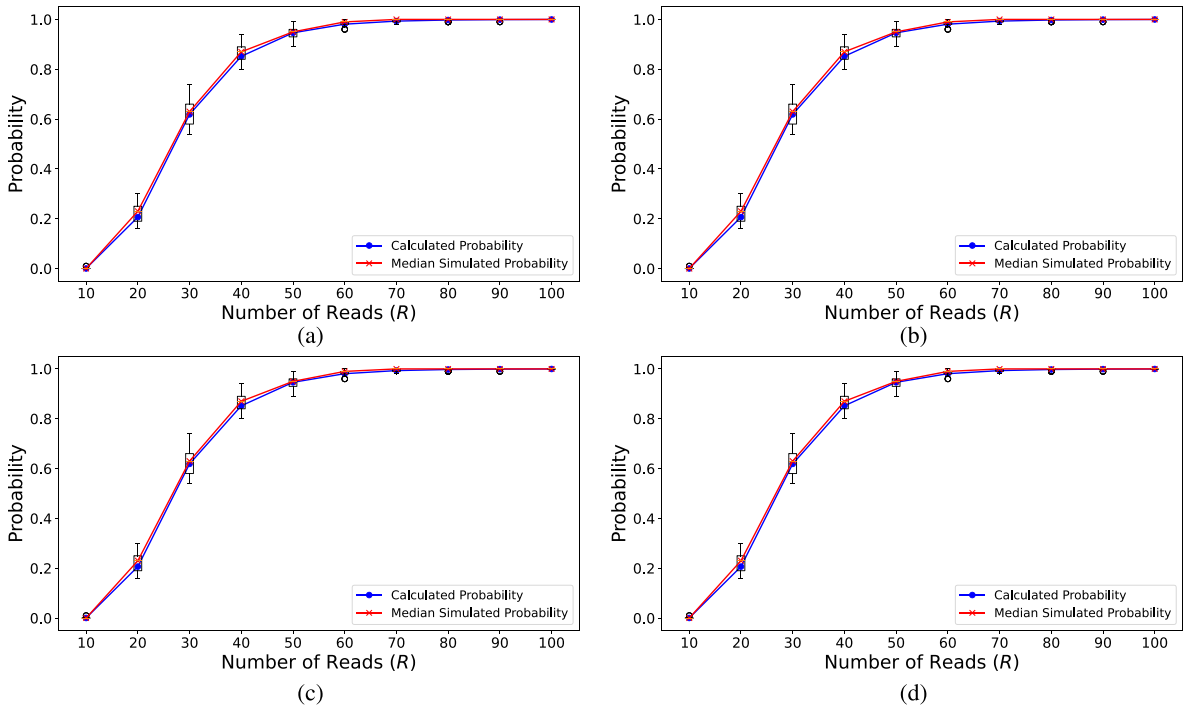


Fig. 11. Decoding probability for a varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analysis was conducted for $K = 10$ and $\epsilon = 0.01$. (a) Results for $t = 1$, the blue line corresponds to the calculated probability 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, respectively.

Expressing λ using the primal feasibility (68):

$$p_1 + (1-l)p_i = 1 \tag{72}$$

Substituting p_i and p_1 from (66) and (67):

$$\lambda = \log\left(\frac{l}{e^{\mu} + l - 1}\right) + 1 \tag{73}$$

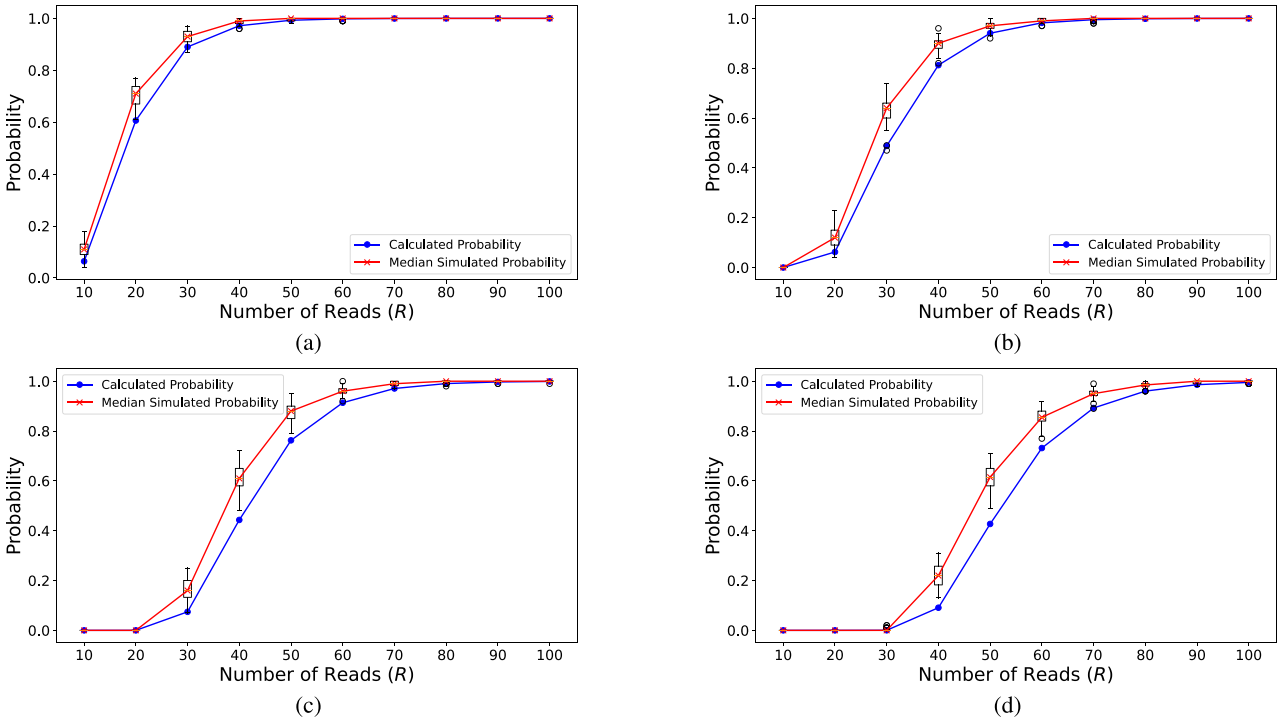


Fig. 12. Decoding probability for a varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analysis was conducted for $K = 7$ and $\epsilon = 0.1$. (a) Results for $t = 1$, the blue line corresponds to the calculated probability 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, respectively.

Expressing p_1 with λ from (73):

$$p_1 = \frac{e^{\mu}}{l + e^{\mu} - 1} \tag{74}$$

Expressing μ:

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

If $p_1 = \alpha$, we substitute α for p_1 in (74):

$$\alpha = \frac{e^{\mu}}{l + e^{\mu} - 1} \rightarrow \mu = \log\left(\frac{\alpha(l-1)}{1-\alpha}\right) \tag{75}$$

Using the original expressions for p_i from (66), and substituting μ we expressed in (75), we get:

$$\text{for } i > 1, \quad p_i = \frac{1}{l + \frac{\alpha(l-1)}{1-\alpha} - 1} = \frac{1-\alpha}{l-1} \tag{76}$$

and recall that $p_1 = \alpha$.

If $p_1 \neq \alpha$, we substitute $\mu = 0$ for p_1 in (67):

$$p_1 = \frac{1}{l} \tag{77}$$

Using the original expressions for p_i from (66), and substituting $\mu = 0$, we get:

$$\text{for } i > 1, \quad p_i = \frac{e^{\lambda-1}}{l} = \frac{e^{\log(l/(e^{\mu}+l-1))}}{l} = \frac{l}{e^{\mu}+l-1}/l = \frac{1}{l} \tag{78}$$

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

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

APPENDIX D
TOOL IMPLEMENTATION

We provide a Python code for calculating the required sequencing depth given the system design parameter. Figure 9 presents an overview of the tool run. The implementation details of the different steps are outlined below.

A. Reconstruction Probability of a Single Combinatorial Letter

For the calculation of the reconstruction probability of a single combinatorial letter, the Markov Chain (MC) representation of the coupon collector's process is used. See Algorithm 6.

B. Reconstruction Probability of a Combinatorial Sequence

For the calculation of the reconstruction probability of a combinatorial sequence, first the reconstruction probability for a single letter, $\pi_{K,t}(R)$, was calculated. Then, the reconstruction probability of the sequence is calculated using the binomial distribution or the normal approximation. See Algorithm 7.

C. Reconstruction Probability of a Complete Combinatorial Message

The calculation of the bound for the reconstruction probability of a complete combinatorial message is performed by splitting the calculation in two. First, a threshold ρ is found such that $P(X_\rho \geq a) \geq \sqrt{1-\delta}$. This is done by an iterative search that uses the decoding probability calculation

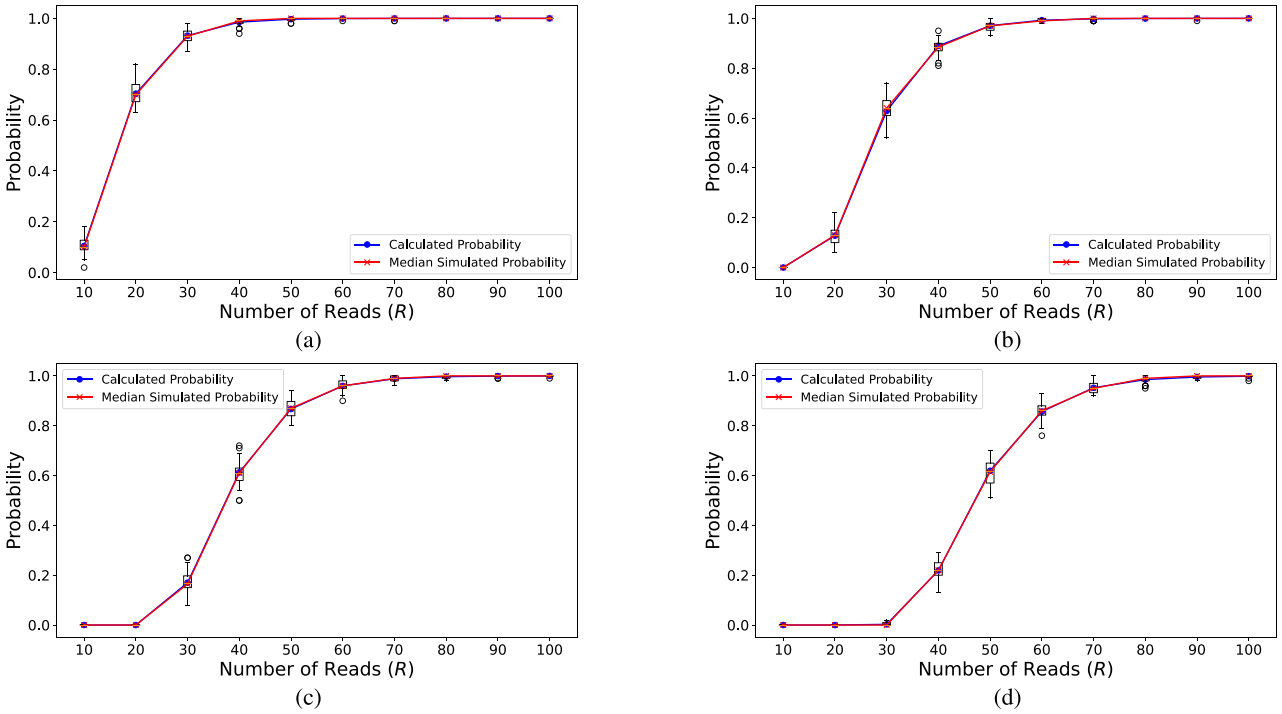


Fig. 13. Decoding probability for a varying number of analyzed reads (R) for different thresholds (t). Each subplot corresponds to a different threshold value (t). The analysis was conducted for $K = 7$ and $\epsilon = 0$. (a) Results for $t = 1$, the blue line corresponds to the calculated probability 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, respectively.

for a single sequence and the binomial model. Next, the required number of total reads R_{all} that ensures that $1 - P(E(\rho)) \geq \sqrt{1-\delta}$ is found either by using Sanov's bound or by a numerical simulation of the uniform distribution. See Algorithm 8.

Appendix E
Examples for Reconstruction of a Single Combinatorial Letter

More examples for decoding probability for varying number of analyzed reads (R) for different thresholds (t), with varying number of K and ϵ.

A. Decoding Probability for Varying Number of Analyzed Reads (R) for Different Thresholds (t), and Varying Numbers of K

See Fig. 10, and Fig. 11.

B. Decoding Probability for Varying Number of Analyzed Reads (R) for Different Thresholds (t), and Varying Numbers of ϵ

See Fig. 12, and Fig 13.

Acknowledgment

The authors of this paper thank the Yakhini Research Group for the fruitful discussions. The authors also thank Daniella Bar-Lev for her insightful comments, the DNA Storage Lab at the Technion (Israel Institute of Technology) for their ongoing collaboration, and the team of the Machine Learning and Data Science program at Reichman University for sharing their broad knowledge with us. No animal or human subjects were involved in this work.

Code Availability

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

References

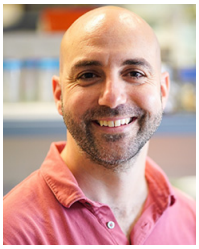
[1] J. Rydning, *Worldwide IDC Global Datasphere Forecast, 2022–2026: Enterprise Organizations Driving Most of the Data Growth*, Int. Data Corp., Needham, MA, USA, 2022.
[2] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.
[3] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *Nature*, vol. 494, no. 7435, pp. 77–80, 2013.
[4] 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.*, vol. 37, no. 10, pp. 1229–1236, 2019.
[5] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, 2017.
[6] L. Organick et al., "Random access in large-scale DNA data storage," *Nat. Biotechnol.*, vol. 36, no. 3, pp. 242–248, 2018.
[7] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 1, p. 5011, 2017.
[8] E. M. LeProust et al., "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process," *Nucl. Acid. Res.*, vol. 38, no. 8, pp. 2522–2540, 2010.
[9] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the dna data storage channel," *Sci. Rep.*, vol. 9, no. 1, p. 9663, 2019.

[10] I. Shomorony and R. Heckel, "Information-theoretic foundations of dna data storage," *Found. Trends® Commun. Inf. Theory*, vol. 19, no. 1, pp. 1–106, 2022.
[11] S. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: Trends and methods," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 1, no. 3, pp. 230–248, 2015.
[12] "Preserving our digital legacy: An introduction to dna data storage," DNA Data Storage Alliance, San Francisco, CA, USA, White Paper, 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," 2023, *arXiv:2305.05656*.
[14] S. Chandak et al., "Improved read/write cost tradeoff in dna-based data storage using LDPC codes," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, 2019, pp. 147–156.
[15] I. Preuss, M. Rosenberg, Z. Yakhini, and L. Anavy, "Efficient DNA-based data storage using shortmer combinatorial encoding," *Sci. Rep.*, vol. 14, p. 7731, Apr. 2024.
[16] N. Roquet et al., "DNA-based data storage via combinatorial assembly," bioRxiv, 2021.
[17] Y. Yan, N. Pinnamaneni, S. Chalapati, C. Crosbie, and R. Appuswamy, "Scaling logical density of DNA storage with enzymatically-ligated composite motifs," *Sci. Rep.*, vol. 13, Sep. 2023, Art. no. 15978.
[18] Y. Choi et al., "High information capacity dna-based data storage with augmented encoding characters using degenerate bases," *Sci. Rep.*, vol. 9, no. 1, p. 6582, 2019.
[19] M. Blawat et al., "Forward error correction for DNA data storage," *Procedia Comput. Sci.*, vol. 80, pp. 1011–1022, Jan. 2016.
[20] S. Chandak et al., "Overcoming high nanopore basecaller error rates for dna storage via basecaller-decoder integration and convolutional codes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 8822–8826.
[21] 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.
[22] W. Feller, *An Introduction to Probability Theory and Its Applications*. Hoboken, NJ, USA: Wiley, vol. 1, no. 2, p. 35, 1967.
[23] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discr. Appl. Math.*, vol. 39, no. 3, pp. 207–229, 1992.
[24] D. J. Newman, "The double dixie cup problem," *Am. Math. Month.*, vol. 67, no. 1, pp. 58–61, 1960.
[25] P. Neal, "The generalised coupon collector problem," *J. Appl. Probabil.*, vol. 45, no. 3, pp. 621–629, 2008.
[26] I. N. Sanov, "On the probability of large deviations of random variables," Office of Sci. Res., United States Air Force, Washington, DC, USA, Rep., 1958.

Inbal Preuss (Student Member, IEEE) received the B.Sc. degree in Computer Science from Reichman University, Herzliya, Israel, in 2019, where she is currently pursuing her Doctoral degree at the Computer Science Department. She is a Research Assistant at the Technion—Israel Institute of Technology, Haifa, and a member of the Yakhini Research Group at Reichman University and the Technion—Israel Institute of Technology. Her research interests include DNA-based data storage, CRISPR tools and methods, coding theory, computational biology, and synthetic biology.

Ben Galili received the B.Sc. degree in computer science and mathematics from Ben Gurion University and the M.Sc. degree in computer science from Reichman University. He is currently pursuing the Ph.D. degree with the Technion—Israel Institute of Technology, Haifa. He is the Academic Director of the M.Sc. degree in machine learning and data science and a Lecturer with the Efi Arazi School of Computer Science, Reichman University. His research interests include machine learning, statistics, algorithms, and computational biology.

Zohar Yakhini (Member, IEEE) received the Ph.D. degree in mathematics from Stanford University in 1996. He is an Associate Professor of Computer Science with Reichman University Herzliya and a Visiting Associate Professor of Computer Science with Technion—Israel Institute of Technology, Haifa. After graduation, he established the bioinformatics and computational biology research with Agilent Laboratories, contributing to Agilent's microarray and synthetic DNA technology and solutions development for over 20 years. With his leadership the Yakhini Research Group has developed several important data analysis methods and tools, including mHG and GOrilla, and made contributions to ground breaking cancer biology and molecular biology research projects, including the computational aspects of DNA copy number measurement and of manufacturing synthetic DNA. He is the Head of the Machine Learning and Data Science M.Sc. Program with the School of Computer Science, Reichman University. His research interests include machine learning, statistics and data analysis, especially methods development and applications to molecular biology and synthetic biology.



Leon Anavy received the B.Sc. degree in industrial engineering and management, the M.Sc. degree in biology, and the Ph.D. degree in computer science from the Technion—Israel Institute of Technology, Haifa, Israel, in 2012, 2015, and 2020, respectively. He is a Lecturer with the Efi Arazi School of Computer Science, Reichman University and with the Faculty of Computer Science at the Technion—Israel Institute of Technology. He is also a Visiting Researcher with Verily Life Science, an Alphabet precision health company where he is part of the AI research group. From 2020 to 2022, he served as the Academic Director of the M.Sc. degree in machine learning and data science with Reichman University. His research interests lie at the intersection of computer science, machine learning, computational biology, and synthetic biology.