Permutation Recovery for DNA Data Storage

Shubhransh Singhvi[*], **Charchit Gupta**[†], **Avital Boruchovsky**[‡], **Han Mao Kiah**[§] and **Eitan Yaakobi**[‡]
[*]Signal Processing & Communications Research Center, IIIT Hyderabad, India
[†]IIIT Hyderabad, India
[‡]Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel
[§]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

Abstract

Owing to its immense storage density and durability, DNA has emerged as a promising storage medium. However, due to technological constraints, data can only be written onto many short DNA molecules called *data blocks* that are stored in an unordered way. To handle the unordered nature of DNA data storage systems, a unique *address* is typically prepended to each data block to form a *DNA strand*. However, DNA storage systems are prone to errors and generate multiple noisy copies of each strand called *DNA reads*. Thus, we study the problem of *permutation recovery* for DNA data storage.

The permutation recovery problem for DNA data storage requires one to reconstruct the addresses or in other words to uniquely identify the noisy reads. By successfully reconstructing the addresses, one can essentially determine the correct order of the data blocks, effectively solving the clustering problem.

We first show that we can almost surely identify all the noisy reads under certain mild assumptions. We then propose a permutation recovery procedure and demonstrate that on average the procedure uses only a fraction of $\mathcal{M}^2$ data comparisons(when $\mathcal{M}$ is the number of reads). Specifically, ...

Index Terms

DNA Data Storage, Permutation Recovery, Clustering, Bee Identification.

I. Introduction

The need for a more durable and compact storage system has become increasingly evident with the explosion of data in modern times. While magnetic and optical disks have been the primary solutions for storing large amounts of data, they still face limitations in terms of storage density and physical space requirements. Storing a zettabyte of data using these traditional technologies would necessitate a vast number of units and considerable physical space.

The idea of using macromolecules for ultra-dense storage systems was recognized as early as the 1960s when physicist Richard Feynman outlined his vision for nanotechnology in his talk 'There is plenty of room at the bottom'. Using DNA is an attractive possibility because it is extremely dense (up to about 1 exabyte per cubic millimeter) and durable (half-life of over 500 years). Since the first experiments conducted by Church et al. in 2012 [4] and Goldman et al. in 2013 [5], there have been a flurry of experimental demonstrations (see [11], [13] for a survey). Amongst the various coding design considerations, in this work, we study the unsorted nature of the DNA storage system [8], [11].

A DNA storage system consists of three important components. The first is the DNA synthesis which produces the oligonucleotides, also called *strands*, that encode the data. The second part is a storage container with compartments which stores the DNA strands, however without order. Finally, to retrieve the data, the DNA is accessed using next-generation sequencing, which results in several noisy copies, called *reads*. The processes of synthesizing, storing, sequencing, and handling strands are all error prone. Due to this unordered nature of DNA-based storage systems, when the user retrieves the information, in addition to decoding the data, the user has to determine the identity of the data stored in each strand. A typical solution is to simply have a set of addresses and store this address information as a prefix to each DNA strand. As the addresses are also known to the user, the user can identify the information after the decoding process. As these addresses along with the stored data are prone to errors, this solution needs further refinements.

In [9], the strands (strand = address + data) are first clustered with respect to the edit distance. Then the authors determine a consensus output amongst the strands in each cluster and finally, decode these consensus outputs using a classic concatenation scheme. For this approach, the clustering step is computationally expensive. When there are $\mathcal{M}$ reads, the usual clustering method involves $\mathcal{M}^2$ pairwise comparisons to compute distances. This is costly when the data strands are long, and the problem is further exacerbated if the metric is the edit distance. Therefore, in [10], a distributed approximate clustering algorithm was proposed and the authors clustered 5 billion strands in 46 minutes on 24 processors.

In [3], the authors proposed and investigated an approach that avoids clustering, by studying a generalisation of the *bee identification problem*. Informally, the bee identification problem requires the receiver to identify M "bees" using a set of

M unordered noisy measurements [12]. Later, in [3], the authors generalized the setup to multi-draw channels where every bee (addresses) results in N noisy outputs (noisy addresses). The task then is to identify each of the M bees from the MN noisy outputs and it turns out that this task can be reduced to a minimum-cost network flow problem. In contrast to previous works, the approach in [3] utilizes only the address information, which is of significantly shorter length, and the method does not take into account the associated noisy data. Hence, this approach involves no data comparisons.

However, as evident, the clustering and bee identification based approaches do not completely take into account the nature of the DNA storage system. In particular, the clustering approaches do not utilise the uncorrupted set of addresses which can be accessed by the receiver and the bee identification approach uses solely the information stored in address and neglects the noisy data strands.

In this work, we overcome these shortcomings by utilising both the address and data information to identify the noisy reads. Specifically for the binary erasure channel, we first show that we can almost surely correctly identify all the reads under certain mild assumptions. Then we propose our permutation recovery procedure and demonstrate that on average the procedure uses only a fraction of $\mathcal{M}^2$ data comparisons (when there are $\mathcal{M}$ reads).

II. Related Work

A brief history of the Bee Identification Problem: Consider M bees flying in a beehive, each tagged with a unique barcode. We take a sanpshot of the bees and obtain an unordered set of noisy barcodes. The *bee-identification problem* - proposed and formally defined by Tandon et al. [] - requires one to uniquely identify each bee from the multiset of the noisy barcodes. The naive way is to separately look at each barcode and decode them *independently*. However, due to noise, certain bees may be assigned to the same barcode, and in this case, we fail to identify all the bees. In contrast, one can *jointly* look at all the barcodes and determine the best way to assign the barcodes so that the accuracy/likelihood of correct identification is maximized. Tandon et al. studied error exponents and showed that decoding the barcodes jointly results in a significantly smaller error exponent compared to decoding the barcodes independently. The authors further posited that joint decoding entails a computationally prohibitive exhausitve search. However, to the contrary, Kiah et al. [] demostrated that efficient joint decoding is achievable. Specifically, for the binary erasure channel (BEC) and binary symmetric channel (BSC), they reduced the bee-identification problem to the problem of finding a perfect matching and minimum-cost matching, respectively. Hence, applying the well-known Hopcraft-Karp algorithm [] and Hungarian method [], respectively, one can identify the bees in time polynomial in M. They further reduce the running times by using classical tools like peeling decoders and list-decoders, and also show that the identifier algorithms when used with Reed-Muller codes terminates in almost linear and quadratic time for BEC and BSC, respectively. Chrisnata et al. [], proposed and analysed a generalization of the Bee Identification problem to the multi-draw channels. For multi-draw deletion channel, similar to [], they reduce the identification problem to the problem of finding a minimum-cost matching. Then, applying the Edmonds-Karp or Tomizawa algorithm [], they show that the bee identification problem for multi-draw deletion channels can be solved in $\mathcal{O}(M^3)$ time.

Clustering Problem in DNA Data Storage: ...

III. Problem Formulation

Let N and M be positive integers. Let $[M]$ denote the set $\{1, 2, \ldots, M\}$. An N-permutation ψ over $[M]$ is an NM-tuple $(\psi(j))_{j\in[MN]}$ where every symbol in $[M]$ appears exactly N times, and we denote the set of all N-permutations over $[M]$ by $\mathbb{S}_N(M)$. Let the set of addresses denoted by $\mathcal{A}$ be a $(n, k, d_{\min})$-binary linear code such that $M = 2^k$, and assume that every codeword $\boldsymbol{x}_i \in \mathcal{A}$ is attached to a length-L data part $\boldsymbol{d}_i \in \{0,1\}^L$ to form a strand, which is the tuple, $(\boldsymbol{x}_i, \boldsymbol{d}_i)$. Let the multiset of data be denoted by $D = \{\{\boldsymbol{d}_i : i \in [M]\}\}$ and the set of strands by $R = \{(\boldsymbol{x}_i, \boldsymbol{d}_i) : i \in [M]\}$. Throughout this paper, we assume that D is drawn uniformly at random over $\{0,1\}^L$. Let $\mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ denote the multiset of channel outputs when $(\boldsymbol{x}, \boldsymbol{d})$ is transmitted N times through the channel $\mathcal{S}$. Assume that the entire set R is transmitted through the channel $\mathcal{S}$, hence an unordered multiset, $R' = \{\{(\boldsymbol{x}'_1, \boldsymbol{d}'_1), (\boldsymbol{x}'_2, \boldsymbol{d}'_2), \ldots, (\boldsymbol{x}'_{MN}, \boldsymbol{d}'_{MN})\}\}$, of MN noisy strands (reads) is obtained, where for every $j \in [MN]$, $(\boldsymbol{x}'_j, \boldsymbol{d}'_j) \in \mathcal{S}_N((\boldsymbol{x}_{\pi(j)}, \boldsymbol{d}_{\pi(j)}))$ for some N-permutation π over $[M]$, which will be referred to as the *true* N*-permutation*. Note that the receiver, apart from the set of reads R', has access to the set of addresses $\mathcal{A}$ but does not know the set of data D.

Let the weight enumerator polynomial of $\mathcal{A}$ be denoted by $W_{\mathcal{A}}(z) = \sum_{i=0}^{n} w_i z^i$, where w_i denotes the number of codewords in $\mathcal{A}$ of hamming weight i.

In this work, we first consider the following problem.

Problem 1. *Given $\mathcal{S}, \epsilon$ and $\mathcal{A}$, find the region $\mathcal{R} \in \mathbb{Z}_+^2$, such that for $(N, L) \in \mathcal{R}$, it is possible to identify the true permutation with probability at least $1 - \epsilon$ when the data D is drawn uniformly at random.*

For $(N, L) \in \mathcal{R}$, we can find the true permutation by making at least $(NM)^2$ data comparisons. This may be expensive when the data parts are long, i.e., when L is large. Therefore, our second objective is to reduce the number of data comparisons.

Problem 2. *Let $\kappa < 1$. Given $\mathcal{S}, \epsilon, \mathcal{A}$ and $(N, L) \in \mathcal{R}$, design an algorithm to identify the true permutation with probability at least $1 - \epsilon$ using $\kappa(NM)^2$ data comparisons. As before, D is drawn uniformly at random.*

In Section IV, we first propose an extension of the Peeling Matching Algorithm [6] to the multi-draw erasure channel. We then demonstrate that the peeling matching algorithm identifies the true permutation with a vanishing probability as n grows. In Section V, we address Problem 1 and identify the region $\mathcal{R}$ for which there exists only one valid permutation, viz. the true permutation. In Section VI, we describe our algorithm that identifies the true-permutation with probability at least $1 - \epsilon$ when $(N, L) \in \mathcal{R}$. In Section VII, we analyse the expected number of data comparisons performed by the algorithm.

IV. Bee-Identification over Multi-Draw Erasure Channel

In this section, we extend the Peeling Matching Algorithm (PMA), presented in [6] for $N = 1$ to a general value of N. The PMA-based approach uses solely the information stored in the addresses to identify the true-permutation, and does not take into consideration the noisy data that is also available to the receiver. The first step in the peeling matching algorithm is to construct a bipartite undirected graph $\mathcal{G} = (\mathcal{X} \cup \mathcal{Y}, E)$, where the left nodes are the addresses ($\mathcal{X} = \mathcal{A}$) and the right nodes are the noisy reads ($\mathcal{Y} = R'$). There exists an edge between $\boldsymbol{x} \in \mathcal{X}$ and $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$ if and only if $P(\boldsymbol{y}|\boldsymbol{x}) > 0$, where $P(\boldsymbol{y}|\boldsymbol{x})$ is the likelihood probability of observing $\boldsymbol{y}$ given that $\boldsymbol{x}$ was transmitted. For $\boldsymbol{x} \in \mathcal{X}$ and $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$, let $E_{\boldsymbol{x}}$ and $E_{(\boldsymbol{y},\boldsymbol{d}')}$ denote the multiset of neighbours of $\boldsymbol{x}$ and the set of neighbours of $(\boldsymbol{y}, \boldsymbol{d}')$ in $\mathcal{G}$, respectively, i.e., $E_{\boldsymbol{x}} = \{\{(\boldsymbol{y}, \boldsymbol{d}')|(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}')) \in E\}\}$, $E_{(\boldsymbol{y},\boldsymbol{d}')} = \{\boldsymbol{x}|(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}')) \in E\}$. Note that the degree of every left node is at least N as $\mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d})) \subseteq E_{\boldsymbol{x}}$. For left nodes and right nodes with degrees N and 1 respectively, the corresponding neighbor(s) can be matched with certainty. For ease of exposition, we refer to such nodes as *good* nodes.

Definition 1. *A node $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$ is said to be a **good right node** if $|E_{(\boldsymbol{y},\boldsymbol{d}')}| = 1$. A node $\boldsymbol{x} \in \mathcal{X}$ is said to be a **Type-A good left node** if $|E_{\boldsymbol{x}}| = N$ or a **Type-B good left node** if $|\{(\boldsymbol{y}, \boldsymbol{d}') \in E_x : |E_{(\boldsymbol{y},\boldsymbol{d}')}| = 1\}| = N$.*

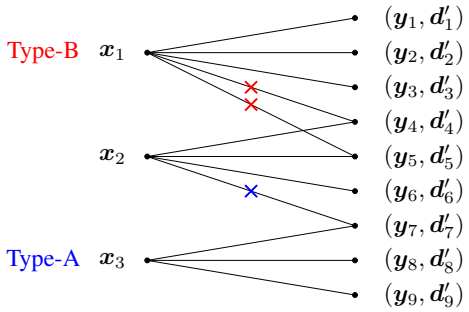

Fig. 1: Let $N = 3$. If $\boldsymbol{x}_1$ is peeled, then $\boldsymbol{x}_2$ becomes a Type-B good left node, and if $\boldsymbol{x}_3$ is peeled then $\boldsymbol{x}_2$ becomes a Type-A good left node.

Let $\mathsf{Y}_\mathsf{g}, \mathsf{X}_{\mathsf{g_A}}$ and $\mathsf{X}_{\mathsf{g_B}}$ denote the set of good right nodes, Type-A good left nodes and Type-B good left nodes, respectively. The peeling matching algorithm when executed over $\mathcal{G}$, finds good left nodes and identifies the corresponding N channel outputs until there are no good left nodes. Let $\mathcal{P}_\mathcal{G} = (\mathcal{X} \cup \mathcal{Y}, \mathcal{P}_E)$ denote the bipartite matching identified by the Peeling Matching Algorithm.

Algorithm 1 Peeling Matching Algorithm

procedure PEEL($\mathcal{P}_\mathcal{G}, \mathcal{G}, \boldsymbol{x}$)
  if $\boldsymbol{x} \in \mathsf{X}_{\mathsf{g_A}}$ **then**
    for $(\boldsymbol{y}, \boldsymbol{d}') \in E_{\boldsymbol{x}}$ **do**
      Remove $\{(\widetilde{\boldsymbol{x}}, (\boldsymbol{y}, \boldsymbol{d}')) : \widetilde{\boldsymbol{x}} \in E_{(\boldsymbol{y},\boldsymbol{d}')}\}$ from E
      Add $(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}'))$ to $\mathcal{P}_E$
      Remove $(\boldsymbol{y}, \boldsymbol{d}')$ from $\mathcal{Y}$
    Remove $\boldsymbol{x}$ from $\mathcal{X}$
  else if $\boldsymbol{x} \in \mathsf{X}_{\mathsf{g_B}}$ **then**
    Add $\{(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}')) : (\boldsymbol{y}, \boldsymbol{d}') \in E_x \cap \mathsf{Y}_\mathsf{g}\}$ to $\mathcal{P}_E$
    Remove $\{(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}')) : (\boldsymbol{y}, \boldsymbol{d}') \in E_x\}$ from E and remove $E_x \cap Y_g$ from $\mathcal{Y}$
    Remove $\boldsymbol{x}$ from $\mathcal{X}$
procedure PMA($\mathcal{P}_\mathcal{G}, \mathcal{G}$)
  for $\boldsymbol{x} \in \mathsf{X}_{\mathsf{g_A}} \cup \mathsf{X}_{\mathsf{g_B}}$ **do**

14: PEEL($\mathcal{P}_{\mathcal{G}}, \mathcal{G}, \boldsymbol{x}$)
15: **if** $|\mathcal{P}_E| = N2^n$ **then**
16: **return** $\mathcal{P}_{\mathcal{G}}$
17: **else**
18: **return** FAILURE

Note that as shown in Fig. 1, peeling Type-A and Type-B good left nodes might generate new Type-A and Type-B good left nodes, respectively. Thus, at any instant during the course of the algorithm, we assume $\mathsf{X}_{\mathsf{g_A}}, \mathsf{X}_{\mathsf{g_B}}$ to reflect the Type-A and Type-B good left nodes, respectively, at that instant.

Proposition 1. *[7] Algorithm 1 finds the true permutation if only if there are no cycles in $\mathcal{G}$.*

Let the multiset of right nodes that are in a cycle be denoted by $\mathsf{Y}_{\mathsf{cycle}}$. In the next lemma, we derive a lower bound on the probability of observing at least one cycle in $\mathcal{G}$.

Lemma 1. *For $\mathcal{A} = \{0,1\}^n$, the probability of observing at least one cycle in $\mathcal{G}$ is lower bounded by*

$$P(|\mathsf{Y}_{\mathsf{cycle}}| > 1) > 1 - \frac{\mathcal{U}_{\mathsf{cycle}}}{N2^n(1-\mathcal{U}_{\mathsf{cycle}})},$$

where $\mathcal{U}_{\mathsf{cycle}} \triangleq 2^{-N\left((1+p^2)^n-1\right)}$.

Proof. Let $\mathsf{Y}^*_{\mathsf{cycle}}$ denote the multiset of right nodes that are in a cycle of size four. Let $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ and $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in \mathcal{S}_N((\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{d}}))$ such that $d_H(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = r > 0$. It can be verified that $\{\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}'), \widetilde{\boldsymbol{x}}, (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}')\}$ forms a cycle with probability p^{2r}. Therefore, the probability that $(\boldsymbol{y}, \boldsymbol{d}')$ is not in a cycle is

$$\begin{aligned} P\left((\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{Y}^*_{\mathsf{cycle}}\right) &= \prod_{r=1}^{n} \left(1 - p^{2r}\right)^{N\binom{n}{r}} \\ &= 2^{\log\left(\prod_{r=1}^{n}\left(1-p^{2r}\right)^{N\binom{n}{r}}\right)} \\ &= 2^{N\sum_{r=1}^{n}\left(\binom{n}{r}\log\left(1-p^{2r}\right)\right)}. \end{aligned}$$

Using Jensen's inequality, it can be verified that

$$\begin{aligned} P\left((\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{Y}^*_{\mathsf{cycle}}\right) &\leq 2^{N(2^n-1)\log\left(\frac{\sum_{r=1}^{n}\binom{n}{r}(1-p^{2r})}{2^n-1}\right)} \\ &= 2^{N(2^n-1)\log\left(1-\frac{(1+p^2)^n-1}{2^n-1}\right)}. \end{aligned}$$

Using Taylor series expansion, for $|a| < 1$, $\log(1+a) \leq a$. Therefore, we have that

$$P\left((\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{Y}^*_{\mathsf{cycle}}\right) \leq \mathcal{U}_{\mathsf{cycle}} \triangleq 2^{-N\left((1+p^2)^n-1\right)}.$$

From linearity of expectation, we have that

$$\begin{aligned} \mathbb{E}\left[|\mathsf{Y}^*_{\mathsf{cycle}}|\right] &= \sum_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}} \mathbb{E}[\mathbb{I}_{\{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{Y}^*_{\mathsf{cycle}}\}}] \\ &= N2^n\left(1 - \prod_{r=1}^{n}\left(1-p^{2r}\right)^{N\binom{n}{r}}\right). \end{aligned}$$

Further, from the linearity of variances of indicator random variables, we have that

$$\begin{aligned} \mathsf{Var}\left[|\mathsf{Y}^*_{\mathsf{cycle}}|\right] &= \sum_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}} \mathsf{Var}[\mathbb{I}_{\{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{Y}^*_{\mathsf{cycle}}\}}] \\ &= N2^n\left(1 - \prod_{r=1}^{n}\left(1-p^{2r}\right)^{N\binom{n}{r}}\right)\left(\prod_{r=1}^{n}\left(1-p^{2r}\right)^{N\binom{n}{r}}\right). \end{aligned}$$

Next, from Chebyshev's inequality and the upper bound on $P\left((\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{Y}_{\mathsf{cycle}}\right)$, it can be verified that

$$P(|\mathsf{Y}^*_{\mathsf{cycle}}| < 1) < \frac{\mathsf{Var}\left[|\mathsf{Y}^*_{\mathsf{cycle}}|\right]}{(\mathbb{E}\left[|\mathsf{Y}^*_{\mathsf{cycle}}|\right])^2} < \frac{\mathcal{U}_{\mathsf{cycle}}}{N2^n(1-\mathcal{U}_{\mathsf{cycle}})}.$$

Therefore, the results follows. □

Hence, from Proposition 1 and Lemma 1, it is highly improbable (vanishingly low probability) to find the true permutation using only the addresses and its noisy measurements (i.e., $\boldsymbol{x}$ and $\boldsymbol{y}$'s). In the next section, we see that by making use of the data parts, we can find the true permutation under certain mild assumptions.

V. Uniqueness of the N-Permutation

The task of identifying the true permutation π, can be split into two steps. We can first identify the *partitioning* $\{\mathcal{S}_N((\boldsymbol{x}_i, \boldsymbol{d}_i)) : i \in [M]\}$ and then for each *partition* $(\mathcal{S}_N((\boldsymbol{x}_i, \boldsymbol{d}_i)))$ identify the *label*, viz. the channel input $(\boldsymbol{x}_i)$, where $i \in [M]$. Hence, given R' and $\mathcal{A}$, we are able to find the true permutation if and only if there exists only one valid partitioning and one valid labelling.

In this section, we study Problem 1 when $\mathcal{S} = \mathsf{BEC}(p)$. Specifically, in Lemmas 3 and 4, we determine the values L_{Th} and N_{Th}, respectively, such that for all $L \geq L_{\mathsf{Th}}$ and $N \geq N_{\mathsf{Th}}$, we are able to find the true permutation with high probability. The result is formally stated in Theorem 1.

Before formally defining partitioning and labelling, we introduce some notations. Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \{0,1\}^\ell$. For $i \in \{1,2\}$, let $\boldsymbol{b}_i \in \{0,1,*\}^\ell$ be the output of $\boldsymbol{x}_i$ through $\mathsf{BEC}(p)$. We denote the event of $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ agreeing at the non-erased positions by $\boldsymbol{b}_1 \cong \boldsymbol{b}_2$. For example, let $\boldsymbol{x}_1 = \texttt{00000}, \boldsymbol{x}_2 = \texttt{00011}$ and let $\boldsymbol{b}_1 = \texttt{0000}*, \boldsymbol{b}_2 = \texttt{000}*\texttt{1}$ then $\boldsymbol{b}_1 \tilde{=} \boldsymbol{b}_2$. Furhter, by abuse of notation, we would denote the event of all sequences in $A \subseteq \mathcal{S}_N(\boldsymbol{x}_1)$ agreeing at the non-erased positions with all sequences in $B \subseteq \mathcal{S}_N(\boldsymbol{x}_2)$ by $A \cong B$, where $\mathcal{S}_N(\boldsymbol{x}_i)$ denotes the multiset of channel outputs when $\boldsymbol{x}_i$ is transmitted N times through the channel $\mathcal{S}$, $i \in \{1,2\}$. A read $(\boldsymbol{y}, \boldsymbol{d}') \in R'$ is said to be **faulty** if there exists some other read $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in R' \backslash \{(\boldsymbol{y}, \boldsymbol{d}')\}$ with $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ and $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in \mathcal{S}_N((\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{d}}))$ such that $(\boldsymbol{y}, \boldsymbol{d}') \cong (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}')$. Let $\mathsf{R}_{\mathsf{faulty}}$ denote the multiset of such faulty reads. In the next lemma, we calculate the probability of a read being faulty.

Lemma 2. *For $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$,*

$$P(\mathbb{I}_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{R}_{\mathsf{faulty}}}) = 1 - \prod_{r=1}^{n} \left(1 - (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L\right)^{N w_r}.$$

Furthermore,

$$P(\mathbb{I}_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{R}_{\mathsf{faulty}}}) \leq N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(W_{\mathcal{A}}(2p - p^2) - 1\right).$$

Proof. For $(\boldsymbol{x}, \boldsymbol{d}), (\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{d}}) \in R$, let $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ and $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in \mathcal{S}_N((\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{d}}))$ with $d_H(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = r > 0$. For $\boldsymbol{y} \cong \widetilde{\boldsymbol{y}}$, the positions where $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ differ must be erased in at least one of them, which happens with probability $(1 - (1-p)^2)^r = (2p - p^2)^r$. For index $i \in [L]$, the probability that both $\boldsymbol{d}'$ and $\widetilde{\boldsymbol{d}}'$ are not erased and disagree on i is $\frac{1}{2}(1-p)^2$. Therefore, $P(\boldsymbol{d}' \cong \widetilde{\boldsymbol{d}}') = \left(1 - \frac{1}{2}(1-p)^2\right)^L$. Hence, $P\left((\boldsymbol{y}, \boldsymbol{d}') \cong (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}')\right) = (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L$. Thus,

$$\begin{aligned} P(\mathbb{I}_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{R}_{\mathsf{faulty}}}) &= 1 - \prod_{(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in R'/\mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}'))} \left(1 - P\left((\boldsymbol{y}, \boldsymbol{d}') \cong (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}')\right)\right) \\ &= 1 - \prod_{r=1}^{n} \left(1 - (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L\right)^{N w_r}. \end{aligned}$$

Using Weierstrass Inequality, we have that

$$\begin{aligned} P(\mathbb{I}_{(\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{R}_{\mathsf{faulty}}}) &\leq \sum_{r=1}^{n} N w_r (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L \\ &= N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(\sum_{r=1}^{n} w_r (2p - p^2)^r\right) \\ &= N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(W_{\mathcal{A}}(2p - p^2) - 1\right). \end{aligned}$$

□

Definition 2. *A **partitioning** $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ of R' is defined as the collection of disjoint submultisets of R', each of size N, such that for $i \in [M]$, for $(j,k) \in \binom{[N]}{2}$, $(\boldsymbol{y}_j, \boldsymbol{d}'_j) \cong (\boldsymbol{y}_k, \boldsymbol{d}'_k)$, where $(\boldsymbol{y}_j, \boldsymbol{d}'_j), (\boldsymbol{y}_k, \boldsymbol{d}'_k) \in P_i$.*

We will refer to $\mathcal{P}^* \triangleq \{\mathcal{S}_N((\boldsymbol{x}_i, \boldsymbol{d}_i)) : i \in [M]\}$ as the *true partitioning* of R'. Let $\mathbb{P}_{R'}$ denote the set of all possible partitionings of R'. Note that if $|\mathbb{P}_{R'}| = 1$ then $\mathbb{P}_{R'} = \{\mathcal{P}^*\}$. Let $\mathcal{G}' = (\mathcal{Y}, E')$, where $\mathcal{Y} = R'$. For $(\boldsymbol{y}, \boldsymbol{d}'), (\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}') \in \mathcal{Y}, ((\boldsymbol{y}, \boldsymbol{d}'), (\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}')) \in E'$ if $(\boldsymbol{y}, \boldsymbol{d}') \tilde{=} (\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}')$. Note that a partitioning $\mathcal{P} \in \mathbb{P}_{R'}$ corresponds to partitioning the graph $\mathcal{G}'$ into M cliques each of size N.

Proposition 2. *$|\mathbb{P}_{R'}| = 1$ if and only if there exists a unique partitioning of the graph $\mathcal{G}'$ into M cliques each of size N.*

In the next lemma, we derive a threshold on L such that for $L \geq L_{\mathsf{Th}}, \mathbb{P}_{R'} = \{\mathcal{P}^*\}$ with probability at least $1 - \epsilon_1$.

Lemma 3. *For $L \geq L_{\mathsf{Th}} \triangleq \log_{(1-\frac{1}{2}(1-p)^2)} \left(\frac{\sqrt[N]{\epsilon_1/2^k}}{N(W_{\mathcal{A}}(2p-p^2)-1)} \right)$, we have that $\mathbb{P}_{R'} = \{\mathcal{P}^*\}$ with probability at least $1 - \epsilon_1$.*

Proof. Note that for every $\boldsymbol{x} \in \mathcal{A}$, if there exists at least one $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ such that $(\boldsymbol{y}, \boldsymbol{d}')$ is not faulty, then the only valid partitioning is $\mathcal{P}^*$. Let $\mathsf{X}_{\mathsf{faulty}}$ denote the set of left nodes with $\mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d})) \subset \mathsf{R}_{\mathsf{faulty}}$. From Markov Inequality,

$$\begin{aligned} P(\boldsymbol{x} \in \mathsf{X}_{\mathsf{faulty}}) &= P(\mathbb{I}_{\mathcal{S}_N((\boldsymbol{x},\boldsymbol{d})) \subset \mathsf{R}_{\mathsf{faulty}}} \geq 1) \\ &\leq \mathbb{E}\left[\mathbb{I}_{\mathcal{S}_N((\boldsymbol{x},\boldsymbol{d})) \subset \mathsf{R}_{\mathsf{faulty}}}\right] = \left(P(\mathbb{I}_{(\boldsymbol{y},\boldsymbol{d}') \subset \mathsf{R}_{\mathsf{faulty}}} = 1)\right)^N. \end{aligned}$$

Therefore, from Lemma 2, $P(\boldsymbol{x} \in \mathsf{X}_{\mathsf{faulty}})$ is at most

$$\left(N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(W_{\mathcal{A}}(2p-p^2) - 1\right) \right)^N.$$

From linearity of expectation, $\mathbb{E}\left[|\mathsf{X}_{\mathsf{faulty}}|\right]$ is at most

$$2^k \left(N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(W_{\mathcal{A}}(2p-p^2) - 1\right) \right)^N.$$

From Markov inequality, $P(|\mathsf{X}_{\mathsf{faulty}}| \geq 1) \leq \mathbb{E}\left[|\mathsf{X}_{\mathsf{faulty}}|\right]$. Hence, $P(|\mathsf{X}_{\mathsf{faulty}}| < 1)$ is at least

$$= 1 - 2^k \left(N \left(1 - \frac{1}{2}(1-p)^2\right)^L \left(W_{\mathcal{A}}(2p-p^2) - 1\right) \right)^N.$$

Lastly, it can be verified that $P(|\mathsf{X}_{\mathsf{faulty}}| < 1) \geq 1 - \epsilon_1$ if $L > \log_{(1-\frac{1}{2}(1-p)^2)} \left(\frac{\sqrt[N]{\epsilon_1/2^k}}{N(W_{\mathcal{A}}(2p-p^2)-1)} \right)$. □

The next corollary follows from the fact that $W_{\{0,1\}^n}(z) = \sum_{i=0}^{n} \binom{n}{i} z^i = (1+z)^n$.

Corollary 1. *For $\mathcal{A} = \{0,1\}^n, L_{\mathsf{Th}} = \log_{(1-\frac{1}{2}(1-p)^2)} \left(\frac{\sqrt[N]{\epsilon_1/2^n}}{N((1+2p-p^2)^n-1)} \right)$.*

Corollary 2. *For given $n, \epsilon_1, p, \mathcal{A} = \{0,1\}^n$, L_{Th} is minimum at $N = \ln\left(\frac{2^n}{\epsilon_1}\right)$.*

Proof. From Corollary 1, observe that

$$\arg\min_{N \in \mathbb{Z}_+} L_{\mathsf{Th}} = \arg\min_{N \in \mathbb{Z}_+} \log_2 \left(\frac{N((1+2p-p^2)^n - 1)}{\sqrt[N]{\epsilon_1/2^n}} \right),$$

which can be readily verified to be minimum at $N = \ln\left(\frac{2^n}{\epsilon_1}\right)$. □

Corollary 3. *For given N, ϵ_1, p, $\mathcal{A} = \{0,1\}^n$, $L_{\mathsf{Th}} = \mathcal{O}(n)$. Specifically,*

$$L_{\mathsf{Th}} < \left[\frac{\log_2(N(1+2p-p^2)) + \frac{1-\log_2(\epsilon_1)}{N}}{1 - \log_2(1+2p-p^2)} \right] n.$$

Proof. From Corollary 1, $L_{\mathsf{Th}} = \frac{\log_2\left(\frac{N((1+2p-p^2)^n-1)}{\sqrt[N]{\epsilon_1/2^n}}\right)}{(1-\log_2(1+2p-p^2))}$. Therefore, we have that

$$\frac{L_{\mathsf{Th}}}{n} < \frac{\log_2\left(\frac{N(1+2p-p^2)^n}{\sqrt[N]{\epsilon_1/2^n}}\right)}{n(1-\log_2(1+2p-p^2))}$$

$$
\begin{aligned}
&= \frac{\log_2(N) + n\log\left(1+2p-p^2\right) - \left(\frac{\log(\epsilon_1)-n}{N}\right)}{n(1-\log_2(1+2p-p^2))} \\
&= \frac{\log_2(1+2p-p^2) + \frac{1}{N}}{(1-\log_2(1+2p-p^2)} + \frac{\log_2(N) - \frac{\log_2(\epsilon_1)}{N}}{n(1-\log_2(1+2p-p^2)} \\
&< \frac{\log_2(N(1+2p-p^2)) + \frac{1-\log_2(\epsilon_1)}{N}}{1-\log_2(1+2p-p^2)}.
\end{aligned}
$$

□

Definition 3. *Given a partitioning $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$, we define a **labelling**, denoted by $\mathcal{L}$, as a length-M vector of distinct addresses from $\mathcal{A}$ such that $\mathcal{L}[i] \in \{\boldsymbol{x} : \forall(\boldsymbol{y}, \boldsymbol{d}') \in P_i, P(\boldsymbol{x}|\boldsymbol{y}) > 0\}$, where $\mathcal{L}[i]$ denotes the i-th element of $\mathcal{L}$, and $i \in [M]$.*

We denote the set of all possible labellings for a given partitioning $\mathcal{P}$ by $\mathbb{L}_{\mathcal{P},R'}$. Given the true partitioning $\mathcal{P}^*$, we define the *true labelling*, denoted by $\mathcal{L}^*$, as the labelling in which for each partition $\mathcal{S}_N((\boldsymbol{x}_i, \boldsymbol{d}_i))$, the assigned label is $\boldsymbol{x}_i$, where $i \in [M]$. Note that if $\mathcal{P} \neq \mathcal{P}^*$ then $\mathcal{L}^* \notin \mathbb{L}_{\mathcal{P},R'}$. Further, if $|\mathbb{L}_{\mathcal{P}^*,R'}| = 1$ then $\mathbb{L}_{\mathcal{P}^*,R'} = \{\mathcal{L}^*\}$. Let $\mathcal{G}'' = (\mathcal{X}, E'')$, where $\mathcal{X} = \mathcal{A}$. There is a directed edge $\boldsymbol{x} \to \widetilde{\boldsymbol{x}}$ if all of the N channel outputs of $\boldsymbol{x}$ are erased at the positions where $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ differ, i.e., $\{\widetilde{\boldsymbol{x}}\} \in \{\bigcap_{(\boldsymbol{y},\boldsymbol{d}')\in\mathcal{S}_N((\boldsymbol{x},\boldsymbol{d}))} E_{(\boldsymbol{y},\boldsymbol{d}')}\}$.

Proposition 3. *$|\mathbb{L}_{\mathcal{P}^*,R'}| = 1$ if and only if there are no directed cycles in $\mathcal{G}''$.*

In the next lemma, we derive a threshold on N such that for $N \geq N_{\mathsf{Th}}, \mathbb{L}_{\mathcal{P}^*,R'} = \{\mathcal{L}^*\}$ with probability at least $1-\epsilon_2$.

Lemma 4. *For $N \geq N_{\mathsf{Th}} \triangleq \log_p\left(W_{\mathcal{A}}^{-1}\left(1+\frac{\epsilon_2}{2^k}\right)\right)$, we have that $\mathbb{L}_{\mathcal{P}^*,R'} = \{\mathcal{L}^*\}$ with probability at least $1-\epsilon_2$.*

Proof. Let $\mathsf{X}_{\mathsf{faulty}}$ denote the set of nodes in $\mathcal{G}''$ that have at least one outgoing edge. For $\boldsymbol{x}, \widetilde{\boldsymbol{x}} \in \mathcal{X}$, let $r = d_H(\boldsymbol{x}, \widetilde{\boldsymbol{x}})$. Note that $\boldsymbol{x} \to \widetilde{\boldsymbol{x}}$ happens with probability p^{rN}. Therefore, the probability that $\boldsymbol{x}$ has no outgoing edges is $\prod_{r=1}^{n}(1-p^{rN})^{w_r}$. Hence, from linearity of expectation,

$$
\begin{aligned}
\mathbb{E}\left[|\mathsf{X}_{\mathsf{faulty}}|\right] &= \sum_{\boldsymbol{x}\in\mathcal{X}} \mathbb{E}[\mathbb{I}_{\{\boldsymbol{x}\in\mathsf{X}_{\mathsf{faulty}}\}}] \\
&= 2^k\left(1 - \prod_{r=1}^{n}\left(1-p^{rN}\right)^{w_r}\right).
\end{aligned}
$$

From Weierstrass inequality, we have that $\prod_{r=1}^{n}\left(1-p^{rN}\right)^{w_r} \geq 1 - \sum_{r=1}^{n} w_r p^{rN} = 2 - W_{\mathcal{A}}(p^N)$. Therefore,

$$
\mathbb{E}\left[|\mathsf{X}_{\mathsf{faulty}}|\right] \leq 2^k\left(W_{\mathcal{A}}(p^N) - 1\right).
$$

From Markov inequality, $P(|\mathsf{X}_{\mathsf{faulty}}| \geq 1) \leq \mathbb{E}\left[|\mathsf{X}_{\mathsf{faulty}}|\right]$. Hence,

$$
P(|\mathsf{X}_{\mathsf{faulty}}| < 1) \geq 1 - 2^k\left(W_{\mathcal{A}}(p^N) - 1\right).
$$

Lastly, it can be verified that $P(|\mathsf{X}_{\mathsf{faulty}}| < 1) \geq 1-\epsilon_2$ if $N \geq \log_p\left(W_{\mathcal{A}}^{-1}\left(1+\frac{\epsilon_2}{2^k}\right)\right)$. □

The next corollary follows from the fact that $W_{\{0,1\}^n}(z) = \sum_{i=0}^{n}\binom{n}{i}z^i = (1+z)^n$.

Corollary 4. *For $\mathcal{A} = \{0,1\}^n, N_{\mathsf{Th}} \triangleq \log_p\left(\sqrt[n]{1+\frac{\epsilon_2}{2^n}} - 1\right)$.*

Corollary 5. *For given p, ϵ_2, $\mathcal{A} = \{0,1\}^n$, $N_{\mathsf{Th}} = \Theta(n)$. Specifically,*

$$
\frac{n + \log_2\left(\frac{n}{\epsilon_2\ln(2)}\right)}{\log_2\left(\frac{1}{p}\right)} > N_{\mathsf{Th}} > \frac{n + \log_2\left(\frac{n}{\epsilon_2}\right)}{\log_2\left(\frac{1}{p}\right)}.
$$

Proof. From Lemma 4, $N_{\mathsf{Th}} = \frac{\log_2\left(\sqrt[n]{1+\frac{\epsilon_2}{2^n}}-1\right)}{\log_2(p)}$. For $0 < a < 1, 0 < b < 1$, it can be verifed that $1 + ab\ln(2) < 2^{ab} < (1+a)^b < 1 + ab$. Therefore, we have that

$$
\begin{aligned}
&\frac{\log_2\left(\frac{\epsilon_2\ln(2)}{n2^n}\right)}{\log_2(p)} > N_{\mathsf{Th}} > \frac{\log_2\left(\frac{\epsilon_2}{n2^n}\right)}{\log_2(p)} \\
\iff &\frac{n + \log_2\left(\frac{n}{\epsilon_2\ln(2)}\right)}{\log_2\left(\frac{1}{p}\right)} > N_{\mathsf{Th}} > \frac{n + \log_2\left(\frac{n}{\epsilon_2}\right)}{\log_2\left(\frac{1}{p}\right)}.
\end{aligned}
$$

□

Thus, we define the region $\mathcal{R}$ as $\mathcal{R} \triangleq \{(\beta, N) : \beta \geq \beta_{\mathsf{Th}}, N \geq N_{\mathsf{Th}}\}$. In the next theorem, we give a sufficient condition for the existence of a unique N-permutation.

Theorem 1. *For $(L, N) \in \mathcal{R}$, it is possible to identify the true permutation with probability at least $1-\epsilon$, when $\epsilon_1, \epsilon_2 < \frac{\epsilon}{2}$.*

Proof. From Lemma 3 and 4, it follows that for $L > L_{\mathsf{Th}}$ and $N > N_{\mathsf{Th}}$, $\mathbb{P}_{R'} = \{\mathcal{P}^*\}$ with probability $(1-\epsilon_1)$ and $\mathbb{L}_{\mathcal{P}^*,R'} = \{\mathcal{L}^*\}$ with probability $(1-\epsilon_2)$, respectively. Hence, for $L > L_{\mathsf{Th}}$ and $N > N_{\mathsf{Th}}$, there exists only one valid permutation with probability $(1-\frac{\epsilon}{2})^2 > (1-\epsilon)$. □

From Corollaries 3 and 5, for $\mathcal{A} = \{0,1\}^n$, we observe that $L_{\mathsf{Th}} < \lambda^* n$ and $N_{\mathsf{Th}} < \nu^* n$ for some constants λ^* and ν^*. This means that we only require data parts to be of length $L = \lambda^* n$ and the number of reads to be $N = \nu^* n$ so that correct identification occurs with high probability. In the next section, we design an algorithm to find the true permutation with a small number of data comparisons.

VI. Achievabilty: Permutation Recovery Algorithm

As the receiver has access to the set of addresses, we design an algorithm that reduces the number of data comparisons by comparing a pair of reads if and only if they agree at the positions that are not erased in the address part. Hence, similar to the peeling matching algorithm, we first build the bipartite graph $\mathcal{G} = (\mathcal{X} \cup \mathcal{Y}, E)$ as described in Section IV. Let $\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}$ denote the two-hop neighborhood of $(\boldsymbol{y}, \boldsymbol{d}')$ in $\mathcal{G}$.

Proposition 4. *For $(\boldsymbol{y}, \boldsymbol{d}'), (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}}') \in \mathcal{Y}$, $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{N}_{(\widetilde{\boldsymbol{y}},\widetilde{\boldsymbol{d}}')}$ if and only if $\boldsymbol{y} \cong \widetilde{\boldsymbol{y}}$.*

In the next lemma, we calculate the expected value of $|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|$.

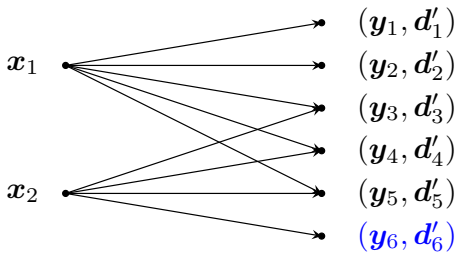

Fig. 2: Let $N = 3$. For $(\boldsymbol{y}_6, \boldsymbol{d}'_6)$, we can potentially identify the remaining 2 copies by performing only $|\mathcal{N}_{(\boldsymbol{y}_6,\boldsymbol{d}'_6)}| = 3$ data comparisons.

Lemma 5. *Let $(\boldsymbol{x}, \boldsymbol{d}) \in R, (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}((\boldsymbol{x}, \boldsymbol{d}))$. Then, we have that*

$$E[\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}] = NW_{\mathcal{A}}(2p - p^2) - 1$$

Proof. Let $(\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) \in R$, such that $d_H(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = r$. Let $(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}') \in \mathcal{S}((\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}))$. Then for $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{N}_{(\tilde{\boldsymbol{y}},\tilde{\boldsymbol{d}}')}$, there must be at least one erasure in $\boldsymbol{x}$ or $\widetilde{\boldsymbol{x}}$ at all the positions where they differ, which happens with probability $(1-(1-p)^2)^r = (2p-p^2)^r$. Hence, $E[\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}] = \sum_{r=1}^{n} W_r (2p-p^2)^r N + (N-1) = NW_{\mathcal{A}}(2p-p^2) - 1$. □

Lemma 6. *For $\mathcal{A} = \{0,1\}^n$ and $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$,*

$$\mathbb{E}[|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| \mid (\boldsymbol{y}, \boldsymbol{d}')] = N2^r(1+p)^{n-r} - 1,$$

where r denotes the number of erasures in $\boldsymbol{y}$. Further, $\mathbb{E}[|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|] = N(1+2p-p^2)^n - 1$.

Proof. Let the number of erasures in $\boldsymbol{y}$ be r. Let $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$.

1) For $(\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) \in \mathcal{X}$ that does not differ from $(\boldsymbol{x}, \boldsymbol{d})$ at the non-erased positions in $(\boldsymbol{y}, \boldsymbol{d}')$, $\mathcal{S}_N((\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}))) \subset \mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}$. Note that there are $2^r - 1$ such $(\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}})$.
2) For $(\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) \in \mathcal{X}$ that differ from $(\boldsymbol{x}, \boldsymbol{d})$ at i out of the $n-r$ non-erased positions in $(\boldsymbol{y}, \boldsymbol{d}')$, we have that for $(\widetilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}') \in \mathcal{S}_N((\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}))$, $P((\widetilde{\boldsymbol{y}}, \tilde{\boldsymbol{d}}') \in \mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}) = p^i$. Note that there are $2^r\binom{n-r}{i}$ such $(\widetilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}})$.

Hence, from linearity of expectations,

$$\mathbb{E}[|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| \mid (\boldsymbol{y}, \boldsymbol{d}')] = (N(2^r - 1) + N - 1) + N2^r \sum_{i=1}^{n-r} \binom{n-r}{i} p^i$$

$$= N2^r(1+p)^{n-r} - 1.$$

□

The permutation recovery algorithm as described below, iteratively selects the right node $(\boldsymbol{y}, \boldsymbol{d}')$ with the smallest two-hop neighborhood in $\mathcal{Y}$ and then as shown in Fig. 2, performs $|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|$ data comparisons to identify the remaining $N-1$ copies. Note that the algorithm finds the remaining $N-1$ copies if and only if $(\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{R}_{\mathsf{faulty}}$. Let $\mathcal{P}_{\mathcal{G}} = (\mathcal{X} \cup \mathcal{Y}, \mathcal{P}_E)$ denote the bipartite matching identified by the permutation recovery algorithm.

Algorithm 2 Permutation Recovery Algorithm

procedure PRUNE($\mathcal{G}, (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'})$)
  $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'}) \longrightarrow \mathsf{Pruned}$, $\mathcal{T} = \{\}$
  for $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{N}_{(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'})}$ **do**
    if $(\boldsymbol{y}, \boldsymbol{d}') \cong (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'})$ **then**
      $(\boldsymbol{y}, \boldsymbol{d}') \longrightarrow \mathcal{T}$
  if $|\mathcal{T}| = N - 1$ **then**
    Let $\mathcal{X}^* = \bigcap_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{T}} E_{(\boldsymbol{y},\boldsymbol{d}')}$
    for $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{T}$ **do**
      Remove $\{(\boldsymbol{x}, (\boldsymbol{y}, \boldsymbol{d}')) : \boldsymbol{x} \notin \mathcal{X}^*\}$ from E
      $(\boldsymbol{y}, \boldsymbol{d}') \longrightarrow \mathsf{Pruned}$

procedure PRUNING ALGORITHM($\mathcal{P}_{\mathcal{G}}, \mathcal{G}$)
  $\mathsf{Pruned} = \{\}$
  while $|\mathsf{Pruned}| < N2^n$ **do**
    $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'}) = \arg\min\{|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| : (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}\}$
    PRUNE $(\mathcal{G}, (\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'}))$
  return $\mathsf{PMA}(\mathcal{P}_{\mathcal{G}}, \mathcal{G})$

Proposition 5. *For $(L, N) \in \mathcal{R}$, Algorithm 2 finds the true permutation with probability at least $1 - \epsilon$, when $\epsilon_1, \epsilon_2 < \frac{\epsilon}{2}$.*

Proof. For $L > L_{\mathsf{Th}}$, every left node has at least one non-faulty channel output with probability at least $(1 - \epsilon_1)$. Thus, the permutation recovery algorithm identifies the true partitioning with probability at least $(1 - \epsilon_1)$. For $N > N_{\mathsf{Th}}$, there exists only one valid labelling, viz. the true labelling with probability at least $(1 - \epsilon_2)$. Thus, the permutation recovery algorithm identifies the true permutation with probability at least $(1 - \frac{\epsilon}{2})^2 > (1 - \epsilon)$. □

VII. Analysis of permutation recovery Algorithm for BEC

In this section, we analyse the expected number of data comparisons performed by Algorithm 2 for three subregions of $\mathcal{R}$. In the next lemma, we give an upper bound on the expected number of data comparisons performed by Algorithm 2 when $(L, N) \in \mathcal{R}$.

Lemma 7. *The expected number of data comparisons performed by Algorithm 2 when $(L, N) \in \mathcal{R}$ is at most*

$$\mathcal{U}_0 \triangleq N^2 2^n \left(1 + 2p - p^2\right)^n.$$

Proof. Note that the number of data comparisons performed by Algorithm 2 is at most $\sum_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{Y}} |\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|$. From linearity of expectations, $\mathbb{E}\left[\sum_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{Y}} |\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|\right] = \sum_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{Y}} \mathbb{E}\left[|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|\right]$. From Lemma 6, the result follows. □

Let β_0 be a threshold on β such that for $\beta \geq \beta_0$, $P(|\mathsf{R}_{\mathsf{faulty}}| > 1) < \epsilon_1$. In the next lemma, we derive this threshold β_0.

Lemma 8. *For $\beta \geq \beta_0 \triangleq \frac{\log_2\left(\frac{\epsilon_1}{2^n N^2((1+2p-p^2)^n - 1)}\right)}{n \log_2\left(1 - \frac{1}{2}(1-p)^2\right)}$, $P(|\mathsf{R}_{\mathsf{faulty}}| > 1) < \epsilon_1$.*

Proof. If $\mathbb{E}\left[|\mathsf{R}_{\mathsf{faulty}}|\right] < \epsilon_1$ then from Markov inequality the result follows. From Lemma 2, we have that

$$\mathbb{E}\left[|\mathsf{R}_{\mathsf{faulty}}|\right] = N2^n \left(1 - \prod_{r=1}^{n} \left(1 - (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L\right)^{N\binom{n}{r}}\right).$$

Hence, to show that $\mathbb{E}\left[|\mathsf{R}_{\mathsf{faulty}}|\right] < \epsilon_1$ it is sufficient to show that

$$1 - \frac{\epsilon_1}{2^n N} < \prod_{r=1}^{n} \left(1 - (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L\right)^{N\binom{n}{r}}.$$

Using Weierstrass inequality we have that

$$\begin{aligned}
&\prod_{r=1}^{n} \left(1 - (2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L\right)^{N\binom{n}{r}} \\
&\geq 1 - \sum_{r=1}^{n} N\binom{n}{r}(2p - p^2)^r \left(1 - \frac{1}{2}(1-p)^2\right)^L \\
&= 1 - N\left(\left(1 - \frac{1}{2}(1-p)^2\right)^L \left((1 + 2p - p^2)^n - 1\right)\right),
\end{aligned}$$

and thus it is enough to show that

$$N\left(\left(1 - \frac{1}{2}(1-p)^2\right)^L \left((1 + 2p - p^2)^n - 1\right)\right) < \frac{\epsilon_1}{2^n N}.$$

Lastly, it can be verified that $\mathbb{E}\left[|\mathsf{R}_{\mathsf{faulty}}|\right] < \epsilon_1$ if $\log_{(1-\frac{1}{2}(1-p)^2)}\left(\frac{\epsilon_1}{2^n N^2((1+2p-p^2)^n-1)}\right) < L = \beta n$.

□

We define $\mathcal{R}' \subseteq \mathcal{R}$ as $\mathcal{R}' \triangleq \{(\beta, N) : \beta \geq \beta_0, N \geq N_{\mathsf{Th}}\}$. To analyse the expected number of data comparisons performed by Algorithm 2 when $(\beta, N) \in \mathcal{R}'$, we define the notion of order of a left node.

Definition 4. *A node $\boldsymbol{x} \in \mathcal{X}$ has order s if $\min\{|E_{(\boldsymbol{y},\boldsymbol{d}')}| : (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))\} = s$.*

For $s \in [2^n]$, let X_s denote the set of left nodes with order s. In the next lemma, we calculate the probability that a left node has order s.

Lemma 9. *For $\boldsymbol{x} \in \mathcal{X}$, before the initiation of Algorithm 2, $P(\boldsymbol{x} \in \mathsf{X}_s)$ is*

$$\begin{cases} \left(\sum_{i=\ell}^{n}\binom{n}{i}p^i(1-p)^{n-i}\right)^N - \left(\sum_{i=\ell+1}^{n}\binom{n}{i}p^i(1-p)^{n-i}\right)^N & s \in \{2^\ell, \ell \in [0:n]\} \\ 0 & \textit{otherwise}. \end{cases}$$

Proof. Note that before any edges are removed from $\mathcal{G}$, the degree of a right node can take values only from the set $\{2^\ell, \ell \in [0:n]\}$. Therefore, for $s \notin \{2^\ell, \ell \in [0:n]\}, P(x \in \mathsf{X}_s) = 0$. For $s \in \{2^\ell, \ell \in [0:n]\}$, it can be verified that $P(\boldsymbol{x} \in \mathsf{X}_s) = P(\bigcup_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x},\boldsymbol{d}))} |E_{(\boldsymbol{y},\boldsymbol{d}')}| \geq s) - P(\bigcup_{(\boldsymbol{y},\boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x},\boldsymbol{d}))} |E_{(\boldsymbol{y},\boldsymbol{d}')}| \geq s+1)$. Therefore, the result follows. □

In the next lemma, we derive an upper bound on the expected number of data comparisons performed by Algorithm 2 when $(\beta, N) \in \mathcal{R}'$.

Lemma 10. *For $(\beta, N) \in \mathcal{R}'$, the expected number of data comparisons performed by Algorithm 2 is at most*

$$\mathcal{U}_1 \triangleq \sum_{r=0}^{n} \mathbb{E}[|\mathsf{X}_{2^r}|] N 2^r ((1+p)^{n-r}).$$

Proof. Note that for $\beta > \beta_0$, there are no faulty right nodes with probability at least $1-\epsilon_1$. Hence, the expected number of data comparisons performed by Algorithm 2 to identify the N channel outputs of $\boldsymbol{x}$ is at most $\mathbb{E}\left[\min\{|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| : (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))\}\right]$. For $\boldsymbol{x} \in \mathcal{X}$, by law of total expectation

$$\begin{aligned}
&\mathbb{E}\left[\min\{|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| : (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))\}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\min\{|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| : (\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))\}\right] \mid x \in \mathsf{X}_s\right]
\end{aligned}$$

From Lemma 5, $\mathbb{E}[|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}| \mid (\boldsymbol{y}, \boldsymbol{d}')] = N 2^r (1+p)^{n-r} - 1$,

$$= \sum_{r=0}^{n} P(x \in \mathsf{X}_{2^r})(N 2^r (1+p)^{n-r} - 1).$$

From linearity of expectation, the result follows. □

We now define the notion of confusability for left nodes.

Definition 5. *Let $\boldsymbol{x}, \widetilde{\boldsymbol{x}} \in \mathcal{X}$ then $\boldsymbol{x}$ is **confusable** with $\widetilde{\boldsymbol{x}}$, denoted by $\boldsymbol{x} \to \widetilde{\boldsymbol{x}}$, if there exists at least one $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'}) \in \mathcal{S}_N((\widetilde{\boldsymbol{x}}, \widetilde{\boldsymbol{d}'}))$ such that $E_{(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{d}'})} = \{\boldsymbol{x}, \widetilde{\boldsymbol{x}}\}$.*

Next, we build a graph of left nodes, $T = (\mathcal{X}, \mathsf{E}_{\mathsf{conf}})$. Let $\boldsymbol{x}, \widetilde{\boldsymbol{x}}, \boldsymbol{x}' \in \mathcal{X}$. Note that before the initiation of Algorithm 2, for $\boldsymbol{x} \to \widetilde{\boldsymbol{x}}$, it must be that $d_H(\boldsymbol{x}, \widetilde{\boldsymbol{x}}) = 1$. For ease of analysis, we do not consider the confusable edges that would be generated over the course of Algorithm 2. Thus, there is an edge $\boldsymbol{x} \to \widetilde{\boldsymbol{x}} \in \mathsf{E}_{\mathsf{conf}}$ if and only if $\boldsymbol{x}$ is confusable with $\widetilde{\boldsymbol{x}}$ before the initiation of the algorithm. In the next lemma, we derive the probability that $\boldsymbol{x}$ has edges to all nodes in $S \subseteq \{\boldsymbol{x}' : d_H(\boldsymbol{x}, \boldsymbol{x}') = 1\}$.

Lemma 11. *Let $\boldsymbol{x} \in \mathcal{X}$ and let $S \subseteq \{\boldsymbol{x}' : d_H(\boldsymbol{x}, \boldsymbol{x}') = 1\}$. Then,*

$$P\left(\bigcup_{j=1}^{|S|} (\boldsymbol{x} \to \boldsymbol{x}_i)\right) = \prod_{j=1}^{|S|} \left(1 - \left(1 - p(1-p)^{n-1}\right)^{N-j+1}\right),$$

where $\boldsymbol{x}_i \in S$ for $i \in [|S|]$.

Proof. From Definition 5, $\boldsymbol{x} \to \widetilde{\boldsymbol{x}}$ there must exist a $\boldsymbol{y} \in \mathcal{S}_N((\boldsymbol{x}, \boldsymbol{d}))$ such that $E_{(\boldsymbol{y}, \boldsymbol{d}')} = \{\boldsymbol{x}, \widetilde{\boldsymbol{x}}\}$, which happens if and only if $\boldsymbol{y}$ is erased only at the position where $\boldsymbol{x}$ and $\widetilde{\boldsymbol{x}}$ differ, which happens with the probability $p(1-p)^{n-1}$. Therefore, $P\left(\bigcup_{j=1}^{|S|} (\boldsymbol{x} \to \boldsymbol{x}_i)\right) = \prod_{j=1}^{|S|} \left(1 - \left(1 - p(1-p)^{n-1}\right)^{N-j+1}\right)$. □

Next, let $G_A = (\mathcal{X}, \mathcal{E})$ be a directed n-cube [2]. A vertex $\boldsymbol{x} \in \mathcal{X}$ has outgoing edges to the vertices $\{\boldsymbol{x}' : d_H(\boldsymbol{x}, \boldsymbol{x}') = 1, \boldsymbol{x}' \in \mathcal{X}\}$. Let $G_A(p_e)$ denote a random sub-graph of G_A where every edge in $\mathcal{E}$ is selected with probability p_e.

Proposition 6. *The probability of the appearance of a connected component is greater in T than in $G_A(p_T)$, where $p_T \triangleq \left(1 - \left(1 - p(1-p)^{n-1}\right)^{N-n+1}\right)$.*

Proof. Since the probability that there is an edge $\boldsymbol{x} \to \boldsymbol{x}'$ is independent of the existence of the edge $\widetilde{\boldsymbol{x}} \to \boldsymbol{x}'$, the proposition follows form Lemma 11. □

Lemma 12. *For $N > N_0 \triangleq n - \frac{1}{\log(1-p(1-p)^{n-1})} = \mathcal{O}_p\left(\frac{1}{p(1-p)^{n-1}}\right)$, T is almost surely connected.*

Proof. From [2], we know that $G_A(p_e)$ is almost surely connected if $p_e > \frac{1}{2}$. It can be verified that for $N > n - \frac{1}{\log(1-p(1-p)^{n-1})}$, $p_e = p_T > \frac{1}{2}$. Then from Proposition 6, the result follows. □

We define region $\mathcal{R}'' \subseteq \mathcal{R}'$ as $\mathcal{R}'' \triangleq \{(\beta, N) : \beta \geq \beta_0, N \geq N_0\}$.

Lemma 13. *The expected number of data comparisons performed by Algorithm 2 when $(\beta, N) \in \mathcal{R}''$ is at most*

$$\mathcal{U}_2 \triangleq N 2^n (1+p)^n.$$

Proof. Since, the graph T is connected, Algorithm 2 will always prune an order 1 node. Hence, the result follows. □

Hence, from Lemmas 7, 10 and 13, the expected number of data comparisons performed by Algorithm 2 is only a $\kappa_{\beta,N}$-fraction of data comparisons required by clustering based approaches agnostic to the nature of DNA storage systems, $\left(\frac{1+2p-p^2}{2}\right)^n \geq \kappa_{\beta,N} \geq \left(\frac{1+p}{2}\right)^n$.

REFERENCES

[1] S. Singhvi, A. Boruchovsky, H. M. Kiah and E. Yaakobi, "Data-Driven Bee Identification for DNA Strands", *arXiv preprint*, arXiv:2305.04597, 2023.

[2] B. Bollobás, C. Gotsman, and E. Shamir, "Connectivity and dynamics for random subgraphs of the directed cube," *ISRAEL JOURNAL OF MATHEMATICS*, vol 83, pp 321–328, 1993.

[3] J. Chrisnata, H. M. Kiah, A. Vardy, and E. Yaakobi, "Bee identification problem for DNA strands," *IEEE International Symposium on Information Theory (ISIT)*, pp. 969–974, June, 2022.

[4] G. M. Church, Y. Gao, and S. Kosuri. "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.

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

[6] H. M. Kiah, A. Vardy, and H. Yao, "Efficient bee identification," *IEEE International Symposium on Information Theory (ISIT)*, pp. 1943–1948, July, 2021.

[7] H. M. Kiah, A. Vardy, and H. Yao, "Efficient algorithms for the bee-identification problem," *arXiv preprint* arXiv:2212.09952, 2022.

[8] A. Lenz, P. H. Siegel, A. Wachter-Zeh and E. Yaakobi, "Coding over sets for DNA storage," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2331–2351, April 2020.

[9] L. Organick, S. Ang, Y.J. Chen, R. Lopez, S.Yekhanin, K. Makarychev, M. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. Takahashi, S. Newman, H. Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss, "Random access in largescale DNA data storage," *Nature Biotechnology*, vol. 36, no. 3, pp 242–248, 2018.

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

[11] I. Shomorony, and R. Heckel, "Information-theoretic foundations of DNA data storage," Foundations and Trends®in Communications and Information Theory, 19(1), 1–106, 2022

[12] A. Tandon , V.Y.F. Tan, and L.R. Varshney, "The bee-identification problem: Bounds on the error exponent," *IEEE Transactions on Communications*, vol. 67, issue no.11, pp. 7405–7416, November, 2019.

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