Data-Driven Bee Identification for DNA Strands

Shubhransh Singhvi[*], **Avital Boruchovsky**[†], **Han Mao Kiah**[‡] and **Eitan Yaakobi**[†]
[*]Signal Processing & Communications Research Center, International Institute of Information Technology, 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*—We study a data-driven approach to the bee identification problem for DNA strands. The bee-identification problem, introduced by Tandon et al. (2019), requires one to identify M bees, each tagged by a unique barcode, via a set of M noisy measurements. Later, Chrisnata et al. (2022) extended the model to case where one observes N noisy measurements of each bee, and applied the model to address the unordered nature of DNA storage systems.**

In such systems, a unique address is typically prepended to each DNA data block to form a DNA strand, but the address may possibly be corrupted. While clustering is usually used to identify the address of a DNA strand, this requires $\mathcal{M}^2$ data comparisons (when $\mathcal{M}$ is the number of reads). In contrast, the approach of Chrisnata et al. (2022) avoids data comparisons completely. In this work, we study an intermediate, data-driven approach to this identification task.

For the binary erasure channel, we first show that we can almost surely correctly identify all DNA strands under certain mild assumptions. Then we propose a data-driven pruning procedure and demonstrate that on average the procedure uses only a fraction of $\mathcal{M}^2$ data comparisons. Specifically, for $\mathcal{M} = 2^n$ and erasure probability p, the expected number of data comparisons performed by the procedure is $\kappa\mathcal{M}^2$, where $\left(\frac{1+2p-p^2}{2}\right)^n \leq \kappa \leq \left(\frac{1+p}{2}\right)^n$.

I. Introduction

Existing storage technologies cannot keep up with the modern data explosion. Current solutions for storing huge amounts of data uses magnetic and optical disks. Despite improvements in optical discs, storing a zettabyte of data would still take many millions of units, and use significant physical space. Certainly, there is a growing need for a significantly more durable and compact storage system. The potential of macromolecules in ultra-dense storage systems was recognized as early as in the 1960s, when the celebrated physicists Richard Feynman outlined his vision for nanotechnology in the 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 an approach that avoids clustering. 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 (strand) results in N noisy outputs (reads). 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 noisy addresses, which are of significantly shorter length, and the method does not take into account the associated noisy data. Hence, this approach involves no data comparisons.

The work of Han Mao Kiah was supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 Award MOE-T2EP20121-0007.

The work of Shubhransh Singhvi, Avital Boruchovsky, and Eitan Yaakobi was funded by NSF Grant CCF 2212437 and by the European Union (ERC, DNAStorage, 865630). Views and opinions expressed are however those of the author(s) 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.

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

In this work, we consider an intermediate, data-driven approach to the identification task by drawing ideas from the clustering and the bee identification problems. Specifically, we focus on the case of the binary erasure channel and the case where the addresses are uncoded. We first show that we can almost surely correctly identify all DNA strands under certain mild assumptions. Then we propose a data-driven pruning procedure and demonstrate that on average the procedure uses only a fraction of $\mathcal{M}^2$ data comparisons (when there are $\mathcal{M}$ reads). We formally define our problem in the next section. Due to space limitations, all the proofs have been omitted and are presented in the full version of the paper [1].

II. 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 $C \subseteq \{0,1\}^n$ be a binary code of length n and size M (the addresses), and assume that every codeword $\boldsymbol{x}_i \in C$ is attached to a length-L data part $\boldsymbol{d}_i \in \{0,1\}^L$ to form a strand, which is the tuple, $(\boldsymbol{x}_i, \boldsymbol{d}_i)$. We denote the ratio of the length of the data part to the length of the address part by β, i.e., $L \triangleq \beta n$, where $\beta \in \mathbb{R}_+$. 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 $\{0,1\}^L$ and that C is the whole space. 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, $R'_N = \{\{(\boldsymbol{y}_1, \boldsymbol{d}'_1), (\boldsymbol{y}_2, \boldsymbol{d}'_2), \ldots, (\boldsymbol{y}_{MN}, \boldsymbol{d}'_{MN})\}\}$, of MN noisy strands (reads) is obtained, where for every $j \in [MN]$, $(\boldsymbol{y}_j, \boldsymbol{d}'_j) \in \mathcal{S}_N((\boldsymbol{x}_{\pi(j)}, \boldsymbol{d}_{\pi(j)}))$ 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'_N, has access to the set of addresses C but does not know the set of data D. In this work, we first consider the following problem.

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

For $(N, \beta) \in \mathcal{R}$, we can find the true permutation by making at least $(N|C|)^2$ 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 $\mathcal{S}$ and ϵ and $(N, \beta) \in \mathcal{R}$, design an algorithm to identify the true permutation with probability at least $1-\epsilon$ using $\kappa(N|C|)^2$ data comparisons. As before, C is the whole space and D is drawn uniformly at random.*

Unless otherwise stated, we assume that $\mathcal{S}$ is the $\mathsf{BEC}(p)$ channel with $0 < p < 1$. In Section III, 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 IV, we address Problem 1 and identify the region $\mathcal{R}$ for which there exists only one valid permutation, viz. the true permutation. In Section V, we propose a data-driven pruning algorithm that identifies the true-permutation with probability at least $1-\epsilon$ when $(N, \beta) \in \mathcal{R}$. In Section VI, we analyse the expected number of data comparisons performed by the data-driven pruning algorithm.

III. The Peeling Matching Algorithm (PMA)

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 ($\mathcal{X} = C$) and the right nodes are the noisy reads ($\mathcal{Y} = R'_N$). 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 right nodes and left nodes with degrees 1 and N 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 $|\{(\boldsymbol{y}, \boldsymbol{d}') \in E_x : |E_{(\boldsymbol{y},\boldsymbol{d}')}| = 1\}| = N$.*

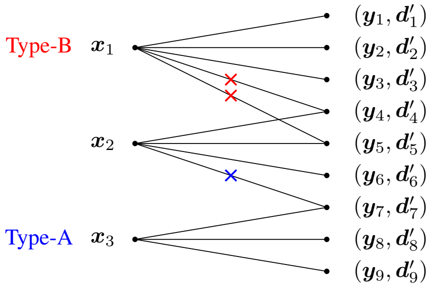
Type-B
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
Type-A
$\boldsymbol{x}_3$
$(\boldsymbol{y}_1, \boldsymbol{d}'_1)$
$(\boldsymbol{y}_2, \boldsymbol{d}'_2)$
$(\boldsymbol{y}_3, \boldsymbol{d}'_3)$
$(\boldsymbol{y}_4, \boldsymbol{d}'_4)$
$(\boldsymbol{y}_5, \boldsymbol{d}'_5)$
$(\boldsymbol{y}_6, \boldsymbol{d}'_6)$
$(\boldsymbol{y}_7, \boldsymbol{d}'_7)$
$(\boldsymbol{y}_8, \boldsymbol{d}'_8)$
$(\boldsymbol{y}_9, \boldsymbol{d}'_9)$

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 $\mathsf{Y}_{\mathsf{g}}, \mathsf{X}_{\mathsf{g}_\mathsf{A}}$ and $\mathsf{X}_{\mathsf{g}_\mathsf{B}}$ 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
        PEEL($\mathcal{P}_{\mathcal{G}}, \mathcal{G}, \boldsymbol{x}$)
    if $|\mathcal{P}_E| = N2^n$ then
        return $\mathcal{P}_{\mathcal{G}}$
    else
        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. *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 $\mathcal{U}_{\mathsf{cycle}} \triangleq 2^{-N\left((1+p^2)^n - 1\right)}$.

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.

IV. Uniqueness of the N-Permutation

In this section, we study Problem 1 when $\mathcal{S} = \mathsf{BEC}(p)$. Specifically, in Lemmas 3 and 4, we determine the values β_{Th} and N_{Th}, respectively, such that for all $\beta \geq \beta_{\mathsf{Th}}$ 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.

For $\mathcal{S} = \mathsf{BEC}(p)$, 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'_N and C, we are able to find the true permutation if and only if there exists only one valid partitioning and one valid labelling.

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

Lemma 2. *For $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$, $P((\boldsymbol{y}, \boldsymbol{d}') \in \mathsf{Y}_{\mathsf{faulty}})$ is*

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

Definition 2. *A **partitioning** $\mathcal{P} = \{P_1, P_2, \ldots, P_M\}$ of $\mathcal{Y}$ is defined as the collection of disjoint submultisets of $\mathcal{Y}$, 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 $\mathcal{Y}$. Let $\mathbb{P}_{\mathcal{Y}}$ denote the set of all possible partitionings of $\mathcal{Y}$. Note that if $|\mathbb{P}_{\mathcal{Y}}| = 1$ then $\mathbb{P}_{\mathcal{Y}} = \{\mathcal{P}^*\}$. Let $\mathcal{G}' = (\mathcal{Y}, E')$. Now consider the graph, $\mathcal{G}' = (\mathcal{X}, E')$, where $\mathcal{Y} = R'_N$. 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 $\mathcal{P} \in \mathbb{P}_{\mathcal{Y}}$ corresponds to partitioning the graph $\mathcal{G}'$ into M cliques each of size N.

Proposition 2. $|\mathbb{P}_{\mathcal{Y}}| = 1$ *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 β such that for $\beta \geq \beta_{\mathsf{Th}}, \mathbb{P}_{\mathcal{Y}} = \{\mathcal{P}^*\}$ with probability at least $1 - \epsilon_1$.

Lemma 3. *For $\beta \geq \beta_{\mathsf{Th}} \triangleq \frac{\log_2\left(\frac{N((1+2p-p^2)^n - 1)}{\sqrt[N]{\epsilon_1/2^n}}\right)}{n(1 - \log_2(1+2p-p^2))}$, we have that $\mathbb{P}_{\mathcal{Y}} = \{\mathcal{P}^*\}$ with probability at least $1 - \epsilon_1$.*

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 C 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 $\mathbb{L}_{\mathcal{P},\mathcal{Y}}$. 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 $\mathcal{L}^* \notin \mathbb{L}_{\mathcal{P},\mathcal{Y}}$. Further, if $|\mathbb{L}_{\mathcal{P}^*,\mathcal{Y}}| = 1$ then $\mathbb{L}_{\mathcal{P}^*,\mathcal{Y}} = \{\mathcal{L}^*\}$. Let $\mathcal{G}'' = (\mathcal{X}, E'')$, where $\mathcal{X} = C$. 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. *$|\mathbb{L}_{\mathcal{P}^*,\mathcal{Y}}| = 1$ 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 $N \geq N_{\mathsf{Th}}, \mathbb{L}_{\mathcal{P}^*,\mathcal{Y}} = \{\mathcal{L}^*\}$ with probability at least $1-\epsilon_2$.

Lemma 4. *For $N \geq N_{\mathsf{Th}} \triangleq \frac{\log_2\left(\sqrt[n]{\frac{\epsilon_2+2^n}{2^n}}-1\right)}{\log_2(p)}$, we have that $\mathbb{L}_{\mathcal{P}^*,\mathcal{Y}} = \{\mathcal{L}^*\}$ with probability at least $1-\epsilon_2$.*

Thus, we define the region $\mathcal{R}$ as $\mathcal{R} \triangleq \{(\beta, N) : \beta \geq \beta_{\mathsf{Th}}, N \geq N_{\mathsf{Th}}\}$. The next theorem follows from Lemma 3 and 4.

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

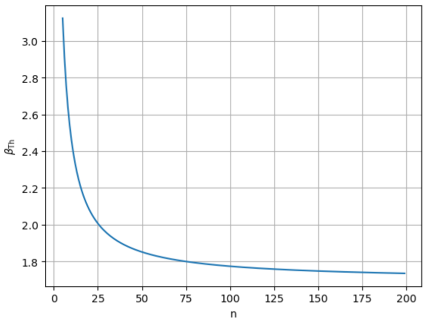

Fig. 2: Plot of β_{Th} versus n for $N = 2, p = 0.2, \epsilon_1 = 0.01$.

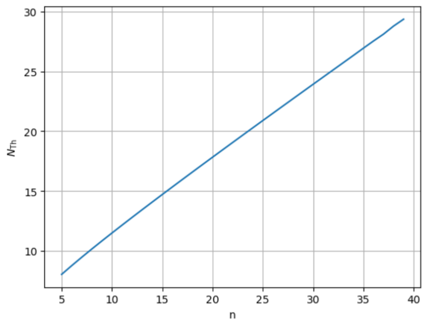

Fig. 3: Plot of N_{Th} versus n for $p = 0.3, \epsilon_2 = 0.01$.

From Lemma 3 and 4, we observe that $\beta_{\mathsf{Th}} < \beta^*$ and $N_{\mathsf{Th}} < \nu^* n$ for some constants β^* and ν^*. This means that we only require data parts to be of length $L = \beta^* n$ 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.

V. Data-driven Pruning 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 III. Let $\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}$ denote the two-hop neighborhood of $(\boldsymbol{y}, \boldsymbol{d}')$ in $\mathcal{G}$. Note that 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 $\boldsymbol{y} \ncong \widetilde{\boldsymbol{y}}$. In the next lemma, we calculate the expected value of $|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|$.

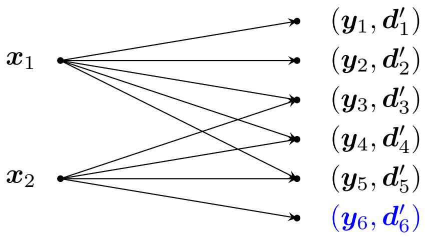
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$(\boldsymbol{y}_1, \boldsymbol{d}'_1)$
$(\boldsymbol{y}_2, \boldsymbol{d}'_2)$
$(\boldsymbol{y}_3, \boldsymbol{d}'_3)$
$(\boldsymbol{y}_4, \boldsymbol{d}'_4)$
$(\boldsymbol{y}_5, \boldsymbol{d}'_5)$
$(\boldsymbol{y}_6, \boldsymbol{d}'_6)$

Fig. 4: 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. *For a given $(\boldsymbol{y}, \boldsymbol{d}') \in \mathcal{Y}$,*

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

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

The data-driven pruning 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. 4, performs $|\mathcal{N}_{(\boldsymbol{y},\boldsymbol{d}')}|$ data comparisons to identify the remaining $N-1$ copies. Note that this pruning procedure finds the remaining $N-1$ copies if and only if $(\boldsymbol{y}, \boldsymbol{d}') \notin \mathsf{Y}_{\mathsf{faulty}}$. Let $\mathcal{P}_{\mathcal{G}} = (\mathcal{X} \cup \mathcal{Y}, \mathcal{P}_E)$ denote the bipartite matching identified by the data-driven pruning algorithm.

Proposition 4. *For $(\beta, N) \in \mathcal{R}$, Algorithm 2 finds the true permutation with probability at least $1-\epsilon$, when $\epsilon_1, \epsilon_2 < \frac{\epsilon}{2}$.*

VI. Analysis of Data-driven Pruning Algorithm

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 $(\beta, N) \in \mathcal{R}$.

Algorithm 2 Data-driven Pruning 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 $\mathsf{PMA}(\mathcal{P}_{\mathcal{G}}, \mathcal{G})$
```

Lemma 6. *The expected number of data comparisons performed by Algorithm 2 when $(\beta, N) \in \mathcal{R}$ is at most*

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

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

Lemma 7. *For $\beta \geq \beta_0 \triangleq \frac{\log_2\left(\frac{\epsilon_1}{2^n N^2((1+2p-p^2)^n-1)}\right)}{n \log_2\left(1-\frac{1}{2}(1-p)^2\right)}$, $P(|\mathsf{Y}_{\mathsf{faulty}}| > 1) < \epsilon_1$.*

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

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

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

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

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 10. *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 $i \in [|S|]$.

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

Lemma 11. *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.*

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

Lemma 12. *The expected number of data comparisons performed by Algorithm 2 when $(\beta, N) \in \mathcal{R}''$ is at most*

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

Hence, from Lemmas 6, 9 and 12, 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, where $\left(\frac{1+2p-p^2}{2}\right)^n \leq \kappa_{\beta,N} \leq \left(\frac{1+p}{2}\right)^n$.

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.