An Instance-Based Approach to the Trace Reconstruction Problem

Kayvon Mazooji and Ilan Shomorony
University of Illinois, Urbana-Champaign
mazooji2@illinois.edu, ilans@illinois.edu

***Abstract*—In the trace reconstruction problem, one observes the output of passing a binary string $s \in \{0,1\}^n$ through a deletion channel T times and wishes to recover s from the resulting T "traces." Most of the literature has focused on characterizing the hardness of this problem in terms of the number of traces T needed for perfect reconstruction either in the worst case or in the average case (over input sequences s). In this paper, we propose an alternative, instance-based approach to the problem. We define the "Levenshtein difficulty" of a problem instance (s, T) as the probability that the resulting traces do not provide enough information for correct recovery with full certainty. One can then try to characterize, for a specific s, how T needs to scale in order for the Levenshtein difficulty to go to zero, and seek reconstruction algorithms that match this scaling for each s. For a class of binary strings with alternating long runs, we precisely characterize the scaling of T for which the Levenshtein difficulty goes to zero. For this class, we also prove that a simple "Las Vegas algorithm" has an error probability that decays to zero with the same rate as that with which the Levenshtein difficulty tends to zero.**

I. Introduction

In the trace reconstruction problem, originally proposed by Levenshtein in 1997 [1, 2], there exists a binary source string $s \in \{0,1\}^n$ and we are given a set $\mathcal{T}$ of traces of s, where a trace is simply a subsequence of s. We are then asked to reconstruct s from $\mathcal{T}$. The problem received considerable attention in the last few years [3–14], partially due to its applications in nanopore sequencing [6], DNA-based storage [15, 16] and personalized immunogenomics [17].

Most of the work on the trace reconstruction problem has focused on characterizing the minimum number of traces T required to guarantee perfect reconstruction of s with high probability. Batu et al. [3] studied the problem in the setting where each trace is generated by passing s through a deletion channel that deletes every bit in s independently with probability p (and coined the name "trace reconstruction"). The authors derived lower and upper bounds on the number of traces T needed to guarantee correct reconstruction with high probability in two different problem settings: in the *worst-case* trace reconstruction problem, an algorithm is required to reconstruct *any* sequence $s \in \{0,1\}^n$ with high probability; in the *average-case* trace reconstruction problem, s is assumed to be chosen uniformly at random from $\{0,1\}^n$ (equivalently, each symbol is independently chosen as $\mathrm{Bern}(1/2)$).

Since the work by Batu et al., significant development has been made for both the worst-case and the average-case versions of the problem. For the average-case version, assuming s is chosen uniformly at random from $\{0,1\}^n$, it is known that $T = \exp(O(\log^{1/3} n))$ traces are sufficient [9], while at least $\tilde{\Omega}(\log^{5/2}(n))$ traces are required [18]. However, in many practical settings the assumption that s is an i.i.d. random string is unrealistic (e.g., DNA sequences exhibit long repeat patterns [19–21]). On the other hand, for the worst-case version of the problem, it is known that $T = \exp(\tilde{O}(n^{1/5}))$ traces suffice [22], while the best known lower bound on the minimum number of traces required is $\tilde{\Omega}(n^{3/2})$ [18]. The gap between the upper and lower bounds in this case is striking. This suggests that the landscape of problem instances is too diverse, some easy and some very hard, and requiring an algorithm to reconstruct any s correctly may be too strict.

Inspired by the reconstruction condition in Levenshtein's original work [1, 2], we propose an instance-based approach to the trace reconstruction problem. The approach is based on a feasibility question: if the length of the source string s is known, when does the set of traces $\mathcal{T}$ contain enough information to allow the unambiguous recovery of s with full certainty? More precisely, we say that $\mathcal{T}$ is Levenshtein sufficient if s is the only length-n sequence that could have generated $\mathcal{T}$. The term *Levenshtein sufficiency* is used because reconstruction with full certainty is required in Levenshtein's original study of the problem [1, 2]. More random traces are generally required for Levenshtein-sufficiency to hold compared to the commonly studied goal of reconstruction with high probability [18, 22], since the latter does not require that the string has been reconstructed with full certainty. Current state of the art algorithms for worst-case trace reconstruction with high probability can use a metric, such as likelihood, to choose among a set of possible candidate reconstructions that are all consistent with $\mathcal{T}$. In contrast, Levenshtein-sufficiency requires that there is only one possible reconstruction from the set of traces.

For a problem instance defined by the a pair (s, T) (where T is the number of traces, not the traces themselves), a natural definition for the difficulty of (s, T) is then

$$\mathcal{D}(s, T) = \Pr\left(\mathcal{T} \text{ is not sufficient} \,|\, s, T\right), \tag{1}$$

The work of K.M. and I.S. was supported in part by the National Science Foundation (NSF) under grants CCF-2007597 and CCF-2046991. Part of this paper was presented at the 58th Annual Conference on Information Sciences and Systems (CISS 2024).

where $\mathcal{T}$ is a set of T traces of s generated with deletion probability p. We refer to $\mathcal{D}(s,T)$ as the Levenshtein difficulty of (s,T).

For example, the sequence $s = 00\ldots0011\ldots11$ (similar to the sequences used to prove the lower bound in [3]) has a high value of $\mathcal{D}(s,T)$, since it is unlikely that the traces will reveal the right number of zeros and ones. Levenshtein difficulty provides an algorithm-independent measure of difficulty for reconstructing a particular source string s that other frameworks do not explicitly provide, and thus gives rise to an instance-based approach to trace reconstruction.

Intuitively, a good reconstruction algorithm should have a small error probability for instances (s,T) for which $\mathcal{D}(s,T)$ is small, but should not be heavily penalized if its error probability is large on an instance with large $\mathcal{D}(s,T)$. In particular, if we consider the asymptotic regime $n \to \infty$ (considered in most studies of the trace reconstruction problem), a natural goal is to design an algorithm $\mathcal{A}$ satisfying

$$\Pr(\text{error} \,|\, \mathcal{A}, s, T) \to 0 \;\text{ whenever }\; \mathcal{D}(s,T) \to 0, \qquad (2)$$

as $n \to \infty$. This means that, if T scales with n fast enough so that $\mathcal{D}(s,T) \to 0$ (i.e., T is scaling in a way that makes the problem feasible with high probability), then the algorithm's error probability also goes to zero. Because our goal is reconstruction with full certainty in this framework, we are interested in algorithms that either output the source string, or report an error if the source string is not the only possible reconstruction. The probability of failure of such an algorithm then serves as an upper bound on $\mathcal{D}(s,T)$. If in addition, the algorithm is computationally efficient given the number of traces T, it can be classified as a Las Vegas algorithm [23].

We initiate the investigation of this framework for the trace reconstruction problem by deriving general necessary conditions and sufficient conditions for $\mathcal{T}$ to be Levenshtein sufficient for s. These conditions are in terms of specific sequence patterns that must be preserved in the traces, and they translate into bounds on $\mathcal{D}(s,T)$. We use these necessary conditions to prove that a broad class of strings containing consecutive repeated sequences requires $\exp(\text{poly}(n))$ traces for Levenshtein sufficiency to hold with high probability. We then focus on a class of strings s (strings that are formed by a fixed number of alternating runs of zeros and ones) and show that, for this class, $\mathcal{D}(s,T)$ exhibits a phase transition: if T grows faster than $\exp(c^* n)$, where c^* is a positive constant that depends on the length of the runs, $\mathcal{D}(s,T) \to 0$; if T grows slower than that, then $\mathcal{D}(s,T) \to 1$. We then propose a simple Las Vegas algorithm (based on observing the maximum run lengths in traces) for which the goal in (2) is achieved. Therefore, for this class of strings, this algorithm is optimal, in the sense that its error probability goes to zero whenever we have a sequence of problems whose difficulty tends to zero. Moreover, we show that in those cases the error probability decays with a similar rate to the instance difficulty $\mathcal{D}(s,T)$. An appendix containing omitted derivations is available in a longer version of this paper [24].

II. Preliminaries

Strings in this paper are binary and indexed starting from 1. For a given string x, we let $|x|$ denote the length of x. A *subsequence* of x is a string that can be formed by deleting elements from x, and a *supersequence* of x is a string that can be formed by inserting elements into x. This is in comparison to a *substring* of x, which is a string that appears contiguously in x. We let $x[i,j] = (x_i, x_{i+1}, \ldots, x_j)$ be the substring of x that begins at position i and ends at position j. For a string a and positive integer r, we let a^r be the length $r|a|$ string $aa..a$ that has a repeated r times. A *run* in a string s is a substring in s of the form 0^r that has a 1 or nothing on either side, or 1^r that has a 0 or nothing on either side. For example, for the string 010011, the runs in order are 0, 1, 00, and 11. For a string x, we denote the ith run in x by $r_i(x)$. If x is clear from context, we denote $r_i(x)$ by r_i. Following standard notation, for two real-valued sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n \sim b_n$ if $\lim_{n\to\infty} a_n/b_n = 1$. We let $\log(x)$ denote the natural logarithm of x. For a matrix A, we denote the transpose of A by $A^\dagger$. Following standard notation, for a positive integer M, we let $[M]$ denote the set $\{1,\ldots,M\}$.

Let $s = (s_1, s_2, \ldots, s_n) \in \{0,1\}^n$ be the length-n string we are trying to recover. s will be called the *source string*. A *trace* of s is any subsequence of s. For our probabilistic analysis, we denote the *channel* that s is passed through by $\mathcal{C}$. In this paper, $\mathcal{C}$ is the *deletion channel* Del_p that deletes each bit of the source string s independently with probability p. In our probabilistic analysis, a trace t of s is the output of channel $\mathcal{C}$ when s is passed through it, i.e., $t \leftarrow \mathcal{C}(s)$.

Definition 1. *A set of traces $\mathcal{T}$ is Levenshtein sufficient for reconstructing a string s if the only length-n string that could have given rise to $\mathcal{T}$ is s itself.*

Given $\mathcal{T}$ that is sufficient for s, we wish to reconstruct s from $\mathcal{T}$. A problem instance is defined by a pair (s,T).

Definition 2. *The Levenshtein difficulty of instance (s,T) is defined as*

$$\mathcal{D}(s,T) = \Pr\left(\mathcal{T} \textit{ is not sufficient} \,|\, s, T\right). \qquad (3)$$

An algorithm $\mathcal{A}$ is *Levenshtein efficient* for a sequence of strings $\{s_n\}$ (where n indicates the length of s_n) and number of traces $\{T_n\}$ if it always either outputs the correct source string, or does not output a string and reports an error, and as n increases, the probability that $\mathcal{A}$ fails to reconstruct s_n approaches zero whenever the instance difficulty approaches zero; i.e., if

$$\Pr(\text{error} \,|\, \mathcal{A}, s_n, T_n) \to 0 \text{ whenever } \mathcal{D}(s_n, T_n) \to 0, \qquad (4)$$

as $n \to \infty$. Here, the probability of error for $\mathcal{A}$ depends on any randomness used in $\mathcal{A}$ in addition to randomness in trace generation. In principle, we can always find the set of all possible length-n strings that could have given rise to $\mathcal{T}$ by taking the intersection of all sets S_i for $i \in [T]$, where S_i is the set of length-n strings that are supersequences of trace t_i. This brute-force "algorithm" reconstructs the source

strings correctly whenever $\mathcal{T}$ is Levenshtein sufficient, and is therefore a Levenshtein efficient reconstruction algorithm for any sequence of source strings $\{s_n\}$ and number of traces $\{T_n\}$. However, it is not computationally efficient since in the worst case; it runs in time exponential in n. Furthermore, such an algorithm is difficult to analyze and provides no insight as to which scaling of T_n is needed in order for $\mathcal{D}(s, T_n) \to 0$.

III. Main Results

Our first main result provides a lower bound on how T_n must scale with the string length n to guarantee that $D(s_n, T_n) \to 0$ when s_n belongs to the class of sequences of strings defined below.

Definition 3. *We denote by $\mathcal{Q}(r_n, f_n)$ the set of sequences of strings $\{s_n\}$ such that s_n is a string of length n that contains a substring of the form A^{f_n} for some string A where $|A| = r_n$. Note that if for some n, f_n is not an integer, the substring in s_n can be $A^{\lceil f_n \rceil}$ or $A^{\lfloor f_n \rfloor}$, so long as $|s_n| = n$.*

For example, the class $\mathcal{Q}(2, n/4)$ contains the sequence of strings of the form $(01)^{n/4}0^{n/2}$ where A is taken to be 01 in this case.

Theorem 1. *Let $c^* = \ell \log(\frac{1}{1-p^r})$. For a sequence of strings $\{s_n\} \in \mathcal{Q}(r, \ell n)$ where r, ℓ are constants such that $r \geq 1$ and $0 < \ell \leq 1$, the instance difficulty satisfies, as $n \to \infty$,*

$$\mathcal{D}(s_n, T_n) \to 1 \text{ if } T_n = O(\exp(cn)), \quad c < c^*. \tag{5}$$

This theorem shows that any string in $\mathcal{Q}(r, \ell n)$ requires an exponential number of traces in n, and can be further generalized to strings in $\mathcal{Q}(r, \ell n^a)$ for $a \leq 1$ as shown in the next section. Interestingly, the string pairs that are used to calculate lower bounds on trace reconstuction with high probability [18] use strings in $\mathcal{Q}(r, \ell n)$. Theorem 1 is proved by establishing necessary conditions for a set of traces to be Levenshtein sufficient, and calculating the probability that the conditions are satisfied.

We use Theorem 1 to give an asymptotic characterization of $\mathcal{D}(s_n, T_n)$ as the string length n increases when $\{s_n\}$ belongs to the more restrictive class of sequences of strings $\mathcal{S}(M, \ell^*)$ defined below.

Definition 4. *We denote by $\mathcal{S}(M, \ell^*)$ the set of sequences of strings $\{s_n\}$ such that s_n has M runs for all n, the ith run has length $\ell_i n$ with $\sum_{i=1}^{M} \ell_i = 1$, and $\ell^* = \max_{i \in [M]} \ell_i$. Note that if for some n, there exists a subset $G \subseteq [M]$ such that $\ell_i n$ is not an integer for $i \in G$, the length of the ith run can be chosen to be $\lceil \ell_i n \rceil$ or $\lfloor \ell_i n \rfloor$ for each $i \in G$, in any way such that $|s_n| = n$.*

For example, the class $\mathcal{S}(3, 1/2)$ contains strings of the form $0 \ldots 01 \ldots 10 \ldots 0$ and $1 \ldots 10 \ldots 01 \ldots 1$, with a maximum run of length $n/2$. Notice that $\mathcal{S}(M, \ell^*) \subset \mathcal{Q}(1, \ell^* n)$. While somewhat restrictive, this class serves as a way to study the impact of the number of runs and the run lengths on the problem difficulty.

Theorem 2. *Let $c^* = \ell^* \log(\frac{1}{1-p})$. For a sequence of strings $\{s_n\} \in \mathcal{S}(M, \ell^*)$, the instance difficulty satisfies, as $n \to \infty$,*

$$\mathcal{D}(s_n, T_n) \to 0 \text{ if } T_n = \Omega(\exp(cn)), \quad c > c^*, \tag{6}$$

$$\mathcal{D}(s_n, T_n) \to 1 \text{ if } T_n = O(\exp(cn)), \quad c < c^*. \tag{7}$$

We prove Theorem 2 in the next sections by applying Theorem 1, and deriving a sufficient condition for $\mathcal{T}$ to be Levenshtein sufficient. The sufficient conditions can also be seen as the sufficient conditions for a simple algorithm called Maximal Runs (Algorithm 1) to reconstruct s from $\mathcal{T}$.

Theorem 2 establishes a critical phenomenon for the problem difficulty for sequences in $\mathcal{S}(M, \ell^*)$ and precisely characterizes the regime where an algorithm can be expected to perform well. It may seem counterintuitive that $T = \exp(\Omega(n))$ traces are required for a set of traces to be Levenshtein-sufficient with high probability for a source string in $\mathcal{S}(M, \ell^*)$, while $T = \exp(\tilde{O}(n^{1/5}))$ traces are known to be sufficient for reconstructing any source string with high probability [22]. This difference in trace complexity occurs because algorithms for reconstructing a string with high probability can select one among a set of possible reconstructions based on some specific criterion (such as likelihood or another statistic of the traces), without the set of traces necessarily being Levenshtein sufficient.

In Theorem 3, we state that the Maximal Runs algorithm (Algorithm 1) performs well precisely when T_n scales so that $\mathcal{D}(s_n, T_n) \to 0$. While very simple, the Maximal Runs algorithm is computationally efficient in that it runs in $O(nT)$ time. Moreover, it is easy to see that this algorithm can only recover s correctly when $\mathcal{T}$ is Levenshtein sufficient.

Theorem 3. *Let $c^* = \ell^* \log(\frac{1}{1-p})$, and let $\mathcal{A}$ be the Maximal Runs algorithm (Algorithm 1). For $\{s_n\} \in \mathcal{S}(M, \ell^*)$, $\mathcal{A}$ satisfies, as $n \to \infty$,*

$$\Pr(\text{error} \mid \mathcal{A}, s_n, T_n) \to 0, \tag{8}$$

$$\frac{\log(\Pr(\text{error} \mid \mathcal{A}, s_n, T_n))}{\log(\mathcal{D}(s_n, T_n))} \to 1, \tag{9}$$

as long as $T_n = \Omega(\exp(cn))$, for $c > c^*$.

Observe that c^* is a sharp threshold on the exponent c in $T = \exp(cn)$, below which the problem becomes infeasible as n increases, and above which, the problem is feasible and the Maximal Runs algorithm outputs the correct answer as n increases. Furthermore, (9) implies that error probability $\Pr(\text{error} \mid \mathcal{A}, s_n, T_n)$ goes to zero at the same exponential rate as the rate with which $\mathcal{D}(s, T)$ goes to zero.

IV. Concluding remarks

In this paper, we proposed an instance-based approach to the trace reconstruction problem based on a new notion of instance-specific difficulty. For a class of strings with a fixed number of runs, we precisely characterized how the number of traces needs to grow as a function of the run lengths in order for the instance difficulty to go to zero. While the class of strings considered is somewhat restrictive, we obtain

Algorithm 1: Maximal Runs

Data: $n, \mathcal{T}$
Result: $\hat{s}$

$\hat{s} \leftarrow$ empty string;
$\hat{M} \leftarrow$ maximum number of runs in any trace in $\mathcal{T}$;
$S \leftarrow$ set of all traces with $\hat{M}$ runs;
if $t_1[1] = t_2[1]$ *for all* $t_1, t_2 \in S$ **then**
  for $i \in [\hat{M}]$ **do**
    $t^* \leftarrow \arg\max_{t \in S} |r_i(t)|$;
    $x_i \leftarrow r_i(t^*)$;
  if $\sum_{i \in [\hat{M}]} |x_i| = n$ **then**
    $\hat{s} \leftarrow x_1 x_2 ... x_{\hat{M}}$;

sharp bounds on the required T, in contrast to most existing results for trace reconstruction. In addition, we derived a lower bound on the number of traces for the instance difficulty to go to zero for a much broader class of strings. This work can thus be seen as developing the initial tools for a more general characterization of the instance-based hardness of trace reconstruction. We note that our Theorem 1 can be generalized to the following as proved in the following section.

Theorem 4. *Let* $c^* = \ell \log(\frac{1}{1-p^r})$. *For a sequence of strings* $\{s_n\} \in \mathcal{Q}(r, \ell n^a)$ *where* $r \geq 1$ *and* $0 < \ell, a \leq 1$, *the instance difficulty satisfies, as* $n \to \infty$,

$$\mathcal{D}(s_n, T_n) \to 1 \text{ if } T_n = O(\exp(cn^a)), \quad c < c^* \tag{10}$$

This shows that any string that contains a constant length string repeated consecutively a polynomial number of times in n requires a superpolynomial number of traces in n for Levenshtein sufficiency to hold with high probability.

Finally, we point out that other interesting definitions for the "sufficiency" of $\mathcal{T}$ are possible. For example, one could say $\mathcal{T}$ is sufficient for s if the maximum likelihood source string given $\mathcal{T}$ is s.

V. Proof of Main Results

We begin by introducing necessary and sufficient conditions for $\mathcal{T}$ to be Levenshtein sufficient for s, which we will use to prove the main results. Out of the conditions in Lemma 1, we only need condition (1) for our analysis of Levenshtein sufficiency, but we include the other two conditions to show additional requirements for Levenshtein sufficiency.

Lemma 1. *For any two distinct binary strings* A, B *such that* $|A| \leq |B|$, *we have the following necessary conditions on the set of traces* $\mathcal{T}$ *to be Levenshtein sufficient for* s. *Let* $a, b \in \mathbb{N}$ *such that* $a, b \geq 1$.

1) *If* A^a *is a substring of* s, *it cannot happen that for every trace, there exists a copy of* A *in this substring that is deleted.*
2) *If* $B^a A B^b$ *is a substring of* s, *it cannot happen that for every trace, there exists a copy of* B *in this substring that is deleted.*
3) *If* $A^a B^b$ *or* $B^b A^a$ *is a substring of* s, *it cannot happen that for every trace, there exists a copy of* A *or* B *that is deleted from this substring.*

Proof: (1) If a copy of A is deleted from this substring in every trace, then $\mathcal{T}$ could arise from s with the substring A^a replaced by DA^{a-1} for any string D such that $|D| = |A|$ and $D \neq A$.

(2) If a copy of B is deleted from this substring in every trace, then $\mathcal{T}$ could arise from s with the substring $B^a A B^b$ replaced by $G = B^{a-1} A B A 1^{|B|-|A|} B^{b-1}$. To see this, notice that if a trace of s has a copy of B deleted from B^a in $B^a A B^b$, then the same trace can be formed if $B^a A B^b$ is replaced by G in s since $A1^{|B|-|A|)}$ can be deleted from G. A similar argument holds if a trace of s has a copy of B deleted from B^b in $B^a A B^b$.

(3) If a copy of A or B is deleted from the substring $A^a B^b$ in every trace, then $\mathcal{T}$ could arise from s with the substring $A^a B^b$ replaced by $A^{a-1} B A B^{b-1}$. The $B^b A^a$ case is analogous. ∎

Lemma 2. $\mathcal{T}$ *is Levenshtein sufficient for* s *if for each run* i, *there exists a trace such that no run is fully deleted and run* i *is fully preserved.*

The fact that the conditions in Lemma 2 imply Levenshtein sufficiency is straightforward to verify. Moreover, it is easy to see that these conditions guarantee that Algorithm 1 recovers s correctly. Also notice that the condition for Levenshtein sufficiency in Lemma 2 is very similar to the necessary condition (1) in Lemma 1 when A is a single bit, and this observation forms the basis of our result.

We now prove the main theorems. For a source string s_n where $\{s_n\} \in \mathcal{Q}(r, f_n)$ and r is a constant, let $E_1^{x_n}$ denote the event that necessary condition (1) holds for the substring x_n in s_n where $x_n = A^{f_n}$ for some string A of length r. In other words, $E_1^{x_n}$ is the event that for the substring $x_n = AA...A$ of interest, we have that there exists a trace where no copy of A in x_n is fully deleted.

Let E_2 denote the event that the sufficient condition in Lemma 2 holds for (s_n, T_n). Notice that for any s_n and T_n such that $\{s_n\} \in \mathcal{Q}(r, f_n)$ and x_n is a substring of s_n of the form described above, it follows that

$$\Pr(\bar{E}_1^{x_n}) \leq \mathcal{D}(s_n, T_n) \leq \Pr(\bar{E}_2). \tag{11}$$

Therefore, by proving that $\lim_{n\to\infty} \Pr(\bar{E}_1^{x_n}) = 1$ for a pair of sequences $\{s_n\}, \{T_n\}$, we prove that $\lim_{n\to\infty} \mathcal{D}(s_n, T_n') = 1$ for any $\{T_n'\}$ such that $T_n' = O(T_n)$ since $\mathcal{D}(s, T)$ can only increase for fixed s if T decreases. Also, by proving that $\lim_{n\to\infty} \Pr(\bar{E}_2) = 0$ for $\{s_n\}, \{T_n\}$, we prove that $\lim_{n\to\infty} \mathcal{D}(s_n, T_n') = 0$ for any $\{T_n'\}$ such that $T_n' = \Omega(T_n)$ since $\mathcal{D}(s, T)$ can only decrease for fixed s if T increases.

Lemma 3 gives an asymptotic characterization of $\Pr(\bar{E}_1^{x_n})$ for $\{s_n\} \in \mathcal{Q}(r, \ell n^a)$ which immediately yields Theorems 1 and 4. Lemma 4 gives an asymptotic characterization of $\Pr(\bar{E}_2)$, which immediately yields Theorems 2 and 3 by the logic above along with the fact that $\mathcal{S}(M, \ell^*) \subset \mathcal{Q}(1, \ell^* n)$.

Lemma 3. *Suppose* $\{s_n\} \in \mathcal{Q}(r, \ell n^a)$ *where* r, ℓ, a *are constants such that* $0 < \ell, a \leq 1$*, and let* x_n *be a substring of* s_n *of the form* $A^{\ell n^a}$ *where* $|A| = r$*. Let* $c^* = \ell \log(\frac{1}{1-p^r})$*. Then, as* $n \to \infty$*,*

- $\Pr(\bar{E}_1^{x_n}) \to 0$ *if* $T_n = \Omega(\exp(cn^a)), \quad c > c^*$ (12)
- $\Pr(\bar{E}_1^{x_n}) \to 1/e$ *if* $T_n = \exp(c^* n^a)$ (13)
- $\Pr(\bar{E}_1^{x_n}) \to 1$ *if* $T_n = O(\exp(cn^a)), \quad c < c^*$ (14)

Lemma 4. *Suppose* $\{s_n\} \in \mathcal{S}(M, \ell^*)$ *and let* x_n *be a run in* s_n *of length* $\ell^* n$*. Let* $c^* = \ell^* \log(\frac{1}{1-p})$*. Then, as* $n \to \infty$*,*

$$\frac{\log(\Pr(\bar{E}_2))}{\log(\Pr(\bar{E}_1^{x_n}))} \to 1 \text{ if } T_n = \Theta(\exp(cn)), \quad c > c^*. \tag{15}$$

VI. Proof of Lemmas 3 and 4

For ease of presentation in this proof, we write T_n as T. For the source string s_n, let x_n be a substring of s_n of the form A^{f_n} where A is of length r. We have that

$$\Pr(E_1^{x_n}) = 1 - \Pr(\bar{E}_1^{x_n}) = 1 - (1 - (1-p^r)^{f_n})^T.$$

Let E_2^i be the event that there is at least one trace that has the ith run fully preserved and has no run fully deleted. With slight abuse of notation, let r_i be the length of the ith run. Let a be the $T \times 1$ binary vector that has a 1 in the ith position if the ith trace has no run fully deleted, and has a 0 in the ith position otherwise. We then have that

$$\begin{aligned}
\Pr(E_2) &= \sum_{a \in \{0,1\}^{T\times 1}} \Pr(E_2|a) \Pr(a) \\
&= \sum_a \left(\prod_{i=1}^M \Pr(E_2^i|a)\right) \left(\prod_{i=1}^M (1-p^{r_i})\right)^{1^\dagger a} \\
&\quad \times \left(1 - \prod_{i=1}^M (1-p^{r_i})\right)^{T-1^\dagger a} \\
&= \sum_a \left(\prod_{i=1}^M \left(1 - \left(1 - \frac{(1-p)^{r_i}}{1-p^{r_i}}\right)^{1^\dagger a}\right)\right) \\
&\quad \times \left(\prod_{i=1}^M (1-p^{r_i})\right)^{1^\dagger a} \left(1 - \prod_{i=1}^M (1-p^{r_i})\right)^{T-1^\dagger a} \\
&= \sum_{j=0}^T \binom{T}{j} \left(\prod_{i=1}^M \left(1 - \left(1 - \frac{(1-p)^{r_i}}{1-p^{r_i}}\right)^{j}\right)\right) \\
&\quad \times (\prod_{i=1}^M (1-p^{r_i}))^j (1 - \prod_{i=1}^M (1-p^{r_i}))^{T-j}
\end{aligned} \tag{16}$$

where the third equality follows because

$$\begin{aligned}
&\Pr(E_2^i|a) = 1 - \Pr(\bar{E}_2^i|a) \\
&= 1 - \frac{\Pr(\bar{E}_2^i \cap a)}{\Pr(a)} \\
&= 1 - \frac{(1-p^{r_i} - (1-p)^{r_i})^{1^\dagger a}}{\left(\prod_{j=1}^M (1-p^{r_j})\right)^{1^\dagger a} \left(1 - \prod_{j=1}^M (1-p^{r_j})\right)^{T-1^\dagger a}} \\
&\quad \times \left(\prod_{j\neq i}^M (1-p^{r_j})\right)^{1^\dagger a} \left(1 - \prod_{j=1}^M (1-p^{r_j})\right)^{T-1^\dagger a} \\
&= 1 - \frac{(1-p^{r_i} - (1-p)^{r_i})^{1^\dagger a}}{(1-p^{r_i})^{1^\dagger a}} \\
&= 1 - \left(1 - \frac{(1-p)^{r_i}}{1-p^{r_i}}\right)^{1^\dagger a}.
\end{aligned} \tag{17}$$

Observe that

$$\Pr(E_2) = \mathsf{E}\left[\prod_{i=1}^M \left(1 - \left(1 - \frac{(1-p)^{r_i}}{1-p^{r_i}}\right)^X\right)\right] \tag{18}$$

where $X \sim \text{Bin}(T, \prod_{i=1}^M (1-p^{r_i}))$ is a binomial random variable with T trials and probability parameter $\prod_{i=1}^M (1-p^{r_i})$.

A. Analysis of $\Pr(\bar{E}_1^x)$

Suppose the string $x_n = AA...A$ that we are analyzing is such that A is repeated ℓn^a times where $0 < a, \ell \leq 1$ are constants, and $|A| = r$. where $r > 0$ is constant.

Suppose the number of traces is $T = \exp(cn^a)$ where c is a positive constant. We will write n in terms of T in the expression for $\Pr(\bar{E}_1^{x_n})$ to perform asymptotic analysis. According to the formula in the previous section, and letting $q = \ell/c$, we have

$$\begin{aligned}
\Pr(\bar{E}_1^{x_n}) &= (1 - (1-p^r)^{\ell n^a})^T \\
&= (1 - (1-p^r)^{q \log(T)})^T \\
&\sim \exp\left(-T^{q \log(1-p^r)+1}\right).
\end{aligned} \tag{19}$$

We prove the asymptotic expression in (19) in the appendix [24].

If $c > c^* = \ell \log(\frac{1}{1-p^r})$, which is equivalent to $\frac{\ell}{c} \log(1 - p^r) + 1 > 0 \quad \forall i \in [M]$, then

$$\lim_{n\to\infty} \Pr(\bar{E}_1^{x_n}) = 0. \tag{20}$$

If T grows faster than $\exp(c^* n^a)$, i.e., $T = \Omega(\exp(cn^a))$ for $c > c^*$, then $P(\bar{E}_1^{x_n})$ approaches zero because for fixed n, having more traces can only cause $P(\bar{E}_1^{x_n})$ to decrease. This proves (12). On the other hand, if $c < c^*$, then clearly

$$\lim_{n\to\infty} \Pr(\bar{E}_1^{x_n}) = 1. \tag{21}$$

If $T = O(\exp(cn^a))$ where $c < c^*$, we have that $\lim_{n\to\infty} \Pr(\bar{E}_1^{x_n}) = 1$ since for any such c, there exists a larger value of c that also satisfies the property and for fixed n, having less traces can only cause $P(\bar{E}_1^{x_n})$ to increase. This proves (14). Finally, if $c = c^*$, then clearly

$$\lim_{n\to\infty} \Pr(\bar{E}_1^{x_n}) = 1/e. \tag{22}$$

B. Analysis of $\Pr(\bar{E}_2)$

In this section suppose that $\{s_n\} \in \mathcal{S}(M, \ell^*)$. Notice that $\mathcal{S}(M, \ell^*) \subset \mathcal{Q}(1, \ell^* n)$, so the analysis of $\Pr(\bar{E}_1^{x_n})$ in the previous subsection can be applied to any string in $\mathcal{S}(M, \ell^*)$.

Suppose the number of traces is $T_n = \exp(cn)$ where c is a positive constant. For ease of presentation, let $q_i = \ell_i/c$

and $u_i = \frac{\ell_i}{c}\log(T)$. Let $X \sim \text{Bin}(T,\ \prod_{i=1}^{M}(1-p^{u_i}))$ be the binomial random variable with T trials and probability parameter $p_X = \prod_{i=1}^{M}(1-p^{u_i})$. We have that

$$\begin{aligned}
\Pr(\bar{E}_2) &= 1 - \mathsf{E}\left[\prod_{i=1}^{M}\left(1-\left(1-\frac{(1-p)^{u_i}}{1-p^{u_i}}\right)^X\right)\right] \\
&= 1 - \mathsf{E}\left[1-\sum_{y=1}^{M}\sum_{\substack{K\subseteq[M]:\\ |K|=y}}(-1)^{y+1}\prod_{i\in K}\left(1-\frac{(1-p)^{u_i}}{1-p^{u_i}}\right)^X\right] \\
&= \sum_{y=1}^{M}\sum_{K\subseteq[M]:\,|K|=y}(-1)^{y+1}\mathsf{E}\left[\prod_{i\in K}\left(1-\frac{(1-p)^{u_i}}{1-p^{u_i}}\right)^X\right] \\
&= \sum_{y=1}^{M}\sum_{K\subseteq[M]:\,|K|=y}(-1)^{y+1} \\
&\quad\times \mathsf{E}\left[\exp\left(\log\left(\prod_{i\in K}\left(1-\frac{(1-p)^{u_i}}{1-p^{u_i}}\right)\right)X\right)\right] \\
&= \sum_{y=1}^{M}\sum_{K\subseteq[M]:\,|K|=y}(-1)^{y+1}\left(1-p_X+p_X\prod_{i\in K}\left(1-\frac{(1-p)^{u_i}}{1-p^{u_i}}\right)\right)^T
\end{aligned} \tag{23}$$

from the moment-generating function of a binomial random variable. Letting $N = |\{i \ : \ \ell_i = \ell_{i^*}\}|$ where $i^* = \arg\max_i \ell_i$, we have that $\Pr(\bar{E}_2)$ is asymptotically given by

$$N\exp\left(-\left(\prod_{k=1}^{M}(1-T^{q_k\log(p)})\right)\left(\frac{T^{q_{i^*}\log(1-p)+1}}{1-T^{q_{i^*}\log(p)}}\right)\right) \tag{24}$$

as proved in the appendix [24]. Therefore, if $c > \ell_{i^*}\log(\frac{1}{1-p}) = \ell^*\log(\frac{1}{1-p}) = c^*$,

$$\frac{\log(\Pr(\bar{E}_2))}{\log(\Pr(\bar{E}_1^{r_{i^*}(s_n)}))} \sim \frac{\prod_{k=1}^{M}(1-T^{q_k\log(p)})}{1-T^{q_{i^*}\log(p)}} \sim 1 \tag{25}$$

as proved in the appendix [24]. This concludes the proof of Lemma 4.

REFERENCES

[1] V. Levenshtein, "Reconstruction of objects from a minimum number of distorted patterns," *Doklady Mathematics*, vol. 55, no. 3, pp. 417–420, 1997.

[2] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Transactions on Information Theory*, vol. 47, no. 1, pp. 2–22, 2001.

[3] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," *Departmental Papers (CIS)*, p. 173, 2004.

[4] M. Abroshan, R. Venkataramanan, L. Dolecek, and A. G. i Fàbregas, "Coding for deletion channels with multiple traces," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1372–1376, IEEE, 2019.

[5] A. Magner, J. Duda, W. Szpankowski, and A. Grama, "Fundamental bounds for sequence reconstruction from nanopore sequencers," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 92–106, 2016.

[6] W. Mao, S. N. Diggavi, and S. Kannan, "Models and information-theoretic bounds for nanopore sequencing," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3216–3236, 2018.

[7] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results.," in *SODA*, vol. 8, pp. 389–398, 2008.

[8] A. De, R. O'Donnell, and R. A. Servedio, "Optimal mean-based algorithms for trace reconstruction," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1047–1056, 2017.

[9] L. Hartung, N. Holden, and Y. Peres, "Trace reconstruction with varying deletion probabilities," in *2018 Proceedings of the Fifteenth Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, pp. 54–61, SIAM, 2018.

[10] M. Cheraghchi, R. Gabrys, O. Milenkovic, and J. Ribeiro, "Coded trace reconstruction," *IEEE Transactions on Information Theory*, 2020.

[11] F. Nazarov and Y. Peres, "Trace reconstruction with exp (o(n1/3)) samples," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1042–1046, 2017.

[12] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "On maximum likelihood reconstruction over multiple deletion channels," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 436–440, IEEE, 2018.

[13] S. R. Srinivasavaradhan, M. Du, S. Diggavi, and C. Fragouli, "Symbol-wise map for multiple deletion channels," in *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 181–185, IEEE, 2019.

[14] O. Sabary, A. Yucovich, G. Shapira, and E. Yaakobi, "Reconstruction algorithms for dna-storage systems," *bioRxiv*, 2020.

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

[16] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, and C. N. Takahashi, "Random access in large-scale dna data storage," *Nature Biotechnology*, 2018.

[17] Y. Safonova and P. A. Pevzner, "De novo inference of diversity genes and analysis of non-canonical v (dd) j recombination in immunoglobulins," *Frontiers in immunology*, vol. 10, p. 987, 2019.

[18] Z. Chase, "New lower bounds for trace reconstruction," in *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, vol. 57, pp. 627–643, Institut Henri Poincaré, 2021.

[19] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, 2013.

[20] I. Shomorony, T. Courtade, S. Kim, and D. Tse, "Information-optimal genome assembly via sparse read-overlap graphs," *Bioinformatics*, vol. 37, no. 17, 2016.

[21] G. M. Kamath, I. Shomorony, F. Xia, T. A. Courtade, and N. T. David, "Hinge: long-read assembly achieves optimal repeat resolution," *Genome research*, vol. 27, no. 5, pp. 747–756, 2017.

[22] Z. Chase, "Separating words and trace reconstruction," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 21–31, 2021.

[23] L. Babai, "Monte-carlo algorithms in graph isomorphism testing," *Université tde Montréal Technical Report, DMS*, no. 79-10, 1979.

[24] K. Mazooji and I. Shomorony, "An instance-based approach to the trace reconstruction problem," *Avaiable on arXiv: https://arxiv.org/abs/2401.14277*.