Substring Density Estimation from Traces

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

***Abstract*—In the trace reconstruction problem, one seeks to reconstruct a binary string s from a collection of traces, each of which is obtained by passing s through a deletion channel. It is known that $\exp(\tilde{O}(n^{1/5}))$ traces suffice to reconstruct any length-n string with high probability. We consider a variant of the trace reconstruction problem where the goal is to recover a "density map" that indicates the locations of each length-k substring throughout s. We show that $\epsilon^{-2} \cdot \mathbf{poly}(n)$ traces suffice to recover the density map with error at most ϵ. As a result, when restricted to a set of source strings whose minimum "density map distance" is at least $1/\mathbf{poly}(n)$, the trace reconstruction problem can be solved with polynomially many traces.**

1. Introduction

In the trace reconstruction problem, there is an unknown binary string $s \in \{0,1\}^n$, which we wish to reconstruct based on T subsequences (or traces) of s. Each trace is generated independently by passing the *source string* s through a deletion channel, which deletes each bit of s independently with probability p. The main question of interest is how many traces are needed to reconstruct s correctly. In the more general problem, traces can be affected by both insertions and deletions, but we restrict our discussion to the deletions case.

The problem of reconstructing a string from its subsequences and supersequences was first studied by Levenshtein [1–3], while the problem of reconstructing a string from randomly generated traces as described above was first studied by Batu et al. [4], motivated by problems in sequence alignment, phylogeny, and computational biology [5]. Most of the work on the trace reconstruction problem has focused on characterizing the minimum number of traces needed for reconstructing the source string s exactly. The most common formulation of the problem, known as *worst-case trace reconstruction* [6], requires the reconstruction algorithm to recover $s \in \{0,1\}^n$ exactly with high probability as $n \to \infty$ for *any* string $s \in \{0,1\}^n$. While this problem has received considerable attention, there is still a significant gap between upper and lower bounds on the number of traces needed. Currently, the best lower bound is $\tilde{\Omega}(n^{3/2})$, while the best upper bound is $\exp(\tilde{O}(n^{1/5}))$, both due to Chase [7, 8].

The exponential gap between the best known lower and upper bounds has motivated the formulation of several variants of the trace reconstruction problem where tighter bounds can hopefully be obtained. For example, in the *average-case trace reconstruction* problem, s is assumed to be drawn uniformly at random from all $\{0,1\}^n$ strings. In this case, it is known that only $T = \exp(O(\log^{1/3}(n)))$ traces are sufficient [9]. An *approximate trace reconstruction* problem, where a fraction of the recovered bits is allowed to be incorrect, has also been formulated [10], and the problem of finding the maximum likelihood sequence s from a small number of traces (possibly insufficient for exact reconstruction) has been recently studied [11]. We can also consider a more modest goal than the reconstruction of the source sequence s itself. One example that is particularly relevant to our discussion is the reconstruction of the k-subword deck of s [12, 13].

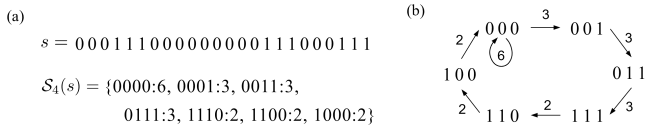


Fig. 1. (a) Example of a source binary string s and its k-subword deck (or k-spectrum) $\mathcal{S}_k(s)$, for $k = 4$. (b) Given $\mathcal{S}_4(s)$ in (a), one can build a de Bruijn graph where the elements in $\mathcal{S}_4(s)$ correspond to edges (with multiplicities) and the nodes correspond to 3-mers. Notice that s corresponds to an Eulerian path on the de Bruijn graph, but such a path is not unique; for example, $s' = 000111000111000000000111$ corresponds to another Eulerian path.

The k-subword deck of a binary sequence $s \in \{0,1\}^n$ is the multiset of all length-k substrings, i.e., $\{s[i : i+k-1] : i = 1, \ldots, n-k+1\}$. Equivalently, the k-subword deck can be defined by the counts $N_{s,x}$ of the number of times x appears in s as a substring:

$$\mathcal{S}_k(s) = [N_{s,x} : x \in \{0,1\}^k]. \tag{1}$$

As shown in [13], for $k = O(\log n)$, the k-subword deck $\mathcal{S}_k(s)$ can be recovered with $\text{poly}(n)$ traces. The k-subword deck of a sequence is an important object in bioinformatics, with applications in error correction [14, 15], sequence assembly [16, 17], and genomic complexity analysis [18, 19]. In these contexts, the k-subword deck $\mathcal{S}_k(s)$ is often referred to as the k-spectrum, and each length-k substring is called a k-mer. Intuitively, as long as k is large enough, the k-subword deck can uniquely determine the source sequence s. In fact, a classical result by Ukkonen [20] provides a necessary and sufficient condition for $\mathcal{S}_k(s)$ to uniquely determine s based on the length of the "interleaved repeats" in s [21]. In particular, if there are no repeats of length $k-1$ in s, one can reconstruct s from $\mathcal{S}_k(s)$ by simply merging k-mers with a prefix-suffix match of length $k-1$. More generally, given $\mathcal{S}_k(s)$, one can build the *de Bruijn graph*, where nodes correspond to $(k-1)$-mers and edges correspond to k-mers, and s is guaranteed to be an *Eulerian path* in the graph [16, 22] (see Figure 1).

While the k-subword deck is a natural intermediate goal towards the reconstruction of s (and can be recovered with only $\text{poly}(n)$ traces), it does not capture all the informa-

tion present in the traces. For example, the k-subword deck $\mathcal{S}_k(s)$ in Figure 1 also admits the reconstruction $s' = 000111000111000000000111$, even though s' should be easy to distinguish from s based on traces (by estimating the length of the second and third runs of zeros). Motivated by this shortcoming of the k-subword deck, we propose the idea of a k-mer density map, as a kind of localized k-subword deck where, in addition to knowing the number of times a given k-mer appears in s, we have some information about where it occurs.

For a k-mer $x \in \{0,1\}^k$, let $\mathcal{I}_{s,x} \in \{0,1\}^{n-k+1}$ be the indicator vector of the occurrences of x in s; i.e., $\mathcal{I}_{s,x}[j] = \mathbb{1}\{s[j : j+k-1] = x\}$, as illustrated in Figure 2. Notice that recovering the k-subword deck can be seen as recovering $\sum_j \mathcal{I}_{s,x}[j]$ for each $x \in \{0,1\}^k$. Also notice that recovering s is equivalent to recovering $\mathcal{I}_{s,x}$ for all $x \in \{0,1\}^k$. A k-mer density map can be obtained by computing

$$K_{s,x}[i] = \sum_{j=1}^{n-k+1} h(i,j)\mathcal{I}_{s,x}[j] \tag{2}$$

for $i \in \{1, ..., n-k+1\}$ and some "smoothing kernel" $h(i,j)$, as illustrated in Figure 2. Intuitively, for a given x, $K_{s,x}$ gives a coarse indication of the occurrences of x in s. Moreover, if h is such that $\sum_i h(i,j) = 1$ for each j, it holds that

$$\sum_i K_{s,x}[i] = \sum_j \mathcal{I}_{s,x}[j] \sum_i h(i,j) = \sum_j \mathcal{I}_{s,x}[j],$$

which means that the k-subword deck $\mathcal{S}_k(s)$ is a function of $K_{s,x}$, and the density map $K_{s,x}$ can be thought of as a generalization of the k-subword deck that provides information about k-mer location.

We will focus on a specific choice of $h(i,j)$ that will render $K_{s,x}$ easier to estimate from the traces. We will let $h(i,j)$ be the probability that a binomial random variable with $j-1$ trials and probability parameter $1-p$ is equal to $i-1$; i.e., $h(i,j) = \binom{j-1}{i-1}(1-p)^{i-1}p^{j-i}$. This is also the probability that the jth bit of s (if not deleted) ends up as the ith bit of a trace. Hence we have

$$K_{s,x}[i] = \sum_{j=1}^{n-k+1} \binom{j-1}{i-1}(1-p)^{i-1}p^{j-i}\mathcal{I}_{s,x}[j]. \tag{3}$$

for $i \in \{1, \ldots, n-k+1\}$. Notice that the maximum value of $h(i,j)$ for a fixed j occurs when $i \approx j(1-p)$ so the kernel $h(\cdot, j)$ has its peak shifted to the left and $K_{s,x}$ is a density map of occurrences of x in s shifted to the left. Operationally, $(1-p)^k K_{s,x}[i]$ is the probability that a fully preserved copy of x in s appears in position i on a given trace of s.

We define the k-mer density map of s as $K_s = [K_{s,x} : x \in \{0,1\}^k]$ (the concatenation of all vectors $K_{s,x}$). If the k-mer density map K_s is known *exactly*, s can also be recovered exactly. This can be seen by noticing the invertibility of the upper-triangular matrix F that transforms the binary vector $\mathcal{I}_{s,x}$ into the vector $K_{s,x}$ (for a fixed x). The matrix F is upper triangular with non-zero entries on the main diagonal, which makes it invertible. While invertible, F is ill-conditioned since

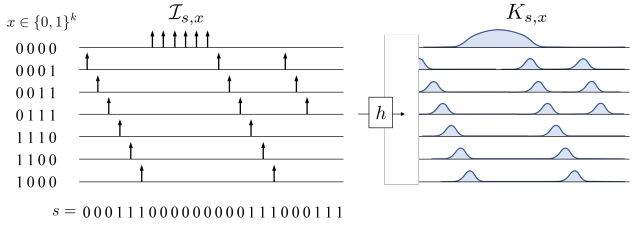


Fig. 2. For each $x \in \{0,1\}^k$, $\mathcal{I}_{s,x}$ indicates the occurrences of x in s. The density map $K_{s,x}$ can be obtained via $K_{s,x}[i] = \sum_{j=1}^{n-k+1} h(i,j)\mathcal{I}_{s,x}[j]$.

some of the entries on the diagonal are close to 0, making the transformation from $K_{s,x}$ to $\mathcal{I}_{s,x}$ sensitive to noise in $K_{s,x}$. To see this, notice that the condition number is at least $(1-p)^{k-n}$, which is exponentially large in n (this holds because $F_{1,1} = 1$ and $F_{n-k+1,\, n-k+1} = (1-p)^{n-k}$).

We present an algorithm that, given T traces, constructs an estimate $\hat{K}_s$ for the k-mer density map. Our main result establishes that we can achieve estimation error

$$\|\hat{K}_s - K_s\|_\infty = \max_{x,i} |\hat{K}_{s,x}[i] - K_{s,x}[i]| < \epsilon$$

using $T = \epsilon^{-2} \cdot \text{poly}(n)$ traces. Hence, the density map K_s can be estimated with maximum error $\epsilon_n = 1/g(n)$ for $g(n) \in \text{poly}(n)$ using polynomially many traces. In particular, given a set of candidate source strings $\mathcal{A} \subset \{0,1\}^n$ such that, for any $s, s' \in \mathcal{A}$,

$$\|K_s - K_{s'}\|_\infty \geq 2\epsilon,$$

the true source sequence $s \in \mathcal{A}$ can be recovered with $\epsilon^{-2} \cdot \text{poly}(n)$ traces. This adds to the existing literature on classes of strings recoverable/distinguishable with polynomially many traces [23–25].

Since $\mathcal{I}_{s,x}$ and $K_{s,x}$ are related through an invertible (albeit ill-conditioned) linear transformation $K_{s,x} = F\mathcal{I}_{s,x}$, the approximate recovery of the k-mer density map $\hat{K}_{s,x}$ suggests natural reconstruction algorithms for $\mathcal{I}_{s,x}$, e.g., based on a regularized least squares problem

$$\min_{\hat{\mathcal{I}}_{s,x}} \|\hat{K}_{s,x} - F\hat{\mathcal{I}}_{s,x}\|_2^2 + \delta\|\hat{\mathcal{I}}_{s,x}\|_2^2,$$

which is a convex program if $\hat{\mathcal{I}}_{s,x}$ is allowed to be real-valued. The solution $\hat{\mathcal{I}}_{s,x}$ can then be converted into a reconstructed string $\hat{s} \in \{0,1\}^n$ through a majority voting across candidate k-mers for each position. Hence, in contrast to much of the theoretical literature on the trace reconstruction problem, the k-mer density map leads to new reconstruction approaches.

Our main result relies on a nontrivial estimator for $K_{s,x}$ that simultaneously uses count information for all binary strings y that are supersequences of x. The estimator is obtained by first deriving a recursive formula for $K_{s,x}$, then applying a known result in the combinatorics of strings on the expansion of the recursive formula to obtain a non-recursive formula. An application of McDiarmid's inequality is then used to prove the estimator is successful with high probability. To the best of our knowledge, these techniques have not appeared in the

trace reconstruction literature, where most recent results have been based on complex analysis [6–9, 26]. Our techniques also lead to an improvement on a previously known upper bound [13] on the number of traces needed for reconstructing the k-subword deck of s for $p < 0.5$. A longer version of this paper [27] provides omitted proofs.

A. Notation

Strings in this paper are binary and indexed starting from 1. If the index i is negative, $x[i]$ is the $(-i)$th element starting from the right end of x. For example, if $s = 1001$, then $s[1] = 1$, $s[2] = 0$, $s[-1] = 1$, and $s[-2] = 0$. Let $s \in \{0,1\}^n$ be the length-n string we are trying to recover. The string s will be called the *source string*. A *trace* of s is denoted by $\tilde{S}$, and is generated by deleting each bit of s independently with probability p. Let $q = 1-p$ be the probability a bit is retained.

For a given string x, we let $|x|$ denote the length of x. For a string a and integer r, a^r denotes the string formed by concatenating r copies of a. 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 contrast to a *substring* of x, which is a string that appears in x. We let $x[i,j] = (x[i], x[i+1], \ldots, x[j])$ be the substring of x the begins at position i and ends at position j. For example, if $s = 10101$, then $s[1:3] = 101$ and $s[2:-2] = 010$. For two stings x and y, the number of ways we can make $|y| - |x|$ deletions on y to form x is denoted by $\binom{y}{x}$ [11, 28]. This is a generalization of the binomial coefficient for strings. Observe that $\binom{y}{x} \le \binom{|y|}{|x|}$.

2. Main Results

Let $s \in \{0,1\}^n$ denote the source string and $x \in \{0,1\}^k$ denote the target k-mer, whose density $K_{s,x}$ we wish to estimate. To simplify the notation, we fix a constant $c > 0$ and define

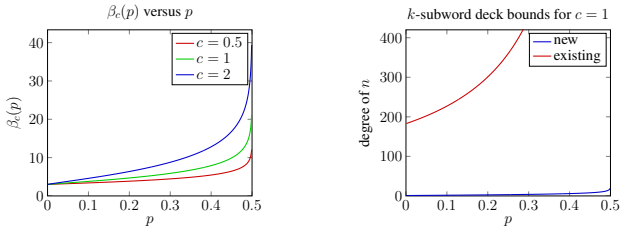
$$f_c(n) = \frac{\left(1 + 2n^{\alpha_c(p)}\right)^2}{2n^{2c\log(q)-1}} \quad \text{and}$$

$$\alpha_c(p) = 1 + c\log\left(\frac{q}{p}\right) + \frac{cH(1-\frac{p}{q}) + c\log\left(\frac{p}{q}\right)}{1-\frac{p}{q}} \tag{4}$$

where $H(\cdot)$ is the binary entropy function. The function $f_c(n)$ can be upper bounded by a polynomial of degree $\beta_c(p) = 2\alpha_c(p) - 2c\log(1-p) + 1$, which can be numerically computed (see Figure 3(a)). Letting $\delta \in (0,1)$, and $\{\epsilon_n\}$ be a positive-valued sequence, the following is proved in Section 3.

Theorem 1. *Suppose $p < 0.5$ and $k = c\log n$. Given $\log\left(\frac{2}{\delta}\right) \cdot \epsilon_n^{-2} \cdot f_c(n)$ traces, an estimator $\hat{K}_{s,x}[i]$ for the ith entry of $K_{s,x}$ can be constructed so that $|\hat{K}_{s,x}[i] - K_{s,x}[i]| < \epsilon_n$ with probability $1-\delta$. Moreover, given $\log\left(\frac{2n^{1+c\log 2}}{\delta}\right) \cdot \epsilon_n^{-2} \cdot f_c(n)$ traces, an estimator for the entire density map $\hat{K}_s$ can be constructed so that $\|\hat{K}_s - K_s\|_\infty < \epsilon_n$ with probability $1-\delta$.*

In particular, Theorem 1 implies that for $p < 0.5$ and $\epsilon_n = 1/g(n)$ where $g(n) \in \text{poly}(n)$, all entries of the $(c\log n)$-mer density map $K_{s,x}$ can be estimated with error at most ϵ_n using $\text{poly}(n)$ traces (and $\text{poly}(n)$ time as discussed in Section 3). Theorem 1 also implies the following result on the trace reconstruction problem restricted to a set of binary strings with a bounded minimum density map distance.



(a) (b)

Fig. 3. (a) Plot of $\beta_c(p)$ for various c. Observe that as p increases, the algorithm requires more traces to achieve the same level of performance. Similarly, as c increases, more traces are needed. For all values of c, the limit of $\beta_c(p)$ as $p \to 0$ is equal to 3. (b) Plot of exponent in (6) versus plot of exponent in (7) for $c = 1$. Observe that our bound is significantly tighter than the existing bound.

Corollary 1. *Suppose $p < 0.5$ and $k = c\log n$, and let $\mathcal{A} \subset \{0,1\}^n$ be such that, for any $s, s' \in \mathcal{A}$,*

$$\|K_s - K_{s'}\|_\infty \ge 2\epsilon_n.$$

Given $\log\left(\frac{2n^{1+c\log 2}}{\delta}\right) \cdot \epsilon_n^{-2} \cdot f_c(n)$ traces from some source string $s \in \mathcal{A}$, s can be correctly identified with probability $1-\delta$.

Consequently, the trace reconstruction problem restricted to a set of binary strings with minimum density map distance $1/g(n)$ where $g(n) \in \text{poly}(n)$ can be solved with $\text{poly}(n)$ traces. We have not been able to find a pair of strings s and s' so that $\|K_s - K_{s'}\|_\infty = o(1/g(n))$ for any $g(n) \in \text{poly}(n)$, and to the best of our knowledge, no such example of s and s' is known.

Recall that, from [13], one can recover the $(c\log n)$-subword deck using $\text{poly}(n)$ traces with high probability. A pair of strings s, s' can be distinguished based on their $(c\log n)$-subword deck alone if and only if their $(c\log n)$-subword decks are distinct, which is equivalent to requiring

$$|N_{s,x} - N_{s',x}| = |\|K_{s,x}\|_1 - \|K_{s',x}\|_1| \ge 1$$

for some $x \in \{0,1\}^{c\log n}$, since $N_{s,x} = \|K_{s,x}\|_1$. In contrast, our main result implies that as long as $|\|K_{s,x}\|_1 - \|K_{s',x}\|_1| \ge 1/\text{poly}(n)$ for some x, s and s' can be distinguished with $\text{poly}(n)$ traces, since due to the equivalence of ℓ_∞ and ℓ_1 norms and the reverse triangle inequality,

$$\|K_{s,x} - K_{s',x}\|_\infty \ge \frac{1}{n}|\|K_{s,x}\|_1 - \|K_{s',x}\|_1|. \tag{5}$$

This further establishes the k-mer density map as a generalization of the k-subword deck.

A special case of Corollary 1 with an explicit condition is given below and proved in the longer version [27].

Corollary 2. *Suppose $p < 0.5$ and $k = c\log n$. If the strings s, s' are such that $x \in \{0,1\}^k$ begins at position i*

in s and x does not appear in s' at an index in the range $[i-f(n), i+f(n)]$ for $f(n)=\Omega(n^a)$ where $a>0.5$, then s can be distinguished from s' with high probability using poly(n) traces.

Corollaries 1 and 2 are an addition to the literature on conditions for distinguishing strings from traces, which includes the facts that strings at constant Hamming distance can be distinguished using poly(n) traces [23, 24], and strings at constant Levenshtein distance can be distinguished using poly(n) traces [25]. Observe that there are string pairs that our results immediately prove are poly-trace distinguishable that the results of [13, 23–25] do not. Consider the string pair $s=0^m1^{\log(m)}0^{1.1m}$ and $s'=0^{1.1m}1^{\log(m)}0^m$ where $n=2.1m+\log(m)$. Observe that s, s' do not have constant Hamming or Levenshtein distance, and have the same $c\log(n)$ spectrum for any constant c, so previous results [13, 23–25] do not apply.

Improved upper bound for k-subword deck reconstruction: Using our proof technique for estimating $K_{s,x}$, we also give a novel proof that the $(c\log n)$-subword deck can be reconstructed using poly(n) traces for $p<0.5$, which yields an improved upper bound on the required number of traces compared to the analysis of the algorithm for $p<0.5$ in [13].

Theorem 2. *For $p<0.5$, we can reconstruct the $(c\log n)$-subword deck of any source string $s\in\{0,1\}^n$ from*

$$\tilde{O}\left(n^{1+c\left(\frac{qH(1-p/q)+p\log(p/q)}{1/2-p}+2\log\left(\frac{1}{q}\right)\right)}\right) \tag{6}$$

traces in poly(n) time with high probability.

In contrast, the analysis in [13] proves that

$$\tilde{O}\left(n^{4+12c\left(\frac{e^2}{1/2-p}\right)+c\log(4)}\right) \tag{7}$$

traces are sufficient for this task.

For $p<0.5$, the exponent in (6) is strictly less than that in (7) as shown in the longer version of this paper [27]. This shows that asymptotically, our upper bound is tighter for any c and any $p<0.5$. See Figure 3(b) for a plot showing the comparison. In particular, for c close to zero, (6) is close to linear in n, nearly matching the following lower bound that we prove in [27].

Theorem 3. *For deletion probability p and any source string $s\in\{0,1\}^n$, we have that $\Omega(npq)$ traces are necessary for recovering the $g(n)$-subword deck for any function g such that $g(n)\geq 2$.*

We did not compare our upper bound on trace complexity to the analysis of the algorithm in [13] that reconstructs the $(\log n)$-subword deck using poly(n) traces for $p<1$ because an explicit upper bound for this algorithm has not appeared in the literature to the best of our knowledge. In [27], we also prove that the initial algorithm for k-subword deck reconstruction presented in [13] only needs poly(n) traces for reconstructing the $(\log n)$-subword deck of a string for $p<0.5$, which was previously unknown.

3. An Estimator for the k-mer Density Map

In this section, we describe our estimator for the k-mer density map and prove Theorem 1. We first introduce some additional notation. For strings x, y, we let $\binom{y}{x}' = \binom{y[2:-2]}{x[2:-2]}$. For a string x, let $Y_i(x)$ be the set of length-i supersequences of x that have the same first and last bit of x. For example, if $x=101$, then $Y_4(x)=\{1011, 1101, 1001\}$.

For source string s and k-mer x, let $P_{s,x}[i]=\Pr(\tilde{S}[i:i+|x|-1]=x)$, i.e., the probability that x appears at position i in a trace $\tilde{S}$ of s. Notice that it is straightforward to estimate $P_{s,x}[i]$ from the set of traces as $\hat{P}_{s,x}[i]=\frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\{\tilde{S}_t[i:i+|x|-1]=x\}$. Recall that the entry in the k-mer density map K_s corresponding to the substring x at index i of s is defined as

$$K_{s,x}[i]=\sum_{j=i}^{n}\binom{j-1}{i-1}q^{i-1}p^{j-i}\mathbb{1}\{s[j:j+\ell-1]=x\}.$$

In order to estimate $K_{s,x}[i]$, we first write it in terms of $P_{s,x}[i]$ and $P_{s,y}[i]$ for all y of length greater than k. This will then allow us to estimate $K_{s,x}[i]$ using the estimates of $P_{s,x}[i]$ and $P_{s,y}[i]$, which can be obtained directly from the set of traces.

Lemma 1. *For source string s and k-mer x, we have that $K_{s,x}[i]$ is given by*

$$\frac{1}{q^k}\left(P_{s,x}[i]-\sum_{\ell=k+1}^{n}\sum_{y\in Y_\ell(x)}(-1)^{|y|-|x|+1}P_{s,y}[i]\binom{y}{x}'\left(\frac{p}{q}\right)^{\ell-k}\right).$$

Proof: We begin by deriving a recursive formula for $K_{s,x}[i]$. We first notice that

$$\begin{aligned}P_{s,x}[i]&=\sum_{\ell=k}^{n}\sum_{y\in Y_\ell(x)}\binom{y}{x}'p^{\ell-k}q^k\\&\quad\times\sum_{j=i}^{n}\binom{j-1}{i-1}q^{i-1}p^{j-i}\mathbb{1}\{s[j:j+\ell-1]=y\}\\&=\sum_{\ell=k}^{n}\sum_{y\in Y_\ell(x)}\binom{y}{x}'p^{\ell-k}q^kK_{s,y}[i].\end{aligned} \tag{8}$$

This follows because, in order for x to appear at position i in a trace, a superstring $y\in Y_\ell(x)$ must appear at position $j\geq i$ in s, $\binom{y}{x}'$ bits from y must be deleted, and $\binom{j-1}{i-1}$ bits in front of y must be deleted. Notice that $\binom{y}{x}'p^{\ell-k}q^k$ is the probability that a copy of y in s becomes x in $\tilde{S}$, and $K_{s,y}[i]$ is the probability that the beginning of a copy of y in s is shifted to position i in $\tilde{S}$.

Notice that, for $\ell=k$, the only term in the summation in (8) is $q^kK_{s,x}[i]$. This allows us to rewrite (8) as

$$\begin{aligned}K_{s,x}[i]&=\frac{1}{q^k}\left(P_{s,x}[i]-\sum_{\ell=k+1}^{n}\sum_{y\in Y_\ell(x)}\binom{y}{x}'p^{\ell-k}q^kK_{s,y}[i]\right)\\&=\frac{1}{q^k}\left(P_{s,x}[i]-\sum_{\ell=k+1}^{n}\sum_{y\in Y_\ell(x)}q^\ell\binom{y}{x}'K_{s,y}[i]\left(\frac{p}{q}\right)^{\ell-k}\right).\end{aligned} \tag{9}$$

By recursively applying (9) into itself, we write $K_{s,x}[i]$ in terms of $P_{s,x}[i]$ terms. This yields

$$K_{s,x}[i] = \frac{1}{q^k}\left(P_{s,x}[i] - \sum_{\ell=k+1}^{n}\sum_{y\in Y_\ell(x)} P_{s,y}[i] a_{s,x,y}\left(\frac{p}{q}\right)^{\ell-k}\right) \quad (10)$$

where $a_{s,x,y} \in \mathbb{Z}$ is a constant that depends on s, x, y. Observe that $a_{s,x,y}$ obeys the following recursion: for $y \in Y_\ell(x)$, we have that

$$a_{s,x,y} = \binom{y}{x}' - \sum_{k+1\le j<\ell}\sum_{z\in Y_j(x)} a_{s,x,z}\binom{y}{z}'. \quad (11)$$

This is because as we expand (9) one step at a time to eventually obtain (10), we observe that every time we obtain a new term involving $P_{s,y}[i]$ in the expansion with coefficient $c_y\left(\frac{p}{q}\right)^{|y|-k}$, in the next step of the expansion we obtain a term involving $P_{s,z}[i]$ with coefficient $-c_y\binom{z}{y}'\left(\frac{p}{q}\right)^{|z|-k}$ for every $z \in \cup_{\ell=|y|+1}^{n} Y_\ell(y)$. One step of the expansion is shown in [27] to illuminate this argument. We proceed to prove via induction on $|y|-|x|$ that for any s, x, y, we have that

$$a_{s,x,y} = (-1)^{|y|-|x|+1}\binom{y}{x}'. \quad (12)$$

We will use the following lemma, which appears as Corollary (6.3.9) in [28].

Lemma 2. *For any two strings f, g over an alphabet Σ,*

$$\sum_{h}(-1)^{|g|+|h|}\binom{f}{h}\binom{h}{g} = \delta_{f,g} \quad (13)$$

where $\delta_{f,g} = 0$ if $f \neq g$, and $\delta_{f,g} = 1$ if $f = g$.

If $|y|-|x| = 1$, (11) implies that $a_{s,x,y} = \binom{y}{x}'$, and (12) clearly holds. Suppose (12) holds for $|y|-|x| < m$. Then if $|y|-|x| = m$, we have that

$$\begin{aligned}
a_{s,x,y} &= \binom{y}{x}' - \sum_{k<j<\ell}\sum_{z\in Y_j(x),} a_{s,x,z}\binom{y}{z}' \\
&= \binom{y}{x}' - \sum_{k<j<\ell}\sum_{z\in Y_j(x),} (-1)^{|z|-|x|+1}\binom{z}{x}'\binom{y}{z}' \\
&= \binom{y}{x}' + \sum_{k<j<\ell}\sum_{z\in Y_j(x),} (-1)^{|z|+|x|}\binom{z}{x}'\binom{y}{z}' \\
&= \binom{y}{x}' + \left(\sum_{k\le j\le\ell}\sum_{z\in Y_j(x),} (-1)^{|z|+|x|}\binom{z}{x}'\binom{y}{z}'\right) \\
&\quad - \binom{x}{x}'\binom{y}{x}' - (-1)^{|y|+|x|}\binom{y}{x}'\binom{y}{y}' \\
&= \binom{y}{x}' - \binom{y}{x}' - (-1)^{|y|+|x|}\binom{y}{x}' \quad (14) \\
&= (-1)^{|y|-|x|+1}\binom{y}{x}'
\end{aligned}$$

where (14) follows from Lemma 2. By plugging in this formula for $a_{s,x,y}$ into (10), we obtain the result. ∎

Lemma 1 allows us to obtain an unbiased estimator $\hat{K}_{s,x}[i]$ for $K_{s,x}[i]$ given by

$$\frac{1}{q^k}\left(\hat{P}_{s,x}[i] - \sum_{\ell=k+1}^{n}\sum_{y\in Y_\ell(x)} (-1)^{|y|-|x|+1}\hat{P}_{s,y}[i]\binom{y}{x}'\left(\frac{p}{q}\right)^{\ell-k}\right) \quad (15)$$

where $\hat{P}_{s,x}[i] = \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}\{\tilde{S}_t[i : i+|x|-1] = x\}$.

One way to analyze the performance of our estimator $\hat{K}_{s,x}[i]$ would be to apply a standard concentration inequality such as the Chernoff bound to each of the terms $\hat{P}_{s,y}[i]$ and use that to bound the error of $\hat{K}_{s,x}[i]$. However, this yields a sub-optimal analysis as we do not need to guarantee the accuracy of each $\hat{P}_{s,y}[i]$ term. Directly analyzing the accuracy of $\hat{K}_{s,x}[i]$ is more subtle, as $\hat{K}_{s,x}[i]$ is not a sum of independent random variables. To that end, we apply McDiarmid's inequality to analyze the deviation of $\hat{K}_{s,x}[i]$ from $K_{s,x}[i]$ directly.

Lemma 3. *For source string s, a k-mer x with $|x| = c\log n$, and a set of T traces, we have*

$$\begin{aligned}
&\Pr\left(|\hat{K}_{s,x}[i] - K_{s,x}[i]| \ge \epsilon\right) \\
&\le 2\exp\left(-\frac{2T\epsilon^2}{n\left(\frac{1}{n^{c\log(q)}}\left(1+2n^{\alpha_c(p)}\right)\right)^2}\right).
\end{aligned}$$

Setting $\delta = \Pr\left(|\hat{K}_{s,x}[i] - K_{s,x}[i]| \ge \epsilon\right)$, we conclude that

$$T = \log(2/\delta)\frac{n}{2\epsilon^2}\left(\frac{1}{n^{c\log(q)}}\left(1+2n^{\alpha_c(p)}\right)\right)^2 \quad (16)$$

traces suffices for recovering $K_{s,x}[i]$ with error less than ϵ with probability at least $1-\delta$.

We can compute the estimator in (15) efficiently by iterating through all T traces, and for each $\ell \in \{k+1, ..., n\}$, constructing a linked list that stores each length-ℓ string that starts at position i in at least one trace, along with the number of times it is seen in the set of traces. After generating this set of linked lists, for each string y present in a linked list, we compute the corresponding term in (15), which requires the computation of $\binom{y}{x}'$. The entire process requires $O(T^2n^3)$ time since we iterate through $O(n)$ strings that start at position i in each of the T traces, and for each observed string y, we must check if there is already a node corresponding to y in a linked list using $O(T)$ time, and compute $\binom{y}{x}'$ in $O(n^2)$ time using dynamic programming (see Proposition 6.3.2 in [28]).

In conjunction with our trace complexity analysis, the above approach for computing the estimator implies that for x of length $c\log(n)$, we can estimate $K_{s,x}[i]$ with error at most $1/\text{poly}(n)$ with high probability in $\text{poly}(n)$ time, and can therefore estimate the $c\log(n)$-density map with maximum error $1/\text{poly}(n)$ with high probability in $\text{poly}(n)$ time.

ACKNOWLEDGMENTS

The work of K.M. and I.S. was supported by the NSF under grants CCF-2007597 and CCF-2046991.

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] V. I. Levenshtein, "Efficient reconstruction of sequences from their subsequences or supersequences," *Journal of Combinatorial Theory, Series A*, vol. 93, no. 2, pp. 310–332, 2001.

[4] T. Batu, S. Kannan, S. Khanna, and A. Mcgregor, "Reconstructing strings from random traces," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, vol. 15, 01 2004.

[5] V. Bhardwaj, P. A. Pevzner, C. Rashtchian, and Y. Safonova, "Trace reconstruction problems in computational biology," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3295–3314, 2020.

[6] 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.

[7] 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.

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

[9] N. Holden, R. Pemantle, and Y. Peres, "Subpolynomial trace reconstruction for random strings and arbitrary deletion probability," in *Conference On Learning Theory*, pp. 1799–1840, PMLR, 2018.

[10] S. Davies, M. Z. Rácz, B. G. Schiffer, and C. Rashtchian, "Approximate trace reconstruction: Algorithms," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2525–2530, IEEE, 2021.

[11] 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.

[12] 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.

[13] X. Chen, A. De, C. H. Lee, R. A. Servedio, and S. Sinha, "Polynomial-time trace reconstruction in the smoothed complexity model," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 54–73, SIAM, 2021.

[14] Y. Liu, J. Schröder, and B. Schmidt, "Musket: a multistage k-mer spectrum-based error corrector for illumina sequence data," *Bioinformatics*, vol. 29, no. 3, pp. 308–315, 2013.

[15] I. Shomorony, T. A. Courtade, and D. Tse, "Fundamental limits of genome assembly under an adversarial erasure model," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 2, pp. 199–208, 2016.

[16] P. A. Pevzner, H. Tang, and M. S. Waterman, "An eulerian path approach to dna fragment assembly," *Proceedings of the National Academy of Sciences*, vol. 98, no. 17, pp. 9748–9753, 2001.

[17] S. Marcovich and E. Yaakobi, "Reconstruction of strings from their substrings spectrum," *IEEE Transactions on Information Theory*, vol. 67, no. 7, pp. 4369–4384, 2021.

[18] B. Chor, D. Horn, N. Goldman, Y. Levy, and T. Massingham, "Genomic dna k-mer spectra: models and modalities," *Genome biology*, vol. 10, no. 10, pp. 1–10, 2009.

[19] P. A. Noble *et al.*, "Tetranucleotide frequencies in microbial genomes," *Electrophoresis*, vol. 19, Apr 1998.

[20] E. Ukkonen, "Approximate string-matching with q-grams and maximal matches," *Theoretical computer science*, vol. 92, no. 1, pp. 191–211, 1992.

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

[22] I. Shomorony, S. H. Kim, T. A. Courtade, and D. N. Tse, "Information-optimal genome assembly via sparse read-overlap graphs," *Bioinformatics*, vol. 32, no. 17, pp. i494–i502, 2016.

[23] A. Mcgregor, E. Price, and S. Vorotnikova, "Trace reconstruction revisited," in *ESA*, 2014.

[24] E. Grigorescu, M. Sudan, and M. Zhu, "Limitations of mean-based algorithms for trace reconstruction at small edit distance," *IEEE Transactions on Information Theory*, 2022.

[25] J. Sima and J. Bruck, "Trace reconstruction with bounded edit distance," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2519–2524, IEEE, 2021.

[26] 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*, STOC 2017, (New York, NY, USA), p. 1042–1046, Association for Computing Machinery, 2017.

[27] K. Mazooji and I. Shomorony, "Substring density estimation from traces," *arXiv preprint arXiv:2210.10917*, 2022.

[28] *Combinatorics on Words*. Cambridge Mathematical Library, Cambridge University Press, 2 ed., 1997.