Trace Reconstruction Revisited

Andrew McGregor[1]*, Eric Price[2], and Sofya Vorotnikova[1]

[1] University of Massachusetts Amherst
{mcgregor,svorotni}@cs.umass.edu
[2] IBM Almaden Research Center
ecprice@mit.edu

Abstract. The trace reconstruction problem is to reconstruct a string x of length n given m random subsequences where each subsequence is generated by deleting each character of x independently with probability p. Two natural questions are a) how large must m be as a function of n and p such that reconstruction is possible with high probability and b) how can this reconstruction be performed efficiently. Existing work considers the case when x is chosen uniformly at random and when x is arbitrary. In this paper, we relate the complexity of both cases; improve bounds by Holenstein et al. (SODA 2008) on the sufficient value of m in both cases; and present a significantly simpler analysis for some of the results proved by Viswanathan and Swaminathan (SODA 2008), Kannan and McGregor (ISIT 2005), and Batu et al. (SODA 2004). In particular, our work implies the first sub-polynomial upper bound (when the alphabet is $\operatorname{polylog} n$) and super-logarithmic lower bound on the number of traces required when x is random and p is constant.

1 Introduction

The basic trace reconstruction problem is to infer a string x of length n from m random subsequences $y^1, \ldots, y^m$ where each subsequence is generated by deleting each character of x independently with probability p. The random subsequences are referred to as *traces*. Two natural questions are a) how many traces (as a function of n and p) are required such that reconstruction is possible with high probability and b) how can this reconstruction be performed efficiently. Note that both questions are trivial if the entries of x were being substituted, rather than deleted. In that case, if $p < 1/2$ is constant and $m = O(\log n)$ then $x_i = \text{mode}(y_i^1, \ldots, y_i^m)$ with high probability. However, when there are deletions, there is no longer any clear way to align the subsequence and thereby decompose the problem into inferring each entry of x independently.

The original motivation for the problem was from computational biology where an active area of research is to reconstruct ancestral DNA sequences given the DNA sequences of the descendants. The above abstraction is a simplification of this problem in which we essentially restrict the possible mutations and assume the descendants are independent. The abstraction serves to both demonstrate

* Supported by NSF CAREER Award CCF-0953754.

why the original problem is hard and exposures our lack of good algorithmic techniques for even basic inference problems. For example, for $p = 1/3$, it is not at all obvious whether the minimal sufficient value m has a polylogarithmic, polynomial, or exponential dependence on n.

Previous Work. The problem was introduced by Batu et al. [1] where they considered both an "average" case when x is chosen uniformly at random (in this case the probability of successful reconstruction is over both the choice of x and the deletions) and the case when x is chosen arbitrarily. In the average case, it was shown that $m = O(\log n)$ is sufficient if $p = O(1/\log n)$. This result was then extended to also handle insertions and substitutions by Kannan and McGregor [4] and Viswanathan and Swaminathan [8]. For small constant deletion probability, Holenstein et al. [3] showed that poly(n) traces was sufficient. While this last result represented a major step forward, it leaves open the question whether a polynomial number of traces is actually necessary or whether a logarithmic number would suffice, as in the case when there was only substitutions. For reconstructing an arbitrary x, Batu et al. [1] showed that $O(n \operatorname{polylog} n)$ traces suffices if $p = O(1/\sqrt{n})$ and Holenstein et al. [3] showed that $\exp(\sqrt{n} \cdot \operatorname{polylog} n)$ traces suffices if p is any constant.

A separate line of work considers the related problem of determining the value k such that the *k-deck* of any x uniquely determines x. The k-deck of x is the number of times each string of length k appears as a subsequence of x. Given a sufficient number of traces of length greater than k, we can compute the k-deck and thereby determine x if k is large enough. Scott [7] proved that $k = O(\sqrt{n \log n})$ and Dudik and Schulman [2] showed that $k = \exp(\Omega(\log^{1/2} n))$. We will make use of the first of these results in the last section.

Our Results. Our main results in the average case are that a) a sub-polynomial number of traces is sufficient if we consider a slightly larger alphabet and b) a super-logarithmic number of traces is necessary. In particular, if x is chosen uniformly from $[\sigma]^n$ where $\sigma = \Theta(\log n)$ and p is a small constant then

$$m = \exp(\sqrt{\log n} \cdot \operatorname{poly}(\log \log n))$$

traces are sufficient which contrasts with the bound $m = \exp(O(\log n))$ that was shown by Holenstein et al. for the binary case. We prove this result by establishing an almost tight relationship between the complexity in the average case to the complexity in the worst case. To do this, we first present a significantly simpler proof of the results of Batu et al. [1] and Viswanathan and Swaminathan [8]. It is then possible to extend the alternative approach to be robust to deletions that occur with constant probability.

In the case of arbitrary strings (binary or otherwise), we show that $m = \exp(\sqrt{n} \cdot \operatorname{polylog} n)$ traces are sufficient for all $p \leq 1 - c/\sqrt{n/\log n}$ for some constant $c > 0$. This result improves upon the other result by Holenstein et al. The previous result showed that the same number of traces were sufficient when the traces are random subsequences of length $\Theta(n)$. The new result shows that reconstruction is still possible even if the traces are only of length $\Theta(\sqrt{n \log n})$.

2 Preliminaries and Terminology

Given a string $x_1x_2\ldots x_n \in [\sigma]^n$, a *trace* generated with deletion probability p is a random subsequence of x, $y = x_{i_1}x_{i_2}x_{i_3}\ldots$ where $i_1 < i_2 < i_3 < \ldots$ and each $i \in [n]$ is present in the set $\{i_1, i_2, \ldots\}$ independently with probability $1-p$. It will sometimes be helpful to refer to the trace y as being *received* when x is *transmitted.* We are interested in whether it is possible to infer x with high probability from multiple independently traces $y^1, y^2, \ldots, y^m$.

We define $f(n, p, \sigma)$ to be the smallest value of m such that for any string $x \in [\sigma]^n$, m traces are sufficient to reconstruct x with high probability[3]. Define $g(n, p, \sigma)$ to be the smallest value of m such that for a *random* string $x \in_R [\sigma]^n$, m traces are sufficient to reconstruct x with high probability where the probability is taken over both the randomness of x and the generation of the traces. For example, existing results show that for small constant $c > 0$:

$$g(n, p, 2) = \begin{cases} O(\log n) & \text{if } p \le c/\log n \\ \operatorname{poly} n & \text{if } p \le c \end{cases}.$$

We present a simple proof of the first part of this result and then prove that $g(n, p, \sigma)$ is sub-polynomial for small constant values of p if $\sigma = \Omega(\log n)$. To prove this result we show that for sufficiently large σ, $f(\log n, p, \sigma) \approx g(n, p, \sigma)$. Lastly, we prove $f(n, p, 2) = \exp(\sqrt{n}\operatorname{polylog} n)$ for all $p \le 1 - O(1/\sqrt{n/\log n})$ whereas it was previously only known for constant p.

Note that any reconstruction algorithm for binary strings can be extended to a larger alphabet of size σ while increasing the number of traces by a factor of $O(\log \sigma)$. The following simple lemma includes the necessary details.

Lemma 1. *$f(n, p, \sigma) = O(\log \sigma) f(n, p, 2)$. If m traces suffice to reconstruct a random string in $\{0,1\}^n$ with probability $1-\delta$, then m traces also suffice to reconstruct a random string in $[\sigma]^n$ with probability $1 - O(\delta \log \sigma)$.*

Proof. Suppose there exists an algorithm for arbitrary binary sequence reconstruction that uses m traces and has failure probability at most δ. By repeating the algorithm $O(\log \sigma)$ times and taking the modal answer we may reduce the failure probability to $\delta / \binom{\sigma}{2}$ at the expense of increasing the number of traces by a factor $O(\log \sigma)$. We will use the resulting algorithm to reconstruct a sequence x from a larger alphabet as follows. For each pair $i, j \in [\sigma]$, if we delete all occurrences of other characters in the traces then we can reconstruct the subsequence $x^{i,j}$ of x consisting of i's and j's. By the union bound we can do this for all pairs with probability of failure at most δ. For the resulting subsequences it is possible to construct x, e.g., we can learn the position of the kth j in x by summing over i the number of occurrences of i's before the kth j in $x^{i,j}$.

The same approach works to prove the bound for random strings except that since the failure probability in this case is taken over both the randomness of the initial string and the traces, we can't first boost the probability of success. □

[3] That is, probability at least $1 - 1/\operatorname{poly}(n)$

Notation. We denote the Hamming distance between two strings u, v by $\Delta(u, v) = |\{i : u_i \neq v_i\}|$. We write $e \in_R S$ to denote that the element e is chosen uniformly at random from the set S. A t-substring of x is a string consisting of t consecutive characters of x. Given a substring w of a trace, we define the pre-image of w, to be the range of indices of x under consideration when w was generated, e.g.,

$$w = x_{i_j} x_{i_{j+1}} \dots x_{i_k} \qquad \text{and} \qquad I(w) = \{i_j, i_j + 1, i_j + 2, \dots, i_k\} \ .$$

We say substrings u, v of two different traces *overlap* if $I(u) \cap I(v) \neq \emptyset$. Lastly, we use the notation $x_{[a,b]}$ to denote the substring $x_a x_{a+1} \dots, x_b$. Let $\mathcal{B}_{n,p}$ denote the binomial distribution with n trials and probability p.

3 Average Case Reconstruction

In this section we assume that the original string x is chosen uniformly at random from the set $[\sigma]^n$. We first present a simpler approach to reconstruction when the deletion probability is $O(1/\log n)$. Previous approaches were generally based on determining the characters of x from left to right, e.g., trying to maintain pointers to corresponding characters in the different traces and using the majority to determine the next character of x. While the resulting algorithms were relatively straight-forward, the analysis was rather involved.

In contrast, our approach is based on finding all sufficiently-long substrings of x independently and the analysis for our approach is significantly shorter and intuitive. While this simplicity is appealing in its own right, it also allows us to generalize the algorithm in the following section and prove a new result in the case of constant deletion probability. We start with a simple lemma about random binary strings.

Lemma 2. *With high probability, every pair of t-substrings of a random sequence $x \in \{0,1\}^n$ differ in at least $t/3$ positions if $t > 94 \ln n$.*

Proof. Consider two arbitrary substrings $u = x_i \dots x_{i+t-2}$ and $v = x_j \dots x_{j+t-2}$. Let $z \in \{0,1\}^t$ be defined by $z_k = x_{i+k-1} \oplus x_{j+k-1}$. Note that $\Delta(u, v) = \sum_k z_k$ and that bits of z_i are fully independent. Hence, $E\left[\sum_k z_k\right] = t/2$ and by an application of the Chernoff bound, $\Pr\left[\sum_k z_k \leq t/3\right] \leq \exp(-t/24)$. Therefore, if $t > 94 \ln n$, then $\Pr\left[\Delta(u, v) \leq t/3\right] \leq 1/n^4$. Applying the union bound over all $\binom{n}{2}$ choices for substrings u and v establishes that the Hamming distance between all pairs is at least $t/3$ with probability at least $1 - 1/n^2$. □

3.1 Warmup: Inverse Logarithmic Deletion Probability

In this section we present a simple proof of the results by Batu et al. [1] when $p = O(1/\log n)$. For the rest of the section we let the deletion probability be $p \leq c_1/\log n$, number of traces be $m = c_2 \log n$ for constants $c_1, c_2 > 0$.

Basic Idea and Algorithm. The idea behind the approach is simple and intuitive. For $t = c_3 \log n$ where c_3 is some sufficiently large constant, the following statements hold with high probability:

1. The set of all t-substrings of a random string x, uniquely defines x.
2. w is a t-substring of x iff w is a t-substring of at least $3/4$ of the traces.

Therefore, it is sufficient to check each t-substring of each trace to see whether it appears in at least $3/4$ of all the traces. The next lemma establishes the first statement.

Lemma 3. *The set of t-substrings of $x \in_R [\sigma]^n$ uniquely define x whp.*

Proof. If all $(t-1)$-substrings of x are unique, then for a t-substring w starting at index i in x, there is a unique t-substrings starting at $i+1$. By repeating this process, we can recover the original string x. The fact that all $(t-1)$-substrings are unique with high probability follows from Lemma 2. □

The next two lemmas establish the if-and-only-if of the second bullet point.

Lemma 4. *Every $4t$-substring of x passes the test with high probability. In particular, none of the characters of any $4t$-substring are deleted in at least $3m/4$ of the traces.*

Proof. Let w be a $4t$-substring of x. Let F be the number of traces where w appears. The probability that w appears in a particular trace is at least $(1-p)^{4t} > 1 - 4pt > 7/8$ if $pt = c_1 c_3 < 1/32$. Hence, $E[F] > 7m/8$ and $\Pr[F \leq 3m/4] < e^{-m/168}$ by an application of the Chernoff bound. If $m > 2 \cdot 168 \ln n$, this probability is at most $1/n^2$. □

Lemma 5. *Any t-string that passes the test is a t-substring of x whp.*

Proof. We start with two simple claims that each hold with high probability:

1. *For any t-substrings w and v of different traces, if $w = v$ then w and v have overlapping pre-images.* This follows because the probability that two non-overlapping t-substrings are equal is $1/2^t$ by considering the randomness of x. There are less than $(mn)^2$ pairs of t-substrings and hence the claim doesn't hold with probability at most $(mn)^2/2^t \leq 1/n^2$ if $t > 4 \log n + 2 \log(c_2 \log n)$.
2. *The pre-image of any t-substring w of a trace has length at most $2t$.* The follows because the probability that more than half of the characters in a $2t$-substring of x are deleted is at most $\exp(-t/12)$ by an application of the Chernoff bound. Since there are at most mn^2 such sequences, the claim doesn't hold with probability at most $mn^2 \exp(-t/12) \leq 1/n^2$ if $t > 48 \ln n + 48 \ln(c_2 \log n)$.

Suppose w equals the substrings $w^1, w^2, \ldots, w^h$ in the other traces for $h \geq 3m/4 - 1$. It follows from the above claims that the pre-images of $w, w^1, w^2, \ldots, w^h$ are contained in a contiguous region of x of size $4t$. However, by Lemma 4 we know the corresponding substring of x was transmitted with deletions in at most $m/4$ of the traces. Therefore, at least $h - m/4 > 1$ of the substrings $w^1, w^2, \ldots, w^h$ correspond exactly to a t-strings of x. Hence w equals a substrings of x. □

Insertions, Deletions, and Substitutions. Viswanathan and Swaminathan [8] extended the above result to handle the case where, in addition to deletions, each character is substituted by a random character with probability α and random characters are inserted with probability q. Specifically, each character x_i is transformed independently as follows:

$$g(x_i) = \begin{cases} Sx_i \text{ with probability } 1-p-\alpha(1-p) \\ Sc \text{ with probability } \alpha(1-p) \\ S \text{ with probability } p \end{cases}$$

where $c \in_R [\sigma]$, $S \in_R [\sigma]^k$ and k is a random variable distributed as a Geometric random variable with parameter $1-q$. In particular,

$$\Pr[g(x_i) = x_i] \geq (1-p-\alpha(1-p))(1-q) \ ,$$

which is $1-\alpha-o(1)$ if $p, q = o(1)$. In this section we present a simple proof of Viswanathan and Swaminathan's result that $O(\log n)$ traces are sufficient for reconstruction if $p, q < c/\log n$ and $\alpha < c$ for some sufficiently small constant c.

Basic Idea and Algorithm. We extend the substring test as follows: w is a t-substring of x iff for some t-substring w' of a trace, there exists t-substrings $w_1, \ldots w_{3m/4}$ in different traces such that $\Delta(w', w_i) \leq 3\alpha t$ for all $i \in [3m/4]$ and

$$w = \text{average}(w_1, \ldots, w_{3m/4})$$

where *average* is taking the mode of each of the t character positions.

Lemma 6. *Every t-substring of x passes the test with high probability.*

Proof. Let w be an arbitrary t-substring of x and let $w' = g(w)$ be the resulting substring in some specific trace. In what follows we assume the constant c governing the deletion and insertion probabilities is sufficiently small. The probability no insertions or deletions occurred during the transmission of w is $(1-q-p+pq)^t \geq 6/7$ and by an application of the Chernoff bound, the number of substitutions is at most $3\alpha t/2$. Hence, by a further application of the Chernoff bound there are at least $5m/6$ traces that contain a t-substring whose Hamming distance is at most $3\alpha t/2$ from w. The Hamming distance between these traces is at most $3\alpha t$ by the triangle inequality. Lastly if these t-substrings are averaged character-wise then the resulting string equals w because with high probability each character of w is flipped in at most $1/3$ of the transmissions. □

Lemma 7. *Every t-string that passes the test is a t-substring of x whp.*

Proof. Suppose a trace contains a t-substring w' such that for some $h \geq 3m/4$, there exists $w_1, \ldots w_h$ in different traces such that $\Delta(w', w_i) \leq 3\alpha t < t/3$ for sufficiently small α. We infer that each w_i overlaps with w' since otherwise w_i and w' are random strings and will differ in at least $t/3$ places with high probability. Hence, each w_i comes from substring x' of x of length $4t$. When x'

was transmitted, it was transmitted without any insertions or deletions in at least of 9/10 of the traces with high probability. Hence, all but at most $m/10$ of the w_i resulted from transmission with no insertions or deletions. But appealing to Lemma 2 we deduce that these w_i actually correspond to the same t-substring of x; otherwise there would be a pair of different t-substrings of x that were sufficiently similar that after bits were flipped with only probability α then the strings would be closer than $6\alpha t$ apart. Hence, when averaging $w_1, \ldots w_h$ character-wise at most a $2\alpha + (m/10)/h \leq 2\alpha + 2/15 < 1/2$ fraction of characters will not be correct. Hence, the majority will be correct. □

3.2 Constant Deletion Probability

In this section we again restrict our attention to the deletion case but now consider p to be a small constant. In the previous two results, the crucial step was being able to identify t-substrings in different traces that were overlapping. Initially, it was sufficient to look for identical t-substrings but then we had to relax this to finding pairs of substrings that were close in Hamming distance. The main idea in this section is the observation that it is possible to find overlapping t-substrings by computing the length of the longest common subsequence between the substrings.

Lemma 8. *If $t = c \log n$ for some large constant $c > 0$, the following claims hold with high probability:*

- *For any two traces y and y' and any t-substring w in y, there exists a t-substring w' in y' such that $\mathrm{lcs}(w, w') \geq 0.99t$.*
- *For any non-overlapping t-substrings w and v in different traces $\mathrm{lcs}(w, v) < 0.99t$.*

Proof. For the first part of the lemma, note that the expected number of deletions during the transmission of a t-substring of x is pt and by an application of the Chernoff bound we may assume it is never larger than $2pt$ with high probability if t is sufficiently large multiple of $\log n$. Therefore, there are at least $(1-2p)t$ characters of some t-substring u of x in w. But any t-substring of y' whose pre-image covers u, will also have $(1-2p)t$ characters of u. Let w' be such a string. Then, $\mathrm{lcs}(w, w') \geq (1-4p)t \geq 0.99t$ for sufficiently small constant p.

To prove the second part of the lemma suppose w, v are non-overlapping t-substrings. Because x is random and w, v are non-overlapping, w, v are independent random strings. Therefore,

$$\Pr\left[\mathrm{lcs}(w, v) \geq 0.99t\right] < \binom{t}{0.99t}^2 1/2^{0.99t} < 2^{2tH(0.99)-0.99t} < 2^{-0.8t}$$

where $H(p) = -p \log p - (1-p)\log(1-p)$. The first inequality follows by considering the $\binom{t}{0.99t}$ subsequences of each segment that might be equal. □

It is likely that the constants in the above lemma can be improved. However, one of the main ingredients in the proof is determining the length of the longest common subsequence of two random strings. Determining even the expected length is a long standing open question (see, e.g., [5] and references therein).

Reduction to Short Sequence Reconstruction. To prove the constant deletion result, the strategy is to reduce the problem of reconstructing a random $x \in_R [\sigma]^n$ to reconstructing $O(n)$ arbitrary strings each of length $O(\log n)$. To do this, we will have to assume that x is chosen randomly from a larger alphabet $\sigma = \Theta(\log n)$. It will then follow that

$$g(n, p, \sigma) \leq f(O(\log n), p, \sigma) = \exp(\sqrt{\log n}\,\mathrm{poly}(\log\log n)) \ ,$$

by appealing to the bounds on the function f established in the next section. To establish this reduction we need the notion of a *useful character*.

Definition 1. *We say a character x_i from x is a* useful character *if:*

1. *The character was not deleted when generating the first trace.*
2. *$x_i \neq x_j$ for all $|i - j| \leq 8t$, i.e., x_i is locally unique.*

The goal is to identify the occurrence of useful characters in the traces and then determine with high probability which characters correspond to the same x_i. The next lemma establishes that the number of non-useful characters between two consecutive useful characters is $O(\log n)$. Since each useful character will occur in all but about a p fraction of the traces, there are roughly a $(1-p)^2$ fraction of traces that have any pair of consecutive useful characters. We then use the substrings of the traces between these useful characters to reconstruct the substring of x between the useful characters. We can then solve the sequence reconstruction problem on these substrings.

Note that because there are $O(n)$ substrings and each has length $O(\log n)$ we now need a reconstruction algorithm that works for all strings (rather than working for random strings with high probability). Note that the algorithm for reconstruction of arbitrary strings presented in Section 4 can be assumed to have exponentially small failure probability without any significant change in the number of traces required (i.e., repeating the algorithm $\mathrm{poly}(n)$ times to boost success probability is not a significant increase when the number of traces is already super-polynomial). This is important since we need the failure probability on length $O(\log n)$ instances to be $1/\mathrm{poly}(n)$ since there are $O(n/\log n)$ such instances.

Lemma 9. *With high probability, there exists a useful character in every r-substring of x if $r = 8t = 8c\log n$.*

Proof. Consider an arbitrary r-substring $x_{[i,i+r-1]}$. With high probability there exists more than $2r/3$ distinct characters in this substring if the alphabet is sufficiently large. Of these, at most $r/2$ can occur twice or more. Hence, there are at least $r/6$ characters that occur exactly once. Of these, $(1-p)r/6 > r/7$ occur in the first trace in expectation and hence the probability that none of them appear in the first trace is at most $p^{r/7} < 1/n^2$ for sufficiently small constant p. □

Algorithm. The algorithm for finding corresponding characters is as follows:

- For each character a in the first trace, consider the t-substring w_1 of the first trace centered at this character (or as close as possible in the case when is a is near the start of end of the trace).
- *Find Overlapping Substrings:* Identify t-substrings of the other traces $w_2, \ldots, w_m$ such that each satisfies $\mathrm{lcs}(w_1, w_i) \geq 0.99t$.
- *Check Local Uniqueness:* For each w_i consider the $8t$-substring w_i' of the same trace centered at w_i. If a occurs twice in any w_i' abort.
- *Match:* Otherwise, conclude that any occurrence of a in w_i corresponds to the same character of x.

The correctness of the algorithm follows from Lemma 8. Specifically, the lemma implies that with high probability the pre-images of every w_i are contained in a contiguous set of at most $4t$ indices. However, this contiguous set is a subset of the pre-image of each w_i'. Hence, if a occurs twice within the contiguous set the algorithm will abort. Otherwise, all occurrences of a in the w_i must correspond to the same index.

Relationship between f and g. We conclude this section by showing that the above relationship between f and g is almost tight.

Lemma 10. *For any p, $f(\frac{1}{2}\log_\sigma n, p, \sigma) \leq g(n, p, \sigma)$.*

Proof. By definition, there exists a reconstruction algorithm $\mathcal{A}$ that recovers a random n character string with high probability using $g(n, p, 2)$ traces.

Given a set of traces of an unknown string x, it is easy to simulate an equal number of traces of the concatenated string $a|x|b$ for arbitrary strings a and b. Given successful recovery of $a|x|b$, we can of course extract x.

Let $B = \frac{1}{2}\log_\sigma n$. To recover $x \in [\sigma]^B$ from a set of traces, we first uniformly at random choose integers $c, d \in \{0, 1, \ldots, n/B - 1\}$ subject to $c + d = n/B - 1$. We then choose $a \in [\sigma]^{cB}$ and $b \in [\sigma]^{dB}$ uniformly at random. We simulate the traces of $a|x|b$, run $\mathcal{A}$ on the results, and extract x.

This succeeds whenever $\mathcal{A}$ successfully recovers $a|x|b$. Let μ be the uniformly random distribution on $[\sigma]^n$ and μ' be the distribution of $a|x|b$. Because $\mathcal{A}$ succeeds with high probability on μ, it suffices to show that the total variation distance between μ and μ' is polynomially small.

By thinking of the n character string as n/B blocks of length B, another way to draw from μ would be to (1) let k be drawn from $\mathcal{B}_{n/B,1/\sigma^B}$, the binomial random variable with n/B trials of probability $1/\sigma^B$; (2) set k random blocks to have value x; (3) set every other block independently to have a uniform value other than x. One can draw from μ' in the same way, but setting $k = 1 + \mathcal{B}_{n/B-1,1/\sigma^B}$ in the first step. Therefore the total variation distance between μ and μ' is at most the distance between $\mathcal{B}_{n/B,1/\sigma^B}$ and $1 + \mathcal{B}_{n/B-1,1/\sigma^B}$. This is $O(1/\sqrt{n/(B\sigma^B)}) < O(n^{-1/3})$, which is polynomially small. □

3.3 Lower Bound

In this section we prove the first super-logarithmic lower bound on the value of $g(n, p, 2)$ for constant p. To do this we introduce two specific binary strings of length $2r$ where $r = O(\log n)$:

1. $w \in \{0,1\}^{2r}$ is the all zero string expect for a single 1 at position r
2. $w' \in \{0,1\}^{2r}$ is the all zero string expect for a single 1 at position $r+1$

The proof relies on the fact that distinguishing w and w' with probability greater than $1-\delta$ requires $\Omega(r \log(1/\delta))$ traces (this will be implied by Corollary 1) and each of w and w' occur $n^{\Omega(1)}$ times in a random binary string of length n. The intuition is then that δ needs to be inversely polynomial in n otherwise one of the occurrences of w will be confused with an occurrence of w' (or vice versa). The following theorem formalizes this argument.

Theorem 1. *$g(n, p, 2) = \Omega(\log^2 n)$ for constant $p > 0$.*

Proof. Set the length of w and w' to be $B = c \log n$ for some small constant c, i.e., $r = (c \log n)/2$. By Corollary 1, if $m < c_2 \log^2 n$ for sufficiently small constant c_2, then the total variation distance between (m traces of w) and (m traces of w') is less than $1 - 1/\sqrt{n}$. Thus we can draw a set of m traces of a uniformly random choice between w or w' by choosing something independent of that choice with probability $1/\sqrt{n}$.

We partition our vector of length n into n/B blocks of length B. For a random bit vector and sufficiently small $c < 1/2$ we have with high probability that more than $\sqrt{n}$ blocks will equal one of w and w'. Therefore the algorithm must succeed with high probability on a random bit vector conditioned on having more than $\sqrt{n}$ blocks of value w or w'.

Now, the trace of a bit vector is just the concatenation of the trace of the component blocks. We could sample a set of m traces by first deciding which blocks are one of w or w', then choosing for each such block whether it is w or w', then taking the m traces. The resulting set of m traces is independent of block's choice between w and w' with probability $1/\sqrt{n}$; hence with at least $1 - (1 - 1/\sqrt{n})^{\sqrt{n}} > 1/2$ probability, the set of m traces will be independent of the choice of at least one of the $\sqrt{n}$ blocks of value w or w'. If this is true, the algorithm can give the correct output with probability at most $1/2$; hence the algorithm can give the correct output with probability at most $3/4$ overall. Therefore we need $m = \Omega(\log^2 n)$ for correct recovery with high probability. □

What remains is to prove Corollary 1. We make use of the Hellinger distance, a convenient measure of distance between distributions. For two discrete distribution $P = (p_1, p_2, p_3, \ldots)$ and $Q = (q_1, q_2, q_3, \ldots)$, the *squared Hellinger distance* between P and Q is defined as $\mathrm{H}^2(P, Q) = \frac{1}{2} \sum_i (\sqrt{p_i} - \sqrt{q_i})^2$.

Hellinger distance has two nice properties: first, squared Hellinger distance is subadditive over product measures, so the squared Hellinger distance between (m samples of P) and (m samples of Q) is at most $mH^2(P, Q)$; and second, if $H(P, Q) = o(1)$ then the total variation distance between P and Q is $o(1)$. Hence if $H(P, Q) \leq \varepsilon$, then it requires $\Omega(1/\varepsilon)$ samples to distinguish P and Q.

Lemma 11. *For any deletion probability $p = \Omega(1)$, the squared Hellinger distance between the distribution of a trace of w and the distribution of a trace of w' is $O(1/r)$.*

Proof. The distribution of a trace of w is

$$\mathrm{Tr}(w) \sim \begin{cases} \underbrace{0\ldots0}_{\mathcal{B}_{2r-1,1-p}} & \text{with probability } p \\ \underbrace{0\ldots0}_{\mathcal{B}_{r-1,1-p}}\,1\,\underbrace{0\ldots0}_{\mathcal{B}_{r,1-p}} & \text{with probability } 1-p \end{cases}$$

while the distribution of a trace of w' is the same, except swapping $\mathcal{B}_{r-1,p}$ and $\mathcal{B}_{r,p}$. Hence the squared Hellinger distance between the two traces is

$$\begin{aligned} H^2(\mathrm{Tr}(w), \mathrm{Tr}(w')) &= (1-p)H^2\left((\mathcal{B}_{r-1,1-p}, \mathcal{B}_{r,1-p}), (\mathcal{B}_{r,1-p}, \mathcal{B}_{r-1,1-p})\right) \\ &\leq 2(1-p)H^2(\mathcal{B}_{r-1,1-p}, \mathcal{B}_{r,1-p}) \leq O(1/r)\ . \end{aligned}$$

□

Corollary 1. *Consider any $r > 1$, $\delta < 1$, and deletion probability $p = \Omega(1)$. For some small constant $c > 0$, the total variation distance between $m = c^2 r \log(1/\delta)$ traces of w and m traces of w' is at most $1 - \delta$.*

Proof. Let $y_1, \ldots, y_m$ be traces of w and $z_1, \ldots, z_m$ be traces of w' for $m = c^2 r \log(1/\delta)$ and a sufficiently small constant c. We will show that the total variation distance between $(y_1, \ldots, y_m)$ and $(z_1, \ldots, z_m)$ is less than $1 - \delta$.

We partition $[m]$ into k groups of size cr, for $k = c\log(1/\delta)$. Within each group, by subadditivity of squared Hellinger distance and appealing to Lemma 11, we have that

$$H^2((y_1, \ldots, y_{cr}), (z_1, \ldots, z_{cr})) \leq crH^2(\mathrm{Tr}(w), \mathrm{Tr}(w')) = O(c) < 1/10$$

for sufficiently small c. Then the total variation distance between $(y_1, \ldots, y_{cr})$ and $(z_1, \ldots, z_{cr})$ is bounded by $2H((y_1, \ldots, y_{cr}), (z_1, \ldots, z_{cr})) \leq 2/3$.

Hence we may sample $(y_1, \ldots, y_{cr})$ and $(z_1, \ldots, z_{cr})$ in such a way that the two distributions are identical with probability at least $1/3$. If we do this for all k groups, we have that $(y_1, \ldots, y_m) \sim (z_1, \ldots, z_m)$ with probability at least $1/3^k > 2\delta$ for sufficiently small constant c. □

4 Arbitrary String Reconstruction

In this last section, we consider the problem of reconstructing an arbitrary binary[4] string $x \in \{0,1\}^n$ from random subsequences of length $\Theta(\sqrt{n \log n})$ or equivalently when the deletion probability of each bit is $1 - c\sqrt{\log n / n}$ for some constant c. We prove the following result.

[4] Recall that Lemma 1 shows that this result can be extended to the non-binary case.

Theorem 2. *$f(n,p,2) \leq e^{\sqrt{n}\,\mathrm{polylog}\,n}$ if $p \leq 1 - c\sqrt{\frac{\log n}{n}}$ for some constant $c > 0$.*

Our result uses the following combinatorial result by Scott [7]. For $i \in \{1,2,3,\ldots\}$, let n_i be the number of length i subsequences of x that end with a 1, i.e., $n_i = \sum_{j=1}^{n} x_j \binom{j-1}{i-1}$. Scott showed that if $k \geq (1+o(1))\sqrt{n \log n}$, then there exists a unique binary solution to the equation $Px^T = n^T$ where $n = (n_1, n_2, \ldots, n_k)$ and P is the $k \times n$ matrix with $P_{ij} = \binom{j-1}{i-1}$. The next theorem follows immediately.

Theorem 3 (Scott [7]). *$\{n_i\}_{i\in[k]}$ uniquely define x if $k \geq (1+o(1))\sqrt{n \log n}$.*

Therefore it is sufficient to determine each n_i. To do this is we pick a random subsequence of length i from each of the m traces and let m_i be the number of them that end with a 1. We then estimate n_i by $\tilde{n}_i = \frac{m_i}{m}\binom{n}{i}$. The next lemma shows that if m is sufficiently large then $n_i = \tilde{n}_i$ with high probability.

Lemma 12. *If $m \geq 2n^{2i}\log(2n)$ then $\Pr[n_i \neq \tilde{n}_i] \leq 1/n^2$.*

Proof. First note that $E[m_i/m] = n_i/\binom{n}{i}$ and that m_i is the sum of independent boolean trials. By applying the Chernoff bound,

$$\Pr[|\tilde{n}_i - n_i| \geq 1] = \Pr\left[\left|m_i - \frac{mn_i}{\binom{n}{i}}\right| \geq \frac{m}{\binom{n}{i}}\right] \leq 2\exp\left(-\frac{m}{3n_i\binom{n}{i}}\right) < 2\exp\left(\frac{-m}{n^{2i}}\right).$$

Hence, $m > 2n^{2i}\log 2n$ ensures this probability is less than $1/n^2$. □

Therefore, by an application of the union bound $2n^{2(1+o(1))\sqrt{n\log n}}$ traces are sufficient to compute all the necessary n_i with high probability.

References

1. T. Batu, S. Kannan, S. Khanna, and A. McGregor. Reconstructing strings from random traces. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 910–918, 2004.
2. M. Dudík and L. J. Schulman. Reconstruction from subsequences. *J. Comb. Theory, Ser. A*, 103(2):337–348, 2003.
3. T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder. Trace reconstruction with constant deletion probability and related results. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 389–398, 2008.
4. S. Kannan and A. McGregor. More on reconstructing strings from random traces: Insertions and deletions. In *IEEE International Symposium on Information Theory*, pages 297–301, 2005.
5. J. Lember and H. Matzinger. Standard deviation of the longest common subsequence. *The Annals of Probability*, 37(3):1192–1235, 05 2009.
6. D. Pollard. *Asymptopia*. http://www.stat.yale.edu/ pollard/, 2000.
7. A. D. Scott. Reconstructing sequences. *Discrete Mathematics*, 175(1-3):231–238, 1997.
8. K. Viswanathan and R. Swaminathan. Improved string reconstruction over insertion-deletion channels. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 399–408, 2008.