Approximate Trace Reconstruction: Algorithms

Sami Davies
University of Washington
Seattle, WA, USA
Email: daviess@uw.edu

Miklós Z. Rácz and Benjamin G. Schiffer
Princeton University
Princeton, NJ, USA
Email: {mracz, bgs3}@princeton.edu

Cyrus Rashtchian
University of California San Diego
La Jolla, CA, USA
Email: crashtchian@eng.ucsd.edu

2021 IEEE International Symposium on Information Theory (ISIT) | 978-1-5386-8209-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISIT45174.2021.9517926

***Abstract*—We introduce *approximate trace reconstruction*, a relaxed version of the trace reconstruction problem. Here, instead of learning a binary string perfectly from noisy samples, as in the original trace reconstruction problem, the goal is to output a string that is close in *edit distance* to the original string using few traces. We present several algorithms that can approximately reconstruct strings that belong to certain classes, where the estimate is within $n/\mathrm{polylog}(n)$ edit distance and where we only use $\mathrm{polylog}(n)$ traces (or sometimes just a single trace). These classes contain strings that require a linear number of traces for exact reconstruction and that are quite different from a typical random string. From a technical point of view, our algorithms approximately reconstruct consecutive substrings of the unknown string by aligning dense regions of traces and using a run of a suitable length to approximate each region.**

A full version of this paper is accessible at: https://arxiv.org/abs/2012.06713.pdf

I. Introduction

In the *trace reconstruction problem*, an unknown string on n bits is passed through a deletion channel many times independently, producing a set of traces (i.e., random subsequences of the string). The deletion channel deletes each bit independently with constant probability q. The goal is to exactly reconstruct the original string from the traces [1], [2]. Exact reconstruction is proving difficult to study, so far; currently there is an exponential gap between the best known upper and lower bounds [3]–[7]. Here, we relax the problem and instead study whether it is possible to approximately reconstruct an unknown string with much less information than in exact reconstruction. The algorithm should output a string that is close to the original string. We consider *edit distance*, which measures the minimum number of insertions, deletions, and substitutions between a pair of strings. For an unknown string of length n, we investigate the number of traces needed to approximate the string up to εn edit distance; we call this *εn-approximate reconstruction*.

Trace reconstruction has become popular partially due to its connection with DNA data storage, where reconstruction algorithms are used to recover stored data [8]–[13]. In these reconstruction algorithms, error-correcting codes handle missing data, and so approximate reconstruction algorithms are practically useful. A single trace from a string X is, in expectation and with high probability, an εn-approximation to X for any ε larger than the deletion probability. However, the interesting regime is for much smaller ε, such as when ε is a very small constant compared to q or even going to 0 with n. Theoretically, an eventual goal for this line of work would be to find the smallest ε such that any string can be εn-approximately reconstructed with $\mathrm{poly}(n)$ traces, where ε might depend on n and q. Designing algorithms to find approximate solutions for general strings may in fact require fundamentally different methods than all previous work on exact reconstruction or on the maximum likelihood solution. Our results begin to tackle this challenge; they exhibit the ability to approximately reconstruct strings based on various run-length or density assumptions. For these classes of strings, we develop new polynomial-time, alignment-based algorithms, and we show that $O(\log(n)/\varepsilon^2)$ traces suffice.

II. Related work

In trace reconstruction, the main theoretical question is whether $\mathrm{poly}(n)$ traces suffice for exact reconstruction. For an arbitrary string, $\exp(\widetilde{O}(n^{1/5}))$ traces suffice [5], improving the previous bound of $\exp(O(n^{1/3}))$ [3], [4]. On the other hand, at least $\widetilde{\Omega}(n^{3/2})$ traces are required [6], [7].

Others have studied forms of approximate trace reconstruction but with different goals in mind. For instance, the maximum likelihood decoding of average-case strings has been studied given only a constant number of traces [14]–[16].

Other work related to ours surrounds attempts to distinguish strings close in edit distance [17]. Our work also resembles coded trace reconstruction, though we point out that we study classes of strings that are very different from codes (e.g., pairs of strings in our classes can be very close) [18], [19]. A more complete comparison to and background on coded trace reconstruction are included in the full version.

III. Our results

In our results, the deletion probability q is a fixed constant, and we let $p := 1-q$ be the retention probability. The variables $C, C', C'', C_1, C_2, \ldots$ are constants, and $O(\cdot)$ hides constants that may depend on p, q. We use $\log(\cdot)$ with base $1/q$. The phrase *with high probability* means probability at least $1 - O(1/n)$. A *run* in a string is a substring of consecutive bits of the same value, and we often refer specifically to 0-runs and 1-runs. We use bold $\mathbf{r}$ to denote runs, or more generally substrings, and let $|\mathbf{r}|$ denote its length (number of bits). We assume that the algorithms know n, q, ε, and the class that the unknown string comes from. In Section IV, we also state the basic ideas for our "warm-up" algorithms, which are simpler

and introduce some of the techniques we use in our other algorithms; details are included in the full version.

Theorem 1 only requires 1-runs to be long, while the length of 0-runs is more flexible.

Theorem 1. *Let X be a string on n bits such that all of its 1-runs have length at least $C'\log(n)/\varepsilon$ and none of its 0-runs have length between $C'\log(n)$ and $3C'\log(n)$. If $C' \geqslant 100/p$, then X can be εn-approximately reconstructed with $O(\log(n)/\varepsilon^2)$ traces.*

Theorem 2 extends Theorem 1 to a wider class of strings by allowing many of the bits in the runs to be arbitrarily flipped.

Theorem 2. *Suppose that $p > 3\varepsilon$. Let Y be a string on n bits such that all of its 1-runs have length at least $C'\log(n)/\varepsilon$ and none of its 0-runs have length between $C'\log(n)$ and $3C'\log(n)$. Suppose that X is formed from Y by modifying at most $\varepsilon C'\log(n)$ arbitrary bits in each run of Y. If $C' \geqslant 1000/p$, then X can be εn-approximately reconstructed with $O(\log(n)/\varepsilon^2)$ traces.*

For the final class, we consider a slightly different relaxation of having long runs. We impose a local density or sparsity constraint on contiguous intervals. Here, a single trace suffices.

Theorem 3. *For $C' \geqslant 50/p^2$, if X can be divided into contiguous intervals $I_1, \ldots, I_m$ with all I_i having length at least $C'\log(n)/\varepsilon^2$ and density at least $1 - \frac{\varepsilon}{12}$ of 0s or 1s, then X can be εn-approximately reconstructed with a single trace in polynomial time.*

The algorithm for Theorem 3 extends to handle independent insertions at a rate of $O(\varepsilon)$, since the proof relies on finding high density regions, which are unchanged by such insertions.

We provide some justification for the classes of strings considered in the above theorems. Strings that either contain long runs or that are locally dense are a natural class to examine in order to understand the advantage gained by approximate reconstruction over exact. Strings with sufficiently long runs require $\Omega(n)$ traces to reconstruct exactly, as exact reconstruction for this set involves distinguishing between the strings $1^{n/2}01^{n/2-1}$ and $1^{n/2-1}01^{n/2}$, for example. These strings can be approximately reconstructed with substantially less traces for large enough values of ε. We then relax the condition that strings have long runs to the condition that strings are locally dense. Strings with long runs and strings that are locally dense also look very different than average-case (i.e., uniformly random) strings, which can be exactly reconstructed with $O(\exp(\log^{1/3}(n)))$ traces [20].

IV. Preliminaries

We let $d_{\mathrm{E}}(X, X')$ be the edit distance metric between X and X', which is the minimum number of insertions, deletions, and substitutions required to transform X into X'. For each class of strings that we consider, we present an algorithm and argue that it can εn-approximately reconstruct any string from the class. Our algorithms output a string $\widehat{X}$, an approximation of X, satisfying $d_{\mathrm{E}}(X, \widehat{X}) \leqslant \varepsilon n$ with high probability.

We denote a single run by $\mathbf{r}$ and a set of runs by $\mathbf{r}_1, \ldots, \mathbf{r}_k$. Our convention is to let X denote the unknown string that we wish to reconstruct, and Y will sometimes denote a modified version. A single trace will be denoted by $\widetilde{X}$ and a set of traces by $\widetilde{X}_1, \ldots, \widetilde{X}_T$. Tildes will also be used to mark runs and intervals of traces. Some strings X we partition into ℓ substrings $X^1, \ldots, X^\ell$; their concatenation to form X is denoted as $X = X^1X^2\cdots X^\ell$.

Some of our algorithms reconstruct X by partitioning it into substrings $X^1, \ldots, X^\ell$ and reconstructing these substrings approximately. Specifically, we will find strings $\widehat{X}^i$ such that the edit distance between $\widehat{X}^i$ and X^i is at most $\varepsilon|X^i|$, and then we will invoke the following lemma to see that $X = X^1\cdots X^\ell$ and $\widehat{X} = \widehat{X}^1\cdots\widehat{X}^\ell$ have edit distance at most εn.

Lemma 4. *Let $X = X^1X^2\cdots X^\ell$ and $\widehat{X} = \widehat{X}^1\cdots\widehat{X}^\ell$ be strings on n bits. If the edit distance between X^i and $\widehat{X}^i$ is at most $\varepsilon|X^i|$ for all $i \in [\ell]$, then $d_{\mathsf{E}}(X, \widehat{X}) \leqslant \varepsilon n$.*

In the full version, as a warm-up we present two simple algorithms that reconstruct simple classes of strings with very long runs. If all of the runs in X have length at least $5\log(n)$, then by Chernoff bounds all $O(\log n/\varepsilon^2)$ traces have the same number of runs as the original string with high probability. By aligning the traces by run and scaling the average run lengths across traces by $1/p$, X can be εn-approximately reconstructed with high probability. Similarly, if X has 1-runs with length at least $C'\log(n)/\varepsilon^2$ for sufficiently large C', then simply scaling the length of every run in the trace by $1/p$ gives an εn-approximation. See the full version for the warm-up algorithms and the associated proofs.

V. Algorithms and Proofs

A. Identifying long runs

We begin with an algorithm that builds on the ideas described for our warm-up algorithms; though here when we relax the length restriction on the 0-runs, entire runs of 0s may be deleted, combining consecutive 1-runs and making it difficult to identify which runs align together between traces. To still use an alignment algorithm that averages run lengths, we impose the weaker condition on the 0-runs that they must be divided into short 0-runs and long 0-runs. As long as there is a gap of sufficiently large size such that there are no 0-runs with length in the gap, then in the traces we can identify which 0-runs are long and which are short.

The following points outline the reconstruction algorithm used in the proof of Theorem 1.

1) **Set-up:** String X on n bits such that all of its 1-runs have length at least $C'\log(n)/\varepsilon$, where $C' \geqslant 100/p$, and all of its 0-runs have length either greater than $3C'\log n$ or less than $C'\log n$.
2) Sample $T = \frac{2}{p^2\varepsilon^2}\log(n)$ traces, $\widetilde{X}_1, \ldots, \widetilde{X}_T$, from the deletion channel with probability q.
3) Define $L := 2C'p\log n$, and for all $j \in [T]$, index the 0-runs in $\widetilde{X}_j$ with length at least L as $\widetilde{\mathbf{r}}_1^j, \ldots, \widetilde{\mathbf{r}}_{k_j}^j$. For $i \in [k_j - 1]$, let $\widetilde{\mathbf{s}}_i^j$ be the bits between $\widetilde{\mathbf{r}}_i^j$ and $\widetilde{\mathbf{r}}_{i+1}^j$ in

$\widetilde{X}_j$ and let $\widetilde{\mathbf{s}}_0^j$ be the bits before $\widetilde{\mathbf{r}}_1^j$ and $\widetilde{\mathbf{s}}_{k_j+1}^j$ the bits after $\widetilde{\mathbf{r}}_{k_j}^j$ for all $j \in [T]$.

4) If there is $j \neq j' \in [T]$ with $k_j \neq k_{j'}$, then fail without output. Otherwise, let $k := k_1 = k_2 = \cdots = k_T$.
5) Compute $\widetilde{\mu}_i^{\mathbf{r}} = \frac{1}{T}\sum_{j=1}^T |\widetilde{\mathbf{r}}_i^j|$ for all $i \in [k]$ and $\widetilde{\mu}_i^{\mathbf{s}} = \frac{1}{T}\sum_{j=1}^T |\widetilde{\mathbf{s}}_i^j|$ for all $i \in \{0\} \cup [k+1]$.
6) Output $\widehat{X} = \widehat{1}_0\widehat{0}_1\widehat{1}_1 \cdots \widehat{1}_k\widehat{0}_k\widehat{1}_{k+1}$, where $\widehat{1}_i$ is a 1-run, length $\frac{\widetilde{\mu}_i^{\mathbf{s}}}{p}$, and $\widehat{0}_i$ is a 0-run, length $\frac{\widetilde{\mu}_i^{\mathbf{r}}}{p}$.

This algorithm is inherently approximate, since we fill gaps between the long 0-runs with 1-runs, omitting short 0-runs.

Proof of Theorem 1. Let X be a string on n bits such that all of its 1-runs have length at least $C'\log(n)/\varepsilon$, where $C' \geqslant 100/p$, and all of its 0-runs have length either greater than $3C'\log n$ or less than $C'\log n$. Take $T = \frac{2}{p^2\varepsilon^2}\log(n)$ traces of X. By a Chernoff bound, with probability at least $1 - \frac{1}{n^2}$, no 1-run is fully deleted in any trace; in the following we assume that we are on this event.

We will justify that in the traces we can identify all 0-runs that had length at least $3C'\log(n)$ in X. Let $\mathbf{r}$ be a 0-run from X with length $|\mathbf{r}| \geqslant 3C'\log(n)$. Using a Chernoff bound, the probability that in a single trace $\mathbf{r}$ is transformed into a run $\widetilde{\mathbf{r}}$ with $|\widetilde{\mathbf{r}}| \leqslant 2C'p\log(n)$ is bounded by $2n^{-3}$.

It follows that, with probability at least $1 - \frac{4T}{n^2}$, there does not exist any 0-run and any trace such that either of the "unlikely" inequalities above holds. On this event, we have that for any 0-run $\mathbf{r}$ of length at least $3C'\log n$, and any trace $\widetilde{X}_j$, we can identify the image $\widetilde{\mathbf{r}}^j$ of $\mathbf{r}$ in trace $\widetilde{X}_j$. In particular, on this event, the number of 0-runs in each trace that has length at least $2C'p\log(n)$ is equal to the number of 0-runs in X of length at least $3C'\log(n)$; thus $k_1 = k_2 = \cdots k_T =: k$.

Let $L := 2C'p\log n$ and find every 0-run in $\widetilde{X}_j$ with length at least L, indexing them as $\widetilde{\mathbf{r}}_1^j, \ldots, \widetilde{\mathbf{r}}_k^j$. For $i \in [k-1]$, let $\widetilde{\mathbf{s}}_i^j$ be the bits between the last bit of $\widetilde{\mathbf{r}}_i^j$ and the first bit of $\widetilde{\mathbf{r}}_{i+1}^j$ in $\widetilde{X}_j$ and let $\widetilde{\mathbf{s}}_0^j$ be the bits before $\widetilde{\mathbf{r}}_1^j$ and $\widetilde{\mathbf{s}}_{k+1}^j$ the bits after $\widetilde{\mathbf{r}}_k^j$. Let $\mathbf{s}_i$ be the contiguous substring of X from which $\widetilde{\mathbf{s}}_i^1, \ldots, \widetilde{\mathbf{s}}_i^T$ came and $\mathbf{r}_i$ the contiguous substring of X from which $\widetilde{\mathbf{r}}_i^1, \ldots, \widetilde{\mathbf{r}}_i^T$ came.

For all i, we will approximate $\mathbf{r}_i$ with $\widehat{0}_i$ a 0-run of length $\widetilde{\mu}_i^{\mathbf{r}}/p$, for $\widetilde{\mu}_i^{\mathbf{r}} = \frac{1}{T}\sum_{j=1}^T |\widetilde{\mathbf{r}}_i^j|$, and we will approximate $\mathbf{s}_i$ with $\widehat{1}_i$, a 1-run of length $\widetilde{\mu}_i^{\mathbf{s}}/p$, for $\widetilde{\mu}_i^{\mathbf{s}} = \frac{1}{T}\sum_{j=1}^T |\widetilde{\mathbf{s}}_i^j|$. By Chernoff and union bound, $\mathbf{P}(\exists i : |\widetilde{\mu}_i^{\mathbf{r}}/p - |\mathbf{r}_i|| \geqslant \varepsilon|\mathbf{r}_i|) \leqslant 2n^{-3}$ and $\mathbf{P}(\exists i : |\widetilde{\mu}_i^{\mathbf{s}}/p - |\mathbf{s}_i|| \geqslant \varepsilon|\mathbf{s}_i|) \leqslant 2n^{-3}$.

Since $\mathbf{s}_i$ contains alternating 1-runs with length at least $C'\log(n)/\varepsilon$ and 0-runs with length at most $C'\log(n)$, $\mathbf{s}_i$ has at least a $1-\varepsilon$ density of 1s. Therefore $d_{\mathsf{E}}(\mathbf{s}_i, \widehat{1}_i) \leqslant 2\varepsilon|\mathbf{s}_i|$ and $d_{\mathsf{E}}(\mathbf{r}_i, \widehat{0}_i) \leqslant \varepsilon|\mathbf{r}_i|$. Let $\widehat{X} = \widehat{1}_0\widehat{0}_1\widehat{1}_1 \cdots \widehat{1}_k\widehat{0}_k\widehat{1}_{k+1}$ and we see that from Lemma 4 $d_{\mathsf{E}}(X, \widehat{X}) \leqslant 2\varepsilon n$. Applying this with $\varepsilon/2$ instead of ε, the result follows. Constants were taken large enough to account for this factor of 2. □

Note that the above theorem holds when the constant C' is unknown. Given $T = O(\log n/\varepsilon^2)$ traces of X, we can determine whether or not X had such a gap, and the corresponding C' value, with high probability. We can then execute the algorithm as stated.

B. Identifying dense substrings

We extend the class of strings we can approximately reconstruct, proving Theorem 2, which is a robust version of Theorem 1. Specifically, we consider strings with similar properties to those in Theorem 1, but allow for bit flips.

The goal of the algorithm is similar to that of Theorem 1, which is to identify long 0-runs from Y in each trace of X and to align by these 0-runs; then, we approximate the rest of X with 1-runs. Because X and Y have small edit distance, a good approximation for Y is also good for X. Unfortunately the long 0-runs from Y are no longer necessarily 0-runs in X, and therefore they are more difficult to find in the traces. Instead we find long 0-dense substrings in X.

Let X and Y be as in the theorem statement. We also fix $m := C'\varepsilon\log(n)$ throughout this subsection. Fix a trace $\widetilde{X}$ of X, as well as an index ℓ. Let $\widetilde{n}$ denote the length of the trace. Define the indices i_ℓ and j_ℓ to be those that are $(m+1)$ 1s to the left and right of ℓ in $\widetilde{X}$, respectively, if such indices exist. We count the 0s in $\widetilde{X}$ between indices i_ℓ and j_ℓ with

$$S_{\text{int}}(\widetilde{X}, \ell) := \sum_{k=i_\ell}^{j_\ell} \mathbb{1}_{\widetilde{X}[k]=0}.$$

Note that $S_{\text{int}}(\widetilde{X}, \ell)$ is not defined if i_ℓ or j_ℓ are not defined. We use a slightly different quantity on the boundary of the trace to handle this. Letting the definition of i_ℓ and j_ℓ remain the same, if i_ℓ or j_ℓ is not defined, then we consider $S_{\text{L-bound}}(\widetilde{X}, \ell) := \sum_{k=0}^{j_\ell} \mathbb{1}_{\widetilde{X}[k]=0}$ or $S_{\text{R-bound}}(\widetilde{X}, \ell) := \sum_{k=i_\ell}^{\widetilde{n}} \mathbb{1}_{\widetilde{X}[k]=0}$, respectively. Combining the interior and boundary quantities, let $S(\widetilde{X}_j, \ell) = S_{\text{int}}(\widetilde{X}_j, \ell)$ if there are $(m+1)$ 1s to the left and right of ℓ, let $S(\widetilde{X}_j, \ell) = S_{L-\text{bound}}(\widetilde{X}_j, \ell)$ if there are $(m+1)$ 1s to the right of ℓ but not the left, and let $S(\widetilde{X}_j, \ell) = S_{R-\text{bound}}(\widetilde{X}_j, \ell)$ if there are $(m+1)$ 1s to the left of ℓ but not the right.

In each trace we identify a set of substrings of X that are 0-dense, and then decide whether each such substring is long or short using $S(\widetilde{X}_j, \ell)$; that is, whether the corresponding unknown 0-runs in Y are long (length at least the upper bound of the gap) or short (length at most the lower bound of the gap). If the traces all agree on the number of long 0-dense substrings, we align the traces by these substrings and reconstruct in a manner similar to that of Theorem 1.

The following points outline the reconstruction algorithm used in the proof of Theorem 2.

1) **Set-up:** String X on n bits formed by flipping at most $\varepsilon C'\log(n)$ bits in each run of Y, where Y is a string on n bits such that all of its 1-runs have length at least $C'\log(n)/\varepsilon$, for $C' \geqslant 1000/p$, and all of its 0-runs have length either greater than $3C'\log n$ or less than $C'\log n$.
2) Sample $T = \frac{2}{p^2\varepsilon^2}\log n$ traces, $\widetilde{X}_1, \ldots, \widetilde{X}_T$, from the deletion channel with deletion probability q.
3) Set $m := \varepsilon C'\log n$ and $a := pC'\log n$. For each trace $\widetilde{X}_j$, let i be the smallest index of $\widetilde{X}_j$ with $\widetilde{X}_j[i] = 0$ and

$|\{k : \widetilde{X}_j[k] = 0, |i-k| \leqslant a+m\}| \geqslant a$. Let ℓ_1^j be the smallest index such that $\widetilde{X}_j[\ell_1^j] = 0$ and $|\{k : \widetilde{X}_j[k] = 0, i-(a+m) \leqslant k < \ell_1^j\}| = m$. Compute $S(\widetilde{X}_j, \ell_1^j)$. Starting $m+1$ bits to the right of the last bit counted in $S(\widetilde{X}_j, \ell_1^j)$, continue to the right and repeat this process, finding indices ℓ_t^j and computing $S(\widetilde{X}_j, \ell_t^j)$, for $t \geqslant 2$.

4) Set $\bar{G} = 2C'p\log n$. For every trace $\widetilde{X}_j$, let $I_j = \{t : S(\widetilde{X}_j, \ell_t^j) > \bar{G}\}$. If $|I_j|$ is not the same across all T traces, the algorithm fails. Otherwise, define $I = |I_j|$ and for all $t \in [I]$, we let $\widehat{0}_t$ be a 0-run of length $\widetilde{\mu}_t/p$, for $\widetilde{\mu}_t = \frac{1}{T}\sum_{j=1}^{T} S(\widetilde{X}_j, \ell_t^j)$.
5) Define $\widehat{i}_t = \frac{1}{T}\sum_{j'=1}^{T} i_{\ell_t^{j'}}$ and $\widehat{j}_t = \frac{1}{T}\sum_{j'=1}^{T} j_{\ell_t^{j'}}$, for $i_{\ell_t^{j'}}$ and $j_{\ell_t^{j'}}$ as in the definition of $S(\widetilde{X}_{j'}, \ell_t^{j'})$. Let $\widehat{1}_0, \ldots, \widehat{1}_I$ be 1-runs where $\widehat{1}_t$ has length $|\widehat{i}_{t+1} - \widehat{j}_t|/p$ for $t \in [I-1]$, $\widehat{1}_0$ has length $\widehat{i}_1/p$, and $\widehat{1}_I$ has length $|pn - \widehat{j}_I|/p$.
6) Output $\widehat{X} = \widehat{1}_0\widehat{0}_1\widehat{1}_1 \cdots \widehat{1}_{I-1}\widehat{0}_{I-1}\widehat{1}_I$.

Let ε, p be fixed with $p > 3\varepsilon$. Suppose X, Y, and C' are as in the algorithm statement. Let $\widetilde{X}$ be a trace of X. A 0-run $\mathbf{r}$ in Y may have some bits flipped from 0 to 1 in X, becoming the substring $\mathbf{r}_X$, so let $|\mathbf{r}_X^0|$ denote the number of 0s in $\mathbf{r}_X$.

Lemma 5. *Let $\widetilde{X}$ be a random trace from X, and let ℓ be an index of $\widetilde{X}$ such that $\widetilde{X}[\ell] = 0$. If the bit at $\widetilde{X}[\ell]$ is from a 0-run $\mathbf{r}$ in Y, then the following holds for the quantity $S(\widetilde{X}, \ell)$:*

1) *(Property 1) With probability at least $1 - n^{-6}$ the bits at indices i_ℓ and j_ℓ come from a 1-run adjacent to $\mathbf{r}$.*
2) *(Property 2) If indices i_ℓ and j_ℓ come from a 1-run adjacent to $\mathbf{r}$, then $S(\widetilde{X}, \ell)$ is upper bounded by a random variable from the distribution $\mathrm{Bin}(|\mathbf{r}_X^0|, p) + \mathrm{Bin}(2m, p)$.*
3) *(Property 3) If $|\mathbf{r}| \geqslant C'\log n$ and the bits at indices i_ℓ and j_ℓ come from a 1-run adjacent to $\mathbf{r}$, then with probability at least $1-n^{-6}$, $|S(\widetilde{X}, \ell) - p|\mathbf{r}|| \leqslant \frac{p|\mathbf{r}|}{4} + 3m$.*

Proof of Property 1. It suffices to prove the claim for i_ℓ. Index i_ℓ is $m+1$ 1s to the left of ℓ, and therefore not from $\mathbf{r}$, since at most m 0s of $\mathbf{r}$ were flipped to 1s. Further, by a Chernoff bound, with probability at least $1-n^{-6}$ the 1-run left-adjacent to $\mathbf{r}$ in Y has at least $2m+1$ bits surviving in $\widetilde{X}$. At most m bits of the left-adjacent 1-run to $\mathbf{r}$ in Y are flipped to 0, so at least $m+1$ 1s from this 1-run survive in $\widetilde{X}$. It follows that i_ℓ came from the left adjacent 1-run to $\mathbf{r}$ in Y. □

Proof of Property 2. Recall that $|\mathbf{r}_X^0|$ is the number of 0s in $\mathbf{r}$ that were not flipped to 1 in X. This component of $S(\widetilde{X}, \ell)$ is from the distribution $\mathrm{Bin}(|\mathbf{r}_X^0|, p)$. Let the contribution to $S(\widetilde{X}, \ell)$ by any 0s not from $\mathbf{r}$ be the random variable $Z_{\mathbf{r}}(\ell)$. Each bit that was flipped to 0 in either 1-run adjacent to $\mathbf{r}$ in Y can contribute 1 with probability at most p to $Z_{\mathbf{r}}(\ell)$. From the assumption on i_ℓ and j_ℓ, any other 0 from X will be outside of the range $[i_\ell, j_\ell]$. Therefore we can upper bound the contribution of $Z_{\mathbf{r}}(\ell)$ by a random variable sampled from $\mathrm{Bin}(2m, p)$. □

Proof of Property 3. By Property 2, $S(\widetilde{X}, \ell)$ is upper bounded by a random variable from the distribution $\mathrm{Bin}(|\mathbf{r}_X^0|, p) +$ $Z_{\mathbf{r}}(\ell)$. By a Chernoff bound, with probability $1 - n^{-6}$ the first binomial term varies from its mean by at most $p|\mathbf{r}|/4$. The second binomial term is upper bounded by $2m$ and $||\mathbf{r}_X^0| - |\mathbf{r}|| \leqslant m$. □

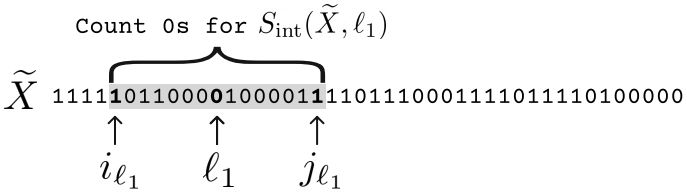

Fig. 1. Example for Theorem 2. For a trace $\widetilde{X}$, the index ℓ_1 is carefully chosen so that $\widetilde{X}[\ell_1]$ is from a 0-run in Y with high probability.

Proof of Theorem 2. Define $a := pC'\log(n)$. Take $T = \frac{2}{p^2\varepsilon^2}\log n$ traces of X, $\widetilde{X}_1, \ldots, \widetilde{X}_T$, and fix a trace $\widetilde{X}_j$. Our first goal is to find long 0-dense substrings in X; we can also think of these long 0-dense substrings as corresponding to long 0-runs in Y. Let i be the smallest index of $\widetilde{X}_j$ such that $\widetilde{X}_j[i] = 0$ and there are at least a 0s in $\widetilde{X}_j$ within $a+m$ indices of i, i.e.

$$|\{k : \widetilde{X}_j[k] = 0, |i-k| \leqslant a+m\}| \geqslant a.$$

Next find the index ℓ_1^j such that $\widetilde{X}_j[\ell_1^j] = 0$ and there are exactly m 0s in $\widetilde{X}_j$ within the interval of indices $[i-(a+m), \ell_1^j]$, i.e. $|\{k : \widetilde{X}_j[k] = 0,\ i-(a+m) \leqslant k < \ell_1^j\}| = m$. See Figure 1. The goal of this procedure is to find an index ℓ_1^j such that the bit at $\widetilde{X}_j[\ell_1^j]$ is from a 0-run in Y with high probability.

With probability at least $1-n^{-6}$, every 1-run in Y is reduced to a substring with at least $2(a+m)$ 1s in $\widetilde{X}_j$. This implies that the length $2(a+m)$ interval $\widetilde{X}_j[i-(a+m), i+a+m]$ contains bits from at most two 1 runs in Y and at most one 0 run with probability $1-n^{-6}$. By construction, this interval contains at least $a > 3m$ 0s (the inequality coming from the fact that $p > 3\varepsilon$). Since each 1-run had at most m bits flipped to 0, there must be at least $a - 2m > m$ 0s in the interval $\widetilde{X}_j[i-(a+m), i+a+m]$ that came from some 0-run $\mathbf{r}$ in Y. In this construction, the 0s from the $\mathbf{r}$ that survived in $\widetilde{X}_j$ are nested between at most m 0s that were flipped from the left-adjacent 1-run to $\mathbf{r}$ in Y and at most m 0s that were flipped from the right-adjacent 1-run to $\mathbf{r}$ in Y. This implies that the $(m+1)$th 0 in this interval must be from the 0-run $\mathbf{r}$.

Compute $S(\widetilde{X}_j, \ell_1^j)$. Note that with high probability, if a trace does not have $(m+1)$ 1s to the right of ℓ_1^j, the original string can be well-approximated by outputting the all 0s string with length $\frac{1}{T}\sum_{j=1}^{T}|\widetilde{X}_j|/p$. Starting $m+1$ bits to the right of the last bit counted in $S(\widetilde{X}_j, \ell_1^j)$, continue scanning to the right and repeat this process, finding indices ℓ_t^j and computing $S(\widetilde{X}_j, \ell_t^j)$, for $t \geqslant 2$. We jump ahead $m+1$ bits to the right between iterations because this forces the next bit i that satisfies the condition $|\{k : \widetilde{X}_j[k] = 0, |i-k| \leqslant a+m\}| \geqslant a$ to not overlap with the previous 0-run with high probability by Property 1.

We justify that this process succeeds, meaning that it catches all long 0-runs from Y, in all T traces, with high probability. For 0-run $\mathbf{r}$ in Y such that $|\mathbf{r}| \geqslant 3C' \log(n)$, with probability at least $1-n^{-6}$ at least $a+m$ bits from all such 0-runs survive in all T traces. Further there are at most m 1s among these bits. Therefore, with probability at least $1-n^{-6}$, we have at least a 0s that have at most m 1s inserted among them, and this triggers the calculation of ℓ_t^j for some t.

By the theorem assumptions, there exists an interval $[C' \log n, 3C' \log n]$ such that no 0-run $\mathbf{r}$ in Y has $|\mathbf{r}|$ in the gap $[C' \log n, 3C' \log n]$. Let $\bar{G}$ be the middle of the gap scaled by p, so $\bar{G} = 2C'p \log n$. By Property 3 and a union bound, with probability at least $1-n^{-4}$, all 0-runs $\mathbf{r}$ in Y with $|\mathbf{r}| \geqslant 3C' \log n$ will trigger the calculation of an ℓ_t^j with $S(\widetilde{X}_j, \ell_t^j) > \bar{G}$ in all traces, and all 0-runs $\mathbf{r}$ in Y with $|\mathbf{r}| < C' \log n$ will either not trigger an ℓ_t^j calculation, or if they do, ℓ_t^j will have $S(\widetilde{X}_j, \ell_t^j) < \bar{G}$ for all traces.

For every trace $\widetilde{X}_j$, let $I_j = \{t : S(\widetilde{X}_j, \ell_t^j) > \bar{G}\}$. If $|I_j|$ is not the same across all T traces, the algorithm fails. Otherwise let $I = |I_j|$ for all j, and for each trace $\widetilde{X}_j$ relabel the ℓ_t^j with $S(\widetilde{X}_j, \ell_t^j) > \bar{G}$ as $\ell_1^j, \ldots, \ell_I^j$.

The proof now proceeds similarly to that of Theorem 1. We approximate long 0-runs $\mathbf{r}_t$ in Y, which are close to some long 0-dense substrings of X with high probability, with 0-runs, and the rest is approximated with 1-runs. We first estimate the distance between the 0-runs in Y. Consider a 0-run $\mathbf{r}_t$ that generates an estimate of $\widetilde{\mu}_t^{\mathbf{r}}/p$, and take $\widehat{i}_t = \frac{1}{T}\sum_{j'=1}^{T} i_{\ell_t^{j'}}$ and $\widehat{j}_t = \frac{1}{T}\sum_{j'=1}^{T} j_{\ell_t^{j'}}$, for $i_{\ell_t^{j'}}$ and $j_{\ell_t^{j'}}$ as in the definition of $S(\widetilde{X}_{j'}, \ell_t^{j'})$. The average of the indices $\widehat{i}_t$ can be at most m bits to the left of the first 0 from $\mathbf{r}_t$, and therefore is at most off by m bits. The same is true for $\widehat{j}_t$. By a Chernoff bound, $|\widehat{i}_{t+1} - \widehat{j}_t|/p$ is an estimate of the distance between 0-runs with accuracy $2\varepsilon|\mathbf{r}_t|$ with probability at least $1-n^{-6}$. The substring between these 0-runs also has at least a $1-\varepsilon$ density of 1s, so we can fill with 1-runs for a good approximation. Let $\widehat{1}_0, \ldots, \widehat{1}_I$ be 1-runs where $\widehat{1}_t$ has length $|\widehat{i}_{t+1} - \widehat{j}_t|/p$ for $t \in [I-1]$, $\widehat{1}_0$ has length $\widehat{i}_1/p$, and $\widehat{1}_I$ has length $|pn - \widehat{j}_I|/p$. Hence by Lemma 4 the 1-runs contribute at most $3\varepsilon n$ to the edit distance error.

It remains to estimate the lengths of the long 0-runs in Y $\mathbf{r}_1, \ldots, \mathbf{r}_I$. Fix $t \in [I]$, let $\widehat{0}_t$ be a 0-run of length $\widetilde{\mu}_t^{\mathbf{r}}/p$, for $\widetilde{\mu}_t^{\mathbf{r}} = \frac{1}{T}\sum_{j=1}^{T} S(\widetilde{X}_j, \ell_t^j)$. For every $\mathbf{r}_t \in \{\mathbf{r}_1, \ldots, \mathbf{r}_I\}$, define $\mathbf{r}_{t_X}^0$ as above (the number of 0s from $\mathbf{r}_t$ in X). With probability at least $1-n^{-6}$ the average of $\mathrm{Bin}(|\mathbf{r}_{t_X}^0|, p)$ over $T = O(\log(n)/\varepsilon^2)$ traces is within $\varepsilon p|\mathbf{r}_{t_X}^0|$ of the mean $p|\mathbf{r}_{t_X}^0|$. Combining this with Property 2, with probability at least $1-n^{-3}$,

$$|\widetilde{\mu}_t^{\mathbf{r}} - p|\mathbf{r}_{t_X}^0|| \leqslant \varepsilon p|\mathbf{r}_{t_X}^0| + 2m.$$

Since $||\mathbf{r}_{t_X}^0| - |\mathbf{r_t}|| \leqslant m$, we have that

$$|p|\mathbf{r_t}| - \widetilde{\mu}_t^{\mathbf{r}}| \leqslant \varepsilon p|\mathbf{r}_t| + 2m + pm = \varepsilon p|\mathbf{r}_t| + 3m.$$

This is at worst an approximation of $p|\mathbf{r}_t|$ with edit distance error at most $\varepsilon + \frac{9\varepsilon}{p^2} \leqslant C''\varepsilon$ where we use $a > 3m$ and $C'' = 1 + \frac{9}{p^2}$. Taking a union bound over all $\mathbf{r}_t \in \{\mathbf{r}_1, \ldots, \mathbf{r}_I\}$, and applying Lemma 4, with probability at least $1-n^{-2}$ the long 0-run estimates contribute at most error $C''\varepsilon n$. Putting this all together, we output the string $\widehat{X} = \widehat{1}_0\widehat{0}_1\widehat{1}_1 \cdots \widehat{1}_{I-1}\widehat{0}_{I-1}\widehat{1}_I$. One more application of Lemma 4 implies that $d_{\mathsf{E}}(Y, \widehat{X}) \leqslant (C''+3)\varepsilon n$. Since Y is within εn edit distance from X, we can conclude that $d_{\mathsf{E}}(X, \widehat{X}) \leqslant (C''+4)\varepsilon n$.

If we apply this algorithm and analysis with $\frac{\varepsilon}{C''+4}$ instead of ε, the result follows. Constants were taken large enough to account for this factor of $C''+4$. □

C. Majority voting in substrings

A natural follow-up question to the previous theorems is what happens when the string no longer has long runs, but instead has long dense regions. This question is addressed by Theorem 3; the proof can be found in the full version.

The following points outline the reconstruction algorithm used in the proof of Theorem 3.

1) **Set-up:** String X on n bits such that X can be divided into contiguous intervals all of length at least $L = 50 \log n/(p^2\varepsilon^2)$ and density at least $1 - \frac{\varepsilon}{12}$ of 0s or 1s.
2) Sample $\widetilde{X}$ from the deletion channel with probability q.
3) Uniformly partition $\widetilde{X}$ into contiguous substrings of length $w = \varepsilon pL$, so $\widetilde{X} = \widetilde{X}_1 \cdots \widetilde{X}_{\lceil n/w \rceil}$, with a shorter last interval if needed.
4) Output $\widehat{X} = \widehat{X}_1 \cdots \widehat{X}_{\lceil n/w \rceil}$, where $\widehat{X}_i$ is a run of length w/p with value the majority bit of $\widetilde{X}_i$ for $i \in [\lceil n/w \rceil]$.

VI. Conclusion

We studied the challenge of determining the relative trace complexity of approximate versus exact string reconstruction. We present algorithms for classes of strings, where these classes lend themselves to techniques in every theoretician's toolbox, while introducing new alignment techniques that may be useful for other algorithms. Our algorithms output a string within edit distance εn from the original string using $O(\log n/\varepsilon^2)$ traces for classes of strings; these classes of strings are hard to reconstruct exactly. We leave as open work constructing algorithms for approximating arbitrary strings.

Algorithms with small sample complexity for the approximate trace reconstruction problem could also provide insight into exact solutions. If we know that the unknown string belongs to a specified Hamming ball of radius k, then one can recover the string exactly with $n^{O(k)}$ traces by estimating the histogram of length k subsequences [21], [22]. It is an open question whether an analogous claim can be proven for edit distance [17]. Do $n^{O(k)}$ traces suffice if we know an edit ball of radius k that contains the string? If so, then an algorithm satisfying our notion of edit distance approximation would imply an exact reconstruction result.

Acknowledgments

M.Z.R. was supported in part by NSF grant DMS 1811724 and by a Princeton SEAS Innovation Award. We thank João Ribeiro and Josh Brakensiek for discussions on coded trace reconstruction and the anonymous reviewers for feedback on an earlier version of the paper.

REFERENCES

[1] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004, pp. 910–918. [Online]. Available: http://dl.acm.org/citation.cfm?id=982792.982929

[2] 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.

[3] A. De, R. O'Donnell, and R. A. Servedio, "Optimal mean-based algorithms for trace reconstruction," *The Annals of Applied Probability*, vol. 29, no. 2, pp. 851–874, 2019.

[4] F. Nazarov and Y. Peres, "Trace reconstruction with $\exp(O(n^{1/3}))$ samples," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2017, pp. 1042–1046. [Online]. Available: http://doi.acm.org/10.1145/3055399.3055494

[5] Z. Chase, "New upper bounds for trace reconstruction," 2020, preprint available at https://arxiv.org/abs/2009.03296.

[6] N. Holden and R. Lyons, "Lower bounds for trace reconstruction," *Annals of Applied Probability*, vol. 30, no. 2, pp. 503–525, 2020.

[7] Z. Chase, "New lower bounds for trace reconstruction," *Annales de l'Institut Henri Poincaré (to appear)*, 2020, preprint at https://arxiv.org/abs/1905.03031.

[8] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen, C. N. Takahashi, S. Newman, H.-Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 36, pp. 242–248, 2018. [Online]. Available: https://www.nature.com/articles/nbt.4079

[9] G. M. Church, Y. Gao, and S. Kosuri, "Next-Generation Digital Information Storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, 2012.

[10] V. Bhardwaj, P. A. Pevzner, C. Rashtchian, and Y. Safonova, "Trace Reconstruction Problems in Computational Biology," *IEEE Transactions on Information Theory*, pp. 1–1, 2020.

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

[12] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific reports*, vol. 7, no. 1, pp. 1–6, 2017.

[13] R. Lopez, Y.-J. Chen, S. D. Ang, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Seelig, K. Strauss, and L. Ceze, "DNA assembly for nanopore data storage readout," *Nature Communications*, vol. 10, no. 1, pp. 1–9, 2019.

[14] O. Sabary, E. Yaakobi, and A. Yucovich, "The error probability of maximum-likelihood decoding over two deletion channels," 2020, preprint available at https://arxiv.org/abs/2001.05582.

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

[16] ——, "Algorithms for reconstruction over single and multiple deletion channels," 2020, preprint available at https://arxiv.org/abs/2005.14388.

[17] E. Grigorescu, M. Sudan, and M. Zhu, "Limitations of Mean-Based Algorithms for Trace Reconstruction at Small Distance," 2020, preprint available at https://arxiv.org/abs/2011.13737.

[18] J. Brakensiek, R. Li, and B. Spang, "Coded trace reconstruction in a constant number of traces," in *IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2020.

[19] M. Cheraghchi, R. Gabrys, O. Milenkovic, and J. Ribeiro, "Coded trace reconstruction," *IEEE Transactions on Information Theory*, vol. 66, no. 10, pp. 6084–6103, 2020.

[20] N. Holden, R. Pemantle, and Y. Peres, "Subpolynomial trace reconstruction for random strings and arbitrary deletion probability," in *Proceedings of the 31st Conference On Learning Theory (COLT)*, 2018, pp. 1799–1840. [Online]. Available: http://proceedings.mlr.press/v75/holden18a.html

[21] I. Krasikov and Y. Roditty, "On a Reconstruction Problem for Sequences," *Journal of Combinatorial Theory, Series A*, vol. 77, no. 2, pp. 344–348, 1997.

[22] A. Krishnamurthy, A. Mazumdar, A. McGregor, and S. Pal, "Trace reconstruction: Generalized and parameterized," 2019, preprint at https://arxiv.org/abs/1904.09618.