Multi-strand Reconstruction from Substrings

Yonatan Yehezkeally
Institute for Communications Engineering
Technical University of Munich
Munich 80333, Germany
yonatan.yehezkeally@tum.de

Sagi Marcovich
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 3200003, Israel
sagimar@cs.technion.ac.il

Eitan Yaakobi
Dept. of Computer Science
Technion-Israel Institute of Technology
Haifa 3200003, Israel
yaakobi@cs.technion.ac.il

***Abstract*—The problem of string reconstruction based on its substrings spectrum has received significant attention recently due to its applicability to DNA data storage and sequencing. In contrast to previous works, we consider in this paper a setup of this problem where multiple strings are reconstructed together. Given a multiset S of strings, all their substrings of some fixed length ℓ, defined as the *ℓ-profile* of S, are received and the goal is to reconstruct all strings in S. A *multi-strand ℓ-reconstruction code* is a set of multisets such that every element S can be reconstructed from its ℓ-profile. Given the number of strings k and their length n, we first find a lower bound on the value of ℓ necessary for existence of multi-strand ℓ-reconstruction codes with non-vanishing asymptotic rate. We then present two constructions of such codes and show that their rates approach 1 for values of ℓ that asymptotically behave like the lower bound.**

I. Introduction

String reconstruction refers to a large class of problems where information about a string can only be obtained in forms other than receiving it as one unit, and allowing for errors. Examples of this set of problems are the *k-deck problem* [7], [8], [20], [26] and the *reconstruction from substring compositions problem* [1], [2], [4], [12], [15], [23], [24], [27], [28]. Similar problems under this paradigm are the *trace reconstruction problem* [3] and the *reconstruction problem* by Levenshtein [16], however in these setups the string is received as one unit multiple times with possible errors.

The problem of string reconstruction from its substring spectrum has received significant interest in the past decade and has been rigorously studied. For a length-n string $\boldsymbol{x}$ and a positive integer ℓ, its *ℓ-profile*, denoted by $\mathcal{L}_\ell(\boldsymbol{x})$, is the multiset of all its length-ℓ substrings. Then, the goal is to reconstruct the string $\boldsymbol{x}$ given only $\mathcal{L}_\ell(\boldsymbol{x})$. If a string can be uniquely reconstructed from its ℓ-profile, then it is called *ℓ-reconstructible*. One of the main problems under this paradigm is to find the minimum value of ℓ, as a function of n, which guarantees that the asymptotic rate of all ℓ-reconstructible strings approaches 1. It was proved by Ukkonen [29] that if all length-ℓ substrings of $\boldsymbol{x}$ are different from each other, then the string $\boldsymbol{x}$ is $(\ell+1)$-reconstructible. A string $\boldsymbol{x}$ that satisfies this constraint is referred to as *ℓ-repeat free*. Based upon this property, it was recently proved in [9] that if $\ell = \lceil a\log(n)\rceil$ for some fixed value of $a > 1$, then the asymptotic rate of all ℓ-reconstructible strings approaches 1. The authors of [9] also proposed two encoding schemes of ℓ-repeat free strings; the first one uses a single redundancy symbol and supports $\ell = 2\lceil\log(n)\rceil + 2$, while the second works for substrings of length $\ell = \lceil\log n\rceil + \lceil 2\log\log n\rceil + 5$ and its asymptotic rate approaches 1. Extensions of this problem to the setup where the ℓ-profiles are not received error-free were also studied recently [5], [10], [21], [30].

In this paper, we extend the problem of ℓ-reconstructible strings to multisets of strings. This extension of the problem is motivated by DNA and polymer-based storage systems, since in both sequencing- and tandem mass spectrometry technologies it is typical that not a single string is read alone, but multiple strings simultaneously [6], [11], [14], [18], [25]. Thus, the ℓ-profiles of all the strings in some multiset S are read and the goal is to reconstruct all the strings in the multiset S. Assuming the multiset S consists of k length-n strings, our first goal is to study the minimum value of ℓ as a function of k and n which guarantees the asymptotic rate of all ℓ-reconstructible multisets approaches 1. We also present two efficient constructions of codes of ℓ-reconstructible multisets where their asymptotic rate approaches 1.

The rest of this paper is organized as follows. Section II presents the definitions that will be used throughout the paper as well as the problem formulation of multi-strand ℓ-reconstruction code. Section III shows that if $\log(nk) - \ell = \omega_{nk}(1)$ then there do not exist positive-rate multi-strand ℓ-reconstruction codes. In Section IV, two efficient constructions of multi-strand reconstruction codes are presented. We summarize and compare between the results of the constructions in the paper and show that, as a result, if $\ell \geqslant \log(nk) + 2\log\log(nk) + 5$ then there exists a family of multi-strand reconstruction codes with asymptotic rate 1.

II. Definitions and Preliminaries

Let Σ be a finite alphabet of size q, and denote some element $0 \in \Sigma$. In our setting, information is stored in an unordered collection of k strings of length n over Σ; it might be allowed for the same string to appear with multiplicity in the collection, which is encapsulated in the following formal definition. Let $\{\{a, a, b, \ldots\}\}$ denotes a multiset; i.e., elements are allowed to appear with multiplicity (for a multiset S, for convenience we let $\|S\|$ denote the number of unique elements in S). Then

$$\mathcal{X}_{n,k} \triangleq \{S = \{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k\}\} : \forall i : \boldsymbol{x}_i \in \Sigma^n\}.$$

Note that $|\mathcal{X}_{n,k}| = \binom{k+q^n-1}{k}$.

This work was supported in part by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Programme under Grant 801434. Y. Yehezkeally was supported by the Alexander von Humboldt Foundation in cooperation with the Carl Friedrich von Siemens Foundation, under a postdoctoral research fellowship. S. Marcovich and E. Yaakobi were supported by the United States-Israel BSF grant no. 2018048.

978-1-6654-0312-2/21/$31.00 ©2021 IEEE

For strings $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$, we denote their concatenation by $\boldsymbol{x} \circ \boldsymbol{y}$. We say that $\boldsymbol{v}$ is a *substring* of $\boldsymbol{x}$ if there exist strings $\boldsymbol{u}, \boldsymbol{w}$ (perhaps empty) such that $\boldsymbol{x} = \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{w}$. If the length of $\boldsymbol{v}$ is ℓ, we specifically say that $\boldsymbol{v}$ is an *ℓ-mer* of $\boldsymbol{x}$. Similarly, if $\boldsymbol{x} = \boldsymbol{u} \circ \boldsymbol{v}$ we say that $\boldsymbol{u}$ ($\boldsymbol{v}$) is a *prefix* (*suffix*, respectively) of x. An ℓ-mer which is also a prefix is an *l-prefix* (similarly, ℓ-suffix). For a multiset $S \in \mathcal{X}_{n,k}$, we let $\mathcal{L}_\ell(S)$ denote its *ℓ-profile*, i.e., the multiset of all ℓ-mers of all elements of S. For example, if $S = \{\{01010, 00101, 11101\}\}$ (which may be thought of as a multiset),

$$\mathcal{L}_3(S) = \{\{010, 101, 010, 001, 010, 101, 111, 110, 101\}\}.$$

By abuse of notation, we let $\mathcal{L}_\ell(\boldsymbol{x}) \triangleq \mathcal{L}_\ell(\{\{\boldsymbol{x}\}\})$, for $\boldsymbol{x} \in \Sigma^n$.

For a window of length ℓ, a code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is said to be a *multi-strand ℓ-reconstruction code* if for all for all $S, S' \in \mathcal{C}$ such that $S \neq S'$ it holds that $\mathcal{L}_\ell(S) \neq \mathcal{L}_\ell(S')$. Define

$$A_{n,k,\ell} \triangleq \{S \in \mathcal{X}_{n,k} : \mathcal{L}_\ell(S) \text{ is unique in } \mathcal{X}_{n,k}\},$$
$$B_{n,k,\ell} \triangleq \{\mathcal{L}_\ell(S) : S \in \mathcal{X}_{n,k}\}.$$

The case of $k = 1$ is of special interest; [9] introduced *repeat-free* strings, which we denote herein for all $\ell < n$ by

$$\mathcal{RF}_\ell^n \triangleq \{\boldsymbol{x} \in \Sigma^n : \|\mathcal{L}_\ell(\boldsymbol{x})\| = n - \ell + 1\}.$$

Observe that $\mathcal{RF}_\ell^n \subseteq A_{n,1,\ell+1}$ [29]. Moreover, an efficient algorithm reconstructs each $\boldsymbol{x} \in \mathcal{RF}_\ell^n$ from $\mathcal{L}_{\ell+1}(\boldsymbol{x})$ as follows. Given any $(\ell+1)$-mer $\boldsymbol{v}$ of $\boldsymbol{x}$, there exists a unique preceding $(\ell+1)$-mer $\boldsymbol{u}$ of $\boldsymbol{x}$ such that the ℓ-suffix of $\boldsymbol{u}$ equals the ℓ-prefix of $\boldsymbol{v}$ (unless $\boldsymbol{v}$ is a prefix of $\boldsymbol{x}$); similarly, a unique following $(\ell+1)$-mer. An extension of the same argument shows that if $S \in \mathcal{X}_{n,k}$ satisfies $\|\mathcal{L}_\ell(S)\| = (n-\ell+1)k$, then S is efficiently reconstructible from $\mathcal{L}_{\ell+1}(S)$, and in particular, $S \in A_{n,k,\ell+1}$.

We note that $A_{n,k,\ell}$ is a multi-strand ℓ-reconstruction code and that for any multi-strand ℓ-reconstruction code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$, $|\mathcal{C}| \leqslant |B_{n,k,\ell}|$. For all $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ (and, by abuse of notation, for $B_{n,k,\ell}$ as well) we denote the *rate* and *redundancy* of $\mathcal{C}$ by $R(\mathcal{C}) \triangleq \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|}$ and $\mathrm{red}(\mathcal{C}) \triangleq \log|\mathcal{X}_{n,k}| - \log|\mathcal{C}|$, respectively. Throughout the paper, we use the base-q logarithms where not otherwise indicated.

Finally, for two positive functions f, g of a common variable n, we say that $f = o_n(g)$ if $\limsup_{n\to\infty} \frac{f(n)}{g(n)} = 0$, $f = \Omega_n(g)$ if $\liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0$, and $f = O_n(g)$ if $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$. If clear from context, we omit the subscript from the aforementioned notations.

The main goal of this work is to find the minimum ℓ, as a function of n and k, such that the asymptotic rate of $A_{n,k,\ell}$ and $B_{n,k,\ell}$ approaches 1. We will also be interested in efficient constructions of multi-strand ℓ-reconstruction codes with asymptotic rate 1 while the value of ℓ will be close to the minimum value that accomplishes this rate result.

III. A Lower Bound on ℓ for Codes with Positive Rate

We begin by analyzing our channel, $\mathcal{X}_{n,k}$. Throughout this work, we assume in asymptotic analysis that as n grows, $\alpha \triangleq \limsup \frac{\log(k)}{n} < 1$.

Lemma 1 $\log|X_{n,k}| = nk - k\log(k/e) + o(k)$.

Proof: We note that

$$\frac{q^{nk}}{k!} \leqslant |\mathcal{X}_{n,k}| = \frac{q^{nk}}{k!} \prod_{j=0}^{k-1} \left(1 + \frac{j}{q^n}\right)$$
$$\leqslant q^{nk} \left(\frac{e}{k}\right)^k \left(\frac{1}{k} \sum_{j=0}^{k-1} \left(1 + \frac{j}{q^n}\right)\right)^k$$
$$\leqslant q^{nk} \left(\frac{e}{k} + \frac{e}{2q^n}\right)^k.$$

Recalling that $\log(k!) \leqslant \log\left(e\sqrt{k}(k/e)^k\right) = k\log(k/e) + O(\log(k))$, and observing for $\alpha < 1$ that $\frac{e}{k} + \frac{e}{2q^n} = \frac{e}{k}\left(1 + O(q^{-(1-\alpha)n})\right)$, the proof is concluded. ∎

We can now observe a lower bound on the required window length ℓ for multi-strand ℓ-reconstruction codes to have positive rate and in particular rate approaching 1.

Lemma 2 *Let $f : (0,\infty) \to (0,\infty)$ be any function satisfying $f(x) < \log(x)$ and $f(x) \underset{x\to\infty}{\longrightarrow} \infty$. Let $\ell \leqslant \log(nk) - f(nk)$. Then,*

$$R(A_{n,k,\ell}) \leqslant R(B_{n,k,\ell}) = o_{nk}(1).$$

Proof: We follow [5], by defining *profile vectors*, as follows. For every $S \in \mathcal{X}_{n,k}$ and $\boldsymbol{v} \in \Sigma^\ell$, let $f_S(\boldsymbol{v})$ count the number of times $\boldsymbol{v}$ appears in $\mathcal{L}_\ell(S)$. Clearly, if $S, S' \in \mathcal{X}_{n,k}$ satisfy $\mathcal{L}(S) \neq \mathcal{L}(S')$, then $f_S \neq f_{S'}$. Since $\sum_{v\in\Sigma^\ell} f_S(\boldsymbol{v}) = k(n-\ell+1)$, this implies that $|B_{n,k,\ell}| \leqslant \binom{k(n-\ell+1)+q^\ell-1}{q^\ell-1}$. Observe that $\frac{q^\ell}{nk} \underset{nk\to\infty}{\longrightarrow} 0$; then, based on $\binom{a}{b} \leqslant 2^{aH_2(b/a)}$, where H_2 is the binary entropy function (see, e.g., [19, Ch.10, Sec.11, Lem.7]), it is possible to derive that $\log|B_{n,k,\ell}| = o(nk)$. Recalling Lemma 1, we have that $\log|\mathcal{X}_{n,k}| \geqslant (n - \log(k))k = \Omega(nk)$, and thus the proposition is proven. ∎

IV. Constructions of Multi-strand Reconstruction Codes

In contrast to Lemma 2, we show in this section that if $\ell \geqslant \log(nk) + 2\log\log(nk) + 5$ as nk grows, then $R(B_{n,k,\ell}) \geqslant R(A_{n,k,\ell}) = 1 - o_{nk}(1)$. We shall do so by presenting two explicit constructions of multi-strand ℓ-reconstruction codes with efficient encoders and decoders.

Our constructions will be generic in the sense that they will apply any repeat-free encoding/decoding algorithm, and more specifically the ones from [9]. Since the algorithms from [9] use another component of run-length limited constrained (RLL) codes (for discussion of their use there, see the proof of Theorem 7), we first begin with an examination of known encoders for the well-studied $(0, M-1)_q$-RLL constraint (see, e.g., [22, Sec. 1.2]). The formal definition of these codes is given as follows.

Definition 3 *Let $Z(N, M)$ denote the set of length-N strings over Σ containing no zero-run of length M.*

While most previous works studied the case where M is fixed with respect to N, [17] studied this constraint when $M = \log(N) + O(1)$, and showed that $\mathrm{red}(Z(N,M)) = \Theta\big(N/q^M\big)$. Even though we could use the results from [17] for the derivation of the results in our paper, we next show how they can yet be improved. These improvements will be beneficial when deriving the parameters of the multi-strand reconstruction codes generated by our two constructions in this section. We start with the following lemma on the redundancy of the set $Z(N,M)$.

Lemma 4 $\mathrm{red}(Z(N,M)) \leqslant \frac{q-1}{q}(1+o_M(1))\frac{N}{q^M}$.

Proof: It is well-known (see, e.g., for the binary case [22, Exm. 3.3]), that for a fixed $M > 0$, $\lim \frac{1}{N}\log|Z(N,M)| = \log(\lambda)$, where λ is the unique real root strictly greater than 1 of the polynomial $p(x) = x^{M+1} - qx^M + q - 1$. Since $\log|Z(N,M)|$ is sub-additive in N, by Fekete's lemma (as was observed in [9]) $\lim \frac{1}{N}\log|Z(N,M)| = \inf \frac{1}{N}\log|Z(N,M)|$ for any fixed M. Hence, in particular, for all N it may be observed $\log|Z(N,M)| \geqslant N\log(\lambda)$. Equivalently, $\mathrm{red}(Z(N,M)) \leqslant N(1-\log(\lambda))$. Based on [13, App.], $1-\log(\lambda) \leqslant \frac{q-1}{q}\cdot\frac{1+o_M(1)}{q^M}$, which completes the proof. ∎

The authors of [17] also presented an algorithm with efficient encoder/decoder pair from $\Sigma^{N'}$ into $Z(N,M)$ (see Alg. 1 and the discussion at the end of Section III in [17]). It analyzed the case $q = 2$, and the resulting encoder requires $N - N' = 2\lceil N/q^{M-1}\rceil$ redundant symbols, which is the optimal order of magnitude. We can slightly improve upon this algorithms, when $q > 2$.

Lemma 5 *If $q > 2$, an efficient encoder/decoder pair into $Z(N,M)$ exists, requiring $\lceil N/\big(q^{M-1}(q-2)+M-1\big)\rceil$ redundant symbols.*

Proof: The concept of the encoder is similar to [17, Alg. 1]. First, the information string $\boldsymbol{x} \in \Sigma^{N'}$ is divided into blocks of length n, to be determined later. Then, in each block:

1) Append a 1.
2) From left to right, search for zero-runs of length M; if one is encountered, remove it, and append the index of its incidence to the block using M symbols, such that the last symbol is restricted not be be either $\{0,1\}$.
3) Continue, until no further zero-runs of length M exist.

Note that this process concludes in finite time (as in each iteration of step 2 it must advance in finite time, and the string length is preserved). Further, with the given restriction, M symbols may index a total of $q^{M-1}(q-2)$ positions for the beginning of the zero M-mer. It is therefore required to set $n = q^{M-1}(q-2)+M-1$.

Also observe that a possible decoder can use the last symbol to indicate whether a zero-run of length M was removed and indexed (which it can then inject in the correct place, discarding the index), or if the process is concluded (in which case the 1 suffix should also be discarded).

Next, since every encoded block ends with a nonzero symbol, these blocks can be concatenated without violating the constraint. Observe, then, that a single redundant symbol is added per block, hence the claimed overall redundancy.

Finally, note that both encoder and decoder operate in polynomial time in the input length. ∎

We are now ready to present two distinct constructions for multi-strand reconstruction codes. For convenience, we assume all quantities to have integer values; a straightforward adjustment of the described methods applies for all values.

A. Construction A

Our first construction of multi-strand reconstruction codes is next presented.

Construction A Let $\boldsymbol{x} \in \Sigma^m$ be an arbitrary information string, and encode it into an ℓ'-repeat-free string $\boldsymbol{c} = E(\boldsymbol{x}) \in \Sigma^{n'}$ using any known repeat-free encoder. Take $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_k \in \Sigma^{n'/k}$ such that $\boldsymbol{c} = \boldsymbol{c}_1 \circ \boldsymbol{c}_2 \circ \cdots \circ \boldsymbol{c}_k$. Let $f(i) \in \Sigma^{\log(k)}$ be a q-ary expansion of $i \in [1,k]$. Denote $\widetilde{\boldsymbol{c}}_i \triangleq f(i)\circ\boldsymbol{c}_i$, and let $n \triangleq n'/k + \log(k)$. Then,

$$\mathrm{Enc}_A(\boldsymbol{x}) \triangleq \{\{\widetilde{\boldsymbol{c}}_i : i = 1,\ldots,k\}\} \in \mathcal{X}_{n,k}.$$

□

The decoding success of Construction A follows from the next lemma.

Lemma 6 *For all $\boldsymbol{x} \in \Sigma^m$, it holds that $\mathrm{Enc}_A(\boldsymbol{x}) \in A_{n,k,\ell+1}$, where $\ell = \ell' + \log(k)$.*

Proof: Note that $\boldsymbol{c} = \boldsymbol{c}_1 \circ \boldsymbol{c}_2 \circ \cdots \circ \boldsymbol{c}_k \in \mathcal{RF}_{\ell'}^{n'}$ and thus $\|\mathcal{L}_{\ell'}(\{\{\boldsymbol{c}_i : i = 1,\ldots,k\}\})\| = (n'-\ell'+1)k$. It follows that $\{\{\boldsymbol{c}_i : i = 1,\ldots,k\}\} \in A_{n',k,\ell'+1}$.

Now, let $\boldsymbol{u},\boldsymbol{v}$ be $(\ell+1)$-mers of $\widetilde{\boldsymbol{c}}_i,\widetilde{\boldsymbol{c}}_j$ respectively; note that the $(\ell'+1)$-suffixes of $\boldsymbol{u},\boldsymbol{v}$ are $(\ell'+1)$-mers of $\boldsymbol{c}_i,\boldsymbol{c}_j$ respectively, and hence if $\boldsymbol{u} = \boldsymbol{v}$ then $i = j$ and the positions of both $(\ell'+1)$-mers agree. It follows that the positions of $\boldsymbol{u},\boldsymbol{v}$ agree as well, and the claim follows. ∎

Recall, then, that given $\mathcal{L}_{\ell+1}(\mathrm{Enc}_A(\boldsymbol{x}))$, an efficient algorithm can easily produce the set of strings $\widetilde{\boldsymbol{c}}_1,\widetilde{\boldsymbol{c}}_2,\ldots,\widetilde{\boldsymbol{c}}_k$ by simple stitching based on identical prefixes and suffixes of the $(\ell+1)$-mers in $\mathcal{L}_{\ell+1}(\mathrm{Enc}_A(\boldsymbol{x}))$. Then, by ordering and then removing the length-$\log(k)$ indices from these strings, we receive the string $\boldsymbol{c} = E(\boldsymbol{x})$, and consequently, $\boldsymbol{x}$. Note that the role of the indices in this construction is crucial to deduce the string $\boldsymbol{c} = E(\boldsymbol{x})$ from the set of strings $\widetilde{\boldsymbol{c}}_1,\widetilde{\boldsymbol{c}}_2,\ldots,\widetilde{\boldsymbol{c}}_k$. Without indices the order of these k strings could have not been derived so we could only know the string $\boldsymbol{c} = E(\boldsymbol{x})$ up to a permutation of its k sub-strings. The next theorem analyzes the parameters of codes that can be constructed using Construction A based upon the repeat-free encoders from [9].

Theorem 7 *Let n,k grow such that $\alpha = \limsup\frac{\log(k)}{n} < 1$. For the values of ℓ to be specified, take m such that $\mathrm{Enc}_A\colon \Sigma^m \to A_{n,k,\ell+1}$, and denote $\mathcal{C}_A \triangleq \mathrm{rng}(\mathrm{Enc}_A)$. Then*

1) One may choose a regime satisfying $\ell = \log(nk^2) + 2(\log\log(nk)) + O(1)$, while assuring $\mathrm{red}(\mathcal{C}_A) \leqslant \frac{q}{q-2}\cdot$

$\frac{nk}{\log^2(nk)} + k\log(e) + o(k)$. *(When $q = 2$, this is $2q\frac{nk}{\log^2(nk)} + k\log(e) + o(k)$.)*

2) *Allowing $\ell = (1+\epsilon)\log(nk) + \log(k) + O(1)$ for any $\epsilon > 0$, we have* $\mathrm{red}(\mathcal{C}_A) \leqslant \frac{q(1-\alpha+o(1))^{1-\epsilon}}{q-2} \cdot (nk)^{1-\epsilon} + k\log(e) + o(k)$. *(When $q = 2$, this is $2q(1-\alpha+o(1))^{1-\epsilon} \cdot (nk)^{1-\epsilon} + k\log(e) + o(k)$.)*
3) *Finally, when $\epsilon > 1$ in the definition of the last part, we have that* $\mathrm{red}(\mathcal{C}_A) = k\log(e) + o(k)$.

Note that the resulting redundancies are $o(nk)$.

Proof:

1) We may set, in the parameters of Construction A, $\ell' = \log(n') + 2\log\log(n') + 5$. We then have

$$\begin{aligned}\ell &= \ell' + \log(k) = \log(n'k) + 2\log\log(n') + 5\\ &= \log((n-\log(k))k^2) + 2\log\log((n-\log(k))k) + 5\\ &= \log(nk^2) + 2\log\log(nk) + O(1),\end{aligned}$$

where the last equality relies on $\alpha < 1$.

We now utilize in Construction A the repeat-free encoder described in [9, Alg. 3], which produces strings in $\mathcal{RF}^{n'}_{\log(n')+2\log\log(n')+5}$. It initializes by encoding an information string in Σ^m into $Z(n', 2\log\log(n'))$; this is done so that $0^{2\log\log(n')}$ may be used as a marker later on. Following steps of encoding the result into repeat-free strings interestingly require no further redundancy. We observed that efficient encoders exist into $Z(n', 2\log\log(n'))$ using less than $\frac{q}{q-2} \cdot \frac{n'}{\log^2(n')}$ redundant symbols (in the case of $q = 2$, the coefficient is based on [17] as described above).

It follows that $m = n' - \frac{q}{q-2} \cdot \frac{n'}{\log^2(n')}$, hence

$$\begin{aligned}\mathrm{red}(\mathcal{C}_A) &= \log|\mathcal{X}_{n,k}| - m\\ &= \frac{q}{q-2}\frac{nk - k\log(k)}{(\log(nk) + O(1))^2} + k\log(e) + o(k)\\ &\leqslant \frac{q}{q-2}\frac{nk}{\log^2(nk)} + k\log(e) + o(k).\end{aligned}$$

2) Note that no part of the encoder of [9, Alg. 3] is affected if initialization is done by encoding into $Z(n', \epsilon\log(n'))$ (and makers changed accordingly). It then produces strings in $\mathcal{RF}^{n'}_{(1+\epsilon)\log(n')+5}$. As observed, the initial encoding requires less than $\frac{q}{q-2} \cdot n'^{1-\epsilon}$ redundant symbols (and similarly for $q = 2$). It follows that

$$\begin{aligned}\mathrm{red}(\mathcal{C}_A) &= \log|\mathcal{X}_{n,k}| - m\\ &= \tfrac{q}{q-2}((n-\log(k))k)^{1-\epsilon} + k\log(e) + o(k)\\ &\leqslant \tfrac{q(1-\alpha+o(1))^{1-\epsilon}}{q-2}(nk)^{1-\epsilon} + k\log(e) + o(k).\end{aligned}$$

In this case,

$$\begin{aligned}\ell &= \ell' + \log(k) = (1+\epsilon)\log(n') + \log(k) + 5\\ &= (1+\epsilon)\log(nk) + \log(k) + O(1).\end{aligned}$$

3) We analyze the cases where Construction A may utilize the repeat-free encoder described in [9, Alg. 1], which produces strings in $\mathcal{RF}^{n'}_{2\log(n')+2}$ and requires a single redundant symbol (see [9, Thm. 16]).

Note that $\ell' \geqslant 2\log(n') + 2$ if and only if

$$\begin{aligned}1 + \epsilon &\geqslant \frac{2\log((n-\log(k))k) + 2}{\log(nk)}\\ &= 2(1 - o(1)) + o(1),\end{aligned}$$

hence for sufficiently large n, k and all $\epsilon > 1$, the encoder of [9, Alg. 1] may be used; in this case, $m = n' - 1$, and

$$\mathrm{red}(\mathcal{C}_A) = \log|\mathcal{X}_{n,k}| - m = k\log(e) + o(k).$$

■

B. Construction B

While in Construction A we added indices in order to overcome the lack of ordering when the string $\boldsymbol{c} = E(\boldsymbol{x})$ is partitioned into k strings, in Construction B we tackle this constraint differently. We again partition $\boldsymbol{c} = E(\boldsymbol{x})$ to k strings but with overlapping segments between consecutive strings. The overlapping segments will guarantee in decoding that, given the set of k strings, there will be only one way to concatenate them into one long string. As opposed to Construction A, this also guarantees that there is no need to increase the length of the read ℓ-mers with respect to the one required by the repeat-free encoders.

Construction B Let $\boldsymbol{x} \in \Sigma^m$ be an arbitrary information string, and encode it into an ℓ-repeat-free string $\boldsymbol{c} = E(\boldsymbol{x}) \in \Sigma^{n'}$ using any known repeat-free encoder. For n, k such that $n' = nk - (k-1)\ell = (n-\ell)k + \ell$, define k length-n strings $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k \in \Sigma^n$ by $\boldsymbol{c}_i = (c_{i,1}, \ldots, c_{i,n})$, where

$$c_{i,j} \triangleq c_{(i-1)(n-\ell)+j}; \quad j = 0, \ldots, n-1.$$

Then,

$$\mathrm{Enc}_B(\boldsymbol{x}) \triangleq \{\{\boldsymbol{c}_i : i = 1, \ldots, k\}\} \in \mathcal{X}_{n,k}.$$

□

The decoding correctness of the information string $\boldsymbol{x}$ in Construction B follows from the following simple observation.

Lemma 8 *For all $\boldsymbol{x} \in \Sigma^m$ it holds that $\mathcal{L}_{\ell+1}(E(\boldsymbol{x})) = \mathcal{L}_{\ell+1}(\mathrm{Enc}_B(\boldsymbol{x}))$.*

Proof: Since $\boldsymbol{c}_i$ is a substring of $\boldsymbol{c}$ for all i, it follows that $\mathcal{L}_{\ell+1}(\mathrm{Enc}_B(\boldsymbol{x})) \subseteq \mathcal{L}_{\ell+1}(E(\boldsymbol{x}))$. For the other direction, note that for all $1 \leqslant i < k$, $\boldsymbol{c}_i, \boldsymbol{c}_{i+1}$ are overlapping substrings of $\boldsymbol{c}$, with a common substring of length ℓ; thus all $(\ell+1)$-substrings of $\boldsymbol{c}$ are also substrings of some $\boldsymbol{c}_i$. ■

Lemma 8 immediately implies the next corollary.

Corollary 9 $\mathrm{Enc}_B(\boldsymbol{x}) \in A_{n,k,\ell+1}$ *for all* $\boldsymbol{x} \in \Sigma^m$.

Proof: According to Lemma 8 and since $E(\boldsymbol{x}) \in \mathcal{RF}^{n'}_{\ell}$, the corollary's statement follows. ■

We are now ready to analyze the code parameters that Construction B can achieve, again using the repeat-free encoders from [9].

TABLE I
REDUNDANCY AND WINDOW LENGTH TRADE-OFF COMPARISON

Lower bound	$\log(nk)-\ell=\omega_{nk}(1) \implies R(A_{n,k,\ell}) \leqslant R(B_{n,k,\ell})=o_{nk}(1)$	
Case	Construction A	Construction B
1	$\ell=\log(nk)+\log(k)+2\log\log(nk)+O(1)$ $\mathrm{red}(\mathcal{C}_A) \leqslant \frac{q}{q-2}\cdot\frac{nk}{\log^2(nk)}+k\log(e)+o(k)$	$\ell=\log(nk)+2\log\log(nk)+5$ $\mathrm{red}(\mathcal{C}_B) \leqslant \frac{q}{q-2}\cdot\frac{nk}{\log^2(nk)}+k\log(n)(1+o(1))$
2 ($\epsilon>0$)	$\ell=(1+\epsilon)\log(nk)+\log(k)+O(1)$ $\mathrm{red}(\mathcal{C}_A) \leqslant \frac{q(1-\alpha+o(1))^{1-\epsilon}}{q-2}\cdot(nk)^{1-\epsilon}+k\log(e)+o(k)$	$\ell=(1+\epsilon)\log(nk)+5$ $\mathrm{red}(\mathcal{C}_B) \leqslant \frac{q}{q-2}\cdot(nk)^{1-\epsilon}+k\log\big(n^{1+\epsilon}k^{\epsilon}\big)(1+o(1))$
3	$\ell=(1+\epsilon)\log(nk)+\log(k)+O(1);\ \epsilon>1$ $\mathrm{red}(\mathcal{C}_A)=k\log(e)+o(k)$	$\ell=(1+\epsilon)\log(nk)+5;\ \epsilon\geqslant 1$ $\mathrm{red}(\mathcal{C}_B)=k\log(n^{1+\epsilon}k^{\epsilon})(1+o(1))$

Theorem 10 *Let n,k grow such that $\alpha=\limsup\frac{\log(k)}{n}<1$. For the values of ℓ to be specified, take m such that $\mathrm{Enc}_B\colon\Sigma^m\to A_{n,k,\ell+1}$, and denote $\mathcal{C}_B\triangleq\mathrm{rng}(\mathrm{Enc}_B)$. Then*

1) *Letting $\ell=\log(nk)+2\log\log(nk)+5$, we have $\mathrm{red}(\mathcal{C}_B)\leqslant\frac{q}{q-2}\cdot\frac{nk}{\log^2(nk)}+k\log(n)(1+o(1))$. (When $q=2$, this is $2q\frac{nk}{\log^2(nk)}+k\log(n)(1+o(1))$.)*
2) *Setting $\ell=(1+\epsilon)\log(nk)+5$ for any $\epsilon>0$, we have $\mathrm{red}(\mathcal{C}_B)\leqslant\frac{q}{q-2}\cdot(nk)^{1-\epsilon}+k\log\big(n^{1+\epsilon}k^{\epsilon}\big)(1+o(1))$. (For $q=2$, this is $2q(nk)^{1-\epsilon}+k\log\big(n^{1+\epsilon}k^{\epsilon}\big)(1+o(1))$.)*
3) *Finally, when $\epsilon\geqslant 1$ in the definition of the last part, we have $\mathrm{red}(\mathcal{C}_B)=k\log(n^{1+\epsilon}k^{\epsilon})(1+o(1))$.*

Proof:

1) We start by observing, since $n'\leqslant nk$, that the repeat-free encoder of [9, Alg. 3] may produce strings in $\mathcal{RF}_{\ell}^{n'}$, if initialization is done by encoding into $Z(n',2\log\log(nk))$ (instead of $Z(nk,2\log\log(nk))$). Following steps [9, Lem. 19–20, Lem. 23] are not affected.
   In this case, we have $m=n'-\frac{q}{q-2}\cdot\frac{n'}{\log^2(nk)}$ (again, the case $q=2$ is handled similarly). Therefore
$$\begin{aligned}\mathrm{red}(\mathcal{C}_B)&=\log|\mathcal{X}_{n,k}|-m\\&=\frac{q}{q-2}\cdot\frac{nk}{\log^2(nk)}+k(\ell-\log(k))+O(k)-\ell\\&=\frac{q}{q-2}\cdot\frac{nk}{\log^2(nk)}+k\log(n)(1+o(1)).\end{aligned}$$
2) The next part follows similarly to part 1, based on the adjusted encoder described in part 2 of Theorem 7.
3) Finally, note that is suffices that $\ell=(1+\epsilon)\log(nk)+5>2\log(n')+2$ to utilize the repeat-free encoder of [9, Alg. 1]; in this case, $m=n'-1$, hence
$$\begin{aligned}\mathrm{red}(\mathcal{C}_B)&=\log|\mathcal{X}_{n,k}|-m\\&=k(\ell-\log(k))+k\log(e)+o(k)-\ell+1\\&=k\log(n^{1+\epsilon}k^{\epsilon})(1+o(1)).\end{aligned}$$
■

It should be noted that Theorem 10 does not preclude the possibility that the encoder of part 3 requires more redundancy than that of part 1 (namely, in any asymptotic regime satisfying $\log(k)=\Omega(n)$); this is an oddity since $A_{n,k,\ell_1}\subseteq A_{n,k,\ell_2}$ for all $\ell_1\leqslant\ell_2$. We observe that it is inherent to Construction B that the last step might introduce more redundancy than is required by E, the repeat-free encoder utilized. Nevertheless, Theorem 10 is structured to take into account other asymptotic regimes, and should be applied accordingly in practice.

C. Comparison and Summary

We first seek to give a converse to Lemma 2 and establish the result on the minimum value of ℓ which guarantees that the asymptotic rate of multi-strand ℓ-reconstruction codes (in fact, $R(A_{n,k,\ell})$) is 1. This result is established in the next corollary using the results of Construction B.

Corollary 11 *For sufficiently large n,k satisfying $\limsup\frac{\log(k)}{n}<1$ and for $\ell\geq\log(nk)+2\log\log(nk)+5$, it holds that $R(B_{n,k,\ell})\geqslant R(A_{n,k,\ell})=1-o_{nk}(1)$.*

Proof: Observe that under the assumption $n-\log(k)=\Omega(n)$ we have from Lemma 1 that $\log|\mathcal{X}_{n,k}|=\Omega(nk)$. The proposition of part 1 of Theorem 10 now suffices to establish the corollary's statement. ■

Note that if one aims to achieve the same result using Construction A, then the minimum value of ℓ should be $\log(nk)+\log(k)+2\log\log(nk)+O(1)$ and hence there is a gap of roughly $\log(k)$ with respect to the result in Corollary 11. However, Construction A can support better redundancy for comparable values of ℓ. Since Constructions A and B provide codes with parameters that cannot be directly compared, Table I lists the parameters of each construction for different regimes of ℓ in order to have a better understanding of the trade-offs between the minimal window length ℓ and the constructions' redundancy from Theorems 7 and 10. In general, one can observe that the window length in Construction A should be larger than the one in Construction B but construction's redundancy is smaller.

Before concluding, we suggest that one might consider a slightly different channel definition to the one handled above where the k strands are required to be distinct from one another, i.e., when information is stored in the space $\mathcal{X}_{n,k}^{*}\triangleq\{S\subseteq\Sigma^n:|S|=k\}$. A priori, it seems feasible that the added restriction might allow for lower redundancy (when measured in $\mathcal{X}_{n,k}^{*}$). However, we note that $\left|\mathcal{X}_{n,k}^{*}\right|=\binom{q^n}{k}$, thus a similar development to Lemma 1 yields
$$q^{nk}\left(\frac{e}{k}-\frac{e}{2q^n}\right)^k\leqslant\left|\mathcal{X}_{n,k}^{*}\right|\leqslant\frac{q^{nk}}{k!}.$$
It follows that $\log\left|\mathcal{X}_{n,k}^{*}\right|=nk-k\log(k)+k\log(e)+o(k)$ as well. A careful examination reveals that Constructions A and B actually encode into the set
$$\big\{S\in\mathcal{X}_{n,k}^{*}:S\text{ has a unique }\ell\text{-profile}\big\},$$
and hence the results of this work also hold for this setup of the problem.

REFERENCES

[1] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "On reconstructing a string from its substring compositions," in *Proc. of the IEEE International Symposium on Information Theory*, Austin, Texas, USA, 2010, pp. 1238–1242.

[2] ——, "String reconstruction from substring compositions," *SIAM Journal on Discrete Mathematics*, vol. 29, no. 3, pp. 1340–1371, 2015.

[3] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proc. of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, USA*, 2004, pp. 910–918.

[4] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, vol. 14, 2013.

[5] Z. Chang, J. Chrisnata, M. F. Ezerman, and H. M. Kiah, "Rates of DNA sequence profiles for practical values of read lengths," *IEEE Trans. on Inform. Theory*, vol. 63, no. 11, pp. 7166–7177, Nov. 2017.

[6] C.-S. Chin, D. H. Alexander, P. Marks, A. A. Klammer, J. Drake, C. Heiner, A. Clum, A. Copeland, J. Huddleston, E. E. Eichler, S. W. Turner, and J. Korlach, "Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data," *Nature Methods*, vol. 10, no. 6, pp. 563–569, 2013.

[7] J. Chrisnata, H. M. Kiah, S. Rao, A. Vardy, E. Yaakobi, and H. Yao, "On the number of distinct k-decks: Enumeration and bounds," in *Proceedings of the 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, Vietnam*, Sep. 2019, pp. 519–524.

[8] M. Dudık and L. J. Schulman, "Reconstruction from subsequences," *Journal of Combinatorial Theory, Series A*, vol. 103, no. 2, pp. 337–348, 2003.

[9] O. Elishco, R. Gabrys, M. Médard, and E. Yaakobi, "Repeat-free codes," *IEEE Trans. on Inform. Theory*, 2021.

[10] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded sequences from multiset substring spectra," in *Proc. of the IEEE International Symposium on Information Theory, Vail, Colorado, USA*, Jun. 2018, pp. 2540–2544.

[11] R. Gabrys, S. Pattabiraman, and O. Milenkovic, "Reconstructing mixtures of coded strings from prefix and suffix compositions," in *Proceedings of the 2020 IEEE Information Theory Workshop (ITW'2020), Riva del Garda, Italy*, Apr. 2021.

[12] S. Ganguly, E. Mossel, and M. Racz, "Sequence assembly from corrupted shotgun reads," in *Proc. of the IEEE International Symposium of Information Theory*, Barcelona, Spain, 2016, pp. 265–269.

[13] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Duplication-correcting codes for data storage in the DNA of living organisms," *IEEE Trans. on Inform. Theory*, vol. 63, no. 8, pp. 4996–5010, Aug. 2017.

[14] A. R. Khan, M. T. Pervez, M. E. Babar, N. Naveed, and M. Shoaib, "A comprehensive study of de novo genome assemblers: Current challenges and future prospective," *Evolutionary Bioinformatics*, vol. 14, Jan. 2018, PMID: 29511353.

[15] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA sequence profiles," *IEEE Trans. on Inform. Theory*, vol. 62, no. 6, pp. 3125–3146, Jun. 2016.

[16] 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.

[17] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for DNA storage," *IEEE Trans. on Inform. Theory*, vol. 65, no. 6, pp. 3671–3691, Jun. 2019.

[18] N. Loman, J. Quick, and J. Simpson, "A complete bacterial genome assembled de novo using only nanopore sequencing data," *Nature Methods*, vol. 12, no. 8, pp. 733—735, 2015.

[19] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1978.

[20] B. Manvel, A. Meyerowitz, A. Schwenk, K. Smith, and P. Stockmeyer, "Reconstruction of sequences," *Discrete Mathematics*, vol. 94, no. 3, pp. 209–219, 1991.

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

[22] B. H. Marcus, R. M. Roth, and P. H. Siegel, "An introduction to coding for constrained systems," Oct. 2001, unpublished Lecture Notes. [Online]. Available: www.math.ubc.ca/~marcus/Handbook

[23] A. S. Motahari, G. Bresler, and D. Tse, "Information theory of DNA shotgun sequencing," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6273–6289, 2013.

[24] A. Motahari, K. Ramchandran, D. Tse, and N. Ma, "Optimal DNA shotgun sequencing: Noisy reads are as good as noiseless reads," in *Proc. of the IEEE International Symposium of Information Theory*, Istanbul, Turkey, 2013, pp. 1640–1644.

[25] S. L. Salzberg, "Mind the gaps," *Nature Methods*, vol. 7, no. 2, pp. 105–106, 2010.

[26] A. D. Scott, "Reconstructing sequences," *Discrete Mathematics*, vol. 175, no. 1-3, pp. 231–238, 1997.

[27] I. Shomorony, T. Courtade, and D. Tse, "Do read errors matter for genome assembly?" in *Proc. of the IEEE International Symposium of Information Theory*, Hong Kong, 2015, pp. 919–923.

[28] I. Shomorony, G. Kamath, F. Xia, T. Courtade, and D. Tse, "Partial DNA assembly: A rate-distortion perspective," in *Proc. of the IEEE International Symposium of Information Theory*, Barcelona, Spain, 2016, pp. 1799–1803.

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

[30] Y. Yehezkeally and N. Polyanskii, "On codes for the noisy substring channel," in *Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT'2021), Melbourne, Victoria, Australia*, Jul. 2021.