Adversarial Torn-paper Codes

Daniella Bar-Lev*, **Sagi Marcovich***, **Eitan Yaakobi***, and **Yonatan Yehezkeally**†

*Department of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel

†Institute for Communications Engineering, Technical University of Munich, 80333 Munich, Germany

***Abstract*—This paper studies the *adversarial torn-paper channel*. This problem is motivated by applications in DNA data storage where the DNA strands that carry the information may break into smaller pieces that are received out of order. Our model extends the previously researched probabilistic setting to the worst-case. We develop code constructions for any parameters of the channel for which non-vanishing asymptotic rate is possible and show that our constructions achieve optimal asymptotic rate while allowing for efficient encoding and decoding. Finally, we extend our results to related settings included multi-strand storage, presence of substitution errors, or incomplete coverage.**

I. Introduction

High density and extreme longevity make DNA an appealing medium for data storage, especially for archival purposes [4], [10], [33]. Advances in DNA synthesis and sequencing technologies and recent proofs of concept [6], [10], [11], [14], [15], [23] have ignited active research into the capacity and challenges of data storage in this medium.

An aspect of this medium is that typically only short DNA sequences may be read; information molecules are therefore broken up into pieces and then read out of order, such as in shotgun sequencing [7], [12], [21], [25]. Multiple channel models have recently been suggested and studied based on this property. An assumption of overlap in read substrings and (near) uniform coverage leads to the problem of string reconstruction from substring composition [3], [3], [7], [13], [20], [21], [27], [29]; on the contrary, assuming no overlap in read substrings leads to the *torn-paper problem* [22], [24], [30], a problem closely related to the shuffling channel [16], [17], [28], [32]. This problem is motivated by DNA-based storage systems, where the information is stored in synthesized strands of DNA molecules. However, during and after synthesis, the DNA strands may break into smaller segments and due to the lack of ordering among the strands in these systems, all broken segments can only be read out of order [30]. Thus, the goal is to successfully retrieve the data from this multiset collection of read segments of the broken DNA strands.

In the *torn-paper channel* [24], [30], also known as the *chop-and-shuffle channel* [22], a long information string is segmented into non-overlapping substrings and their length has some known distribution. The channel outputs an unordered collection of the substrings. Given the lengths' distribution, the goal is to determine the channel capacity and devise efficient coding techniques. The geometric distribution was first studied in [30], and later in [22] using the Varshamov-Tenengolts (VT) codes [31]. The work [24] considered almost arbitrary distributions while, additionally, extending the problem by introducing incomplete coverage, i.e., assuming some of the substrings are deleted with some probability.

The torn-paper channel was studied so far only in a probabilistic setting. The goal of this paper is to extend this channel to the worst case, referred to herein as the *adversarial torn-paper channel*. Namely, it is assumed that an information string is adversarially segmented into non-overlapping substrings, where the length of each substring is between $L_{\min}$ and $L_{\max}$, for some given $L_{\min}$ and $L_{\max}$. We note that while the capacity of the probabilistic channel was shown to depend on *average* substring length, this adversarial model is chosen here for ease of analysis. Observe that under this setting the average substring length might indeed approach $L_{\min}$.

We study the noiseless adversarial torn-paper channel for a single information string, as well as multiple strings, which is motivated by DNA sequencing technologies where multiple strings are sequenced simultaneously [9], [19], [26]. In the single string case, we also extend the model to either allow for substitution errors affecting the information string prior to segmentation, or for incomplete coverage due to deletion of several segments after the segmentation. In all cases we investigate the values of $L_{\min}$ and $L_{\max}$ that provide codes with non-vanishing asymptotic rates, and develop constructions of codes with efficient encoding and decoding algorithms, achieving asymptotically optimal rates.

The rest of this paper is organized as follows. In Section II, the definitions and notations that will be used throughout the paper are presented as well as a lower bound on $L_{\min}$ required for the existence of codes for the adversarial torn-paper channel with non-vanishing asymptotic rates. Then, in Section III we present the basic construction used throughout the paper for the noiseless case of the single-strand adversarial torn-paper channel, and extend it to the multi-strand case; in Section IV we extend our construction to the noisy settings, including substitution errors or incomplete coverage. Due to space limitations, some proofs have been delegated to an extended version in preprint [5].

II. Definitions and Preliminaries

Let Σ be a finite alphabet of size q. For convenience of presentation, we assume Σ is equipped with a ring structure, and in particular identify elements $0, 1 \in \Sigma$. For a positive integer n, let $[n]$ denote the set $[n] \triangleq \{0, 1, \ldots, n-1\}$. Let Σ^* denote the set of all finite strings over Σ. The length of a string $\boldsymbol{x} \in \Sigma^*$ is denoted by $|\boldsymbol{x}|$. For strings $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, 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 $|\boldsymbol{v}| = \ell$, we specifically say that $\boldsymbol{v}$ is an *ℓ-segment* of $\boldsymbol{x}$. If $|\boldsymbol{u}| = i$ then it said that i is the *location* of $\boldsymbol{v}$ in $\boldsymbol{x}$. We also consider the cases where $\boldsymbol{v}$ appears cyclically in $\boldsymbol{x}$ and refer to its location accordingly. We reserve the term *index* to elements of presented constructions.

In our setting, information is stored in an unordered collection of strings over Σ; it might be allowed for the same

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. D. Bar-Lev, S. Marcovich, and E. Yaakobi were supported in part by the United States-Israel BSF grant no. 2018048. Y. Yehezkeally was supported by a Carl Friedrich von Siemens postdoctoral research fellowship of the Alexander von Humboldt Foundation.

The first, second, and fourth authors contributed equally to this work.

string to appear with multiplicity in the collection, which is encapsulated in the following formal definition:

$$\mathcal{X}_{n,k} \triangleq \{S = \{\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}\}\} : \forall i, \boldsymbol{x}_i \in \Sigma^n\}.$$

Here, $\{\{a, a, b, \ldots\}\}$ denotes a multiset; i.e., elements appear with multiplicity (but no order). Note that $|\mathcal{X}_{n,k}| = \binom{k+q^n-1}{k}$. It is assumed that a message $S \in \mathcal{X}_{n,k}$ is read by segmenting all elements of S into non-overlapping substrings of lengths between some fixed values $L_{\min}$ and $L_{\max}$, and all segments are received, possibly with multiplicity, without order or information on which element they originated from. More formally, a *segmentation* of the string $\boldsymbol{x}$ is a multiset $\{\{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{m-1}\}\}$, where $\boldsymbol{x}$ can be presented as $\boldsymbol{x} = \boldsymbol{u}_0 \circ \boldsymbol{u}_1 \circ \cdots \circ \boldsymbol{u}_{m-1}$. In case $L_{\min} \leqslant |\boldsymbol{u}_i| \leqslant L_{\max}$ for $0 \leqslant i < m-1$ and $|\boldsymbol{u}_{m-1}| \leqslant L_{\max}$, then the segmentation is called an $(L_{\min}, L_{\max})$*-segmentation*. The multiset of all $(L_{\min}, L_{\max})$-segmentations of $\boldsymbol{x}$ is denoted by $\mathcal{T}_{L_{\min}}^{L_{\max}}(\boldsymbol{x})$ and is referred as the $(L_{\min}, L_{\max})$*-segmentation spectrum of* $\boldsymbol{x}$. These definitions are naturally extended for a multiset $S \in \mathcal{X}_{n,k}$, so a *segmentation* of S is a union (as a multiset) of segmentations of all the strings in S (and the same holds for an an $(L_{\min}, L_{\max})$-segmentation), and $\mathcal{T}_{L_{\min}}^{L_{\max}}(S)$, the $(L_{\min}, L_{\max})$*-segmentation spectrum of* S, is the multiset of all $(L_{\min}, L_{\max})$-segmentations of S.

Note that our channel model only restricts the length of the last segment to be at most $L_{\max}$. Such a relaxation is motivated in applications where segmentation of the strings occurs sequentially, so that it might happen that the last segment is shorter than $L_{\min}$, but not larger than $L_{\max}$.

A code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ is said to be an $(L_{\min}, L_{\max})$*-multistrand torn-paper code* if for any $S, S' \in \mathcal{C}$, $S \neq S'$, it holds that any possible $(L_{\min}, L_{\max})$-segmentations of S, S' are distinct. That is, $\mathcal{T}_{L_{\min}}^{L_{\max}}(S) \cap \mathcal{T}_{L_{\min}}^{L_{\max}}(S') = \emptyset$. For $k = 1$, we simply receive $(L_{\min}, L_{\max})$*-single strand torn-paper codes.*

In case $L_{\min} = L_{\max} = \ell$, then for convenience, we let $\mathcal{T}_\ell(\boldsymbol{x}) \triangleq \mathcal{T}_\ell^\ell(\boldsymbol{x})$ and $\mathcal{T}_\ell(S) \triangleq \mathcal{T}_\ell^\ell(S)$ and note that in this case $|\mathcal{T}_\ell(\boldsymbol{x})| = |\mathcal{T}_\ell(S)| = 1$. For example, if $S = \{\{01010, 00101, 11101\}\}$ (which may be thought of as a multiset), then $\mathcal{T}_2(S) = \{\{01, 01, 0, 00, 10, 1, 11, 10, 1\}\}$.

The definition of $\mathcal{T}_\ell(S)$ should be seen as a technical instrument in our analysis rather than an intrinsic part of the channel model. However, we note that $\mathcal{T}_{L_{\min}}(S) \subseteq \mathcal{T}_{L_{\min}}^{L_{\max}}(S)$ for all S and $L_{\min} \leqslant L_{\max}$, hence every $(L_{\min}, L_{\max})$-multistrand torn-paper code $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ satisfies

$$|\mathcal{C}| \leqslant |\{\mathcal{T}_{L_{\min}}(S) : S \in \mathcal{X}_{n,k}\}|.$$

For all $\mathcal{C} \subseteq \mathcal{X}_{n,k}$ we denote the *rate*, *redundancy* of $\mathcal{C}$ by $R(\mathcal{C}) \triangleq \frac{\log|\mathcal{C}|}{\log|\mathcal{X}_{n,k}|}$, $\operatorname{red}(\mathcal{C}) \triangleq \log|\mathcal{X}_{n,k}| - \log|\mathcal{C}|$, respectively. Throughout the paper, we use the base-q logarithms.

For two non-negative functions f, g of a common variable n, denoting $L \triangleq \limsup_{n\to\infty} \frac{f(n)}{g(n)}$ (in the wide sense) we say that $f = o_n(g)$ if $L = 0$, $f = \Omega_n(g)$ if $L > 0$, $f = O_n(g)$ if $L < \infty$, and $f = \omega_n(g)$ if $L = \infty$. If f is not positive, we say $f(n) = O_n(g(n))$ ($f(n) = o_n(g(n))$) if $|f(n)| = O_n(g(n))$ (respectively, $|f(n)| = o_n(g(n))$). We say that $f = \Theta_n(g)$ if $f = \Omega_n(g)$ and $f = O_n(g)$. If clear from context, we omit the subscript from aforementioned notations.

We conclude this section by observing a lower bound on the required segment length $L_{\min}$ for multi-strand torn-paper codes to achieve non-vanishing rates, and in particular rates approaching one. This result is stated in the next theorem.

Algorithm 1: Encoder for Construction A

```
Input:  $\boldsymbol{x} = (x_0, x_1, \ldots, x_{Km-1}) \in \Sigma^{Km}$
Output: $\mathrm{Enc}_A(\boldsymbol{x})$
for $i \leftarrow 0$ to $K-1$ do
    $\boldsymbol{x}_i \leftarrow (x_{im}, x_{im+1}, \ldots, x_{(i+1)m-1})$ // $|\boldsymbol{x}_i| = m$
    $\boldsymbol{y}_i \leftarrow E_m^{RLL}(\boldsymbol{x}_i)$ // $\boldsymbol{y}_i$ contains no zero runs
        of length $f(n)$
    $\boldsymbol{z}_i \leftarrow \boldsymbol{c}_i'' \circ 10^{f(n)}1 \circ \boldsymbol{y}_i$ // $|\boldsymbol{z}_i| = L_{\min}$
end
$\boldsymbol{z}_K \leftarrow \boldsymbol{c}_K'' \circ 10^{f(n)}10^{N_n(m)}$ // $|\boldsymbol{z}_K| = L_{\min}$
$\boldsymbol{z} \leftarrow \boldsymbol{z}_0 \circ \boldsymbol{z}_1 \circ \cdots \circ \boldsymbol{z}_K \circ 0^{n \bmod L_{\min}}$ // $|\boldsymbol{z}| = n$
return $\boldsymbol{z}$
```

Theorem 1 *Let $\mathcal{C}$ be any $(L_{\min}, L_{\max})$-mulststrand torn-paper code. Assuming $\log(k) = o(n)$, if $L_{\min} \leqslant (a + o_{nk}(1))\log(nk)$, for some $a \geqslant 1$, then $R(\mathcal{C}) \leqslant 1 - \frac{1}{a} + o_{nk}(1)$.*

III. A Construction of Torn Paper Codes

In this section, a construction of single-strand torn-paper codes is presented and is then extended for multiple strands. It is assumed from here on out that $L_{\min} = \lceil a\log(n)\rceil$, for some $a > 1$.

We propose the following construction of length-n $(L_{\min}, L_{\max})$-single strand torn-paper codes.

Construction A Denote $I \triangleq \lceil \log(n/L_{\min}) \rceil$ and $K \triangleq \lfloor n/L_{\min} \rfloor - 1$. Let f be any function satisfying $f(n) = \omega(1)$ and $f(n) = o(\log(n))$. The construction is based on the following two ingredients:

- *Index generation.* Let $(\boldsymbol{c}_i)_{i\in[q^I]}$, $\boldsymbol{c}_i \in \Sigma^I$ be a q-ary Gray code. Denote by $\boldsymbol{c}_i'$ the concatenation of $\boldsymbol{c}_i$ with a single parity symbol (i.e., the sum of the entries in $\boldsymbol{c}_i'$ is zero). Further, denote by $\boldsymbol{c}_i''$ the result of inserting '1's into $\boldsymbol{c}_i'$ at every location divisible by $f(n)$. (Since the locations of substrings start with 0, the first bit of $\boldsymbol{c}_i''$ is always 1.) Note that for all $i \in [q^I]$, $\alpha \triangleq |\boldsymbol{c}_i''| = \left\lceil \frac{f(n)}{f(n)-1}(I+1) \right\rceil$. We refer to $\boldsymbol{c}_i$ (or simply i) as an *index* in the construction and to $\boldsymbol{c}_i''$ as an *encoded index.*
- *Run-length limited (RLL) encoding.* We use an RLL encoder which receives strings of length m and returns strings of length $N_n(m)$ that do not contain length-$f(n)$ '0'-runs. Constructions of such encoders can be taken from [18] or [34, Lem. 5]. Denote this encoder by E_m^{RLL}.

The constructed $(L_{\min}, L_{\max})$-single strand torn-paper code, denoted by $\mathcal{C}_A(n)$, is carried as follows. Let m be an integer such that $N_n(m) = L_{\min} - |\boldsymbol{c}_i''| - f(n) - 2 = L_{\min} - \alpha - f(n) - 2$. The construction's encoder, $\mathrm{Enc}_A : \Sigma^{Km} \to \Sigma^n$, is given in Algorithm 1. □

In the rest of the paper, we call the strings $\boldsymbol{x}_i$ (respectively, $\boldsymbol{y}_i$) in the constructions an *information block* (*encoded block*). The string $\boldsymbol{z}_i$ will be referred to as a segment of $\boldsymbol{z}$. Observe that once the encoded blocks $\boldsymbol{y}_i$'s are obtained, encoding (including the generation of the Gray code) requires a number of operations linear in n. By [18], [34], encoding each $\boldsymbol{x}_i$ into $\boldsymbol{y}_i$ may also be achieved with a linear number of operations. Hence, the complexity of Construction A is linear with n.

Next, it is shown that the constructed code $\mathcal{C}_A(n)$ is an $(L_{\min}, L_{\max})$-single strand torn-paper code.

Theorem 2 *For all admissible values of n, the code $\mathcal{C}_A(n)$ is an $(L_{\min}, L_{\max})$-single strand torn-paper code with a linear-run-time decoder.*

The proof of Theorem 2 is carried by presenting an explicit decoder to $\mathcal{C}_A(n)$. Let $\boldsymbol{z} \in \mathcal{C}_A(n)$ and let $\boldsymbol{z} = \boldsymbol{u}_0 \circ \boldsymbol{u}_1 \circ \cdots \circ \boldsymbol{u}_{s-1}$ so that $\{\{\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_{s-1}\}\}$ is an $(L_{\min}, L_{\max})$-segmentation of $\boldsymbol{z}$. The main task of the decoding algorithm is to successfully retrieve the location within $\boldsymbol{z}$ of each of the s segments of the $(L_{\min}, L_{\max})$-segmentation. For every segment $\boldsymbol{u}_j$, $j \in [s]$, the decoder first finds the location i such that the first (maybe partial) occurrence of an encoded index in the segment $\boldsymbol{u}_j$ is of $\boldsymbol{c}''_i$. Given i and the location of $\boldsymbol{c}''_i$ in $\boldsymbol{u}_j$, the location of the segment $\boldsymbol{u}_j$ within $\boldsymbol{z}$ can be calculated. Then, according to the location in $\boldsymbol{z}$ for each segment in the $(L_{\min}, L_{\max})$-segmentation, one can simply concatenate the segments in the correct order to obtain the code-word $\boldsymbol{z}$. Finally, by removing the markers and the encoded indices and applying the RLL decoder for each of the strings $\boldsymbol{y}_i$'s, the information string $\boldsymbol{x}$ is retrieved.

Consider the case where a segment $\boldsymbol{u}$ is a proper substring of the suffix of $\boldsymbol{z}$ of length $(n \bmod L_{\min}) + N_n(m) + f(n)$, i.e., $\boldsymbol{u}$ begins with a proper suffix of $\boldsymbol{z}_K 0^{n \bmod L_{\min}}$ (note that this does not imply that $\boldsymbol{u}$ is itself a suffix of $\boldsymbol{z}$). Then, $\boldsymbol{u}$ does not intersect $\boldsymbol{y}_i$ for any $i \in [K]$, and may safely be discarded. We see next that these cases may be identified efficiently.

Lemma 3 *Let $\boldsymbol{z} \in \mathcal{C}_A(n)$, and let $\boldsymbol{u}$ be a substring of $\boldsymbol{z}$, $|\boldsymbol{u}| = L$. If $\boldsymbol{u}$ begins with a proper suffix of $\boldsymbol{z}_K 0^{n \bmod L_{\min}}$, then for all sufficiently large n this fact can efficiently be identified.*

By Lemma 3, it is sufficient to retrieve the location of any segment which is not a substring of the suffix of length $(n \bmod L_{\min}) + N_n(m) + f(n)$ of $\boldsymbol{z}$. For any such $\boldsymbol{u}$, the calculation of the index i such that $\boldsymbol{c}''_i$ is the first (perhaps partial) occurrence of an encoded index within $\boldsymbol{u}$, is given in Algorithm 2.

Any L-segment $\boldsymbol{u}$ of $\boldsymbol{z} \in \mathcal{C}_A(n)$, such that $L \geqslant L_{\min}$, contains at least part of one of the encoded indices $\boldsymbol{c}''_i$. If $\boldsymbol{c}''_i$ is the first encoded index to intersect $\boldsymbol{u}$, we denote by $\mathrm{Ind}(\boldsymbol{u}) \triangleq i$ the *index of* $\boldsymbol{u}$. Note that this index does not depend on the information that was encoded in the construction. Algorithm 2 ensures that it is possible to determine the index of every L-segment $\boldsymbol{u}$ of $\boldsymbol{z}$, where $L \geqslant L_{\min}$.

Lemma 4 *Let $\boldsymbol{z} \in \mathcal{C}_A(n)$, $L \geqslant L_{\min}$, and let $\boldsymbol{u}$ be an L-segment of $\boldsymbol{z}$ which is not a substring of the suffix of length $(n \bmod L_{\min}) + N_n(m) + f(n)$ of $\boldsymbol{z}$. Then, Algorithm 2 successfully returns the index $\mathrm{Ind}(\boldsymbol{u})$ of $\boldsymbol{u}$.*

We remark that the described procedure operates in run-time which is linear in the substring length. In addition, if $\boldsymbol{z}$ can be reconstructed from its non-overlapping substrings, then the strings $\boldsymbol{y}_i$'s are readily obtained, and $\boldsymbol{x}$ may be decoded (again, see [18], [34]). These algorithms also require a linear number of operations. This completes the proof of Theorem 2.

Lastly the redundancy of Construction A is analyzed.

```
Algorithm 2: Index retrieval from a segment
  Input: An $L$-segment $\boldsymbol{u}$ of a code-word of $\mathcal{C}_A(n)$,
         where $L \geqslant L_{\min}$.
  Output: The index of $\boldsymbol{u}$ within $\boldsymbol{z}$, $\mathrm{Ind}(\boldsymbol{u})$
  $\boldsymbol{u}' \leftarrow$ the $L_{\min}$-length prefix of $\boldsymbol{u}$
  $j \leftarrow$ the starting index of the unique occurrence of
    $10^{f(n)}1$ within $\boldsymbol{u}'$; if none exists, of the cyclic
    occurrence
  $\boldsymbol{c}'' \leftarrow$ the (cyclic) $\alpha$-substring of $\boldsymbol{u}$ strictly preceding $j$
  $\boldsymbol{c}' \leftarrow$ the non-padded subsequence of $\boldsymbol{c}''$
  $\boldsymbol{c} \leftarrow$ the $I$-prefix of $\boldsymbol{c}'$
  Ind $\leftarrow$ the index of $\boldsymbol{c}$ in the Gray code
  if the last symbol of $\boldsymbol{c}'$ is not the parity of $\boldsymbol{c}$ then
    | Ind $\leftarrow$ Ind $-1$
  end
  return Ind
```

Theorem 5 *For all admissible values of n and $f(n)$ in Construction A, where $f(n) = \omega(1)$, $f(n) = o(\log(n))$ and with the RLL encoders of [18], [34], it holds that*

$$\mathrm{red}(\mathcal{C}_A(n)) \leqslant \frac{n}{a}\left(1 + \frac{f(n)}{\log(n)} + \frac{1 + o(1)}{f(n)} + 2a^2 \frac{\log(n)}{n}\right).$$

In particular, for $f(n) = (1 + o(1))\sqrt{\log(n)}$, $\mathrm{red}(\mathcal{C}_A(n)) \leqslant \frac{n}{a}\left(1 + \frac{2+o(1)}{\sqrt{\log(n)}}\right)$.

Next, we consider the case of $k > 1$ and $\log(k) = o(n)$. We know from Theorem 1 that if $\limsup \frac{L_{\min}}{nk} \leqslant 1$ then any family of $(L_{\min}, L_{\max})$-multistrand torn-paper codes will only achieve vanishing asymptotic rate; hence we assume $L_{\min} = \lceil a \log(nk) \rceil$ for some $a > 1$. The following theorem summarizes our main results regarding $(L_{\min}, L_{\max})$-multistrand torn-paper codes.

Theorem 6 *Take n, k such that $k > 1$, $\log(k) = o(n)$, and let $L_{\min} = \lceil a \log(nk) \rceil$, for $a > 1$. There exists a linear run-time (in the substrings length, i.e., nk) encoder-decoder pair for $(L_{\min}, L_{\max})$-multistrand torn-paper codes achieving $1 - \frac{1}{a} - o_{nk}(1)$ asymptotic rate.*

IV. Error-Correcting Torn-paper Codes

In this section, we extend the study of torn-paper codes to the noisy setup. We consider two models of noise. The first one assumes that the encoded string, before segmentation, suffers at most some t substitution errors. The second model corresponds to the case where some of the segments are deleted during segmentation.

A. Substitution-Correcting Torn-paper Codes

For a string $\boldsymbol{x}$, its *t-error torn-paper ball*, denoted by $\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}; t)$, is defined by all possible $(L_{\min}, L_{\max})$-segmentations after introducing at most t errors to $\boldsymbol{x}$, that is,

$$\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}; t) \triangleq \bigcup_{\boldsymbol{y} \in B_t(\boldsymbol{x})} \mathcal{T}_{L_{\min}}^{L_{\max}}(\boldsymbol{y}),$$

where $B_t(\boldsymbol{x}) = \{\boldsymbol{y} : d_H(\boldsymbol{x}, \boldsymbol{y}) \leqslant t\}$. A code $\mathcal{C}$ is called a *t-error single-strand torn-paper code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ it holds that $\mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_1; t) \cap \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_2; t) = \emptyset$.

Our goal in this section is to show how to adjust Construction A in order to produce t-error single-strand torn-paper codes. We first explain the main ideas of the required modifications. Let $\boldsymbol{z} = \mathrm{Enc}_A(\boldsymbol{x}) \in \mathcal{C}_A(n)$ (encoded with Algorithm 1) and let $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ be an $(L_{\min}, L_{\max})$-segmentation of some word $\boldsymbol{z}'$, where $d_H(\boldsymbol{z}, \boldsymbol{z}') \leqslant t$. The main task of the noiseless decoder of $\mathcal{C}_A(n)$ was to first calculate the index, and thus the location in $\boldsymbol{z}$, of every segment $\boldsymbol{u} \in \mathcal{U}$. However, in the presence of errors, calculating the index of a segment $\boldsymbol{u} \in \mathcal{U}$ based on the first (perhaps partial) occurrence of an encoded index within $\boldsymbol{u}$ might result with the misplacement of all the (perhaps partial) information blocks $\boldsymbol{y}_i$ that are contained in $\boldsymbol{u}$. Hence, a more careful approach is necessary for index decoding.

Before presenting our construction for t-error single-strand torn-paper codes, we introduce additional required definitions. For $\boldsymbol{u} \in \Sigma^*$, define $\mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u})$ to be the multiset of non-overlapping $L_{\min}$-segments of $\boldsymbol{u}$, where the last segment is of length ℓ, $L_{\min} \leqslant \ell < 2L_{\min}$. A segment $\boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u})$ is called *A-decodable* if $\boldsymbol{w}$ satisfies one of the following.

1) $\boldsymbol{w}$ either contains a unique complete occurrence of $10^{f(n)}1$, or it doesn't contain complete occurrences but contains a cyclic occurrence (i.e., has a suffix-prefix pair whose concatenation is $10^{f(n)}1$.
2) $\boldsymbol{w}$ contains precisely two complete occurrences of $10^{f(n)}1$, and there exist a unique pair of occurrences (either complete or complete-to-suffix/prefix) whose locations are at distance precisely $L_{\min}$.

Let $\boldsymbol{w}$ be an A-decodable segment. Then, by definition, there is at least one occurrence (perhaps cyclic) of $10^{f(n)}1$ within $\boldsymbol{w}$ and, if there is more than a single occurrence, then there is exactly one pair of occurrences such that the difference between their locations is $L_{\min}$. Consider the α-segments of $\boldsymbol{w}$ preceding these occurrences as encoded indices; if the (first) occurrence of $10^{f(n)}1$ in $\boldsymbol{w}$ is at location $\ell < \alpha$, concatenate the $(\alpha - \ell)$-segment of $\boldsymbol{w}$ at location $L_{\min} + \ell - \alpha$, to the ℓ-prefix of $\boldsymbol{w}$, and consider the resulting length-α string to be a *cyclic encoded index*.

An A-decodable segment $\boldsymbol{w}$ is called *valid* if it satisfies one of the following conditions:

1) $\boldsymbol{w}$ contains no complete encoded index, hence it contains only a cyclic encoded index.
2) $\boldsymbol{w}$ contains a single complete encoded index, and its parity symbold is correct.
3) $\boldsymbol{w}$ contains two complete encoded indices, and either exactly one of their parity symbols is correct, or both are correct and the indices are consecutive.

Construction B This construction uses Construction A with its coding components and parameters, and in particular the value of K and m, as follows. Let $\mathcal{C}_{\mathrm{EC}}$ be a $(K, q^{mM}, 2t+1)_{q^m}$ error-correcting code with an encoder algorithm $\mathrm{Enc}_{\mathrm{EC}} \colon (\Sigma^m)^M \to (\Sigma^m)^K$. The constructed t-error $(L_{\min}, L_{\max})$-single-strand torn-paper code, denoted by $\mathcal{C}_B(n)$, is defined by the encoder $\mathrm{Enc}_B : \Sigma^{Mm} \to \Sigma^n$ given by, $\mathrm{Enc}_B(\boldsymbol{x}) = \mathrm{Enc}_A(\mathrm{Enc}_{\mathrm{EC}}(\boldsymbol{x}))$, for all $\boldsymbol{x} \in \Sigma^{Mm}$. □

Assume one retrieves a noisy version $\boldsymbol{z}'$ of $\boldsymbol{z} = \mathrm{Enc}_B(\boldsymbol{x})$, such that $\boldsymbol{z}, \boldsymbol{z}'$ agree on all locations containing encoded indices $\boldsymbol{c}_i''$ or markers $10^{f(n)}1$ (as their locations in $\boldsymbol{z}$ do not depend on the information $\boldsymbol{x}$). Thus, one extracts from $\boldsymbol{z}'$ (perhaps erroneous) encoded information blocks, denoted $\boldsymbol{y}_i'$. Denote by e the number of encoded information blocks that were not recovered, and by s the number of encoded blocks that were recovered incorrectly (i.e., $\boldsymbol{y}_i' \neq \boldsymbol{y}_i$). Since the information string is encoded using a $(K, q^{mM}, 2t+1)_{q^m}$ error-correcting code, it suffices that $2s + e \leqslant 2t$ to guarantee correct decoding.

In order to reconstruct a noisy version $\boldsymbol{z}'$ of $\boldsymbol{z}$, we define a modification of Algorithm 2, as follows. First, given $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ we apply the reconstruction algorithm not directly to $\mathcal{U}$, but rather to valid segments in $\mathcal{T}_{L_{\min}}^{+}(\mathcal{U}) \triangleq \left\{\left\{\mathcal{T}_{L_{\min}}^{+}(\boldsymbol{u}) : \boldsymbol{u} \in \mathcal{U}\right\}\right\}$. Secondly, in case a valid $\boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\mathcal{U})$ contains multiple (perhaps cyclic) occurrences of an encoded index, the algorithm selects one to decode by prioritizing complete occurrences over cyclic ones, and then accepting the first containing a correct parity symbol. Decoding of the selected encoded index is then performed as described in Algorithm 2, and denoted by $\mathrm{Ind}'(\boldsymbol{w})$.

For $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$, we define the set $\mathcal{Z}(\mathcal{U}) \triangleq \left\{(\mathrm{Ind}'(\boldsymbol{w}), \boldsymbol{w}) : \boldsymbol{w} \in \mathcal{T}_{L_{\min}}^{+}(\mathcal{U}) \text{ is valid}\right\}$. If $(j, \boldsymbol{w}), (j, \boldsymbol{w}') \in \mathcal{Z}(\mathcal{U})$ for some j and $\boldsymbol{w} \neq \boldsymbol{w}'$, we define a restriction $\mathcal{Z}'(\mathcal{U})$ of $\mathcal{Z}(\mathcal{U})$ by including only the shortest, lexicographically-least, segment (i.e., $\mathcal{Z}'(\mathcal{U})$ defines a proper function). Given the set $\mathcal{Z}'(\mathcal{U})$, a string $\boldsymbol{z}'$ is decoded:

1) Fill the encoded indices and the markers in $\boldsymbol{z}'$ in the correct locations as defined in Algorithm 1 (note again that these locations do not depend on the information).
2) Next, we iterate over any pair $(\mathrm{Ind}'(\boldsymbol{w}), \boldsymbol{w}) \in \mathcal{Z}'(\mathcal{U})$ and update $\boldsymbol{z}'$ with the symbols of the encoded blocks $\boldsymbol{y}_i$'s within $\boldsymbol{w}$; If there is a collision of symbols in the same position within an encoded block $\boldsymbol{y}_i'$ for some i, $i \in [K]$, we erase $\boldsymbol{y}_i'$ completely from $\boldsymbol{z}'$.
3) If an encoded block $\boldsymbol{y}_i'$ is partially filled at the end of the process (i.e., there are missing symbols within $\boldsymbol{y}_i'$) we erase the encoded block $\boldsymbol{y}_i'$.

The output $\boldsymbol{z}'$ of this decoding procedure over the segmentation $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ is denoted by $\mathrm{Dec}_B(\mathcal{U}) \triangleq \boldsymbol{z}'$.

Let $\boldsymbol{z} = \mathrm{Enc}_B(\boldsymbol{x})$, $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$ and let $\boldsymbol{z}' = \mathrm{Dec}_B(\mathcal{U})$ be the noisy version of $\boldsymbol{z}$ reconstructed by the aforementioned algorithm. According to the decoding procedure, $\boldsymbol{z}'$ and $\boldsymbol{z}$ can differ only in the values of the encoded blocks $\boldsymbol{y}_i$. Denote by e the number of encoded blocks that were not recovered in $\boldsymbol{z}'$ and let s denote the number of encoded blocks in $\boldsymbol{z}'$ that were recovered incorrectly with respect to $\boldsymbol{z}$.

Lemma 7 *Let $\boldsymbol{z} = \mathrm{Enc}_B(\boldsymbol{x})$, $\mathcal{U} \in \mathcal{BT}_{L_{\min}}^{L_{\max}}(\boldsymbol{z};t)$, and $\boldsymbol{z}' = \mathrm{Dec}_B(\mathcal{U})$. Then, it holds that $2s + e \leqslant 2t$, where e, s are defined as previously explained.*

Theorem 8 *Denote the redundancy of the code $\mathcal{C}_{EC}$ used in Construction B by $\rho_{EC} \triangleq K - M$. Then, operating Enc_A as in Theorem 5, with $f(n) = (1+o(1))\sqrt{\log(n)}$, we have*

$$\mathrm{red}(\mathcal{C}_B) \leqslant \frac{n}{a}\left(1 + \frac{2+o(1)}{\sqrt{\log(n)}}\right) + \rho_{EC}\Big((a-1)\log(n) - (2+o(1))\sqrt{\log(n)}\Big).$$

Furthermore, when $a > 2$ then the code $\mathcal{C}_{EC}$ can be an MDS code and hence $\rho_{EC} = 2t$.

B. Deletion-Correcting Torn-paper Codes

For a string $\boldsymbol{x}$, its *t-deletion torn-paper ball*, $\mathcal{DT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x};t)$, is defined as all the subsets with at most t missing segments of all the possible $(L_{\min}, L_{\max})$-segmentations of $\boldsymbol{x}$, that is,

$$\mathcal{DT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x};t) \triangleq \bigcup_{S\in\mathcal{T}_{L_{\min}}^{L_{\max}}(\boldsymbol{x})} \{S' \subseteq S : |S|-|S'| \leqslant t\}.$$

A code $\mathcal{C}$ is called a *t-deletion torn-paper code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$ it holds that $\mathcal{DT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_1;t)\cap\mathcal{DT}_{L_{\min}}^{L_{\max}}(\boldsymbol{x}_2;t)=\emptyset$.

In this section, we utilize *burst-erasure-correcting (BEC) codes* in our constructions, which are defined next. For a string $\boldsymbol{x}$, its *t-burst L-erasures ball*, denoted by $\mathcal{B}_{\mathrm{BE}}^{L}(\boldsymbol{x};t)$, is defined as the set of all strings that can be obtained from $\boldsymbol{x}$ by at most t burst of erasures, each of length at most L. A code $\mathcal{C}$ is called a *t-burst L-erasure correcting code* if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{C}$, $\mathcal{B}_{\mathrm{BE}}^{L}(\boldsymbol{x}_1;t)\cap\mathcal{B}_{\mathrm{BE}}^{L}(\boldsymbol{x}_2;t)=\emptyset$. Next, we present a construction of t-deletion torn-paper codes. Let $\hat{L}_{\max} \triangleq L_{\max} - \left\lceil \frac{L_{\max}}{L_{\min}} \right\rceil(\alpha + f(n) + 2)$. This construction is based on Construction A and assumes the existence of a systematic linear t-burst $\hat{L}_{\max}$-erasure correcting code, denoted by $\mathcal{C}_{\mathrm{BEC}}$.

Construction C Let $\rho > 0$ be an integer that is determined next. This construction uses the following family of codes:

Systematic BEC encoding. Let $\mathrm{Enc}_{\mathrm{BEC}} \colon \Sigma^{(K-\rho)N_n(m)} \to \Sigma^{\rho_{\mathrm{BEC}}}$ denote the systematic encoder of the code $\mathcal{C}_{\mathrm{BEC}}$, such that for any string $\boldsymbol{v} \in \Sigma^{(K-\rho)N_n(m)}$, $\boldsymbol{v} \circ \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{v}) \in \mathcal{C}_{\mathrm{BEC}}$ (for convenience we assume that $\mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{v})$ returns only the encoded systematic redundancy symbols). The redundancy of this encoder is denoted by ρ_{BEC}. The parameter ρ is defined $\rho \triangleq \left\lceil \frac{\rho_{\mathrm{BEC}}}{N_n(m)} \right\rceil \cdot \left\lfloor \frac{f(n)}{f(n)-1} \right\rfloor$.

We amend Construction A as follows:

1) *The length of the input string $\boldsymbol{x}$*. The input of this construction is $\boldsymbol{x} \in \Sigma^{(K-\rho)m}$. That is, this construction has additional redundancy of ρm symbols compared to Construction A. The input string is divided to $K-\rho$ information blocks each of length m, denoted by $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{K-\rho-1}$.
2) *The generation of the encoded blocks $\boldsymbol{y}_i$'s*. The first $K-\rho$ blocks are generated from the corresponding $\boldsymbol{x}_i$'s using the RLL encoder E_m^{RLL} similarly to Construction A. Let $\boldsymbol{y}^* \triangleq \boldsymbol{y}_0 \circ \cdots \circ \boldsymbol{y}_{K-\rho-1} \in \Sigma^{(K-\rho)N_n(m)}$ denote their concatenation. Next, we apply $\mathrm{Enc}_{\mathrm{BEC}}$ to obtain $\boldsymbol{w} \triangleq \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{y}^*)$, and denote by $\boldsymbol{w}^*$ the result of inserting '1's into $\boldsymbol{w}$ at every location divisible by $f(n)$ (in particular, $\boldsymbol{y}^* \circ \boldsymbol{w}^*$ does not contain a length-$f(n)$ zero-run). Then, $\boldsymbol{w}^*$ is divided to the remaining segments $\boldsymbol{y}_{K-\rho}, \ldots, \boldsymbol{y}_{K-1} \in \Sigma^{N_n(m)}$ (if $|\boldsymbol{w}^*|$ is not a multiple of $N_n(m)$, $\boldsymbol{y}_{K-1}$ is padded with 1's to length $N_n(m)$). Note that the parameter ρ satisfies $\rho N_n(m) \geqslant \rho_{BEC} \cdot \left\lfloor \frac{f(n)}{f(n)-1} \right\rfloor = |\boldsymbol{w}^*|$.

From here we continue identically to Construction A. That is, an index and a marker are appended to the beginning of each encoded block $\boldsymbol{y}_i$ to construct a segment $\boldsymbol{z}_i$ of length $L_{\min}$. Then, $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{K-1}$ are concatenated along with additional redundancy symbols to construct the output string $\boldsymbol{z}$. □

The correctness of Construction C and redundancy calculation are proved in the next theorem. Let $\mathcal{C}_{\mathrm{del}}(n)$ denote the constructed code.

Theorem 9 *For all admissible values of n, the code $\mathcal{C}_{\mathrm{del}}(n)$ is a t-deletion torn-paper code. Furthermore, it holds that*

$$\mathrm{red}(\mathcal{C}_{\mathrm{del}}(n)) = \mathrm{red}(\mathcal{C}_A(n)) + m\left\lceil \frac{\rho_{\mathrm{BEC}}}{N_n(m)} \right\rceil \left\lfloor \frac{f(n)}{f(n)-1} \right\rfloor.$$

Before concluding, we discuss the cases of $t \in \{1, 2\}$, in which more is known on the construction of BEC codes.

For $t = 1$, we use a systematic interleaving parity BEC code as the code $\mathcal{C}_{\mathrm{BEC}}$. Namely, the redundancy string $\boldsymbol{w} = \mathrm{Enc}_{\mathrm{BEC}}(\boldsymbol{y}^*)$ is of length $\rho_{\mathrm{BEC}} = \hat{L}_{\max}$, and for all $i \in [\hat{L}_{\max}]$, i.e., w_i is a single parity symbol for $\left(y_i^*, y_{i+\hat{L}_{\max}}^*, \ldots\right)$. Denote this code by $\mathcal{C}_{\mathrm{del},1}$.

For $t = 2$, we state for completeness the following basic proposition which draws the connection between burst-error-correcting codes and burst-erasure-correcting codes. We note that this fact has been mentioned before in [8], for a single burst of errors.

Lemma 10 *For $0 < \ell \leqslant n$ and $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^n$, it holds that $\boldsymbol{x}, \boldsymbol{y}$ are confusable under t bursts of errors of lengths at most ℓ if and only if they are confusable under $2t$ bursts of erasures of lengths at most ℓ.*

A construction of 2-deletion torn-paper codes is derived from Construction C, using a BEC code for $t = 2$. Hence, by Lemma 10 one may use an $\hat{L}_{\max}$-burst error-correcting code. Observe Construction C requires a systematic construction of said code, which is guaranteed by several prior works with redundancy at most $\log(N_n(m)) + \hat{L}_{\max}$; see, e.g. [1], [2]. These constructions require the alphabet Σ to be a field, and are linear and cyclic, which ensures the existence of a systematic encoder. For simplicity of derivation we approximate this redundancy by $\hat{L}_{\max} + \log(n)$. Let $\mathcal{C}_{\mathrm{del},2}$ denote this code. The next corollary summarizes these results. For convenience, denote the difference

$$\Delta\,\mathrm{red}(\mathcal{C}(n)) \triangleq \mathrm{red}(\mathcal{C}(n)) - \mathrm{red}(\mathcal{C}_A(n)),$$

for a t-deletion torn-paper code $\mathcal{C}(n) \subseteq \Sigma^n$.

Corollary 11 *For a prime power q and all admissible values of n and $f(n)$ in Construction A, where $f(n) = \omega(1)$, $f(n) = o(\log(n))$ and with RLL encoders of [18], [34], it holds that*

$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},1}(n)) \leqslant \hat{L}_{\max} \cdot \frac{f(n)}{f(n)-1},$$
$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},2}(n)) \leqslant (\hat{L}_{\max} + \log(n)) \cdot \frac{f(n)}{f(n)-1}.$$

In particular, for $f(n) = (1 + o(1))\sqrt{\log(n)}$,

$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},1}(n)) \leqslant \hat{L}_{\max}\left(1 + \frac{1-o(1)}{\sqrt{\log(n)}}\right),$$
$$\Delta\,\mathrm{red}(\mathcal{C}_{\mathrm{del},2}(n)) \leqslant (\hat{L}_{\max} + \log n)\left(1 + \frac{1-o(1)}{\sqrt{\log(n)}}\right).$$

Note that if $L_{\max} = o(n)$ the rates of $\mathcal{C}_{\mathrm{del},1}(n)$ and $\mathcal{C}_{\mathrm{del},2}(n)$ are asymptotically equal to the rate of $\mathcal{C}_A(n)$. Thus, efficient encoding and decoding of t-deletion torn-paper codes, $t = 1, 2$, is possible at rates arbitrarily close to the optimum.

References

[1] K. A. S. Abdel-Ghaffar, "On the existence of optimum cyclic burst correcting codes over GF(q)," *IEEE Trans. on Inform. Theory*, vol. 34, no. 2, pp. 329–332, Mar. 1988.

[2] K. A. S. Abdel-Ghaffar, R. J. McEliece, A. M. Odlyzko, and H. C. A. van Tilborg, "On the existence of optimum cyclic burst-correcting codes," *IEEE Trans. on Inform. Theory*, vol. 32, no. 6, pp. 768–775, Nov. 1986.

[3] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "String reconstruction from substring compositions," *SIAM J. Discrete Math.*, vol. 29, no. 3, pp. 1340–1371, 2015.

[4] F. Balado, "Capacity of DNA data embedding under substitution mutations," *IEEE Trans. on Inform. Theory*, vol. 59, no. 2, pp. 928–941, Feb. 2013.

[5] D. Bar-Lev, S. Marcovich, E. Yaakobi, and Y. Yehezkeally, "Adversarial torn-paper codes," *arXiv preprint arXiv:2201.11150*, 2022. [Online]. Available: https://arxiv.org/abs/2201.11150

[6] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," *ACM SIGPLAN Notices*, vol. 51, no. 4, pp. 637–649, Mar. 2016.

[7] G. Bresler, M. Bresler, and D. Tse, "Optimal assembly for high throughput shotgun sequencing," *BMC Bioinformatics*, vol. 14, no. 5, p. S18, Jul. 2013.

[8] R. T. Chien, L. R. Bahl, and D. Tang, "Correction of two erasure bursts (corresp.)," *IEEE Trans. on Inform. Theory*, vol. 15, no. 1, pp. 186–187, Jan. 1969.

[9] 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, Jun. 2013.

[10] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, pp. 1628–1628, 2012.

[11] Y. Erlich and D. Zielinski, "DNA fountain enables a robust and efficient storage architecture," *Science*, vol. 355, no. 6328, pp. 950–954, Mar. 2017.

[12] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded strings from multiset substring spectra," *IEEE Trans. on Inform. Theory*, vol. 65, no. 12, pp. 7682–7696, Dec. 2019.

[13] S. Ganguly, E. Mossel, and M. Racz, "Sequence assembly from corrupted shotgun reads," in *Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT'2016), Barcelona, Spain*, Jul. 2016, pp. 265–269.

[14] 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, Feb. 2013.

[15] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust chemical preservation of digital information on dna in silica with error-correcting codes," *Angewandte Chemie International Edition*, vol. 54, no. 8, pp. 2552–2555, 2015.

[16] R. Heckel, I. Shomorony, K. Ramchandran, and D. N. C. Tse, "Fundamental limits of DNA storage systems," in *Proceedings of the 2017 IEEE International Symposium on Information Theory (ISIT'2017), Aachen, Germany*, Jun. 2017, pp. 3130–3134.

[17] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "An upper bound on the capacity of the DNA storage channel," in *Proceedings of the 2019 IEEE Information Theory Workshop (ITW'2019), Visby, Sweden*, Aug. 2019.

[18] 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.

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

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

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

[22] S. Nassirpour, I. Shomorony, and A. Vahid, "Reassembly codes for the chop-and-shuffle channel," *arXiv preprint arXiv:2201.03590*, 2022.

[23] L. Organick, S. D. Ang, Y.-J. Chen *et al.*, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 36, no. 3, pp. 242–248, Mar. 2018.

[24] A. N. Ravi, A. Vahid, and I. Shomorony, "Capacity of the torn paper channel with lost pieces," in *Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT'2021), Melbourne, Victoria, Australia*, Jul. 2021, pp. 1937–1942.

[25] ——, "Coded shotgun sequencing," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 1, pp. 147–159, Mar. 2022.

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

[27] I. Shomorony, T. Courtade, and D. Tse, "Do read errors matter for genome assembly?" in *Proceedings of the 2015 IEEE International Symposium on Information Theory (ISIT'2015), Hong Kong, China*, Jun. 2015, pp. 919–923.

[28] I. Shomorony and R. Heckel, "Capacity results for the noisy shuffling channel," in *Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT'2019), Paris, France*, Jul. 2019, pp. 762–766.

[29] I. Shomorony, G. M. Kamath, F. Xia, T. A. Courtade, and D. N. Tse, "Partial DNA assembly: A rate-distortion perspective," in *Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT'2016), Barcelona, Spain*, Jul. 2016, pp. 1799–1803.

[30] I. Shomorony and A. Vahid, "Communicating over the torn-paper channel," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020.

[31] R. R. Varshamov and G. M. Tenengolts, "Code correcting single asymmetric errors (in russian)," *Automatika i Telemkhanika*, vol. 26, no. 2, pp. 288–292, 1965.

[32] N. Weinberger and N. Merhav, "The DNA storage channel: Capacity and error probability," *arXiv preprint arXiv:2109.12549*, Sep. 2021. [Online]. Available: https://arxiv.org/abs/2109.12549

[33] P. C. Wong, K. kwok Wong, and H. Foote, "Organic data memory using the DNA approach," *Communications of the ACM*, vol. 46, no. 1, pp. 95–98, Jan. 2003.

[34] Y. Yehezkeally, S. Marcovich, and E. Yaakobi, "Multi-strand reconstruction from substrings," in *Proceedings of the 2021 IEEE Information Theory Workshop (ITW'2021), Kanazawa, Japan*, Oct. 2021.