Reconstruction and Error-Correction Codes for Polymer-Based Data Storage

Srilakshmi Pattabiraman
ECE Department, UIUC
Urbana, IL, USA
sp16@illinois.edu

Ryan Gabrys
ECE Department, UCSD
San Diego, CA, USA
ryan.gabrys@gmail.com

Olgica Milenkovic
ECE Department, UIUC
Urbana, IL, USA
milenkovic@illinois.edu

Abstract—Motivated by polymer-based data-storage platforms that use chains of binary synthetic polymers as the recording media and read the content via tandem mass spectrometers, we propose a new family of codes that allows for unique string reconstruction and correction of one mass error. Our approach is based on introducing redundancy that scales logarithmically with the length of the string and allows for the string to be uniquely reconstructed based only on its erroneous substring composition multiset. The key idea behind our unique reconstruction approach is to interleave Catalan-type paths with arbitrary binary strings and "reflect" them so as to allow prefixes and suffixes of the same length to have different weights. For error correction, we add a constant number of bits that provides information about the weights of reflected pairs of bits and hence enable recovery from a single mass error. The asymptotic code rate of the scheme is one, and decoding is accomplished via a simplified version of the backtracking algorithm used for the Turnpike problem.

Index Terms—Composition errors; Polymer-based data storage; String reconstruction.

I. INTRODUCTION

Current digital storage systems are facing numerous obstacles in terms of scaling the storage density and allowing for in-memory based computations [1]. To offer storage densities at nanoscale, several molecular storage paradigms have recently been put forward in [2]–[6]. One promising line of work with low storage cost and readout latency is the work in [2], which proposed using synthetic polymers for storing user-defined information and reading the content via tandem mass spectrometry (MS/MS) techniques. More precisely, binary data is encoded using poly(phosphodiester)s, synthesized through automated phosphoamidite chemistry in such a way that the two bits 0 and 1 are represented by molecules of different masses that are stitched together into strings of fixed length. To read the encoded data, inter phosphate bonds are broken, and MS/MS readers are used to estimate the masses of the fragmented polymer and reconstruct the recorded string, as illustrated in Figure 1. Ideally, the masses of all prefixes and suffixes are recovered reliably, allowing one to read the message content by taking the differences of the increasing fragment masses and mapping them to the masses of the 0 or 1 symbol. Polymer synthesis is cost- and time-efficient and MS/MS sequencers are significantly faster than those designed for other macromolecules, such as DNA. Nevertheless, despite
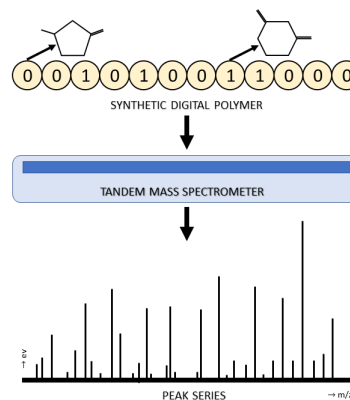
Fig. 1: The scheme is adapted from [2]. The top figure depicts a binary string synthesized using phosphoamide chemistry. The bottom image is an illustration of *peak series* or MS Spectrum obtained by MS/MS readout of the digital polymer. Note that in ideal conditions, the peaks are supposed to correspond to the masses of string fragments, or more precisely, masses of prefixes and suffixes of the string. Due to measurement errors, spurious peaks arise and one needs to apply specialized signal processing techniques to identify the correct peaks.

the fact that the masses of the polymers can be tuned to allow for more accurate mass discrimination, polymer-based storage systems still suffer from large read error-rates. This is due to the fact that MS/MS sequencing methods tend to produce peaks, representing the masses of the fragments that are buried in analogue noise due to atom disassociation during the fragmentation process.

In an earlier line of work, the authors of [7] introduced the problem of *binary string reconstruction from its substring composition multiset* to address the issue of MS/MS readout analysis. The substring composition multiset of a binary string is obtained by writing out all substrings of the string of all possible length and then representing each substring by its composition. As an example, the string 101 contains three substrings of length one - 1, 0, and 1, two substrings of length 2 - 10 and 01, and one substring of length three - 101. The composition multisets of the substrings of length one are $\{0, 1, 1\}$, of length two are $\{0^1 1^1, 0^1 1^1\}$ and of length three $\{0^1 1^2\}$. Note that composition multisets ignore information

about the actual order of the bits and may hence be seen as only capturing the information about the "mass" or "weight" of the string. The problem addressed in [7] was to determine for which string lengths may one guarantee unique reconstruction from an error-free composition multiset up to string reversal. The main results of [7, Theorem 17, 18, 20] asserts that binary strings of length ≤ 7, one less than a prime, or one less than twice a prime are uniquely reconstructable up to reversal.

For our line of work, we also rely on the two modeling assumptions described in [7]:

Assumption 1. We can infer the composition of a polymer substring from its mass. As long as the masses chosen for 0 and 1 are distinct, and the polymer block length is fixed, this assumption is naturally satisfied.

Assumption 2. When a polymer block is broken down for mass spectrometry analysis, we observe the masses of all its substrings with identical frequency. The masses of all binary substrings of an encoded polymer may be abstracted by the composition multiset of a string, provided that Assumption 1 holds. This assumption deviates from the classical ion series theory in so far that the former only provides information about the masses of the prefixes and suffixes, while the abstraction allows one to observe the masses of all substrings, but without a priori knowledge of their order.

Unlike the work in [7] which has solely focused on the problem of unique string reconstruction, we view the problem from a coding-theoretic perspective and ask the following:

Q1. *Can one add asymptotically negligible redundancy to information strings in such a way that unique reconstruction is possible, independent on the length of the strings?* Since only strings of specific lengths are reconstructable up to reversals, we aim to devise an efficient coding scheme that encode all strings of length $k \geq 1$ into strings of a larger length $n \geq k$ that are uniquely and efficiently reconstructable for *all possible string lengths*. Furthermore, we do not allow for both a string and its reversal to be included in the codebook. One simple (non-constructive) means to ensure that a string is uniquely reconstructable up to reversal is to pad the string with bits up to the shortest length of the form $\min\{p-1, 2q-1\}$, where p and q primes. For example, if $k > 89693$, it is known that there exists a prime p such that $k - 1 < p - 1 < \left(1 + \frac{1}{\ln^3 k}\right) k - 1$. Unfortunately, the result only holds for very large k that are beyond the reach of polymer chemistry. Bertrand's postulate [8], applies for shorter lengths $k > 3$, but only guarantees that $k - 1 < p - 1 < 2k - 4$. This implies a possible code rate reduction to $1/2$. Also, eliminating reversals of strings reduces the codebook by less than a half.

Q2. *Can one add asymptotically negligible redundancy to information strings in such a way that unique reconstruction is possible even in the presence of errors, independent on the length of the strings?* For simplicity, we focus on the single deletion-insertion error model, under which the composition (mass) of one substring is erroneously interpreted as a different composition (mass).

We answer both questions affirmatively by describing a coding scheme that allows for unique reconstruction and correction of a single deletion-insertion mass error. Encoding is performed by interleaving symmetric strings with Catalan-type paths, while decoding is accomplished through a modification of the backtracking decoding algorithm described in [7]. Our work extends the existing literature in coded string reconstruction [9], [10].

II. PROBLEM STATEMENT

Let $\mathbf{s} = s_1 s_2 \ldots s_k$ be binary a string of length $k \geq 2$. A substring of $\mathbf{s}$ starting at i and ending at j, where $1 \leq i < j \leq k$, is denoted by $\mathbf{s}_i^j$, and is said to have *composition* $0^z 1^w$, where $0 \leq z, w \leq j - i + 1$ stand for the number of 0s and 1s in the substring, respectively. Note that the composition only conveys information about the weight of the substring, but not the particular order of the bits. Furthermore, let $C_\ell(\mathbf{s})$ stand for the multiset of compositions of substrings of $\mathbf{s}$ of length ℓ, $1 \leq \ell \leq k$; clearly, this multiset contains $k - \ell + 1$ compositions. For example, if $\mathbf{s} = 100101$, then the substrings of length two are $10, 00, 01, 10, 01$, so that $C_2(\mathbf{s}) = \{0^1 1^1, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1\}$. =

The multiset $C(\mathbf{s}) = \cup_{\ell=1}^k C_\ell(\mathbf{s})$ is termed the *composition multiset*. Clearly, the composition multisets of a string $\mathbf{s}$ and its reversal, $\mathbf{s}^r = s_k s_{k-1} \ldots s_1$ are identical and hence these two strings are indistinguishable based on $C(\cdot)$. We define the *cummulative weight* of a composition multiset $C_\ell(\mathbf{s})$, with compositions of the form $0^z 1^w$, where $z + w = \ell$, as $w_\ell(\mathbf{s}) = \sum_{0^z 1^w \in C_\ell(\mathbf{s})} w$. Observe that $w_1(\mathbf{s}) = w_k(\mathbf{s})$, as both equal the weight of the string $\mathbf{s}$. More generally, one also has $w_\ell(\mathbf{s}) = w_{k-\ell+1}(\mathbf{s})$, for all $1 \leq \ell \leq k$. In our subsequent derivations, we also make use of the following notation. For a string $\mathbf{s} = s_1 s_2 \ldots s_k$, we let $\sigma_i = \mathrm{wt}(s_i s_{k-i+1})$ for $i \leq \lfloor \frac{n}{2} \rfloor$, and $\sigma_{\lceil \frac{n}{2} \rceil} = \mathrm{wt}(s_{\lceil \frac{n}{2} \rceil})$, where wt stands for the weight of the string. For our running example $\mathbf{s} = 100101$, $\sigma_1 = 2$, while $\sigma_2 = 0$. We use $\Sigma^{\lceil \frac{n}{2} \rceil}$ to denote the set $\{\sigma_i\}_{i \in [\lceil \frac{n}{2} \rceil]}$, where $[a] = \{1, \ldots, a\}$.

Whenever clear from the context, *we omit the argument $\mathbf{s}$ and the floors/ceiling functions required to obtain appropriate integer lengths.*

The two problem of interests are as follows. The first problem pertains to reconstruction codes: a collection of binary strings of fixed length is called a **reconstruction code** if all the strings in the code can be reconstructed uniquely based on their multiset compositions. We seek reconstruction codes of small redundancy and consequently, large rate.

In the second problem, one is given a valid composition multiset of a string $\mathbf{s}$, $C(\mathbf{s})$. Within the multiset $C(\mathbf{s})$, only one composition is arbitrarily corrupted. We refer to such an error as a **single composition error**, or single insertion-deletion pair. For example, when $\mathbf{s} = 100101$, the multiset $C_2(\mathbf{s}) = \{0^1 1^1, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1\}$ may be corrupted to $C_2(\mathbf{s}) = \{\mathbf{0^2}, 0^2, 0^1 1^1, 0^1 1^1, 0^1 1^1\}$. Single composition errors for strings of even length are detectable, since if an error occurs in only one of the two sets C_ℓ or $C_{k+1-\ell}$, then $w_\ell \neq w_{k+1-\ell}$. We seek reconstruction codes capable of correcting one composition error.

Our main results are summarized below.

Theorem 1. *There exist efficiently encodable and decodable reconstruction codes with information string length k and redundancy at most $\frac{1}{2} \log(k) + 6$.*

Theorem 2. *There exist efficiently encodable and decodable reconstruction code with information string length k capable of correcting a single composition error and redundancy at most $\frac{1}{2} \log(k) + 9$.*

III. SOME TECHNICAL BACKGROUND

Our codebook design relies on the backtracking algorithm [7], motivated by the Turnpike problem. We provide an example illustrating the operation of the algorithm.

Example 1. *Let $s = 1010001010$. It can be shown that the set $\Sigma^5 = \{\sigma_1 = 1, \sigma_2 = 1, \sigma_3 = 1, \sigma_4 = 1, \sigma_5 = 0\}$ is uniquely determined from the composition multiset. For example, $\sigma_1 = 1$ can be deduced from the two compositions of length 9, $0^5 1^4$ and $0^6 1^3$. How to determine $\Sigma^{k/2}$ from the composition multiset will be discussed in more detail in the next section. Backtracking starts by determining the first and last bit of the string and then proceeding with inward bit placements. In our example, $s_1 = 1$ and $s_{10} = 0$. From Σ^5, we easily see that one composition of length 8 equals $0^5 1^3$; removing this set from C_8 allows us to determine $\{\mathrm{wt}(s_1^8), \mathrm{wt}(s_3^{10})\}$. Given C and the previous information, we deduce that $s_2 = 0$ and $s_9 = 1$. Note that these values were determined correctly since $\mathrm{wt}(s_1) \neq \mathrm{wt}(s_{10})$. The same steps can be repeated iteratively, but in general, the algorithm will only be able to determine the compositions of the prefix/suffix extensions, but not their actual placement. This phenomenon can be observed in the next step, since the weights of the currently available prefix and suffix are equal. In this case, the algorithm makes an arbitrary assignment. For instance, the algorithm could make the assignments $s_1^3 = 100$ and $s_8^{10} = 110$. Nevertheless, at some point, combining the information in Σ^5 with the current estimate of the prefix and suffix may produce an invalid composition. In this case, the algorithm backtracks to the first position at which an arbitrary assignment was made and reverses it. Thus, the algorithm will backtrack depending on the weights of the prefixes and suffixes of the same length.*

Theorem. *[7, Theorem 32] Let $\ell_s \overset{def}{=} |\{i \le n/2 : \mathrm{wt}(s_1^i) = \mathrm{wt}(s_{n+1-i}^n) \text{ and } s_{i+1} \neq s_{n-i}\}|$, $E_s \overset{def}{=} \{t : C(t) = C(s)\}$, $\ell_s^* \overset{def}{=} \max_{t \in E_s} \ell_t$. For a given input $C(s)$ and ℓ_s, the backtracking algorithm outputs a set of strings that contains s in time $\mathcal{O}(2^{\ell_s} n^2 \log(n))$. Furthermore, E_s can be recovered in time $\mathcal{O}(2^{\ell_s^*} n^2 \log(n))$.*

Clearly, if the string has a length that does not allow for unique reconstruction, the algorithm will return a set of strings and in the process backtrack multiple times. Backtracking is possible even when the string is uniquely reconstructable, and one condition that ensures non-backtracking is to impose the constraint that no prefix has a matching suffix of the same length and same weight. To see how such strings may be constructed, we introduce strings related to Catalan paths.

Theorem 3. (Bertrand [1887]) *Among all strings comprising a 0s and b 1s, where $a \ge b$, there are $\binom{a+b}{a} - \binom{a+b}{a+1}$ strings in which every prefix has at least as many 0s as 1s. Note that when $a = b = h$, $\binom{a+b}{a} - \binom{a+b}{a+1} = \frac{1}{h+1}\binom{2h}{h} = C_h$. The number C_h is known as the hth Catalan number. The central binomial coefficient $\binom{2h}{h}$, among other things, also counts the number of strings of length $2h$ whose every prefix contains more 0s than 1s. We refer to such strings as Catalan-type.*

The following bounds on the central binomial coefficient will be useful in our subsequent derivations.

Proposition 1. The central binomial coefficient may be bounded as:

$$
\frac{2^{2h}}{\sqrt{\pi h}} \left(1 - \frac{1}{8h}\right) \le \binom{2h}{h} \le \frac{2^{2h}}{\sqrt{\pi h}} \left(1 - \frac{1}{9h}\right), \ \forall \, h \ge 1. \tag{1}
$$

IV. RECONSTRUCTION CODES

In what follows, we describe a family of efficiently encodable and decodable reconstruction codes that map strings of any length k into strings of length $n \le k + 1/2 \log(k) + 6$.

Using C_1 and recalling that $\sigma_i = \mathrm{wt}(s_i, s_{n+1-i})$, we have $\sum_{j=1}^{n/2} \sigma_j = w_1$. When $i = 2$, the bits at positions $1, n$ contribute once to w_2, whereas the bits $2, \dots, n-1$ all contribute twice to w_2. Using C_2, we hence get $\sigma_1 + 2\sum_{j=2}^{n/2} \sigma_j = w_2$. Generalizing for all $C_i, i \le n/2$, we have

$$
\frac{1}{i}\sigma_1 + \frac{2}{i}\sigma_2 + \cdots + \frac{i-1}{i}\sigma_{i-1} + \sigma_i + \sigma_{i+1} + \cdots + \sigma_{n/2} = \frac{1}{i}w_i. \tag{2}
$$

This gives a system of $n/2$ linear equations with $n/2$ unknowns that can be solved efficiently. Thus, for all error-free composition sets, one can find $\Sigma^{n/2}$. Therefore, the problem of interest is to determine s provided $\Sigma^{n/2}$ and $C(s)$. [7, Lemma 31] asserts that when $\mathrm{wt}(s_1^i) \neq \mathrm{wt}(s_{n+1-i}^n)$, then $C(s), s_1^i$, and s_{n-i+1}^n determine the ordered pair (s_{i+1}, s_{n-i}).

The previous lemma will be used to guide our construction of reconstructible code based on Catalan-type strings. We proceed as follows. Let $I \subseteq [n]$. The string formed by concatenating bits at positions in I in-order is denoted by $s|_I$. To construct a string s of a reconstruction code $\mathcal{S}_R(n)$ of even length n we proceed as follows.

$$
\begin{aligned}
\mathcal{S}_R(n) = \{ &s \in \{0,1\}^n, s_1 = 0, s_n = 1, \tag{3} \\
&\exists \, I \subseteq \{2, \dots, n-1\} \text{ such that} \\
&\qquad \text{for all } i \in I, s_i \neq s_{n+1-i}, \\
&\qquad \text{for all } i \notin I, s_i = s_{n+1-i}, \\
&s_{[n/2] \cap I} \text{ is a Catalan-type string.}\}
\end{aligned}
$$

For n odd, we define the codebook as $\mathcal{S}_R(n) = \{s_1^{n/2} 0 \, s_{n/2+1}^n, \, s_1^{n/2} 1 \, s_{n/2+1}^n, s \in \mathcal{S}_R(n-1)\}$.

The following proposition is an immediate consequence of the construction described above.

Lemma 1. *Consider a string $s \in \mathcal{S}_R(n)$. For all prefix-suffix pairs of length $1 \le j \le n/2$, one has $\mathrm{wt}(s_1^j) \neq \mathrm{wt}(s_{n+1-j}^n)$.*

The encoding algorithm that accompanies our reconstruction codebook can be easily implemented using efficient rankings of Catalan strings and symmetric strings that are ordered lexicographically.

The proof of Theorem 1 follows from the fact that $\mathcal{S}_R(n)$ is a reconstruction code, which may be easily established from the guarantees for the backtracking algorithm and Lemma 1.

The size of $\mathcal{S}_R(n)$ may be simply bounded as:

$$|\mathcal{S}_R(n)| \geq \frac{1}{2} \sum_{i=0}^{(n-2)/2} \binom{\frac{n-2}{2}}{i} 2^{\frac{n-2}{2}-i} \binom{i}{\frac{i}{2}} \geq \frac{3\,2^{n-5}}{\sqrt{2\pi(n-2)}}.$$

The first inequality follows from the description of the codebook, while the second follows from Proposition 1 and the binomial theorem. As $2^k \leq |\mathcal{S}_R(n)|$, simple algebraic manipulation reveals that the redundancy of the reconstruction code for information lengths k is at most $1/2 \log(k) + 6$.

V. ERROR-CORRECTING RECONSTRUCTION CODES

Our single composition error-correcting codes use the same interleaving procedure described in the previous section, but require adding a constant number of redundant bits. In particular, let $\mathcal{S}_R(n-2)$ be the code of odd length $n-2$ described in the previous section. Then, a single composition error-correcting code $\mathcal{S}_C(n)$ is constructed by adding two bits to each string in $\mathcal{S}_R(n-2)$ and subsequently fixing the value of one additional bit. These three redundant bits allow us to uniquely recover the set $\Sigma^{n/2}$ in the presence of a single composition error. Consequently, Lemma 3 can be used to show that given $\Sigma^{n/2}$ and the erroneous composition set of $\mathbf{s}$, one can reconstruct $\mathbf{s}$.

To prove Theorem 2, let C' denote the set obtained by introducing a single error in the composition set $C(\mathbf{s})$ of a string $\mathbf{s}$. Furthermore, let w'_j denote the cumulative weight of compositions in C'_j, and recall that w_j stands for the cumulative weight of compositions in C, such that $w_j = w_{n-j+1}$. It is straightforward to prove the following proposition.

Proposition 2. *Let $j \in [n]$. Then,*

$$jw_1 - \sum_{i=1}^{j-1} i\,\sigma_{j-i} - 2 \leq w_j \leq jw_1 - \sum_{i=1}^{j-1} i\,\sigma_{j-i}.$$

This result immediately implies the next proposition.

Proposition 3. *Let $j \in [n]$ and suppose that we are given $w_1, \sigma_1, \ldots, \sigma_{j-1}$. Then, the value $w_j \bmod 3$ uniquely determines w_j.*

We also need the following three propositions.

Proposition 4. *Given $\mathrm{wt}(\mathbf{s}) \bmod 2$, w'_n and w'_1, one can recover w_1.*

Proof. If $w'_n = w'_1$, then clearly $w_1 = w'_n = w'_1$. Hence, suppose that $w'_n \neq w'_1$ and observe that $|w'_1 - w_1| \leq 1$. The last inequality follows since at most one composition error is allowed. If $w'_1 \bmod 2 = \mathrm{wt}(\mathbf{s}) \bmod 2$, then $w_1 = w'_1$; otherwise, $w_1 = w'_n$. ∎

Proposition 5. *Suppose that n is odd and that either $\lceil \frac{n}{2} \rceil + 1$ or $\lceil \frac{n}{2} \rceil$ is divisible by 3. Assume that $\mathbf{s} = s_1 \ldots s_{\lceil \frac{n}{2} \rceil} \ldots s_n$, and let $\mathbf{s}' = s_1 \ldots 1 - s_{\lceil \frac{n}{2} \rceil} \ldots s_n$. Then,*

$$\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\mathbf{s}) \equiv \sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\mathbf{s}') \bmod 3.$$

Proof. Suppose that $s_{\lceil \frac{n}{2} \rceil} = 1$. Then, the bit $s_{\lceil \frac{n}{2} \rceil}$ contributes $\lceil \frac{n}{2} \rceil$ to $w_{\lceil \frac{n}{2} \rceil}$ and $\lceil \frac{n}{2} \rceil - 1$ to $w_{\lceil \frac{n}{2} \rceil - 1}$. In summary, if $s_{\lceil \frac{n}{2} \rceil} = 1$, then

$$\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\mathbf{s}) = \sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\mathbf{s}') + \frac{\lceil \frac{n}{2} \rceil (\lceil \frac{n}{2} \rceil + 1)}{2}.$$

The result follows if either $\lceil \frac{n}{2} \rceil + 1$ or $\lceil \frac{n}{2} \rceil$ is divisible by 3. ∎

Proposition 6. *For odd n, if $s_1 \ldots s_{\lceil \frac{n}{2} \rceil} \ldots s_n \in \mathcal{S}_R(n)$, then $s_1 \ldots 1 - s_{\lceil \frac{n}{2} \rceil} \ldots s_n \in \mathcal{S}_R(n)$.*

Our code for odd n is defined as follows (an almost identical construction is valid for even n):

$$\mathcal{S}_C(n) = \Big\{ \mathbf{s} = s_1 s_1^* s_2 \ldots s_{\lceil \frac{n-2}{2} \rceil} \ldots s_{n-3} s_n^* s_{n-2} \in \{0,1\}^n :$$
$$s_1 \ldots s_{n-2} \in \mathcal{S}_R(n-2), \mathrm{wt}(\mathbf{s}) \bmod 2 \equiv 0,$$
$$\sum_{i=1}^{\frac{n}{2}} w_i(\mathbf{s}) \equiv 0 \bmod 3, \text{ where } s_1^* \leq s_n^* \Big\}.$$

The size of the code $\mathcal{S}_C(n)$ is $\frac{|\mathcal{S}_R(n-2)|}{2}$, which follows since we removed one information symbol from each coded string in $S_R(n-2)$ by requiring $\mathrm{wt}(\mathbf{s}) \bmod 2 \equiv 0$, and then added two more redundant symbols. To construct a string in $\mathcal{S}_C(n)$, we first fix s_1^* and s_n^* so that $\sum_{i=1}^{\lceil \frac{n}{2} \rceil} w_i(\mathbf{s}) \equiv 0 \bmod 3$. Then, we choose $s_{\lceil \frac{n-2}{2} \rceil}$ to satisfy $\mathrm{wt}(\mathbf{s}) \equiv 0 \bmod 2$. From Propositions 5 and 6, the resulting string belongs to $\mathcal{S}_C(n)$.

For the next lemma, recall that $C'(\mathbf{s})$ is the result of a single composition error in $C(\mathbf{s})$.

Lemma 2. *Suppose that $\mathbf{s} \in \mathcal{S}_C(n)$. Then, given $C'(\mathbf{s})$, one can recover $\Sigma^{n/2}$.*

Proof. In order to prove the claim, we show that given $C'(\mathbf{s})$, one can recover $w_1, w_2, \ldots, w_n$, which we know uniquely determine $\Sigma^{n/2}$ according to (2). Let j be such that $w'_j \neq w'_{n+1-j}$. Since at most one single composition error is allowed, there exists at most one such j. It is straightforward to see that due to symmetry, either $w'_j \neq w_j = w_{n+1-j}$ or $w'_{n+1-j} \neq w_j = w_{n+1-j}$. Since $\mathrm{wt}(\mathbf{s}) \bmod 2 \equiv 0$ by construction, it follows that we can determine w_1 based on Proposition 4. Then, according to Proposition 3, we can recover w_j and all of $w_1, \ldots, w_n$. One case left to consider is when $w'_i = w'_{n+1-i}$ for all i. In this case, $w'_{\frac{n}{2}} \neq w_{\frac{n}{2}}$. Applying Proposition 3 allows us to determine $w_{\frac{n}{2}}$ for this case as well, and this completes the proof. ∎

Next, let $\mathcal{T}_i$ be the set of compositions of all substrings s_j^k for which $j < k \leq i$, or $n+1-i \leq j < k$, or $j \leq i$ and $n+1-i \leq k$.

Lemma 3. *Let $s \in \mathcal{S}_C(n)$. Given $C'(s)$, one can uniquely reconstruct the string s.*

Proof. Let j denote the index of the composition multi-set C_j that contains an error. From Lemma 2, $\Sigma^{n/2}$ may be determined in an error-free manner. Using the obtained $\Sigma^{n/2}$, we run the backtracking algorithm and in the process, we may run into non-compatible compositions for $j > \frac{n}{2}$. For the case that backtracking halts for $j = n - i - 1$, the currently reconstructed sub-strings are $\mathbf{s}_1^i, \mathbf{s}_{n+1-i}^n$. Without loss of generality, assume that $\sigma_{i+1} = 1$ as otherwise one can fix the error easily. Furthermore, note that $\mathcal{T}_i$ can be constructed from $\Sigma^{n/2}, \mathbf{s}_1^i$, and $\mathbf{s}_{n+1-i}^n$.

One way in which incompatibility may manifest itself is through $\mathcal{T}_i \not\subset C'$, where $j = n - i - 1$. In this case, we identify the element that is in $\mathcal{T}_i$ but not in C_j', and add its weight to w_j' and compare it with w_{n+1-j}'; this allows us to identify the erroneous composition. Next, suppose that $\mathcal{T}_i \subset C'$. In this case, consider the two longest compositions in $C' \setminus \mathcal{T}_i$. The two longest compositions in $C' \setminus \mathcal{T}_i$ are the compositions of a prefix-suffix pair of length j. Since we have reconstructed the prefix and suffix of length i and we know that $\sigma_{i+1} = 1$, there are two possibilities for compositions compatible with the prefix and two for the suffix of length $i+1$. Out of the six pairs of compositions that may be chosen from the four compositions, only two pairs cannot be directly eliminated as candidates for the correct composition. In this case, the following two prefix-suffix substrings are possible: $\{\mathbf{s}_1^i\, 0, 1\, \mathbf{s}_{n-i+1}^n\}, \{\mathbf{s}_1^i\, 1, 0\, \mathbf{s}_{n-i+1}^n\}$. To show that only one of the constructed prefix-suffix pairs will be valid (compatible), it suffices to show the following: For any two strings $\mathbf{s}_1, \mathbf{s}_2 \in \mathcal{S}_C(n)$ that have the same $\Sigma^{n/2}$, $|C(\mathbf{s}_1) \setminus C(\mathbf{s}_2)| \geq 4$.

Let us assume that on the contrary, there are two strings $\mathbf{s}, \mathbf{t}$ such that $|C(\mathbf{s}) \setminus C(\mathbf{t})| = 2$, and that they differ only in their respective C_j sets (this condition is imposed by the Catalan strings, see Figure 2).

Since the prefixes and suffixes of the strings of length $i = n - j - 1$ are identical, we let $s_1, \ldots, s_i$ and $s_{n+1-i}, \ldots, s_n$ denote the first and last i bits of both strings. Let $c(\mathbf{s})$ denote the composition of the string $\mathbf{s}$. Furthermore, let $c(\mathbf{s}_l^{l'})$ denote the composition of $\mathbf{s}_l^{l'}$, $l \leq l'$.

When $n = 2(i+1)+1$, the strings differ in two compositions in C_{n+1-i} due to the above observations. Note that they also differ in two compositions in their respective multisets C_i.

When $n \geq 2(i+1)+3$ and $\sigma_{i+2} = 1$, we let b_s stand for the $(i+2)^{\text{th}}$ bit in the string $\mathbf{s}$, and b_t stand for the $(i+2)^{\text{th}}$ bit of string $\mathbf{t}$. When $\sigma_{i+2} \in \{0, 2\}$, we let b denote the $(i+2)^{\text{th}}$ bits of the two strings, which are identical. Next, we determine conditions under which $C_{j-1}(\mathbf{s}) = C_{j-1}(\mathbf{t})$. Note that the compositions of substrings of length $n - i - 2$ that contain the bits $i + 1, \ldots, n - i$ are identical for the two strings.

Case 1: $\sigma_{i+2} = 1$. With a slight abuse of notation, we choose to write compositions as sets containing both bits and other compositions. On the left-hand-side of the equation below, the compositions correspond to the substrings of $\mathbf{s}$ of length $n - i - 2$ that *may* differ for the two strings. The right-hand-side

Fig. 2: The figure depicts two strings $\mathbf{s}, \mathbf{t}$ satisfying the assumptions used in the proof.

of the equation corresponds to the same entities in $\mathbf{t}$. If the equation holds, then the multisets $C_{j-1}(\mathbf{s})$ and $C_{j-1}(\mathbf{s})$ are equal.

$$
\begin{cases}
\{c(\mathbf{s}_1^i), 0, b_s, c\}, \\
\{c(\mathbf{s}_2^i), 0, b_s, c, 1-b_s\}, \\
\{c(\mathbf{s}_{j+2}^n), 1, 1-b_s, c\}, \\
\{c(\mathbf{s}_{j+2}^{n-1}), 1, 1-b_s, c, b_s\}
\end{cases}
=
\begin{cases}
\{c(\mathbf{s}_1^i), 1, b_t, c\}, \\
\{c(\mathbf{s}_2^i), 1, b_t, c, 1-b_t\}, \\
\{c(\mathbf{s}_{j+2}^n), 0, 1-b_t, c\}, \\
\{c(\mathbf{s}_{j+2}^{n-1}), 0, 1-b_t, c, b_t\}
\end{cases}
$$

Due to space limitations, we omit the exhaustive case-by-case arguments that show that the above set equality is never true, independently on how b_s and b_t are chosen.

Case 2: $\sigma_{i+2} \in \{0, 2\}$ Similar reasoning leads to a set equality condition in which b_s and b_t are replaced by b. Once again, it can be shown by an exhaustive case-by-case analysis that the set equality never holds, independently on the choice of b. This implies that the composition sets $C_{j-1}(\mathbf{s})$ and $C_{j-1}(\mathbf{t})$ differ, which in turn implies that the composition multisets of the two strings are at distance ≥ 4. ∎

The backtracking string reconstruction process based on an erroneous composition set is straightforward: It takes $\mathcal{O}(n^2)$ time to compute the $\mathcal{T}_k$ multiset, and backtracking performs $\mathcal{O}(n)$ steps. Thus, the decoding algorithm can computes the original string in $\mathcal{O}(n^3)$ time.

REFERENCES

[1] V. Zhirnov, R. M. Zadegan, G. S. Sandhu, G. M. Church, and W. L. Hughes, "Nucleic acid memory," *Nature materials*, vol. 15, no. 4, p. 366, 2016.
[2] A. Al Ouahabi, J.-A. Amalian, L. Charles, and J.-F. Lutz, "Mass spectrometry sequencing of long digital polymers facilitated by programmed inter-byte fragmentation," *Nature communications*, vol. 8, no. 1, p. 967, 2017.
[3] 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, p. 77, 2013.
[4] 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.
[5] S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Scientific reports*, vol. 5, p. 14138, 2015.
[6] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific reports*, vol. 7, no. 1, p. 5011, 2017.
[7] J. Acharya, H. Das, O. Milenkovic, A. Orlitsky, and S. Pan, "String reconstruction from substring compositions," *arXiv preprint arXiv:1403.2439*, 2014.
[8] G. H. Hardy, "An introduction to the theory of numbers," *Bull. Amer. Math. Soc.*, vol. 35, pp. 778–818, 11 1929.
[9] H. M. Kiah, G. J. Puleo, and O. Milenkovic, "Codes for DNA sequence profiles," *IEEE Transactions on Information Theory*, vol. 62, no. 6, pp. 3125–3146, 2016.
[10] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded sequences from multiset substring spectra," in *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2540–2544, IEEE, 2018.