Survey for A Decade of Coding for DNA Storage

Omer Sabary *Student Member, IEEE*, Han Mao Kiah *Senior Member, IEEE*, Paul H. Siegel *Life Fellow, IEEE*, Eitan Yaakobi *Senior Member, IEEE*

***Abstract*—Advancements in DNA synthesis and sequencing technologies have enabled the storage of data on synthetic DNA strands. However, realizing its potential relies on the design of tailored coding techniques and algorithms. This survey paper offers an overview of past contributions, accompanied by a special issue that showcases recent developments in this field.**

I. Background

Increased use of data and energy conservation issues pose new challenges for the storage community in terms of identifying extremely high volume, non-volatile and durable recording media. In addition to progress in conventional data recording techniques using tapes, HDDs, and NAND flash, innovative approaches must be developed to meet these challenges [1]. The potential for using macromolecules for ultra-dense storage was recognized as early as the 1960s [2]. DNA stands out due to its information density, stability, and robustness. Furthermore, current technologies for synthesizing artificial DNA and for sequencing are highly efficient and accurate [3]. These technologies will continue to evolve as DNA is of central interest in medicine and life science more broadly. DNA as a storage medium will therefore never become obsolete. As a result, DNA-based storage is emerging as a technology with applications that will, as evidenced by some proof-of-concept demonstrations, plausibly materialize in the near future. However, to meet this potential, sequencing and synthesis technologies need to be dramatically more cost-effective. This can only be accomplished by developing reduced-cost synthesis and sequencing coupled with the design of appropriate coding techniques and algorithms that will be specifically designed for these methods. This goal of this special issue is to highlight new developments in this direction, while this survey paper provides an overview of some of the past contributions.

A typical DNA storage system consists of three components (see Fig. 1): (1) DNA synthesis that produces the molecules encoding the data. In state of the art technology acceptable error rates are achieved for synthetic *oligonucleotides* (in short - *oligos*) of length no more than 250-300nt [1]; (2) a storage container to store the synthetic DNA strands; (3) a DNA sequencing (aka NGS) device that serves for reading and for retrieving the data. The encoding and decoding are external processes to the storage system that convert the binary data into molecules of DNA in a way that enables reconstruction even in the presence of errors. From a mathematical/coding perspective, DNA as a storage system has several attributes that distinguish it from other storage systems. The most prominent one is that the oligos are not ordered in the memory and thus it is not possible to know the order in which they were stored. One solution uses *indices*, or *barcodes*, that are stored as part of the oligos [1], [4], [5].

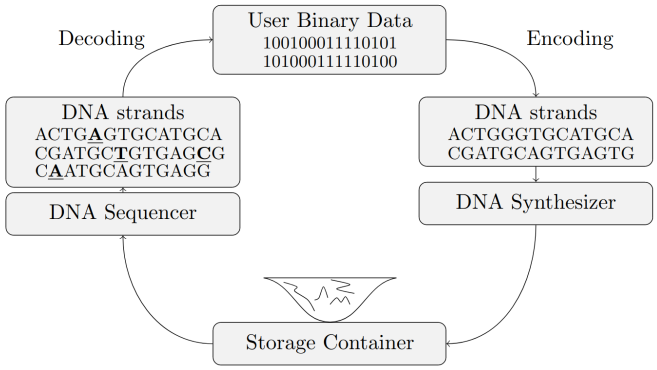

Fig. 1. DNA storage system model.

Errors in DNA are typically substitutions, insertions, and deletions, and most published studies [6], [7] report that either substitutions or deletions are the most prominent ones, depending upon the specific technology used for synthesis and sequencing [8]–[14]. For example, in surface-based DNA oligo synthesis the dominant errors are deletions that result from either failure to remove the dimethoxytrityl (DMT) or combined inefficiencies in the coupling and capping steps [11], while NGS often introduces substitutions [6].

Church et al. [15] and Goldman et al. [16] presented the first DNA storage experiments, in which they stored 643KB and 739KB of digital information in DNA molecules, respectively. However, in both experiments, the data was not recovered completely due to the lack of proper coding solutions. Later, in [17], Grass et al. reported the first DNA storage experiment in which error-correcting codes were used. They were able to perfectly recover 81KB of information. Bornholt et al. similarly retrieved a 42KB message [18]. Later, several groups have managed to successfully store and recover messages

The research of Omer Sabary and Eitan Yaakobi was Funded by the European Union (ERC, DNAStorage, 865630). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

The work of Han Mao Kiah is supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 Award MOE-T2EP20121-0007 and MOE AcRF Tier 1 Award under Grant RG19/23.

Paul Siegel and Eitan Yaakobi were supported in part by NSF Grant CCF-2212437.

Omer Sabary and Eitan Yaakobi are with the Department of Computer Science, Technion — Israel Institute of Technology, Israel (email: {omersabary, yaakobi}@cs.technion.ac.il).

Han Mao Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore (email: hmkiah@ntu.edu.sg).

Paul Siegel is with the Department of Electrical & Computer Engineering, University of California San Diego, La Jolla, CA, USA (email: psiegel@ucsd.edu).

in DNA storage systems. Among these groups, Erlich and Zielinski [10] stored 2.11MB of data with high storage rate, Blawat et al. [8] successfully stored 22MB, and Organick et al. [12] stored the largest experiment so far of 200MB. Yazdi et al. [5], [19] developed a method that supports random access and rewritable storage. More recently, Anavy et al. [4] and Choi et al. [20] suggested a new approach that utilizes the inherent redundancy of synthesis to build *composite symbols*, which are symbols consisting of the composition of more than one base. Hence, the alphabet size can be extended, and the potential information capacity can increase even further. This method was later generalized to work with shortmers, which are short DNA sequences [21]. Deep and machine learning methods for DNA storage were discussed in [22], [23]. For a comprehensive survey on previous works and experiments of DNA storage the reader is referred to [24], where recommended surveys on coding-related problems can be found in [25]–[27].

To date, the papers of Church et al. [15] and Goldman et al. [16] have accumulated more than one thousand citations in a short span of ten years. Hence, our survey is clearly neither exhaustive nor representative. To some extent, the selection of works reflects our personal preferences. Nevertheless, we aim to provide a summary of some beautiful ideas in prior literature and hope to illustrate the interrelationships of these ideas. At the same time, we hope that this survey will be a useful reference for future work in this area.

II. The Sequence Reconstruction Problem

Proposed by Levenshtein in 2001, the *sequence reconstruction problem* addresses a communication scenario where an input is transmitted over several identical channels, resulting in several noisy outputs. In the context of data storage applications, these noisy outputs are often referred to as *reads*. Formally, we use $\mathcal{X}$ and $\mathcal{Y}$ to denote the *input* and *output* space, respectively. To characterize the channel, we use an *error-ball function* $\mathcal{B}$ that maps an *input* $\boldsymbol{x} \in \mathcal{X}$ to a subset $\mathcal{B}(\boldsymbol{x}) \subset \mathcal{Y}$. As always, we have a *codebook* $\mathcal{C} \subseteq \mathcal{X}$ which is a subset of the input space.

A. Maximum Intersection of Error Balls

The original sequence reconstruction problem defined by Levenshtein is as follows. Given a codebook $\mathcal{C} \subseteq \mathcal{X}$ and error-ball function $\mathcal{B}$, determine the following quantity

$$\nu(\mathcal{C};\mathcal{B}) \triangleq \max\{|\mathcal{B}(\boldsymbol{x}) \cap \mathcal{B}(\boldsymbol{x}')| : \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{C}, \boldsymbol{x} \neq \boldsymbol{x}'\}. \quad (1)$$

In other words, $\nu(\mathcal{C};\mathcal{B})$ is the maximum intersection between the error-balls of any two distinct codewords in $\mathcal{C}$. The quantity $\nu(\mathcal{C};\mathcal{B})$ was introduced by Levenshtein [28], where he showed that the number of noisy outputs, or reads, required to uniquely reconstruct a codeword in $\mathcal{C}$ is at least $\nu(\mathcal{C};\mathcal{B})+1$. The sequence reconstruction problem was studied in a variety of storage and communication scenarios. To keep this special issue succinct, we review the instructive case where $\mathcal{X} = \{0,1\}^n$; $\mathcal{C}$ is either the entire input space or a classical error-correcting code; and the error-ball is one of the following three sets: (a) $\mathcal{S}_t$ - all words obtained by at most t substitutions, (b) $\mathcal{D}_t$ - all words obtained by exactly t deletions, (c) $\mathcal{I}_t$ - all words obtained by exactly t insertions. To enable easier comparison across different various scenarios, we provide asymptotic estimates here. Exact formulas can be found in the original works. For our asymptotic regime, we fix t and allow the blocklength n to grow. Here, we use $f(n) \sim g(n)$ to mean that $\lim_{n\to\infty} f(n)/g(n) = 1$.

In Levenshtein's seminal work [28], we have the following result concerning substitutions. If $\mathcal{C}$ is a code with Hamming distance d, then $\nu(\mathcal{C};\mathcal{B}_t) \sim C_{t,d} n^{t-\lceil d/2 \rceil}$ where $C_{t,d}$ is a constant that depends on d and t. Now, a key observation in Levenshtein's derivation is that the intersection size $|\mathcal{S}_t(\boldsymbol{x}) \cap \mathcal{S}_t(\boldsymbol{x}')|$ is completely determined by the Hamming distance $\boldsymbol{x}$ and $\boldsymbol{x}'$. However, this observation is no longer true for the case of deletions and insertions. Specifically, let $\mathcal{C}$ be a code that corrects e deletions. Then, Levenshtein showed that it is both necessary and sufficient that $\mathcal{C}$ corrects e insertions [29]. However, the quantities $\nu(\mathcal{C},\mathcal{I}_t)$ and $\nu(\mathcal{C},\mathcal{D}_t)$ are generally not equal for $t > e$.

Now, when $e = 0$, that is, $\mathcal{C} = \{0,1\}^n$, Levenshtein determined $\nu(\mathcal{C},\mathcal{I}_t)$ and $\nu(\mathcal{C},\mathcal{D}_t)$ in [30]. For $e \geq 1$, the problem was not studied until more than a decade later. The case of insertions was resolved by Sala et al. [31], where the quantity $\nu(\mathcal{C},\mathcal{I}_t)$ was determined for $e \geq 1$. For the case of deletions, the quantity $\nu(\mathcal{C},\mathcal{D}_t)$ was determined by Gabrys and Yaakobi [32] when $e \geq 1$, and by Pham et al. [33] when $e = t-1$. While the exact quantity is not known for other parameters of e and t, Pham et al. provided asymptotically sharp estimates [33]. Coincidentally, both $\nu(\mathcal{C};\mathcal{I}_t)$ and $\nu(\mathcal{C};\mathcal{D}_t)$ are asymptotically equal. In summary, we have that: if $\mathcal{C}$ is a code that corrects exactly $(d-1)$ deletions, then $\nu(\mathcal{C};\mathcal{I}_t) \sim \nu(\mathcal{C};\mathcal{D}_t) \sim \frac{\binom{2d}{d}}{(t-d)!} n^{t-d}$.

B. Code Design

Now, in most storage scenarios, the number of noisy reads N is a fixed system parameter. Therefore, rather than using an existing codebook $\mathcal{C}$ and assessing whether $\nu(\mathcal{C};\mathcal{B})$ is strictly less than N, Cai et al. [34] proposed the following task of *code design*. Given an error-ball function $\mathcal{B}$ and integer N, design a codebook $\mathcal{C}$ such that $\nu(\mathcal{C};\mathcal{B}) < N$. If $\mathcal{X} = \{0,1\}^n$, we say that $\mathcal{C}$ is an $(n, N; \mathcal{B})$-*reconstruction code*.

As always, for fixed n, N and $\mathcal{B}$, we are interested in designing $(n, N; \mathcal{B})$-reconstruction codes of large size. Equivalently, we want to minimize the redundancy of such codebooks. In other words, we are interested in the following quantity of interest that measures the trade-off between codebook redundancy and the number of reads. Given an error-ball function $\mathcal{B}$ and integers n, N, determine

$$\rho(n, N; \mathcal{B}) \triangleq \min\{n - \log|\mathcal{C}| : \mathcal{C} \text{ is an } (n, N; \mathcal{B})\text{-reconstruction code}\}. \quad (2)$$

When $N = 1$, we recover the usual notion of error-correcting codes. Code constructions for substitution errors, equivalently, codes in the Hamming metric, are extensively studied (see for example, [35]). In Sections III-A and III-B, we survey the results of deletion- and insertion-correcting codes.

When the error-ball $B = \mathcal{S}_t$, it follows from earlier discussion that designing a reconstruction code for $\mathcal{S}_t$ is equivalent to designing a code with certain Hamming distance. Specifically, we have the following result. Define $N(n,d;t) \triangleq |\mathcal{S}_t(\boldsymbol{x}) \cap \mathcal{S}_t(\boldsymbol{y})|$ for some $\boldsymbol{x}$ and $\boldsymbol{y}$ with Hamming distance d. Then we have that $\rho(n,N;\mathcal{S}_t) = n - \log A(n,d)$, where d is the smallest integer such that $N(n,t,d) < N$.

When $\mathcal{B} \in \{\mathcal{I}_t, \mathcal{D}_t\}$, upper bounds for optimal redundancy are known via code constructions. When $t \in \{1,2\}$, we list the best known explicit code constructions in Table I. Here, we only state the redundancy values. For the lower bound for redundancy when $N > 1$, very little is known. This is partly because there is no known metric that characterizes the intersection size of the two balls. In [36], Chrisnata et al. used clique covers to showed that $\rho(n,2;\mathcal{D}_1) \geq \log\log(n) - 1 - o(1)$.

C. Efficient Reconstruction

While results from the previous two subsections determine the possibility of unique reconstruction for a given codebook, the solutions may not provide insight into the recovery process of the transmitted word. Therefore, Levenshtein [28] initially proposed the problem (and later revisited by Abu-Sini and Yaakobi [42]) of designing a decoder that efficiently retrieves the transmitted word. Specifically, given an an $(n,N;\mathcal{B})$-reconstruction code $\mathcal{C}$, design a *decoder* Decode such that

$$\mathsf{Decode}(\mathcal{R}) = \boldsymbol{c} \text{ for all } \boldsymbol{c} \in \mathcal{C} \text{ and } \mathcal{R} \subseteq \mathcal{B}(\boldsymbol{c}) \text{ with } |\mathcal{Y}| \geq N\,.$$

Here, $\boldsymbol{c}$ denotes the transmitted codeword, while $\mathcal{R}$ denotes the set of noisy outputs corresponding to $\boldsymbol{c}$. Suppose that the output space is $\{0,1\}^m$. Then the decoder Decode is given at least N words of length m. Hence, we want Decode to run in time polynomial in Nm. If the decoder runs in linear in Nm, then we say that the decoder is *optimal*.

Recall that when the error-ball $B = \mathcal{S}_t$, a code $\mathcal{C}$ with Hamming distance d is an $(n, N(n,d;t)+1, \mathcal{S}_t)$-reconstruction code. Furthermore, $N(n,d;t) = \Theta\left(n^{t-\lceil d/2 \rceil}\right)$. In his seminal paper [28], when $d = 0$, Levenshtein showed by simply applying majority function, we obtain an optimal decoder. However, when $d > 0$, Levenshtein observed that simple majority logic is insufficient. Nevertheless, this problem was later solved in series of papers [42], [43]. Specifically, for fixed values of t and d, given any code $\mathcal{C}$ with Hamming distance d, there is an explicit optimal decoder.

When the error-ball $\mathcal{B} \in \{\mathcal{D}_t, \mathcal{I}_t\}$, much less in known. When $\mathcal{C} = \{0,1\}^n$, Levenshtein in [28] provided efficient decoders using so-called threshold functions. When $\mathcal{C}$ corrects $t-1$ deletions and $\mathcal{B} = \mathcal{D}_t$, Pham et al. provided an efficient decoder under the assumption that $\mathcal{C}$ admits an efficient $(t-1)$-deletion-correcting decoder [33].

Finally, we list certain works related to Levenshtein's sequence reconstruction problem that are relevant to DNA data storage.

- In [44], Junnila et al. consider the scenario where the number of reads is not sufficient to reconstruct a unique codeword. As with classical list-decoding, they determined the size of the list of possible codewords and study algorithmic issues with the list-decoder.
- In addition to the results listed in this section, Abu-Sini and Yaakobi [42] investigated the case where errors are combinations of single substitution and single insertion. Furthermore, they also simplified the reconstruction algorithm when the number of noisy copies exceeds the minimum required, i.e. $N > \nu(\mathcal{C};\mathcal{B}) + 1$.
- Design of reconstruction codes for other variations of the error models has also been studied. Cai et al. looked at single edits and its variants [34]; Wu and Zhang looked at the case where all codewords are balanced [45]; and Sun et al. looked at single bursts of edits and its variants [46].
- Finally, all errors described here are assumed to be adversarial. The scenario where errors are *probabilistic* has been extensively studied and is also known as the *trace reconstruction problem* (see [47] for a survey that is related to computational biology). The problem of code design has also been proposed and is studied in [48], [49].

III. Error-Correcting Codes for DNA Storage: Indel Codes, Coding over Sets, and More Problems

Investigating codes correcting *insertions* and *deletions* has regained significant attention due to emerging applications, including wireless communication, disk and DNA-based data storage, racetrack memories, file synchronization, and compression. Codes correcting substitution errors that only change the value of symbols in sequences are very well understood in the literature [50]–[52] and are applied in communication and storage systems. However, the problem of correcting insertions and deletions has proven to be more challenging. Investigating this problem has started with the work of Levenshtein in the 1960s [53]. Levenshtein derived a lower bound on the redundancy of codes correcting insertions and deletions and proved that the Varshamov-Tenengolts (VT) codes [54] constructed to correct asymmetric substitution errors can, in fact, correct one insertion or one deletion and are asymptotically optimal [55]. Recently, codes correcting an arbitrary fixed number of insertion and deletion errors whose redundancy is a multiplicative factor away from Levensthein's lower bound are constructed in [40], [56], [57]. To approach this problem, simplified versions of those codes are being studied in the hope of building the right toolbox to correct insertions and deletions.

Besides errors arising from insertions and deletions, DNA-based storage is also fundamentally different from other storage media. One key property that sets DNA apart is the lack of ordering among the strands. This new channel model is sometimes referred to as the *shuffling channel* in (see [58], [59] and the references therein), while the code design problem is referred to as *coding over sets* [60]. We elaborate more about the latter in Section III-C.

A. Insertion/Deletion-Correcting Codes

The problem of coding for the *insertion/deletion channel* dates back to the work of Levenshtein and others [61]–[64] in the early 1960s. A code is referred as a *t-deletion-correcting code* if it can correct any t deletions. Similarly, we define *t-insertion-correcting codes* for the correction of t insertions

(a) $\mathcal{B} \in \{\mathcal{D}_1, \mathcal{I}_1\}$

Number of Noisy Reads N	Redundancy	Remark
$N = 2$	$\log\log(n) + O(1)$	Chee et al. [37], Cai et al. [34]
$N \geq 3$	0	Levenshtein [30]

(b) $\mathcal{B} = \mathcal{D}_2$

Number of Noisy Reads N	Redundancy	Remark
$N = 2$	$3\log(n) + 14\log\log(n) + O(1)$	Sun and Ge [38]
$N = 3$	$3\log(n) + O(1)$	Sun and Ge [38]
$N = 4$	$2\log(n) + 6\log\log(n) + O(1)$	Sun and Ge [38]
$5 \leq N \leq 6$	$\log(n) + \log\log(n) + O(1)$	Sun and Ge [38]
$7 \leq N \leq n$	$\log(n) + O(1)$	Gabrys and Yaakobi [32]
$n+1 \leq N \leq 2n-2$	$\log\log(n) + O(1)$	Chrisnata et al. [39]
$N \geq 2n-3$	0	Levenshtein [30]

(c) $\mathcal{B} = \mathcal{I}_2$

Number of Noisy Reads N	Redundancy	Remark
$2 \leq N \leq 4$	$4\log(n) + 10\log\log(n) + O(1)$	Guruswami and Håstad [40]
$5 \leq N \leq 6$	$\log(n) + 14\log\log(n) + O(1)$	Ye et al. [41]
$7 \leq N \leq n+3$	$\log(n) + O(1)$	Ye et al. [41]
$n+4 \leq N \leq 2n+4$	$\log\log(n) + O(1)$	Ye et al. [41]
$N \geq 2n+5$	0	Levenshtein [30]

TABLE I
BEST KNOWN $(n, N; \mathcal{B})$-RECONSTRUCTION CODES FOR $\mathcal{B} \in \{\mathcal{D}_1, \mathcal{I}_1, \mathcal{D}_2, \mathcal{I}_2\}$

and *t-insertion/deletion-correcting code* for the correction of any t insertions and deletions. Since it is well known, as was proved by Levenshtein [62], that a code can correct t deletions if and only if it can correct any combination of t insertions and deletions, we mostly discuss in this review deletion-correcting codes. It is well known that the Varshamov-Tenengolts (VT) code [54] can correct a single deletion and that this code is nearly optimal with respect to the number of redundant bits, while it is strictly optimal for $n \leq 14$ [65]. Levenshtein further showed [62] that the minimum asymptotic redundancy of a t-deletion-correcting code is $t\log(n)$ and by a Gilbert-Varshamov (GV) bound argument there exist codes with asymptotic redundancy $2t\log(n)$. Even starting with the case where $t = 2$, there was a lack of good codes until very recently, when codes correcting insertions and deletions have recently gained interest and attracted significant attention. For example, several specific code constructions [56], [66], [67] have been proposed which generalize the old single-deletion-correcting codes by Varshamov-Tenengolts [54] to correct multiple deletion errors. In [68], Brakensiek et al. presented binary multiple-deletion correcting codes with small asymptotic redundancy. For an explicit small number of deletions, their construction however needs redundancy $c\log n$ where c is a large constant. The recent parallel works by Gabrys et al. [66] and Sima et al. [67] have presented constructions of double-deletion-correcting codes with redundancy $8\log(n) + O(\log(\log(n)))$ [66] and $7\log(n) + o(\log(n))$ [67], respectively. Sima et al. [57] provided a construction of t-deletion-correcting codes with redundancy $4t\log(n) + o(\log(n))$ and this result was also established by Li and Farnoud in [69]. Currently, the best binary double-deletion correcting codes were proposed recently by Guruswami and Håstad with redundancy of $4\log(n)$ [40], which matches the existential lower bound [62].

In addition to the theoretical worst-case codes, several works design codes that correct deletion or *edit* errors (which are insertions, deletions or substitutions) for the probabilistic case with efficient implementation of encoding and decoding algorithms. These codes are of high importance since in many applications, codes correcting a fixed number of deletions are not appropriate since the channel deletes symbols with some given probability. Such works include, for example, HEDGES codes [70], segmented error correcting codes [71], Guess and Check codes [72], polar codes [73], DNA-Aeon codes [74], codes based upon LDPC codes such as [75], and constructions based upon convolutional codes [76].

Another interesting question refers to the maximal deletion fraction δ, for which for every $\epsilon > 0$, there exists a code with rate bounded away from 0 that can handle $\delta - \epsilon$ fraction of adversarial deletions. One can easily see that $\delta = 1/2$ is an upper bound (for any codeword, the adversary can choose the deletions to guarantee the output to be all zeroes or all ones). In [77], the authors presented binary codes that can correct $\sqrt{2} - 1$ fraction of deletions and have a positive rate. On the other hand, it was proven in [78] that there exists a tiny absolute constant δ_0 ($\approx 10^{-40}$) such that any binary code $\mathcal{C} \subseteq \{0,1\}^n$ that can decode from $(1/2 - \delta_0)$ fraction of deletions must satisfy $|\mathcal{C}| \leq 2^{\text{poly}\log n}$. In particular, we cannot hope to decode a fraction of deletions arbitrarily close to $1/2$ with codes of positive rate. We emphasize that determining the exact threshold is still an interesting open question.

There are many other interesting problems regarding insertions and deletions that are out of the scope of this review. One major such problem that was extensively studied in the literature is that of constructing efficient codes that can correct a fraction of insertions and deletions (not a fixed number) [79]–[83], and the state-of-the art binary codes are given in [84], [85]. This problem is also studied for linear codes [86] where it was recently shown that there are asymptotically explicit and efficient good linear binary codes that can correct a fraction of deletions [87]–[89]. Other examples include insertion/deletion burst correcting codes [90], multidimensional deletion correcting codes [91], list decoding from deletions [92], and more.

B. Bounds and the Capacity of the Insertion/Deletion Channel

Let $A_q(n,t)$ be the largest size of a t-deletion-correcting code of length n over an alphabet of size q, and asymptotically, for $\delta \in [0,1]$, let $R_q(\delta) \triangleq \liminf_{n\to\infty} \frac{\log_q(A_q(n,\lfloor \delta n \rfloor))}{n}$. In this section, we review known results on the values of $A_q(n,t)$ and $R_q(\delta)$. First, note that according to Levensthein [62], it is already known that $\Omega_t(\frac{q^n}{n^{2t}}) \leq A_q(n,t) \leq O_t(\frac{q^n}{n^t})$.

The *Levenshtein distance* between two words $\boldsymbol{x}, \boldsymbol{y}$ (not necessarily of the same length) $d_L(\boldsymbol{x},\boldsymbol{y})$ is the minimum number of symbol insertions and deletions to transform $\boldsymbol{x}$ into $\boldsymbol{y}$. The minimum Levenshtein distance of a code $\mathcal{C}$ is the minimum Levenshtein distance between its codewords. It is well known that a code has minimum Levenshtein distance $2t+2$ if and only if it is a t-deletion correcting code. Furthermore, let $\mathcal{D}_t(\boldsymbol{x}), \mathcal{I}_t(\boldsymbol{x})$ be the radius-t deletion, insertion ball of a word $\boldsymbol{x}$, respectively, and more generally for nonnegative integers t_1, t_2, $L_{t_1,t_2}(\boldsymbol{x})$ is the insertion-deletion ball containing all words that can be received by t_1 deletions and t_2 insertions to $\boldsymbol{x}$. One of the important tasks in analyzing bounds for deletion-correcting codes is the study of the sizes of the deletion balls, insertion balls, and insertion-deletion balls. While the size of the insertion ball for any radius is well-known [93] and is the same for all words (i.e., regular), the deletion ball size is not known for all words. However, its minimum, maximum, and average values are known and several upper and lower bound which are based on the number of *runs* (a run is a maximal repeat of the same symbol in the word) were studied; see e.g. [94], [95]. On the other hand, the knowledge on the ball size $L_{t_1,t_2}(\boldsymbol{x})$ is rather limited, while only the single-deletion single-insertion case was studied in [96]–[98], where the ball size was calculated for all words together with the minimum, maximum, and average size values.

For a given word $\boldsymbol{x} \in \{0,1\}^n$ the size of the insertion ball $\mathcal{I}_t(\boldsymbol{x})$ does not depend on $\boldsymbol{x}$ and it is given by $|\mathcal{I}_t(\boldsymbol{x})| \triangleq \mathcal{I}_{n,t} = \sum_{i=0}^{t} \binom{n+t}{i}(q-1)^i$ [93]. Hence, a trivial upper bound on $A_q(n,t)$ is given by $A_q(n,t) \leq \frac{q^{n+t}}{\sum_{i=0}^{t}\binom{n+t}{i}(q-1)^i}$. Asymptotically, this implies the bound $R_q(\delta) \leq (1+\delta)\left(1 - H_q\left(\frac{\delta}{1+\delta}\right)\right)$, where $H_q(x) = -x\log_q(x) - (1-x)\log_q(1-x) + x\log_q(q-1)$. Recently, Yasunaga [99] showed that for any t-deletion-correcting code $\mathcal{C}$ and any integer s such that $s < \frac{n(t+1)}{n-(t+1)}$, it holds that $|\mathcal{C}| \leq \left\lfloor \frac{(n+s)(2t+2)}{(n+s)(2t+2)-2ns} \cdot \frac{q^{n+s}}{\mathcal{I}_q(n,s)} \right\rfloor$, and this bound results with the asymptotic bound $R_q(\delta) \leq \frac{1-H_q(\delta)}{1-\delta}$, which is the best known asymptotic upper bound, at least for the binary case. Explicit upper bounds on $A_q(n,t)$ based on the deletion balls are more challenging to derive since the deletion balls are not regular. However, despite this irregularity, nonasymptotic upper bounds were derived by Kulkarni and Kiyavash [100], based upon a translation of the problem to a hypergraph and finding upper bounds on the independence number of the hypergraph using linear programming. In particular, it was shown in [100] that $A_q(n,1) \leq \frac{q^n-q}{(q-1)(n-1)}$ and in general

$$A_q(n,t) \leq \sum_{r=3}^{n-t} \frac{q(q-1)^{r-1}\binom{n-t-1}{r-1}}{\delta(r,t) + \sum_{i=t+r-(n-t)-1}^{\min(t-2,r-3)} \delta(r-2,i)} + \sum_{r=1}^{2} q(q-1)^{r-1}\binom{n-t-1}{r-1},$$

which also provides the asymptotic bound $A_q(n,t) \leq \frac{t!q^n}{(q-1)^t n^t}$ and the following bound on the rate $R_q(\delta)$ for $0 \leq \delta < 0.5$: $R_q(\delta) \leq \max_{\rho\in[0,1-\delta]}(N(\rho;\delta) - D(\rho;\delta))$, where $N(\rho;\delta) = (1-\delta)h_q(\frac{\rho}{1-\delta})$ and $N(\rho;\delta) = \max_{\max(2\delta+\rho-1,0)\leq\mu\leq\min(\delta,\rho)} \frac{(\rho-\mu)\min(\frac{\mu}{\rho-\mu},0.5)}{\log(q)}$.

As for a lower bound, in order to apply a Gilbert-Varshamov lower bound, one should consider the t-deletion t-insertion ball, i.e., $L_{t,t}(\boldsymbol{x})$. However, since this ball size is not regular, the lower bound should be derived according to [101] which uses the average size of the ball $L_{t,t}(\boldsymbol{x})$ to derive a lower bound on $A_q(n,t)$ and is given by $A_q(n,t) \geq \frac{q^n}{1/q^n \sum_{\boldsymbol{x}\in[q]^n} |L_{t,t}(\boldsymbol{x})|}$. Since the average size of the ball $L_{t,t}(\boldsymbol{x})$ is not known in general, Yasunaga calculated an upper bound on this value and derived the bound $A_q(n,t) \geq \frac{q^n}{q^{-t}I_q(n-t,t)^2 - (1-q^{-n+1})I_q(n-t+1,t-1)}$ [99]. Although this bound improves the one given by Levenshtein [102] $A_q(n,t) \geq \frac{q^n}{q^{-t}\cdot(I_q(n-t,t))^2}$, asymptotically they behave the same. The first asymptotic improvement was given recent by Alon et al. [103] where they used the asymptotic improvement of the Gilbert-Varshamov bound by Jiang and Vardy [104] and derived the following lower bound $A_q(n,t) \geq \Omega_{q,t}(\log(n)\frac{q^n}{n^{2t}})$. Their approach required to calculate the number of triangles in the confusability graph that its vertices set is $[q]^n$ and there is an edge between two nodes if and only if they share a common subsequence of length $n-t$.

Insertion and deletion errors were also studied in the average-case setting. The most studied channel is the *binary deletion channel* (BDC_p) that models the setup where bits of a transmitted message are deleted from the message randomly and independently with probability p. Perhaps the most important question regarding the BDC is determining its capacity, i.e., the maximum achievable transmission rate over the channel that still allows recovering from the errors introduced by the channel, with high probability. In spite of many efforts (see the excellent surveys [105], [106]), the capacity of the BDC_p is still not known and it is an outstanding open challenge to determine it.

In the extremal parameter regimes, the behavior of the capacity of the BDC is partially understood. When $d \to 0$ the capacity approaches $1 - h(d)$ [107]. Mitzenmacher and Drinea [108] and Kirsch and Drinea [109] showed a method by which distributions on run lengths can be converted to codes for the BDC, yielding a lower bound of $\mathbb{C}(\mathrm{BDC}_d) > 0.1185(1-d)$. Fertonani and Duman [110], Dalai [111] and Rahmati and Duman [112] used computer aided analyses based on the Blahut-Arimoto algorithm [113], [114] to prove an upper bound of $\mathbb{C}(\mathrm{BDC}_d) < 0.4143(1-d)$ in the high deletion probability regime ($d > 0.65$). Recently, Rubinstein and Con [115] showed that the Blahut-Arimoto algorithm can be implemented with a lower space complexity, allowing to extend the upper bound analyses, and proved an upper bound of $\mathbb{C}(\mathrm{BDC}_d) < 0.3745(1-d)$ for all $d \geq 0.68$. Furthermore, they also showed that an extension of the Blahut-Arimoto algorithm can be used to select better run length distributions

for Mitzenmacher and Drinea's construction, yielding a lower bound of $\mathbb{C}(\mathrm{BDC}_d) > 0.1221(1-d)$.

Another closely related studied channel is the Poisson repeat channel (PRC). It was first introduced by Mitzenmacher and Drinea [108]. In the PRC with parameter λ, each bit is (randomly and independently) replaced with a discrete number of its copies, distributed according to the Poisson distribution with parameter $0 < \lambda$. In particular, with probability $e^{-\lambda}$ the bit is deleted from the message. This channel is closely related to the BDC as demonstrated in several works [108], [116]–[118], while the lower bound on the capacity of BDC_p relies on a reduction from the PRC_λ [108].

C. Coding over Sets

In this subsection, we discuss coding schemes that address the unordered nature of the DNA storage medium. Specifically, we outline the *coding over sets* framework as formulated by Lenz et al. [60], along with subsequent research and variations.

Throughout this subsection, we consider the set of all q-ary length-L strings Σ_q^L and use $\mathcal{X}_M^L$ the denote the collection of M-subsets of Σ_q^L. In other words, $\mathcal{X}_M^L = \{S \subseteq \Sigma_q^L : |S| = M\}$. Hence, $\mathcal{X}_M^L$ represents the input space and a channel input is a set $S \in \mathcal{X}_M^L$. After passing through the channel, a sequence $\boldsymbol{x}$ in the input S undergoes one of three possibilities.

- It is received correctly without errors and we set ($\boldsymbol{x} \in \mathcal{C}$).
- It is lost and we set $\boldsymbol{x} \in \mathcal{L}$.
- It is received with errors and we set $\boldsymbol{x} \in \mathcal{E}$.

In other words, $(\mathcal{C}, \mathcal{L}, \mathcal{E})$ is a partition of S. We have the formal definition of error-ball.

For $S \in \mathcal{X}_M^L$, the *error-ball* $\mathcal{B}_{s,t,\varepsilon}^{\mathcal{T}}(S)$ is defined to be the collection of received sets S' when s (or fewer) sequences have been lost and t (or fewer) sequences have been distorted by errors of type $\mathcal{T} \in \{\mathcal{S}_\varepsilon, \mathcal{D}_\varepsilon, \mathcal{I}_\varepsilon\}$.

More precisely, let $\mathsf{Part}_{s,t}(S)$ be the set of all partitions $(\mathcal{C}, \mathcal{L}, \mathcal{E})$ of S with $|\mathcal{L}| \le s$ and $|\mathcal{E}| \le t$. Then we define

$$\mathcal{B}_{s,t,\varepsilon}^{\mathcal{T}}(S) = \left\{ S' = \mathcal{C} \cup \mathcal{E}' : \begin{array}{c} (\mathcal{C}, \mathcal{L}, \mathcal{E}) \in \mathsf{Part}_{s,t}(S), \\ \mathcal{E}' = \{\boldsymbol{x}' \in \mathcal{T}(\boldsymbol{x}) : \boldsymbol{x} \in \mathcal{E}\}. \end{array} \right\}$$

Here, $\mathcal{E}'$ denotes the set of all distinct erroneous received sequences $\boldsymbol{x}'$ after removing duplicates. So, $|\mathcal{E}'| \ge |\mathcal{E}|$.

Now, we are ready to define task of *code design*. Given an error-ball type $\mathcal{T} \in \{\mathcal{S}_\varepsilon, \mathcal{D}_\varepsilon, \mathcal{I}_\varepsilon\}$ and integers s, t, ε, design a codebook $\mathcal{C} \subseteq \mathcal{X}_q^L$ such that

$$\mathcal{B}_{s,t,\varepsilon}^{\mathcal{T}}(S) \cap \mathcal{B}_{s,t,\varepsilon}^{\mathcal{T}}(S') = \varnothing \text{ for } S, S' \in \mathcal{C},\ S \neq S'.$$

We say that $\mathcal{C}$ is an (s,t,ε)-*correcting code*.

As always, for fixed (s,t,ε) and input space $\mathcal{X}_M^L$, we are interested in designing (s,t,ε)-correcting code of large size. Equivalently, we want to minimize the redundancy of such codebooks. As before, we define the following quantity of interest.

$$\rho(L, M; s, t, \varepsilon)_{\mathcal{T}} \triangleq \min \left\{ \log \binom{q^L}{M} - \log|\mathcal{C}| : \mathcal{C} \text{ is an } (s,t,\varepsilon;\mathcal{B})_{\mathcal{T}}\text{-correcting code} \right\}. \quad (3)$$

Lower and upper bounds for $\rho(L, M; s, t, \varepsilon)_{\mathcal{T}}$ were studied in [60], and later, some bounds were improved in [119]. Below we list the best known asymptotic estimates for $\rho(L, M; s, t, \varepsilon)_{\mathcal{T}}$ when $\mathcal{T} \in \{\mathcal{S}_\varepsilon, \mathcal{D}_\varepsilon\}$. Here, $q = 2$ and the asymptotics are taken where $M = 2^{\beta L}$ for some constant β and L going to ∞.

(i) We have $\rho(L, M; s, t, L)_{\mathcal{S}_L} \sim (s + 2t)L$.
(ii) We have $\rho(L, M; s, t, L)_{\mathcal{D}_L} \sim (s + t)L$.
(iii) We have $t \log M + t\varepsilon \log L \lesssim \rho(L, M; 0, t, \varepsilon)_{\mathcal{S}_\varepsilon} \lesssim 2t \log M + 2t\varepsilon \log L$.
(iv) We have $\rho(L, M; 0, t, \varepsilon)_{\mathcal{D}_\varepsilon} \gtrsim \lfloor t/2 \rfloor \log M$. Also, $\rho(L, M; 0, t, \varepsilon)_{\mathcal{D}_\varepsilon} \lesssim t \log M + 2t\varepsilon \log(L/2)$ if $t > 1$ and $\lesssim t \log M + 4\varepsilon \log L$ if $t = 1$.

More results about other error types can be found in [60], [119]. In these same works, explicit code constructions with efficient encoding and decoding procedures are also given. Of significance, Wei and Schwartz constructed a class of explicit codes whose redundancy that is logarithmic in M [119].

Next, we list certain related works that are similar to this "coding over sets" framework.

- In [120], Kovačević and Tan considered a more general setup where unordered multisets are transmitted. In this work, both upper and lower bounds for optimal code sizes are obtained and several code constructions were given.
- In [121], [122], Sima et al. considered the case where $s = 0$ and the total number of substitution errors is bounded (but not limited to some number ε). In this regime, Sima et al. provided a class of explicit codes whose redundancy is logarithmic in M and L.
- In [123], Song et al. proposed a new metric to correct both sequence loss and substitution errors. Using this metric, they established Singleton-like and Plotkin-like upper bounds on the code size.

Now, besides the above-mentioned coding solution, an index-based solution is typically adopted in most practical experiments. Here, we have a set of addresses $\mathcal{A} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M\}$ and we store this address information as a prefix to each DNA strand. That is, the information stored in the DNA storage system is an indexed set of strands $\{(\boldsymbol{a}_i, \boldsymbol{d}_i) : i \in [M]\}$, where $\boldsymbol{d}_i$ is the data part of the i-th strand (see also, [58], [124]). As the addresses $\mathcal{A}$ are also known to the user, the user can identify the information after the decoding process. As these addresses along with the stored data are prone to errors, this solution needs further refinements.

Now, in the experiment [12], Organick et al. first clustered the strands with respect to the edit distance. Then they determine a consensus output amongst the strands in each cluster and finally, decode these consensus outputs using a classic concatenation scheme. For this approach, the clustering step is computationally expensive. When there are $\mathcal{M}$ reads, the usual clustering method involves $\mathcal{M}^2$ pairwise comparisons to compute distances. This is costly when the data strands are long, and the problem is further exacerbated if the metric is the edit distance. Therefore, in [125], Rashtchian et al. developed a distributed approximate clustering algorithm and clustered 5 billion strands in 46 minutes on 24 processors.

In [126], Chrisnata et al. proposed an approach that avoids clustering, using *the bee-identification problem*. Informally, the bee-identification problem requires the receiver to identify M "bees" using a set of M unordered noisy measurements [127], [128]. Later, in [126], Chrisnata et al. generalized the setup to multi-draw channels where every bee (strand) results in N noisy outputs (reads). The task then is to identify each of the M bees from the MN noisy outputs and it turns out that this task can be reduced to a minimum-cost network flow problem. In contrast to previous works, the approach in [126] utilizes only the noisy addresses, which are of significantly shorter length, and the method does not take into account the associated noisy data. Hence, this approach involves no data comparisons.

Later on, in [129], Singhvi et al. considered an intermediate, data-driven approach to the identification task by drawing ideas from the clustering and the bee identification problems. By proposing a data-driven pruning procedure, they demonstrated that on average the pruning procedure uses only a fraction of $\mathcal{M}^2$ data comparisons (when there are $\mathcal{M}$ reads).

IV. Constrained Codes

Constrained codes have a long history, dating back to Claude Shannon's landmark 1948 paper, in which he introduced the discrete noiseless channel model [130]. They have found manifold applications in digital communications and data storage systems. Background on the theory and practice of constrained coding for storage can be found in texts, such as Immink [131] and Marcus et al. [132], as well as several expository and survey articles, including [133]–[136]. In magnetic recording, classical runlength-limited (d, k) codes, which require at least d and not more than k zero symbols between consecutive one symbols, improved the system performance by reducing intersymbol interference to support peak and amplitude detection and avoiding synchronization loss in data-driven timing recovery [135]. Codes with spectral nulls at certain frequencies, notably balanced or DC-free codes with a null at zero frequency, found application in magnetic tape recorders and optical disk systems [131], [137]. In multilevel flash memory, constrained coding techniques have been proposed to reduce intercell interference (ICI) along wordlines, bitlines, and, more generally, in two-dimensional arrays to improve the accuracy of data recovery [138].

Constrained coding is also very relevant to DNA storage. Experimental studies have shown that certain properties of DNA sequences increase the likelihood of insertion, deletion, and substitution errors in the synthesis and sequencing steps of the DNA storage process. Repetitions of the same nucleotide, referred to as homopolymer runs, that exceed even as few as four to six nucleotides in length have been observed to significantly increase the likelihood of insertion and deletion errors [139], [140]. The fraction of G and C nucleotides in a strand, referred to as the GC-content, has also been found to contribute to insertion, deletion, and substitution errors when it deviates from the 40% to 60% range [140], [141]. These experimental results have motivated the development of constrained codes satisfying maximum homopolymer runlength (MHR) constraints and balanced GC constraints, either separately or in combination. Secondary structures formed when a DNA strand folds back upon itself are also detrimental to DNA storage. These structures can arise when a strand contains non-overlapping substrings that are reverse complementary, in the sense of the Watson-Crick complement property for nucleotides: $\overline{\mathsf{A}} = \mathsf{T}, \overline{\mathsf{G}} = \mathsf{C}, \overline{\mathsf{C}} = \mathsf{G}, \overline{\mathsf{T}} = \mathsf{A}$. Secondary structure avoidance codes that avoid secondary structures by eliminating or reducing the likelihood of reverse complementary substrings are therefore also of interest.

An early overview paper by Yazdi et al. [19] on DNA-based storage discussed the important role of constrained coding in archival and random access applications and introduced coding schemes that incorporate several sequence and address constraints. More recent surveys of information encoding techniques for DNA are found in [25]–[27]. In the following sections, we summarize the progress that has been made on various aspects of constrained coding.

A. Maximum Homopolymer Runlength (MHR) Constraints

Grass et al. [17] used a mapping of 2 bytes of information to 3 nucleotides that yielded MHR constrained sequences with $m = 3$, for an encoding rate of 1.78 bits/nt. In [18], Bornholt et al. proposed a simple variable-length coding scheme that completely eliminates repetitions of nucleotides; that is, the strings $\mathsf{AA}, \mathsf{GG}, \mathsf{CC}, \mathsf{TT}$ are forbidden. The scheme uses a Huffman code to translate binary bytes into strings of 5 (or occasionally 6) digits over the alphabet $\{0, 1, 2\}$. The ternary strings are then translated into nucleotides using a rotating mapping in which the nucleotide associated with each ternary digit depends on the previous nucleotide. This code has a maximum homopolymer runlength $m = 1$, with a rate of 1.6 bits/nt. Immink and Cai [142] sought to overcome the efficiency loss associated with the coding scheme in [18]. By means of a precoding operation, they associated q-ary sequences satisfying the MHR constraint with $m = k + 1$ to k-constrained q-ary sequences in which nonzero symbols are separated by a run of zero symbols of length 0 to k, and they determined the information capacity of the q-ary k-constrained system. They then presented three methods for converting binary sequences into q-ary k-constrained sequences. The first method employs the classical method of cascadable block codes of maximum size. The second method uses sequence replacement methods to construct rate $\frac{n-1}{n}$, k-constrained codes. The third method generates a 4-ary code sequence from a k'-constrained binary sequence by means of two precoding operations followed by a simple arithmetic mapping from pairs of binary digits to a 4-ary symbol. The resulting symbol sequence satisfies the $m = \lceil k'/2 \rceil$ MHR constraint, for $k' > 2$.

B. Balanced GC Constraints and MHR Constraints

In [19], Yazdi et al. noted that DNA strands with approximately 50% GC content are more stable than those with lower or higher GC-content and have better coverage during sequencing. They considered the well-studied class of binary codes with a D-bounded running digital sum (D-BRDS) constraint, in which any codeword prefix has an imbalance in

the number of zeroes and ones of no more than D. By mapping each binary 0 to one of the nucleotides $\{\mathsf{A}, \mathsf{T}\}$ and each binary 1 to a nucleotide $\{\mathsf{G}, \mathsf{C}\}$, they created a large set of DNA codewords satisfying a D-bounded GC-prefix-balanced (D-GCPB) constraint. Note that these codewords implicitly satisfy a MHR constraint, as well as certain bounds on minimum distance.

Several authors then considered the combination of GC-content constraints and MHR constraints. In [10], Erlich and Zielinski proposed the use of fountain codes, screening the DNA oligos for homopolymer runs of no more than 3 nucleotides and GC-content in the 45% to 55% range. Song et al. [143] further investigated codes satisfying a 3-MHR constraint with GC-content close to 50%. For $3 \leq n \leq 35$, they designed a mapping of $k(2n-1)$ bits into an ordered list of concatenations of k length-n words satisfying the 3-MHR constraint and with GC-content as close to 50% as possible. Simulation studies for $n \in \{5, 6, \ldots, 121\}$ and $k = 20$ confirmed that the GC-content of a large random sample of codewords fell within the 40% to 60% range. For $n \geq 10$, the codes achieve rates at least 1.9 bits/nt. The scheme also ensured error propagation of not more than $2n$ bits due to an erroneous nucleotide.

Wang et al. [144] presented an algorithm for generating content-balanced, runlength-limited (C-RLL) codewords , with 40% to 60% GC-content and MHR constraint $m = 3$. Longer codewords that satisfy the constraints both locally and globally were constructed via concatenation of selected C-RLL codewords. For C-RLL codeword length $n = 12$, a rate 1.917 bits/nt was achieved. Immink and Cai [145] presented a new method for precoding k-constrained DNA sequences into $(k+1)$-MHR sequences with constrained GC-content that facilitates the construction of longer sequences satisfying the $(k+1)$-MHR constraint with GC-content in the range of 40% to 60%. Generating functions were used to enumerate the length-n, k-constrained words of specified GC-content. Most of the constructed codes are highly efficient, with rate $2 - \frac{1}{n}$. In [146], they further investigated properties and constructions for codes with m-MHR and balanced GC-content constraints. Generating functions and approximations for the number of binary and quaternary sequences with maximum runlength and weight constraints were derived. The relative merits of two coding approaches, one of which directly maps binary source sequences to constrained DNA sequences while the other utilizes an intermediate binary mapping, were compared with respect to redundancy and code complexity.

McBain et al. [147] designed a quaternary Markov source that generates sequences satisfying MHR and stochastic GC-content constraints. They designed a homophonic code that maps i.i.d. binary source sequences to the Markov source in such a way that the empirical distribution of the generated sequences approximates the source. Fan et al. [148] proposed a method for computing the channel capacity of m-MHR constraints that are also strong-$(l,\ \delta)$-locally-GC-balanced, meaning that the number of G and C symbols in every subsequence of length $l' \geq l$ is bounded between $\left[\frac{l'}{2} - \delta, \frac{l'}{2} + \delta\right]$. The calculation makes use of the relationship between the capacities of the binary and quarternary constrained channels.

C. Error-Correcting Constrained Codes

Schemes that integrate code constraints with error-correction properties have been proposed. Limbachiya et al. [149] enumerated the sequences of length n with the $m = 1$ MHR constraint (the "no-runlength" constraint) and GC-weight w, where the GC-weight refers to the number of G and C nucleotides. They also provided a lower bound on the number of codewords of length n satisfying these two constraints with minimum Hamming distance d. They provided an altruistic algorithm for constructing a codebook with these properties. Weber et al. [150] provided an easily computed recursive formula to determine the number of quaternary words with any fixed length, GC-weight, and MHR constraint. For the no-runlength constraint, they compute the maximum size of a single error-detecting code, i.e., with minimum distance two, with a specified GC-weight. Deng et al. [151] considered asymmetric error models for DNA sequencers. They combined the MHR constrained code design methodology of Wang et al. [144] with a an optimized protograph low-density parity-check (LDPC) code. This combined coding scheme achieving improved error rate performance in computer simulations with an overall code rate of about 1.98 bits/nt.

Cai et al. [152] studied a segmented edit error model, under which a sequence is divided into segments of fixed length, each of which can undergo at most one insertion or deletion error. For a q-ary alphabet, $q \geq 3$, they proposed reduced redundancy code constructions to correct a single insertion or deletion per code segment. Specializing to the quaternary DNA alphabet, they provide an efficient construction of a segmented error-correcting code that enforces a constraint on GC-content in each segment.

Cai et al. [153], [154] and Nguyen et al. [155] devised coding schemes that combine code constraints with the ability to correct a single indel (insertion or deletion) error or single edit (insertion, deletion, or substitution) error. The former codes are GC-balanced codes, obtained by a modification of the Knuth balancing method. The latter codes satisfy a (ℓ, ϵ) MHR, GC-content constraint where the maximum homopolymer runlength is ℓ, and the GC-content is within $\pm\epsilon$ of 50%. Both of these schemes are asymptotically capacity-achieving and offer linear time encoding, and the latter codes also limit error propagation. Liu et al. [156] provided another construction of (ℓ, ϵ) MHR, GC-content constrained codes based on enumeration coding techniques that increase code efficiency at shorter block lengths. The encoding and decoding have polynomial time complexity. They also introduced codes that satisfy MHR, GC-content constraints on all prefixes of the codewords. Park et al. [157] uses an iterative encoding method based on a greedy algorithm to achieve MHR and GC-content constraints. The short codewords ensure limited error propagation with a reduced number of decoded bit errors per nucleotide error. The iterative coding method consists of a randomization step, M-ary mapping, and a verification step. Simulation results were presented for a code satisfying the 3-MHR constraint with DC-content in the range from 45% to 55% with information density 1.833 bits/nt.

Park et al. [158] proposed a coding technique for m-MHR and GC-content constraints in which binary input data is encoded into a collection of oligo sequences of specified length. The scheme combines the modified Knuth balancing method of [155] with an input-dependent bit insertion-based constrained code (BIC) that limits homopolymer runlengths. Lower and upper bounds on the average information density of the BIC construction were derived and worst-case results were described. Empirical results showed increasing average information densities within 1.5% of capacity for $m = 2, 3, 4$. To provide resilience to insertion, deletion, and substitution errors, inter-oligo LDPC coding based on the 5G New Radio (NR) standard was applied across the collection of oligo sequences. Simulation results for $m = 3$ and GC-content in the 40% to 60% range were provided, along with comparisons to previously proposed methods for combining constraints with error correction.

Weindel et al. [159] compared the error-rate performance of error-mitigating constrained codes that limit homopolymer runlengths and balance GC content to that of schemes which "embrace errors" by combining data randomization with error correction coding for error control. They found that in the error regimes most relevant to current DNA storage systems, the latter approach is more efficient in terms of overall information rate for a given level of performance.

D. Secondary Structure Avoiding Codes

Secondary structures arise when a DNA strand folds back upon itself through complementary base-pair self-hybridization. These structures can take many forms, most notably stem-loop configurations, also referred to as hairpins. These structures render the strand inactive for DNA computing settings and have been shown experimentally to cause read errors in DNA storage applications (See [160], [161] and references therein.) In [160], Milenkovic and Kashyap identified code design criteria that reduced the likelihood of secondary structure formation using an analysis of Nussinov's folding algorithm, a dynamic programming approach to approximately predicting secondary DNA structures. They enumerated and constructed sequences satisfying certain shift properties, and also consider the case of sequences with a constant GC content. The shift properties require that the Watson-Crick distances between a codeword and a specified number of its shifts exceed a given threshold. They also demonstrated a code based on a cyclic simplex code that combines shift properties with a base runlength constraint, constant GC content, Hamming distance constraint, and reverse complement Hamming distance constraint.

Benerjee and Banerjee [162] constructed codes that satisfy constraints on reverse Hamming distance, reverse complement Hamming distance, and GC content. They showed that concatenations of codewords satisfy the 3-MHR constraint and are free of secondary structures with stem length more than 2. They also derived a lower bound on the maximum size of codes with these properties. Their construction uses a modification of Reed-Muller codes defined over the ring $\mathbb{Z}_6$. Another construction inspired by Reed-Muller codes over $\mathbb{Z}_6$ was presented in [163], yielding families of codes satisfying reverse Hamming distance and reverse complement Hamming distance constraints that are free of secondary structures with stem length greater than two. In [164], using the ring $\mathbb{Z}_5$, they constructed DNA codes that are free from secondary structures with stem length more than two with the 4-MHR constraint, while Benerjee et al. [165] constructed *conflict free* codes in which any two consecutive substrings of a codeword are not the same. These codes also satisfy Hamming, reverse Hamming distance, reverse complement Hamming distance, and DC-content constraints and are free of secondary structures with stem length greater than two.

Nguyen et al. [166] gave two construction of codes that avoid secondary structures with stem length greater than m, for any given $m \geq 2$. They referred to such codes as m-SSA codes. The first construction concatenates blocks of length m from a specified set, while the second imposes a symbol composition constraint on every length-m subsequence $\mathbf{z}$ of a codeword $\mathbf{c}$ whereby, for complementary symbol pairs x and $y = \overline{x}$, the subsequence $\mathbf{z}$ must contain at least k symbols x and at most $k-1$ symbols y, for some $0 < k \leq m$. They provided a recursion for the size of codes of length n that satisfy this constraint for general m and $k = 1$ and determined the asymptotic rate. For $m = 3$, their construction yielded an asymptotic rate of about 1.3031 bits/nt, which exceeds the rate of $\frac{1}{2}\log 5 \approx 1.1609$ bits/nt achieved in [164]. For $m \geq 3\log n + 4$, they provided an efficient encoder based on sequence replacement techniques that requires only one redundant symbol.

Chu et al. [167] defined a TC-m-*dominant* sequence to be one for which, in every subsequence of length m, the sum of the number of occurrences of nucleotides T and C is greater than $m/2$. They showed that, for m odd, a code consisting of TC-m-dominant sequences is an m-SSA code. For $m = 3$ and $m = 5$, they used recurrence relations to calculate lower bounds on the capacity of m-SSA codes. They computed upper bounds on the capacity of m-SSA codes in terms of maximal m-SSA generating sets, defined as sets of length-m sequences of maximal size containing no two reverse complement sequences. Using a brute-force search, they showed that for $m = 2$, the capacity is about 1.6719 bits/nt, and for $m = 3$ it is about 1.5515 bits/nt, which is achieved by the TC-3-dominant code construction.

E. Related Works

Finally, we list selected works on other topics related to data encoding for DNA storage.

- Several authors have addressed the high cost of synthesis by proposing source coding algorithms that combine biochemical *constrained coding with compression* methods. The primary application pf these schemes has been the encoding of images data into DNA. See, for example, Yazdi et al. [5], Dimopoulou et al. [168], Pic and Antonini [169], Pic et al, [170], and Biswas et al. [171]
- The reconstruction problem in DNA storage can be formulated in terms of uniquely identifying a string from the multiset of its substrings of a certain length. Motivated

by early work of Gabrys and Milenkovic [172], Elishco et al. [173] studied constructions, bounds, and asymptotic rates for codes containing *k-repeat free sequences* in which no substring of length k is repeated, where $k < n$. They showed that the capacity of k-repeat free sequences is 1 for $k = a\log(n)$ when $a > 1$. For binary sequences and $k = 2\lceil\log(n)\rceil + 2$, they provide an encoder that with only two redundant bits. For $1 < a \leq 2$ they give an encoding algorithm for $k = a\log(n)$. They also calculate the capacity of k-repeat free sequences that satisfy additional local constraints, such as runlength constraints, showing that the k-repeat free constraint does not reduce the capacity when $a > 2\log_\lambda(2)$, where the capacity of the constraint is $\log(\lambda)$. They generalize the capacity analysis to multidimensional k-repeat free arrays.

- In support of their random-access DNA-storage implementation, Yazdi et al. [19] imposed a number of constraints on the short address sequences, including balanced GC-content, large Hamming distance, absence of secondary structures, and *mutual uncorrelatedness*. The latter property requires that, for any two not necessarily distinct length-n sequences, $\mathbf{a}, \mathbf{b}$, no proper prefix of $\mathbf{a}$ is equal to any proper suffix of $\mathbf{b}$. Yazdi et al. [174], Levy and Yaakobi [175], Chee et al. [176], and Cao et al. [177] examined properties and constructions of mutually uncorrelated codes that also possess other features advantageous for reliable random-access DNA storage.

V. Coding for Synthesis

A. Coding for Efficient Synthesis

The synthesis of DNA strands for DNA storage is a time-consuming and costly process. Coding can be used to optimize the efficiency of the synthesis process, providing savings in synthesis time and expense. In [178], Lenz et al. studied a synthesis machine that creates a large number of DNA oligos in parallel, where each oligo is grown by at most one nucleotide in each synthesis cycle. Such a synthesis process has been used in most DNA storage experiments [11], [179], [180] and is also characteristic of photolithographic synthesis [181].

The machine uses a fixed supersequence of possible nucleotides to successively append nucleotides to the strands. At each iteration, the machine adds the next nucleotide in the supersequence to a selected subset of the DNA strands. This process continues until the end of the supersequence is reached. Note that each synthesized DNA strand is a subsequence of the supersequence. Figure 2 represents the synthesis of three strands of variable lengths from the supersequence $\mathbf{S} = \mathsf{ACGTACGT}$ with total synthesis time $T = 8$.

In [178], the problem of finding a *synthesis code* to maximize the amount of information that can be synthesized in a given synthesis time T was posed. Both fixed-length codes and variable-length codes were considered. Maximum synthesis rates, meaning asymptotic information rates measured in the number of information bits per synthesis cycle, were defined. Namely, for a semi-infinite synthesis sequence $\mathbf{S} = S_1, S_2, \ldots$

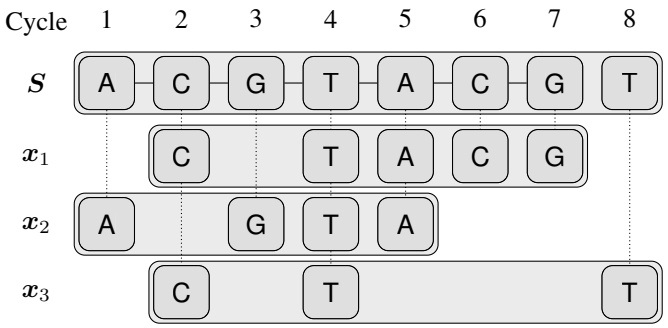

Fig. 2. Synthesis of three strands $\boldsymbol{x}_1 = (\mathsf{CTACG}), \boldsymbol{x}_2 = (\mathsf{AGTA})$, and $\boldsymbol{x}_3 = (\mathsf{CTT})$ using the synthesis sequence $\boldsymbol{S} = (\mathsf{ACGTACGT})$.

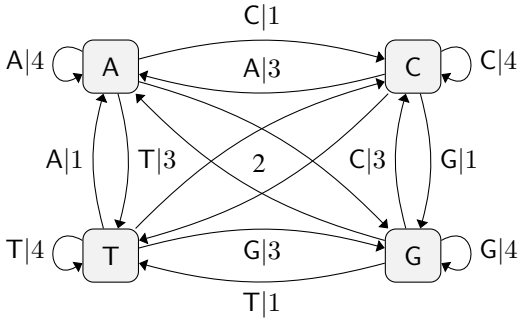

Fig. 3. Synthesis cost graph for DNA synthesis using the alternating sequence ACGT ACGT ….

and $0 \leq \alpha \leq 1$, with $N^{\lfloor \alpha T \rfloor}(\boldsymbol{S}_{1:T}, T)$ defined as the number of sequences of length $\lfloor \alpha T \rfloor$ that can be synthesized from $\mathbf{S}$ in time less that T, the corresponding maximum synthesis rate is

$$R(\mathbf{S}, \alpha) = \limsup_{T\to\infty} \frac{\log(N^{\lfloor \alpha T \rfloor}(\mathbf{S}_{1:T}, T))}{T}.$$

For codes with variable length sequences, with corresponding maximum codebook size $N^*(\boldsymbol{S}_{1:T}, T)$, the maximum synthesis rate is

$$R^*(\mathbf{S}) = \limsup_{T\to\infty} \frac{\log(N^*(\mathbf{S}_{1:T}, T))}{T}.$$

It was shown that these rates can be determined by applying the theory of *discrete noiseless channels* [130], [182], also known as finite-state (noiseless) channels with cost, which can be represented by a finite directed graph with symbol and cost labels on edges. Specifically, the synthesis process described above is modeled using the *synthesis cost graph* shown in Fig. 3, where the labels, such as $\mathsf{C}|1$ on edges represent an appended DNA base and the associated cost (i.e., synthesis time), respectively. For a periodic synthesis sequence $\mathbf{S} = (\mathbf{s}, \mathbf{s}, \ldots)$, it was shown that $R^*(\mathbf{S})$ is the capacity of the associated synthesis cost graph and $R(\mathbf{S}, \alpha)$ is the cost-constrained capacity, both of which can be computed [182]–[184].

Lenz et al. [178] also showed that for a q-ary alphabet $\{0, 1, 2, \ldots, q-1\}$, the *alternating sequence* $\mathbf{S} = \mathbf{A}_q \stackrel{\text{def}}{=} (01\ldots(q-1), 01\ldots(q-1), \ldots)$ that cyclically repeats the symbols in the alphabet in ascending order maximizes both

of the information rates $R(\mathbf{S}, \alpha)$ and $R^*(\mathbf{S})$. Specifically, for $q = 2$, for all $0 \leq \alpha \leq 1$,

$$R(\mathbf{A}_2, \alpha) = \begin{cases} \alpha h(\alpha^{-1} - 1), & \text{if } \alpha \geq \frac{2}{3} \\ \alpha, & \text{otherwise} \end{cases},$$

where $h(x)$ is the binary entropy function. Further, for any $q \geq 2$,

$$R^*(\mathbf{A}_q) = -\log z_q,$$

where z_q is the largest root of the polynomial $\sum_{i=1}^{q} z^i - 1$.

In particular, for $q = 4$, corresponding to the DNA alphabet, the maximum synthesis rate is $R^*(\mathbf{A}_4) \approx 0.9468$. This rate compares favorably to that of the naive synthesis scheme in which oligos of length n are synthesized from the supersequence of length $4n$ that repeats the substring ACGT exactly n times, adding exactly one nucleotide from each repetition of the substring. This scheme achieves an information synthesis rate of 0.5 bits/cycle, since it requires 4 cycles to synthesize each nucleotide corresponding to two information bits. It also compares favorably to the scheme proposed in [178] that achieves an information rate of 0.8 bits/cycle. For a periodic synthesis supersequence $\mathbf{S} = (\boldsymbol{s}, \boldsymbol{s}, \ldots)$, methods for designing codes that approach the maximum achievable rates $R(\mathbf{S}, \alpha)$, including $R^*(\mathbf{S})$, with efficient encoding and decoding algorithms were considered by Liu et al. in [185], [186]

If the goal is to efficiently synthesize constrained sequences, such as those discussed in Section IV, a natural problem is to determine the synthesis sequence that minimizes the average synthesis time for the set of constrained sequences. For a given synthesis sequence, the average synthesis time can be computed from the capacity of a graph that incorporates the constraints into the synthesis cost graph of the synthesis sequence using the techniques discussed above. Finding the optimal synthesis sequence, however, presents a challenging problem that remains unsolved.

A different, but related, problem was considered by Elishco and Huleihel [187]. Motivated by the work of Makarychev et al. [188], they determined asymptotically tight high probability lower and upper bounds on the cost of DNA synthesis for a set of DNA strands randomly drawn from a Markovian distribution modeling a general maximum homopolymer run-length constraint with parameter $m \geq 1$. They concluded that the alternating sequence $\mathbf{A_4} = \mathsf{ACGT\ ACGT}\ \ldots.$ is asymptotically optimal in the sense of achieving the smallest possible cost.

Chrisnata et al. [36] investigated efficient DNA synthesis codes that can correct a single insertion or deletion error. For any synthesis time $n < T \leq 4n$, they considered codes based on Varshamov-Tenengolts codewords of length n with synthesis time at most T. They gave a lower bound on the size of such codes, showing that, in particular, for $2.5n \leq T \leq 4n$, there exists a synthesis-constrained single indel-correcting code with has redundancy at most $3+\log_2(n)$ bits. They provided several explicit encoders that map binary strings into these codewords with rates close to those promised by the bounds.

B. Coding for Emerging Synthesis Methods

The main challenge of making DNA storage systems competitive relative to existing storage technologies is the synthesis cost. The simplest and straightforward approach to reduce the cost is to increase the volume of data coded into a given length of oligos, while the information capacity can be measured by bits/symbol or bits/synthesis-cycle. The naive approach, working over $\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}$ has a theoretical limit of $\log_2 4 = 2$ bits/symbol, while using error-correction codes can significantly decrease this limit. For example, the information rate in [16] and [189] is at most $\log_2 3 \approx 1.58$ since they imposed every two consecutive symbols to be distinct. On the other hand, if additional encoding characters are introduced, it is possible to achieve a logarithmic growth in the information capacity, further reducing the DNA storage cost.

Composite DNA symbols were first introduced in [4], [20] to leverage the significant information redundancies built into the synthesis and sequencing technologies. A composite symbol is a representation of a position in a sequence that does not store just a single nucleotide, but a mixture of the four nucleotides. That is, a composite symbol can be abstracted as a quartet of probabilities $(p_{\mathsf{A}}, p_{\mathsf{C}}, p_{\mathsf{G}}, p_{\mathsf{T}})$, such that p_b describes the fraction of the nucleotide $b \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ present in the mixture, where $0 \leq p_b \leq 1$ and $p_{\mathsf{A}} + p_{\mathsf{C}} + p_{\mathsf{G}} + p_{\mathsf{T}} = 1$. For example, $W = (0.5, 0, 0.5, 0)$ represents a composite symbol in which there is a probability of $0.5, 0, 0.5$, and 0 of seeing $\mathsf{A}, \mathsf{C}, \mathsf{G}$, and T, respectively. In the composite DNA oligo $\mathsf{A}W\mathsf{T}$, two types of DNA sequences will exist in the storage container, namely AAT and AGT. When synthesizing a composite oligo, a cluster of multiple copies of the synthesized oligo are being generated simultaneously, such that roughly p_b of them contain the nucleotide b. Thus, in sequencing, one needs to sequence a fraction of the copies from the cluster and then to estimate $(p_{\mathsf{A}}, p_{\mathsf{C}}, p_{\mathsf{G}}, p_{\mathsf{T}})$ in each position to identify the composite symbols.

The idea of introducing a larger alphabet by composite symbols was later further extended by Preuss et al. [21]. In their paper, the authors introduce the *combinatorial composite synthesis method*, in which the composite symbols are called *combinatorial symbols* and their building blocks are *shortmers*. A shortmer (also known as a *motif*) is a fixed-length sequence that consists of DNA bases. Therefore, the symbols in the alphabet are mixtures of shortmers, where each symbol is represented by a set of $w > 0$ distinct shortmers. Thus, a set of shortmers is synthesized using a standard DNA synthesis technology. Then, data is encoded into combinatorial symbols over the shortmers, and DNA strands are generated by a biochemical process called *ligation* [21], which connects the pre-synthesized shortmers. To improve the data reliability and to allow easier detection of the shortmers, they are selected as a subset of all shortmers of a specific length. Other extensions of this method can be found in [190], [191].

DNA composite introduces several coding and algorithmic challenges. The first work that studied error-correcting codes for composite DNA was done by Zhang et al. [192]. In their work, they assumed that the probability of every composite symbol is a multiple of $1/k$, t is the number of erroneous

composite symbols, and ℓ is a parameter that limits the change in the probabilities of the erroneous symbols. They ignored insertions and deletions and then used a BCH-based scheme to correct the erroneous composite symbols. Another model of the composite synthesis was studied by Walter et al. in [193], where the authors suggested several error-correcting codes for substitution errors, strand loss and deletions.

For the combinatorial composite synthesis, Sabary et al. studied in [194] the case where one or more shortmers are not represented in the sequencing reads. They modeled these cases as asymmetric errors and suggested several error-correcting codes, along with an explicit encoder and decoder.

From a more information-theoretic point of view, Kobovich et al. [195] studied how to choose the probabilities of the composite symbols in order to maximize the DNA composite channel capacity. The authors modelled this channel with composite inputs as a multinomial channel, and proposed an optimization algorithm for its capacity achieving input distribution, for an arbitrary number of output reads. The algorithm is termed multidimensional dynamic assignment Blahut-Arimoto (M-DAB), and is a generalized version of the DAB algorithm, proposed by Wesel et al. [196] developed for the binomial channel. Yet another work [195] studied how to choose these probabilities in order to maximize the decoding success probability of the maximum likelihood decoder.

The required number of sequencing reads to retrieve data, which is encoded with the combinatorial symbols, was studied in [197]. In [197], the authors used a Markov-chain model and other probabilistic methods to compute the probability of successful retrieval and presented an algorithm that calculates the number of reads that guarantee successful decoding with high probability. In [198], the authors studied the expected number of reads in order to decode a composite strand or a group of composite strands.

Additional low-cost synthesis method is the *enzymatic synthesis method*, which was suggested by Lee et al. [189]. This method employs a combination of two enzymes, apyrase and polymerase, to construct a desired DNA sequence. The error characteristic of this method can be described as *repeat* error, where any base in the designed strand is synthesized one or more times consecutively or not synthesized at all. That is, this method can be described as a communication channel with deletions and *sticky insertions* errors. Jain et al. [199] studied the capacity of the enzymatic synthesis to optimize the synthesis time. To do so, they created a graph representing the possible sequences that can be synthesized with this method, considering possible errors. Next, by capturing structural observations on the graph, they were able to minimize the required number of synthesis cycles. They also describe how error-correcting codes can be used to overcome repeat errors. Shafir et al. [200] suggested reconstruction algorithms and designed constrained codes for the enzymatic synthesis method.

Recently, Antkowiak et al. [181] introduced a new synthesis method that improves the costs and the latency of DNA synthesis. The method, termed *photolithographic synthesis* or *light-directed synthesis*, uses UV light and small mirrors (micromirrors), to monitor a stream of DNA bases. In this way, bases can be added to synthesized strands faster and with lower costs. However, this method shows significantly higher error rates compared to the previous synthesis methods [201]. Therefore, it requires a sophisticated coding solution, as suggested in [181].

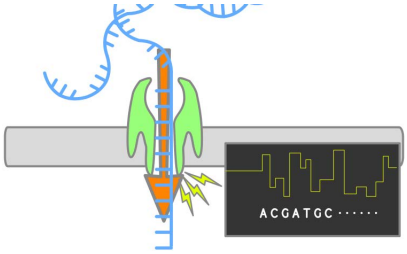

Fig. 4. Description of the nanopore sequencing process.

VI. Coding for Sequencing

One of the main components in any DNA storage system is a DNA sequencer, which reads back the encoded strands that store the user's information. The sequencing output is digital representations of the strands, which are used in the decoding process. Nowadays, DNA sequencing is primarily performed using two main methods, Illumina and nanopore (Oxford Nanopore Technology - ONT), which are named, for simplicity, after the companies that developed them. Illumina sequencing technology employs a method called *sequencing by synthesis*, in which millions of short DNA fragments (strands of lengths which are up to 150 bases) are simultaneously sequenced as they are synthesized on a solid surface. To do so, the bases (nucleotides) of the strands are altered in a way that they are identifiable by a unique fluorescent dye that is attached to each base. Thus, in each cycle, nucleotides are identified by their color, and the complementary strand is synthesized accordingly. This technology has a significant advantage in precision, and it is known that its output suffers from low error-rate [6]. On the other hand, *nanopore sequencing* is an emerging technology that is based on single-strand sequencing. In this technology, a single DNA strand is passing through a small pore (called nanopore). As each DNA nucleotide passes through the nanopore, it induces a unique change in electrical current and thus can be identified. Fig. 4 describes the nanopore sequencing process. The advantages of nanopore sequencing include the increased portability of its platforms and the ability to read longer fragments of DNA. As a result of these benefits, nanopore sequencers have been used extensively in existing DNA storage systems. Unfortunately, one of the potential drawbacks is that these sequencers have exhibited high error rates.

At a conceptual level, there are several aspects of nanopore sequencing that may lead to errors during the readout phase. (i) Inter-symbol Interference (ISI): This occurs because multiple nucleotides may influence the observed current at the same time. (ii) Shift errors: This phenomenon may occur when the strand of nucleotides skips forward or possibly backward in the nanopore. (iii) Noise: Each measurement collected of the modulation signal is subject to random noise. Based on these

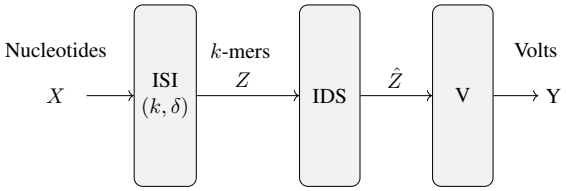

Fig. 5. Description of the nanopore channel as a concatenation of three communication channels.

three aspects, the reading process of a nanopore sequencer can be modeled as the concatenation of three communication channels as illustrated in Fig. 5. The ISI channel is characterized by the parameters (k, δ). Conceptually, it slides a window of size k over an input sequence of length n, generating a sequence of $n-k$ overlapping strings of k consecutive nucleotides, or k-mers.

For an input sequence composed of four nucleotides, there are 4^k possible k-mers. The parameter δ controls how quickly the sequence is read by using δ-decimation of the sequence of k-mer reads. The IDS channel introduces insertion, deletion, and substitution errors into the sampled sequence of k-mer symbols. The V channel then deterministically generates for each k-mer a discrete voltage level according to a function $f : \{\mathsf{A}, \mathsf{T}, \mathsf{C}, \mathsf{G}\}^k \to \{0, 1, \ldots, b-1\}$.

For example, if one considers a noiseless IDS channel, i.e., the IDS channel is the identity channel, then for a given input $x_1, x_2, \ldots, x_n$, and assuming $\delta | (n-k)$, the output of the nanopore channel is then:

$$f(x_1, \ldots, x_k), f(x_{\delta+1}, \ldots, x_{\delta+k}), \ldots f(x_{n-k+1}, \ldots, x_n).$$

The V channel uses the *composition function* f that counts how many of each type of nucleotide the k-mer contains. For example, if $(k, \delta) = (4, 2)$ and the input to the nanopore channel is $(\mathsf{A}, \mathsf{A}, \mathsf{T}, \mathsf{G}, \mathsf{A}, \mathsf{T})$, then the output would convey the information that the first k-mer read contains two A nucleotides, a single G, and a single T, and that the third k-mer read contains a single A, a single G, and two Ts. Note that the ordering of these symbols is not known.

In [202], a deterministic abstracted model of the nanopore channel was discussed. In their model, k-mers are obtained from the nanopore channel, and substitution errors may occur in each k-mer before measuring its voltage with a function similar to f. The authors studied *balanced* functions f, in which the number of k-mers that are mapped to each discrete voltage level is the same. They bounded the information capacity (in bits-per-base) obtained in this nanopore channel as a function of b (the number of voltage levels) and k. They showed that the highest information capacity is given by $\min(\log(b), 2)$, where the lowest capacity is at least $\frac{\log(b)}{k}$ and at most 1 (if $b \leq 2^k$). They also suggested several coding schemes to correct errors in this model. The special case where the function f is defined as the Hamming weight was studied in [203], where the authors studied the capacity of three variations of the ISI channel, based on the values of k and δ. These channels include the cases in which $k \leq \delta$, the cases where k is multiple of δ and the special cases where $\delta = 2$ and $k \in \{3, 5, 7\}$. Furthermore, capacity results for $\delta < k < 2\delta$, as well as an upper bound for $k < \delta$ can be found in [204].

In [205], the authors used a similar model to the one described in Fig. 5. However, they only considered substitution errors. In their model, the substitution errors are interpreted as measurement errors, i.e., after performing the function f (the V channel). They showed that when $\delta = 1$, in order to correct one substitution error, it is required to encode at least $\log\log(n)$ symbols of redundancy. Furthermore, they also studied the case in which the nanopore channel produces two distinct noisy copies of the sequenced strands. In this case, they showed that the minimum redundancy required to correct one substitution error is $\log\log(n) - \log\binom{q}{2} - o(1)$, where q is the alphabet size. Finally, they suggested a single substitution-correcting code, which is optimal up to a constant in the case where $k \geq 3$.

Another more realistic method to model nanopore sequencing process was suggested in [206] and studied in [147], where the authors used a finite-state semi-Markov chain. In their model, the state space, denoted by Ω is defined as all the possible 4^k k-mers. Thus, the bases that pass through the pores induce a finite set of m states (k-mers), which are denoted by $\{S_\ell\}$, $1 \leq \ell \leq m$, where the ℓ-th state corresponds to the k-mer that was detected in the pore after the ℓ-th base from the strand that was transmitted through the pore. Thus, the edges of the Markov chain describe the four bases $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, and they connect the states as follows. For a given state, the next state is defined by an input base, which represents the base detected in the pore. In this model, the noise of the nanopore channel is given by two stages: the first is *sample duplications*, and the second describes the ISI, the measurement of the k-mer, and a possible noise that is added. The sample duplications correspond to the number of samples that are taken per k-mer. They are represented by a duplication channel that receives the ℓ-th state $S_\ell \in \Omega$, $1 \leq \ell \leq m$, and returns K_ℓ duplications of S_ℓ given as a sequence of output states. Thus, in total the duplication channel creates $T_m = \sum_{\ell=1}^{m} K_\ell$ states which are denoted by Z_t, $1 \leq t \leq T_m$. Based on the number of duplicated samples, the output states can be partitioned into m *segmentation point* or *jump times*. Therefore, the Markov property is preserved only per jump-time, and thus, the duplication channel is a semi-Markov chain [207] and can be analyzed accordingly, see e.g., [208]. Following the sample duplications, the ISI and the measurement of the k-mers are modeled similarly to the function f, where an additional layer of noise is added as *Additive Gaussian White Noise (AWGN)*. That is, given a k-mer (state) Z_t, which is the output of the duplication channel, we have that the corresponding voltage is given by $Y_t \triangleq f(Z_t) + N_t$, where $N_t \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian distributed noise. In [147], the authors suggested a coding scheme that employs constrained coding to reduce the errors of the channel and presented an efficient decoding for their scheme, where another scheme was suggested in [209].

One additional model of the nanopore channel was suggested in [210], where the authors focused on base insertion and base deletion errors caused by shift errors, and on the natural decay of the pores that affects the quality of the

measured signals. In their model, the bases are mapped into states as was done in [206], then the mapped states S_ℓ, $1 \leq \ell \leq m$ (which represent k-mers), pass through a deletion channel with probability p_d. The channel deletes each of the states with probability p_d. The set of indices of the deleted states is denoted by $\mathcal{D}$. Since some of the states are deleted, the Markov chain is called a *censored Markov chain [211]*. Lastly, the function f is used to quantize the remaining states into discrete voltage levels, but with the addition of a fading noise, modeling both the errors in the mapping function itself and the natural wear of the pores. Thus, f can be shown to be a discrete memoryless channel (DMC), and for any k-mer state S_ℓ, $f(S_\ell) \in [f(S_\ell) - \Delta, f(S_\ell) + \Delta]$, where Δ is the worst case deviation from the mean.

It should be further mentioned that current sequencing technologies suffer from relatively slow throughput as well as high costs. Hence, designing applicable DNA storage solutions requires a significant reduction in the quantity of DNA oligos that should be sampled and read to retrieve the information. This quantity is also referred to by the *coverage depth*, defined by the ratio between the number of sequenced reads and the number of synthesized oligos [6]. Optimizing the coverage depth can improve the latency of any existing DNA storage system and reduce its costs. Recently, in [212], the *DNA coverage depth problem* was proposed, where the goal is to minimize the coverage depth while maintaining system reliability. To do so, the authors studied the expected required coverage depth as a function of the DNA storage channel, the error-correcting code, and the reconstruction algorithm in order to retrieve the information successfully with high probability. They modeled the problem as follows. They assumed the information is given by k information strands and can be encoded into n encoded strands. The noise and the reconstruction algorithm were modeled together by a parameter $t > 0$ that defines the required number of reads of a specific strand to retrieve it. Lastly, the probability to obtain each of the n strands from sequencing was defined as a vector of probabilities $\boldsymbol{p} \triangleq (p_1, \ldots, p_n)$, $\sum_{i=1}^{n} p_i = 1$, where p_i describes the probability to obtain the i-th encoded strand. In this way, they could define $\nu_t^{\boldsymbol{p}}(n,k)$ as a random variable of the number of samples to retrieve the encoded information using an (n,k) code. In fact, in the more simple cases where $k = n$ and where t the problem can be reduced to the *coupon collector's problem* and *double dixie cup problem* [213]–[215]. Hence, the authors showed that when aiming to reduce the expected coverage depth (i.e. the expectation of $\nu_t^{\boldsymbol{p}}(n,k)$), MDS codes are optimal and showed closed-form upper and lower bounds on $E[\nu_t^{\boldsymbol{p}}(n,k)]$. They also studied this problem in the random access setup, where the goal is to retrieve only a subset of the k information strands. More cases of these two problems were also studied in [216], [217].

Finally, we also mention coding schemes for other DNA sequencing platforms. For example, high-throughput shotgun sequencing platforms is another platform used for reading DNA sequences. To correct errors arising for such platforms, a coding design framework was introduced by Kiah et al. [218] and the associated fundamental limits were later studied in Ravi et al. [219]. For an in-depth tutorial of this coding framework and a survey of related results, we recommend the work of Milenkovic and Pan [25] to the interested reader. In the same survey [25], Milenkovic and Pan also reviewed other sequencing methods and generally classified them as a class of fundamental problems termed as *unique string reconstruction.*

REFERENCES

[1] D. Carmean, L. Ceze, G. Seelig, K. Stewart, K. Strauss, and M. Willsey, "DNA data storage and hybrid molecular–electronic computing," *Proceedings of the IEEE*, vol. 107, no. 1, pp. 63–72, 2018.

[2] R. P. Feynman, "There's plenty of room at the bottom—an invitation to enter a new field of physics," *Caltech Engineering and Science*, vol. 23, p. 5, 1960.

[3] J. Shendure and E. L. Aiden, "The expanding scope of DNA sequencing," *Nature Biotechnology*, vol. 30, no. 11, pp. 1084–1094, 2012.

[4] L. Anavy, I. Vaknin, O. Atar, R. Amit, and Z. Yakhini, "Data storage in DNA with fewer synthesis cycles using composite DNA letters," *Nature Biotechnology*, vol. 37, no. 10, pp. 1229–1236, 2019.

[5] 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.

[6] R. Heckel, G. Mikutis, and R. N. Grass, "A characterization of the DNA data storage channel," *Scientific Reports*, vol. 9, no. 1, p. 9663, 2019.

[7] O. Sabary, Y. Orlev, R. Shafir, L. Anavy, E. Yaakobi, and Z. Yakhini, "SOLQC: Synthetic oligo library quality control tool," *Bioinformatics*, vol. 37, no. 5, pp. 720–722, 2021.

[8] M. Blawat, K. Gaedke, I. Huetter, X.-M. Chen, B. Turczyk, S. Inverso, B. W. Pruitt, and G. M. Church, "Forward error correction for DNA data storage," *Procedia Computer Science*, vol. 80, pp. 1011–1022, 2016.

[9] S. Chandak, J. Neu, K. Tatwawadi, J. Mardia, B. Lau, M. Kubit, R. Hulett, P. Griffin, M. Wootters, T. Weissman *et al.*, "Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8822–8826.

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

[11] S. Kosuri and G. M. Church, "Large-scale de novo DNA synthesis: technologies and applications," *Nature Methods*, vol. 11, no. 5, pp. 499–507, 2014.

[12] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, P. Gopalan, B. Nguyen *et al.*, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 36, no. 3, pp. 242–248, 2018.

[13] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, and D. B. Jaffe, "Characterizing and measuring bias in sequence data," *Genome Biology*, vol. 14, pp. 1–20, 2013.

[14] A. K.-Y. Yim, A. C.-S. Yu, J.-W. Li, A. I.-C. Wong, J. F. Loo, K. M. Chan, S. Kong, K. Y. Yip, and T.-F. Chan, "The essential component in DNA-based information storage system: robust error-tolerating module," *Frontiers in Bioengineering and Biotechnology*, vol. 2, p. 49, 2014.

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

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

[17] 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.

[18] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "Toward a DNA-based archival storage system," *IEEE Micro*, vol. 37, no. 3, pp. 98–104, 2017.

[19] S. M. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Scientific Reports*, vol. 5, no. 1, p. 14138, 2015.

[20] Y. Choi, T. Ryu, A. C. Lee, H. Choi, H. Lee, J. Park, S.-H. Song, S. Kim, H. Kim, W. Park *et al.*, "High information capacity DNA-based data storage with augmented encoding characters using degenerate bases," *Scientific Reports*, vol. 9, no. 1, p. 6582, 2019.

[21] I. Preuss, M. Rosenberg, Z. Yakhini, and L. Anavy, "Efficient DNA-based data storage using shortmer combinatorial encoding," *Scientific Reports*, vol. 14, no. 1, p. 7731, 2024.

[22] D. Bar-Lev, I. Orr, O. Sabary, T. Etzion, and E. Yaakobi, "Deep DNA storage: Scalable and robust DNA storage via coding theory and deep learning," *arXiv preprint arXiv:2109.00031*, 2021.

[23] C. Pan, S. K. Tabatabaei, S. H. Tabatabaei Yazdi, A. G. Hernandez, C. M. Schroeder, and O. Milenkovic, "Rewritable two-dimensional DNA-based data storage with machine learning reconstruction," *Nature Communications*, vol. 13, no. 1, p. 2984, 2022.

[24] D. Landsman and K. Strauss, "The DNA data storage model," *Computer*, vol. 56, no. 7, pp. 78–85, 2023.

[25] O. Milenkovic and C. Pan, "DNA-based data storage systems: A review of implementations and code constructions," *IEEE Transactions on Communications*, 2024.

[26] L. Xiang, Q. Liu, S. Chen, K. Yan, W. Wu, and K. Yang, "A tutorial on coding methods for DNA-based molecular communications and storage," *IEEE Internet of Things Journal*, 2023.

[27] T. Heinis, R. Sokolovskii, and J. J. Alnasir, "Survey of information encoding techniques for dna," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–30, 2023.

[28] V. Levenshtein, "Efficient reconstruction of sequences," *IEEE Transactions on Information Theory*, vol. 47, no. 1, pp. 2–22, 2001.

[29] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.

[30] 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.

[31] F. Sala, R. Gabrys, C. Schoeny, and L. Dolecek, "Exact reconstruction from insertions in synchronization codes," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2428–2445, 2017.

[32] R. Gabrys and E. Yaakobi, "Sequence reconstruction over the deletion channel," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 2924–2931, 2018.

[33] V. L. P. Pham, K. Goyal, and H. M. Kiah, "Sequence reconstruction problem for deletion channels: A complete asymptotic solution," *arXiv preprint arXiv:2111.04255*, 2021.

[34] K. Cai, H. M. Kiah, T. T. Nguyen, and E. Yaakobi, "Coding for sequence reconstruction for single edits," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 66–79, 2021.

[35] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes*. Elsevier, 1977, vol. 16.

[36] J. Chrisnata, H. M. Kiah *et al.*, "Deletion correcting codes for efficient DNA synthesis," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 352–357.

[37] Y. M. Chee, H. M. Kiah, A. Vardy, E. Yaakobi *et al.*, "Coding for racetrack memories," *IEEE Transactions on Information Theory*, vol. 64, no. 11, pp. 7094–7112, 2018.

[38] Y. Sun and G. Ge, "Correcting two-deletion with a constant number of reads," *IEEE Transactions on Information Theory*, 2022.

[39] J. Chrisnata, H. M. Kiah, and E. Yaakobi, "Correcting deletions with multiple reads," *IEEE Transactions on Information Theory*, vol. 68, no. 11, pp. 7141–7158, 2022.

[40] V. Guruswami and J. Håstad, "Explicit two-deletion codes with redundancy matching the existential bound," *IEEE Transactions on Information Theory*, vol. 67, no. 10, pp. 6384–6394, 2021.

[41] Z. Ye and O. Elishco, "Reconstruction of a single string from a part of its composition multiset," *IEEE Transactions on Information Theory*, 2023.

[42] M. Abu-Sini and E. Yaakobi, "On levenshtein's reconstruction problem under insertions, deletions, and substitutions," *IEEE Transactions on Information Theory*, vol. 67, no. 11, pp. 7132–7158, 2021.

[43] E. Yaakobi and J. Bruck, "On the uncertainty of information retrieval in associative memories," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2155–2165, 2018.

[44] V. Junnila, T. Laihonen, and T. Lehtilä, "Levenshtein's reconstruction problem with different error patterns," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 1300–1305.

[45] R. Wu and X. Zhang, "Balanced reconstruction codes for single edits," *Designs, Codes and Cryptography*, pp. 1–19, 2024.

[46] Y. Sun, Y. Xi, and G. Ge, "Sequence reconstruction under single-burst-insertion/deletion/edit channel," *IEEE Transactions on Information Theory*, 2023.

[47] V. Bhardwaj, P. A. Pevzner, C. Rashtchian, and Y. Safonova, "Trace reconstruction problems in computational biology," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3295–3314, 2020.

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

[49] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, and S. Yekhanin, "Trellis bma: Coded trace reconstruction on ids channels for DNA storage," in *IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2453–2458.

[50] S. Lin and D. J. Costello Jr., *Error control coding - fundamentals and applications*. Prentice Hall, 1983.

[51] T. Richardson and R. Urbanke, *Modern Coding Theory*. USA: Cambridge University Press, 2008.

[52] R. Roth, *Introduction to Coding Theory*. Cambridge University Press, 2006.

[53] V. Levenshtein, "Asymptotically optimum binary code with correction for losses of one or two adjacent bits," *Problemy Kibernetiki*, vol. 19, pp. 293–298, 1967.

[54] R. R. Varshamov and G. Tenenholtz, "A code for correcting a single asymmetric error," *Automatica i Telemekhanika*, vol. 26, no. 2, pp. 288–292, 1965.

[55] N. J. Sloane, "On single-deletion-correcting codes," *Codes and Designs*, vol. 10, pp. 273–291, 2000.

[56] J. Sima and J. Bruck, "Optimal k-deletion correcting codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 847–851.

[57] J. Sima, R. Gabrys, and J. Bruck, "Optimal systematic t-deletion correcting codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 769–774.

[58] R. Heckel, I. Shomorony, K. Ramchandran, and N. David, "Fundamental limits of dna storage systems," in *IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 3130–3134.

[59] I. Shomorony, R. Heckel *et al.*, "Information-theoretic foundations of dna data storage," *Foundations and Trends® in Communications and Information Theory*, vol. 19, no. 1, pp. 1–106, 2022.

[60] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding over sets for dna storage," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2331–2351, 2019.

[61] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors," 1961.

[62] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.

[63] F. Sellers, "Bit loss and gain correction code," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 35–38, 1962.

[64] K. Zigangirov, "Sequential decoding for a binary channel with drop-outs and insertions," *Problemy Peredachi Informatsii*, vol. 5, no. 2, pp. 23–30, 1969.

[65] A. Fazeli, A. Vardy, and E. Yaakobi, "Generalized sphere packing bound," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2313–2334, 2015.

[66] R. Gabrys and F. Sala, "Codes correcting two deletions," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 965–974, Feb 2019.

[67] J. Sima, N. Raviv, and J. Bruck, "Two deletion correcting codes from indicator vectors," *IEEE Transactions on Information Theory*, pp. 1–1, 2019.

[68] J. Brakensiek, V. Guruswami, and S. Zbarsky, "Efficient low-redundancy codes for correcting multiple deletions," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3403–3410, 2017.

[69] Y. Li and F. Farnoud, "Linial's algorithm and systematic deletion-correcting codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 2703–2707.

[70] W. H. Press, J. A. Hawkins, S. K. Jones, J. M. Schaub, and I. J. Finkelstein, "HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints," *Procceddings of National Academy of Sciences*, 2020.

[71] Z. Yan, C. Liang, and H. Wu, "A segmented-edit error-correcting code with re-synchronization function for DNA-based storage systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, 2023.

[72] S. K. Hanna and S. El Rouayheb, "Guess & check codes for deletions and synchronization," in *IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2693–2697.

[73] I. Tal, H. D. Pfister, A. Fazeli, and A. Vardy, "Polar codes for the deletion channel: Weak and strong polarization," *IEEE Transactions on Information Theory*, vol. 68, no. 4, pp. 2239–2265, 2021.

[74] M. Welzel, P. M. Schwarz, H. F. Löchel, T. Kabdullayeva, S. Clemens, A. Becker, B. Freisleben, and D. Heider, "DNA-Aeon provides flexible

arithmetic coding for constraint adherence and error correction in DNA storage," *Nature Communications*, vol. 14, no. 1, pp. 1–10, 2023.
[75] J. Jeong, H. Park, H.-Y. Kwak, J.-S. No, H. Jeon, J. W. Lee, and J.-W. Kim, "Iterative soft decoding algorithm for dna storage using quality score and redecoding," *IEEE Transactions on NanoBioscience*, pp. 1–1, 2023.
[76] A. Banerjee, A. Lenz, and A. Wachter-Zeh, "Sequential decoding of convolutional codes for synchronization errors," in *IEEE Information Theory Workshop (ITW)*, 2022, pp. 630–635.
[77] B. Bukh, V. Guruswami, and J. Håstad, "An improved bound on the fraction of correctable deletions," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 93–103, 2016.
[78] V. Guruswami, X. He, and R. Li, "The zero-rate threshold for adversarial bit-deletions is less than 1/2," *IEEE Transactions on Information Theory*, vol. 69, no. 4, pp. 2218–2239, 2022.
[79] V. Guruswami and R. Li, "Efficiently decodable insertion/deletion codes for high-noise and high-rate regimes," in *IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 620–624.
[80] V. Guruswami and C. Wang, "Deletion codes in the high-noise and high-rate regimes," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 1961–1970, 2017.
[81] B. Haeupler and A. Shahrasbi, "Synchronization strings: Codes for insertions and deletions approaching the singleton bound," *Journal of the ACM*, vol. 68, no. 5, 2021.
[82] S. Liu and C. Xing, "Bounds and constructions for insertion and deletion codes," *IEEE Transactions on Information Theory*, vol. 69, no. 2, pp. 928–940, 2023.
[83] L. J. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2552–2557, 1999.
[84] K. Cheng, Z. Jin, X. Li, and K. Wu, "Deterministic document exchange protocols, and almost optimal binary codes for edit errors," in *IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, 2018, pp. 200–211.
[85] B. Haeupler, "Optimal document exchange and new codes for insertions and deletions," in *IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 2019, pp. 334–347.
[86] K. A. Abdel-Ghaffar, H. C. Ferreira, and L. Cheng, "On linear and cyclic codes for correcting deletions," in *IEEE International Symposium on Information Theory*, 2007, pp. 851–855.
[87] K. Cheng, V. Guruswami, B. Haeupler, and X. Li, "Efficient linear and affine codes for correcting insertions/deletions," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2021.
[88] K. Cheng, Z. Jin, X. Li, Z. Wei, and Y. Zheng, "Linear insertion deletion codes in the high-noise and high-rate regimes," *arXiv preprint arXiv:2303.17370*, 2023.
[89] R. Con, A. Shpilka, and I. Tamo, "Explicit and efficient constructions of linear codes against adversarial insertions and deletions," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6516–6526, 2022.
[90] C. Schoeny, A. Wachter-Zeh, R. Gabrys, and E. Yaakobi, "Codes correcting a burst of deletions or insertions," *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 1971–1985, 2017.
[91] L. Welter, R. Bitar, A. Wachter-Zeh, and E. Yaakobi, "Multiple Criss-Cross Insertion and Deletion Correcting Codes," *IEEE Transactions on Information Theory*, vol. 68, no. 6, pp. 3767–3779, 2022.
[92] A. Wachter-Zeh, "List Decoding of Insertions and Deletions," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6297–6304, 2018.
[93] V. I. Levenshtein, "Elements of coding theory," *Diskretnaya matematika i matematicheskie voprosy kibernetiki*, pp. 207–305, 1974.
[94] D. S. Hirschberg and M. Regnier, "Tight bounds on the number of string subsequences," *Journal of Discrete Algorithms*, vol. 1, no. 1, pp. 123–132, 2000.
[95] Y. Liron and M. Langberg, "A characterization of the number of subsequences obtained via the deletion channel," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2300–2312, 2015.
[96] D. Bar-Lev, T. Etzion, and E. Yaakobi, "On the size of balls and anticodes of small diameter under the fixed-length levenshtein metric," *IEEE Transactions on Information Theory*, vol. 69, no. 4, pp. 2324–2340, 2023.
[97] F. Sala and L. Dolecek, "Counting sequences obtained from the synchronization channel," in *IEEE International Symposium on Information Theory*, 2013, pp. 2925–2929.
[98] G. Wang and Q. Wang, "On the size distribution of the fixed-length levenshtein balls with radius one," *Designs, Codes and Cryptography*, 2024. [Online]. Available: https://doi.org/10.1007/s10623-024-01382-1
[99] K. Yasunaga, "Improved asymptotic bounds for codes correcting insertions and deletions," *arXiv preprint arXiv:2107.01785*, 2021.
[100] A. A. Kulkarni and N. Kiyavash, "Nonasymptotic upper bounds for deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 5115–5130, Aug. 2013.
[101] L. Tolhuizen, "The generalized Gilbert-Varshamov bound is implied by Turán's theorem," *IEEE Transactions on Information Theory*, vol. 43, no. 5, pp. 1605–1606, Sep. 1997.
[102] V. I. Levenshtein, "Bounds for deletion/insertion correcting codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2002, p. 370.
[103] N. Alon, G. Bourla, B. Graham, X. He, and N. Kravitz, "Logarithmically larger deletion codes of all distances," *IEEE Transactions on Information Theory*, 2023.
[104] T. Jiang and A. Vardy, "Asymptotic improvement of the gilbert-varshamov bound on the size of binary codes," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1655–1664, 2004.
[105] M. Cheraghchi and J. Ribeiro, "An overview of capacity results for synchronization channels," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3207–3232, 2020.
[106] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels," *Probability Surveys*, vol. 6, pp. 1–33, 2009.
[107] A. Kalai, M. Mitzenmacher, and M. Sudan, "Tight asymptotic bounds for the deletion channel with small deletion probabilities," in *IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2010, pp. 997–1001.
[108] M. Mitzenmacher and E. Drinea, "A simple lower bound for the capacity of the deletion channel," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4657–4660, 2006.
[109] A. Kirsch and E. Drinea, "Directly lower bounding the information capacity for channels with iid deletions and duplications," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 86–102, 2009.
[110] D. Fertonani and T. M. Duman, "Novel bounds on the capacity of the binary deletion channel," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2753–2765, 2010.
[111] M. Dalai, "A new bound on the capacity of the binary deletion channel with high deletion probabilities," in *IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 499–502.
[112] M. Rahmati and T. M. Duman, "Upper bounds on the capacity of deletion channels using channel fragmentation," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 146–156, 2014.
[113] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 14–20, 1972.
[114] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, 1972.
[115] I. Rubinstein and R. Con, "Improved upper and lower bounds on the capacity of the binary deletion channel," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 927–932.
[116] M. Cheraghchi, "Capacity upper bounds for deletion-type channels," *Journal of the ACM (JACM)*, vol. 66, no. 2, pp. 1–79, 2019.
[117] R. Con and A. Shpilka, "Improved constructions of coding schemes for the binary deletion channel and the poisson repeat channel," *IEEE Transactions on Information Theory*, vol. 68, no. 5, pp. 2920–2940, 2022.
[118] I. Rubinstein, "Explicit and efficient construction of (nearly) optimal rate codes for binary deletion channel and the poisson repeat channel," in *International Colloquium on Automata, Languages and Programming*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:240353710
[119] H. Wei and M. Schwartz, "Improved coding over sets for dna-based data storage," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 118–129, 2021.
[120] M. Kovačević and V. Y. Tan, "Codes in the space of multisets—coding for permutation channels with impairments," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5156–5169, 2018.
[121] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2793–2807, 2021.
[122] ——, "Robust indexing for the sliced channel: Almost optimal codes for substitutions and deletions," *arXiv preprint arXiv:2308.07793*, 2023.
[123] W. Song, K. Cai, and K. A. S. Immink, "Sequence-subset distance and coding for error control in DNA-based data storage," *IEEE Transactions on Information Theory*, vol. 66, no. 10, pp. 6048–6065, 2020.

[124] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Anchor-based correction of substitutions in indexed sets," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 757–761.

[125] C. Rashtchian, K. Makarychev, M. Racz, S. Ang, D. Jevdjic, S. Yekhanin, L. Ceze, and K. Strauss, "Clustering billions of reads for dna data storage," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[126] J. Chrisnata, H. M. Kiah, A. Vardy, and E. Yaakobi, "Bee identification problem for dna strands," *IEEE Journal on Selected Areas in Information Theory*, 2023.

[127] A. Tandon, V. Y. Tan, and L. R. Varshney, "The bee-identification problem: Bounds on the error exponent," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 7405–7416, 2019.

[128] H. M. Kiah, A. Vardy, and H. Yaoz, "Efficient algorithms for the bee-identification problem," *IEEE Journal on Selected Areas in Information Theory*, 2023.

[129] S. Singhvi, A. Boruchovsky, H. M. Kiah, and E. Yaakobi, "Data-driven bee identification for DNA strands," *arXiv preprint arXiv:2305.04597*, 2023.

[130] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, Oct. 1948.

[131] K. Schouhamer Immink, *Codes for Mass Data Storage Systems*, 11 2004.

[132] B. H. Marcus, R. M. Roth, and P. H. Siegel, *An Introduction to Coding for Constrained Systems*. Elsevier Press, 2001. [Online]. Available: https://personal.math.ubc.ca/~marcus/Handbook/

[133] P. Siegel and J. Wolf, "Modulation and coding for information storage," *IEEE Communications Magazine*, vol. 29, no. 12, pp. 68–86, 1991.

[134] B. Marcus, P. Siegel, and J. Wolf, "Finite-state modulation codes for data storage," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 1, pp. 5–37, 1992.

[135] K. Schouhamer Immink, P. Siegel, and J. Wolf, "Codes for digital recorders," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2260–2299, 1998.

[136] K. A. S. Immink, "Innovation in constrained codes," *IEEE Communications Magazine*, vol. 60, no. 10, pp. 20–24, 2022.

[137] A. M. Patel, "Zero-modulation encoding in magnetic recording," *IBM Journal of Research and Development*, vol. 19, no. 4, pp. 366–378, 1975.

[138] A. Hareedy, S. Zheng, P. Siegel, and R. Calderbank, "Efficient constrained codes that enable page separation in modern flash memories," *IEEE Transactions on Communications*, vol. 71, no. 12, pp. 6834–6848, 2023.

[139] J. J. Schwartz, C. Lee, and J. Shendure, "Accurate gene synthesis with tag-directed retrieval of sequence-verified DNA molecules," *Nature Methods*, vol. 9, no. 9, pp. 913–915, 2012.

[140] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, and D. B. Jaffe, "Characterizing and measuring bias in sequence data," *Genome Biology*, vol. 14, no. 5, p. R51, May 2013.

[141] P. Yakovchuk, E. Protozanova, and M. D. Frank-Kamenetskii, "Base-stacking and base-pairing contributions into thermal stability of the DNA double helix," *Nucleic Acids Research*, vol. 34, no. 2, pp. 564–574, Jan. 2006.

[142] K. A. Schouhamer Immink and K. Cai, "Design of capacity-approaching constrained codes for dna-based storage systems," *IEEE Communications Letters*, vol. 22, no. 2, pp. 224–227, 2018.

[143] W. Song, K. Cai, M. Zhang, and C. Yuen, "Codes with run-length and gc-content constraints for DNA-based data storage," *IEEE Communications Letters*, vol. 22, no. 10, pp. 2004–2007, 2018.

[144] Y. Wang, M. Noor-A-Rahim, E. Gunawan, Y. L. Guan, and C. L. Poh, "Construction of bio-constrained code for DNA data storage," *IEEE Communications Letters*, vol. 23, no. 6, pp. 963–966, 2019.

[145] K. A. Schouhamer Immink and K. Cai, "Efficient balanced and maximum homopolymer-run restricted block codes for dna-based data storage," *IEEE Communications Letters*, vol. 23, no. 10, pp. 1676–1679, 2019.

[146] K. A. S. Immink and K. Cai, "Properties and constructions of constrained codes for dna-based data storage," *IEEE Access*, vol. 8, pp. 49 523–49 531, 2020.

[147] B. McBain, E. Viterbo, and J. Saunderson, "Homophonic coding for the noisy nanopore channel with constrained markov sources," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 376–381.

[148] K. Fan, H. Wu, and Z. Yan, "Constrained channel capacity for dna-based data storage systems," *IEEE Communications Letters*, vol. 27, no. 1, pp. 70–74, 2023.

[149] D. Limbachiya, M. K. Gupta, and V. Aggarwal, "Family of constrained codes for archival DNA data storage," *IEEE Communications Letters*, vol. 22, no. 10, pp. 1972–1975, 2018.

[150] J. H. Weber, J. A. M. de Groot, and C. J. van Leeuwen, "On single-error-detecting codes for DNA-based data storage," *IEEE Communications Letters*, vol. 25, no. 1, pp. 41–44, 2021.

[151] L. Deng, Y. Wang, M. Noor-A-Rahim, Y. L. Guan, Z. Shi, E. Gunawan, and C. L. Poh, "Optimized code design for constrained DNA data storage with asymmetric errors," *IEEE Access*, vol. 7, pp. 84 107–84 121, 2019.

[152] K. Cai, H. M. Kiah, M. Motani, and T. T. Nguyen, "Coding for segmented edits with local weight constraints," in *IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1694–1699.

[153] K. Cai, X. He, H. M. Kiah, and T. Thanh Nguyen, "Efficient constrained encoders correcting a single nucleotide edit in DNA storage," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8827–8830.

[154] K. Cai, Y. M. Chee, R. Gabrys, H. M. Kiah, and T. T. Nguyen, "Correcting a single indel/edit for dna-based data storage: Linear-time encoders and order-optimality," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3438–3451, 2021.

[155] T. T. Nguyen, K. Cai, K. A. Schouhamer Immink, and H. Mao Kiah, "Constrained coding with error control for DNA-based data storage," in *IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 694–699.

[156] Y. Liu, X. He, and X. Tang, "Capacity-achieving constrained codes with gc-content and runlength limits for DNA storage," in *IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 198–203.

[157] S.-J. Park, Y. Lee, and J.-S. No, "Iterative coding scheme satisfying gc balance and run-length constraints for DNA storage with robustness to error propagation," *Journal of Communications and Networks*, vol. 24, no. 3, pp. 283–291, 2022.

[158] S.-J. Park, H. Park, H.-Y. Kwak, and J.-S. No, "Bic codes: Bit insertion-based constrained codes with error correction for DNA storage," *IEEE Transactions on Emerging Topics in Computing*, vol. 11, no. 3, pp. 764–777, 2023.

[159] F. Weindel, A. L. Gimpel, R. N. Grass, and R. Heckel, "Embracing errors is more effective than avoiding them through constrained coding for DNA data storage," in *59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2023, pp. 1–8.

[160] O. Milenkovic and N. Kashyap, "DNA codes that avoid secondary structures," in *IEEE International Symposium on Information Theory (ISIT)*, 2005, pp. 288–292.

[161] ——, "On the design of codes for DNA computing," in *Coding and Cryptography*, Ø. Ytrehus, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 100–119.

[162] K. G. Benerjee and A. Banerjee, "On homopolymers and secondary structures avoiding, reversible, reversible-complement and gc-balanced DNA codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 204–209.

[163] ——, "On secondary structure avoiding DNA codes with reversible and reversible-complement constraints," in *National Conference on Communications (NCC)*, 2023, pp. 1–6.

[164] ——, "On DNA codes with multiple constraints," *IEEE Communications Letters*, vol. 25, no. 2, pp. 365–368, 2021.

[165] K. G. Benerjee, S. Deb, and M. K. Gupta, "On conflict free DNA codes," *Cryptography Communications*, vol. 13, no. 1, p. 143–171, jan 2021. [Online]. Available: https://doi.org/10.1007/s12095-020-00459-7

[166] T. T. Nguyen, K. Cai, H. M. Kiah, D. Tu Dao, and K. A. Schouhamer Immink, "On the design of codes for DNA computing: Secondary structure avoidance codes," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 573–578.

[167] H. Chu, C. Wang, and Y. Zhang, "Improved constructions of secondary structure avoidance codes for DNA sequences," in *12th International Symposium on Topics in Coding (ISTC)*, 2023, pp. 1–5.

[168] M. Dimopoulou, E. G. S. Antonio, and M. Antonini, "A jpeg-based image coding solution for data storage on dna," in *29th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 786–790.

[169] X. Pic and M. Antonini, "A constrained shannon-fano entropy coder for image storage in synthetic dna," in *30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 1367–1371.

[170] X. Pic, E. G. S. Antonio, M. Dimopoulou, and M. Antonini, "Rotating labeling of entropy coders for synthetic dna data storage," in *24th International Conference on Digital Signal Processing (DSP)*, 2023, pp. 1–5.

[171] S. Biswas, T. Ghosh, and S. Nath, "Selective run-length constrained encoding scheme on extended nucleic acid memory," in *2022 IEEE VLSI Device Circuit and System (VLSI DCS)*, 2022, pp. 148–153.

[172] R. Gabrys and O. Milenkovic, "Unique reconstruction of coded sequences from multiset substring spectra," in *IEEE International Symposium on Information Theory (ISIT)*, 2018, pp. 2540–2544.

[173] O. Elishco, R. Gabrys, E. Yaakobi, and M. Médard, "Repeat-free codes," *IEEE Transactions on Information Theory*, vol. 67, no. 9, pp. 5749–5764, 2021.

[174] S. T. Yazdi, H. M. Kiah, R. Gabrys, and O. Milenkovic, "Mutually uncorrelated primers for dna-based data storage," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6283–6296, 2018.

[175] M. Levy and E. Yaakobi, "Mutually uncorrelated codes for dna storage," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3671–3691, 2018.

[176] Y. M. Chee, H. M. Kiah, and H. Wei, "Efficient and explicit balanced primer codes," *IEEE Transactions on Information Theory*, vol. 66, no. 9, pp. 5344–5357, 2020.

[177] B. Cao, X. Ii, X. Zhang, B. Wang, Q. Zhang, and X. Wei, "Designing uncorrelated address constrain for dna storage by dmvo algorithm," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 2, pp. 866–877, 2022.

[178] A. Lenz, Y. Liu, C. Rashtchian, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding for efficient DNA synthesis," in *IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 2885–2890.

[179] M. H. Caruthers, "The chemical synthesis of DNA/RNA: Our gift to science," *Journal of Biological Chemistry*, vol. 288, no. 2, pp. 1420–1427, 2013.

[180] L. Ceze, J. Nivala, and K. Strauss, "Molecular digital data storage using DNA," *Nature Reviews Genetics*, vol. 20, no. 8, pp. 456–466, Aug. 2019.

[181] P. L. Antkowiak, J. Lietard, M. Z. Darestani, M. M. Somoza, W. J. Stark, R. Heckel, and R. N. Grass, "Low cost DNA data storage using photolithographic synthesis and advanced information reconstruction and error correction," *Nature Communications*, vol. 11, no. 1, p. 5345, 2020.

[182] A. Khandekar, R. J. McEliece, and E. R. Rodemich, "The discrete noiseless channel revisited," in *Coding, Communications, and Broadcasting*. Badlock: Research Studies Series Ltd., UK, 2000, pp. 115–137. [Online]. Available: http://www.mceliece.caltech.edu/publications/PostFinal.ps

[183] J. B. Soriaga and P. H. Siegel, "On the design of finite-state shaping encoders for partial-response channels," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, 2006.

[184] A. Lenz, S. Melczer, C. Rashtchian, and P. H. Siegel, "Exact asymptotics for discrete noiseless channels," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 2494–2498.

[185] Y. Liu, P. Huang, A. W. Bergman, and P. H. Siegel, "Rate-constrained shaping codes for structured sources," *IEEE Transactions on Information Theory*, vol. 66, no. 8, pp. 5261–5281, 2020.

[186] Y. Liu, Y. Li, P. Huang, and P. H. Siegel, "Rate-constrained shaping codes for finite-state channels with cost," in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 1354–1359.

[187] O. Elishco and W. Huleihel, "Optimal reference for DNA synthesis," *IEEE Transactions on Information Theory*, vol. 69, no. 11, pp. 6941–6955, 2023.

[188] K. Makarychev, M. Z. Rácz, C. Rashtchian, and S. Yekhanin, "Batch optimization for DNA synthesis," *IEEE Transactions on Information Theory*, vol. 68, no. 11, pp. 7454–7470, 2022.

[189] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Communications*, vol. 10, no. 1, p. 2383, 2019.

[190] N. Roquet, S. P. Bhatia, S. A. Flickinger, S. Mihm, M. W. Norsworthy, D. Leake, and H. Park, "DNA-based data storage via combinatorial assembly," *bioRxiv*, pp. 2021–04, 2021.

[191] Y. Yan, N. Pinnamaneni, S. Chalapati, C. Crosbie, and R. Appuswamy, "Scaling logical density of DNA storage with enzymatically-ligated composite motifs," *Scientific Reports*, vol. 13, no. 1, p. 15978, 2023.

[192] W. Zhang, Z. Chen, and Z. Wang, "Limited-magnitude error correction for probability vectors in DNA storage," in *IEEE International Conference on Communications (ICC)*, 2022, pp. 3460–3465.

[193] F. Walter, O. Sabary, A. Wachter-Zeh, and E. Yaakobi, "Coding for composite DNA to correct substitutions, strand losses, and deletions," 2024.

[194] O. Sabary, I. Preuss, R. Gabrys, Z. Yakhini, L. Anavy, and E. Yaakobi, "Error-correcting codes for combinatorial DNA composite," 2024.

[195] A. Kobovich, E. Yaakobi, and N. Weinberger, "M-dab: An input-distribution optimization algorithm for composite dna storage by the multinomial channel," *ArXiv*, vol. abs/2309.17193, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263310855

[196] R. D. Wesel *et al.*, "Efficient binomial channel capacity computation with an application to molecular communication," in *ITA*. IEEE, 2018, pp. 1–5.

[197] I. Preuss, B. Galili, Z. Yakhini, and L. Anavy, "Sequencing coverage analysis for combinatorial dna-based storage systems," *bioRxiv*, pp. 2024–01, 2024.

[198] T. Cohen and E. Yaakobi, "Optimizing the decoding probability and coverage ratio of composite dna," 2024.

[199] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Coding for optimized writing rate in DNA storage," in *IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 711–716.

[200] R. Shafir, O. Sabary, L. Anavy, E. Yaakobi, and Z. Yakhini, "Sequence design and reconstruction under the repeat channel in enzymatic DNA synthesis," *IEEE Transactions on Communications*, vol. 72, no. 2, pp. 675–691, 2024.

[201] J. Lietard, A. Leger, Y. Erlich, N. Sadowski, W. Timp, and M. M. Somoza, "Chemical and photochemical error rates in light-directed synthesis of complex DNA libraries," *Nucleic Acids Research*, vol. 49, no. 12, pp. 6687–6701, 2021.

[202] R. Hulett, S. Chandak, and M. Wootters, "On coding for an abstracted nanopore channel for DNA storage," in *IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2465–2470.

[203] Y. M. Chee, A. Vardy, E. Yaakobi *et al.*, "Transverse-read-codes for domain wall memories," *IEEE Journal on Selected Areas in Information Theory*, vol. 4, pp. 784–793, 2023.

[204] O. Yerushalmi, T. Etzion, and E. Yaakobi, "The capacity of the weighted read channel," in *IEEE International Symposium on Information Theory (ISIT)*, 2024.

[205] A. Banerjee, Y. Yehezkeally, A. Wachter-Zeh, and E. Yaakobi, "Error-correcting codes for nanopore sequencing," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 364–369.

[206] B. McBain, E. Viterbo, and J. Saunderson, "Finite-state semi-markov channels for nanopore sequencing," in *IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 216–221.

[207] V. S. Barbu and N. Limnios, *Semi-Markov chains and hidden semi-Markov models toward applications: their use in reliability and DNA analysis*. Springer Science & Business Media, 2009, vol. 191.

[208] V. Girardin and N. Limnios, "Entropy rate and maximum entropy methods for countable semi-markov chains," 2004.

[209] A. Vidal, V. Wijekoon, and E. Viterbo, "Concatenated nanopore DNA codes," *IEEE Transactions on NanoBioscience*, 2024.

[210] W. Mao, S. N. Diggavi, and S. Kannan, "Models and information-theoretic bounds for nanopore sequencing," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3216–3236, 2018.

[211] J. F. C. Kingman, J. G. Kemeny, J. L. Snell, and A. W. Knapp, "Denumerable markov chains," 1969. [Online]. Available: https://api.semanticscholar.org/CorpusID:119064121

[212] D. Bar-Lev, O. Sabary, R. Gabrys, and E. Yaakobi, "Cover your bases: How to minimize the sequencing coverage in DNA storage systems," in *IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 370–375.

[213] P. Erdős and A. Rényi, "On a classical problem of probability theory," *Magyar Tud. Akad. Mat. Kutató Int. Közl*, vol. 6, no. 1, pp. 215–220.

[214] W. Feller, *An introduction to probability theory and its applications, Volume 2*. John Wiley & Sons, 1991, vol. 81.

[215] P. Flajolet, D. Gardy, and L. Thimonier, "Birthday paradox, coupon collectors, caching algorithms and self-organizing search," *Discrete Applied Mathematics*, vol. 39, no. 3, pp. 207–229, 1992.

[216] H. Abraham, R. Gabrys, and E. Yaakobi, "Covering all bases: The next inning in DNA sequencing efficiency," 2024.

[217] A. Gruica, D. Bar-Lev, A. Ravagnani, and E. Yaakobi, "Reducing coverage depth in DNA storage: A combinatorial perspective on random access efficiency," *IEEE International Symposium on Information Theory (ISIT)*, 2024.

[218] 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.

[219] A. N. Ravi, A. Vahid, and I. Shomorony, "Coded shotgun sequencing," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 1, pp. 147–159, 2022.

Omer Sabary (Student Member, IEEE) received the M.Sc. degree from the Computer Science Department, Technion—Israel Institute of Technology, in 2020. He is currently pursuing a Ph.D. degree in the Henry and Marilyn Taub Faculty of Computer Science, Technion — Israel Institute of Technology. His advisor is Prof. Eitan Yaakobi. His research interests include coding techniques and algorithms for DNA storage systems. He was a recipient of the VATAT Excellence Award in data mining in 2020 and 2023. He also received the Jacobs-Qualcomm Excellence Fellowship for PhD Students in 2023/24.

Han Mao Kiah (Senior Member, IEEE) received the Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. From 2014 to 2015, he was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana -Champaign. From 2015 to 2018, he was a Lecturer with the School of Physical and Mathematical Sciences (SPMS), NTU, where he is currently an Assistant Professor with SPMS. His research interests include DNA-based data storage, coding theory, enumerative combinatorics, and combinatorial design theory.

Paul H. Siegel (Life Fellow, IEEE) received the S.B. and Ph.D. degrees in Mathematics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1975 and 1979, respectively. He held a Chaim Weizmann Postdoctoral Fellowship with the Courant Institute, New York University, New York, NY, USA. He was with the IBM Research Division, San Jose, CA, USA, from 1980 to 1995. He joined the faculty at the University of California San Diego (UCSD), La Jolla, CA, USA, in 1995, where he is currently a Distinguished Professor of Electrical and Computer Engineering with the Jacobs School of Engineering. He is affiliated with the Center for Memory and Recording Research where he holds an Endowed Chair and served as Director from 2000 to 2011. His research interests include information theory, coding techniques, and machine learning, with applications to digital data storage and transmission. He is a Member of the National Academy of Engineering. He was a Member of the Board of Governors of the IEEE Information Theory Society from 1991 to 1996 and from 2009 to 2014. He was the 2015 Padovani Lecturer of the IEEE Information Theory Society. He was a co-recipient of the 1992 IEEE Information Theory Society Paper Award, the 1993 IEEE Communications Society Leonard G. Abraham PrizePaper Award, and the 2007 Best Paper Award in Signal Processing and Coding for Data Storage from the Data Storage Technical Committee of the IEEE Communications Society. He served as an Associate Editor of Coding Techniques of the IEEE Transactions on Information Theory from 1992 to 1995, and as the Editor-in-Chief from 2001 to 2004. He served as a Co-Guest Editor of the 1991 Special Issue on Coding for Storage Devices of the IEEE Transactions on Information Theory. He was also a Co-Guest Editor of the 2001 two-part issue on The Turbo Principle: From Theory to Practice and the 2016 issue on Recent Advances in Capacity Approaching Codes of the IEEE Journal on Selected Areas in Communications. He is currently the Lead Guest Editor of the Special Issue on Data Storage of the IEEE BITS the Information Theory Magazine.

Eitan Yaakobi (Senior Member, IEEE)is an Associate Professor at the Computer Science Department at the Technion — Israel Institute of Technology. He also holds a courtesy appointment in the Technion's Electrical and Computer Engineering (ECE) Department. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the department of Electrical Engineering at the California Institute of Technology and at the Center for Memory and Recording Research at the University of California, San Diego. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011. Between 2020 and 2023, he served as an Associate Editor for Coding snd Decoding for the IEEE TRANSACTIONS ON INFORMATION THEORY. Since 2016, he is affiliated with the Center for Memory and Recording Research at the University of California, San Diego, and between 2018–2022, he was affiliated with the Institute of Advanced Studies, Technical University of Munich, where he held a four-year Hans Fischer Fellowship, funded by the German Excellence Initiative and the EU 7th Framework Program. Between August 2023 and January 2024, he was a Visiting Associate Professor at the School of Physical and Mathematical Sciences at Nanyang Technological University. He is a recipient of several grants, including the ERC Consolidator Grant and the EIC Pathfinder Challenge.