

Finite Blocklength Performance Bound for the DNA Storage Channel

Issam Maarouf*, Gianluigi Liva†, Eirik Rosnes*, and Alexandre Graell i Amat‡

*Simula UiB, N-5006 Bergen, Norway

†Institute of Communications and Navigation, German Aerospace Center, 82234 Weßling, Germany

‡Department of Electrical Engineering, Chalmers University of Technology, SE-41296 Gothenburg, Sweden

Abstract—We present a finite blocklength performance bound for a DNA storage channel with insertions, deletions, and substitutions. The considered bound—the dependency testing (DT) bound, introduced by Polyanskiy *et al.* in 2010—, provides an upper bound on the achievable frame error probability and can be used to benchmark coding schemes in the practical short-to-medium blocklength regime. In particular, we consider a concatenated coding scheme where an inner synchronization code deals with insertions and deletions and the outer code corrects remaining (mostly substitution) errors. The bound depends on the inner synchronization code. Thus, it allows to guide its choice. We then consider low-density parity-check codes for the outer code, which we optimize based on extrinsic information transfer charts. Our optimized coding schemes achieve a normalized rate of 87% to 97% with respect to the DT bound for code lengths up to 2000 DNA symbols for a frame error probability of 10^{-3} and code rate $1/2$.

I. INTRODUCTION

Using deoxyribonucleic acid (DNA) as a medium to store data is seen as the next frontier of data storage, providing unprecedented durability and density. Several experiments have already demonstrated the viability of DNA-based data storage, see, e.g., [1], [2].

The DNA storage channel is impaired by insertions, deletions, and substitutions (IDSs) arising from the synthesis and sequencing of DNA sequences [3]. Hence, reliable storage of data in DNA requires the use of error-correcting codes. Designing a code that handles IDS errors jointly is, however, a daunting task. Davey and MacKay [4] proposed a clever solution to this problem by introducing a serially-concatenated coding scheme (for the binary IDS channel) in which the inner code, called *synchronization* code, deals with insertions and deletions, and the outer code (a low-density parity-check (LDPC) code in [4]) corrects remaining errors, mostly in the form of substitutions.

The literature on coding for DNA storage is abundant. Most works consider a very small number of deletions and/or insertions—i.e., an adversarial channel—and a single DNA strand. In DNA-based storage, however, errors occur probabilistically and can be substantial, and the synthesis and sequencing processes result in multiple (noisy) copies of the same DNA strand. The authors in [5] were the first to introduce decoding algorithms for coding schemes exploiting multiple reads of the DNA sequence. The work [5] was followed by [6].

The works [5] and [6] also provided achievable information rates, which give insight into the performance of coding schemes with very large blocklengths. However, current DNA

storage technology only supports the synthesis and sequencing of short-to-medium-length DNA strands, in the range of 100–2000 DNA symbols. Therefore, performance bounds for the finite blocklength regime would be more informative for the DNA channel. To the best of our knowledge, no finite blocklength performance bounds for the DNA storage channel (and IDS channels in general) exist in the literature.

In this paper, we provide a finite blocklength performance bound for a DNA storage channel with IDS errors. Particularly, we consider the dependency testing (DT) bound [7] based on the *random coding* principle, which gives an upper bound on the frame error probability achievable over the DNA storage channel. The bound is tailored to a concatenated coding scheme that uses an inner synchronization code and depends on the inner code. Hence, it can be used as a handy tool to optimize the inner synchronization code for the finite blocklength regime. Further, the bound provides a benchmark to compare coding schemes for DNA storage in the practical short-to-medium blocklength regime. We also consider the optimization of an outer LDPC code for a given inner code using extrinsic information transfer (EXIT) charts, and show that an optimized concatenated coding scheme achieves a normalized rate of 87% to 97% with respect to the DT bound for a frame error probability of 10^{-3} and code rate $1/2$, depending on the sequence length. These values are similar to those of state-of-the-art coding schemes for simpler memoryless channels (such as the Gaussian channel and the binary symmetric channel), highlighting that the scheme in [5] achieves excellent performance for the DNA storage channel in the short-to-medium blocklength regime.

II. SYSTEM MODEL

A. Channel Model

We consider the widely-used simplified channel model depicted in Fig. 1 [4], [8] for the DNA storage channel, where IDS errors are independent and identically distributed. Let $\mathbf{x} = (x_1, \dots, x_N)$, $x_i \in \Sigma_q = \{0, 1, \dots, q-1\}$,¹ be the information DNA sequence of length N to be transmitted over the channel. The sequence can be viewed as a queue of symbols, where each symbol x_i is successively transmitted over the channel. The received sequence $\mathbf{y} = (y_1, \dots, y_{N'})$, where N' may be different to N due to insertions and deletions, is generated state by state and is obtained as follows.

¹For the DNA storage channel, $q = 4$. However, we use q for the sake of generality.

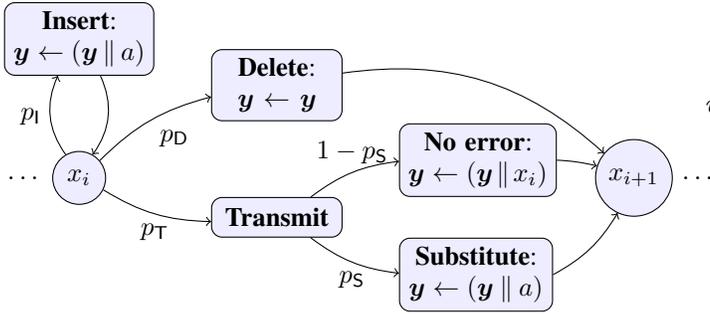


Fig. 1. State-based representation of the DNA storage channel with IDS errors.

Assume x_i is next in queue to be transmitted over the channel. The channel enters state x_i where three events may occur: i) an insertion event, with probability p_I , where a random symbol $a \in \Sigma_q$ is appended to \mathbf{y} instead of x_i . In this case, x_i remains in the queue and the channel returns to state x_i ; ii) a deletion event, with probability p_D , where symbol x_i is deleted from the queue. In this case, nothing is appended to \mathbf{y} , the next symbol x_{i+1} is enqueued, and the channel enters state x_{i+1} ; iii) a transmission event, with probability $p_T = 1 - p_I - p_D$, where x_i is transmitted. In this case, the symbol x_i is either received with no error with probability $1 - p_S$ or in error with probability p_S , in which case x_i is substituted by a random symbol $a \neq x_i$. In either case, the next symbol x_{i+1} is enqueued, and the channel enters state x_{i+1} . The process finishes when the last symbol x_N leaves the queue. The channel output is \mathbf{y} .

The difference $N - N'$ is referred to as the *drift* [4] at the end of the transmitted sequence. We can also define a drift for each symbol x_i to be transmitted, or each time instant i . Formally, the *symbol-level drift* d_i^{sym} , $0 \leq i < N$, is defined as the difference between the number of insertions and the number of deletions that occurred before symbol x_{i+1} is enqueued, while d_N^{sym} is defined as the number of insertions minus deletions that occurred after the last symbol x_N has been transmitted.

Finally, we model the multiple reads of a DNA sequence resulting from the synthesis and sequencing processes as transmitting the DNA sequence \mathbf{x} over M parallel and independent IDS channels, see Fig. 2, resulting in the received sequences $\mathbf{y}_1, \dots, \mathbf{y}_M$.

B. Coding Scheme

We consider a concatenated coding scheme with an inner synchronization code depicted in Fig. 2. First, the information sequence $\mathbf{u} = (u_1, \dots, u_K)$, $u_i \in \mathbb{F}_{q_0}$, is encoded by an $[[N_o, K]]_{q_0}$ outer code to produce a codeword $\mathbf{w} = (w_1, \dots, w_{N_o})$, $w_i \in \mathbb{F}_{q_0}$, where \mathbb{F}_{q_0} is a binary field extension with $q_0 = 2^k$. The codeword \mathbf{w} is then encoded by an inner synchronization code. Here, we consider block and convolutional codes for the inner code. We denote the block code by $[[n, k, t]]_q$, where n and k are the length and dimension of the code, respectively, and t represents the number of different codebooks that are used (see [5] for details). Furthermore, the convolutional code is denoted by $(n, k, m)_q$, where m is the number of memory elements. For simplicity, in the rest of the paper we will consider an inner convolutional code for notations and equations. We denote the codeword of the

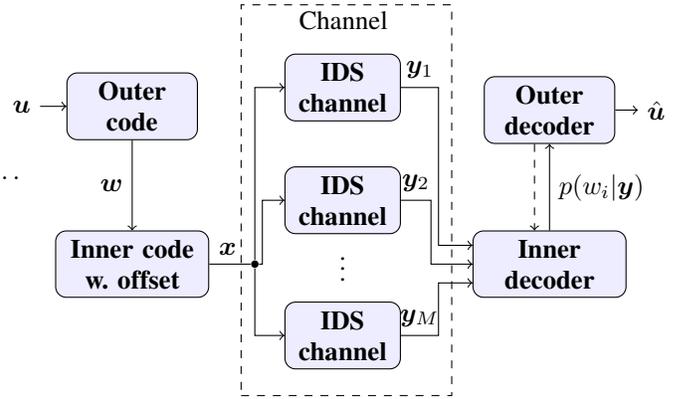


Fig. 2. Block diagram of the encoder and decoder for the DNA storage channel. The DNA storage channel is modeled as multiple reads of the DNA strand transmitted over parallel IDS channels: the channel depicted in Fig. 1 is fed M times with the DNA sequence \mathbf{x} . Here, $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$.

inner code by $\mathbf{v} = (v_1, \dots, v_N)$, $v_i \in \Sigma_q$, which is of length $N = (N_o + m)n$ due to termination of the convolutional code. Finally, a pseudo-random offset sequence is optionally added to \mathbf{v} before transmission for synchronization purposes [4], [9], resulting in the sequence $\mathbf{x} = (x_1, \dots, x_N)$. (A detailed explanation of the role of the random sequence in maintaining synchronization and aiding the decoding of the inner code is given in [5].) The DNA sequence \mathbf{x} is finally stored in the DNA medium.

The coding scheme rate is measured in bits per DNA symbol (i.e., per nucleotide) and is given by $R = R_o R_i = Kk/N$, where $R_o = K/N_o$ and $R_i = N_o k/N$ are the rates of the outer and inner code, respectively. As we will be only concerned with the drift at time instances that are multiples of n , we define the shorthand $d_i \triangleq d_{in}^{\text{sym}}$. Note that $d_0 = 0$ and $d_{N_o+m} = N' - N$, both known to the receiver.

To recover the information sequence \mathbf{u} , the inner decoder uses the (noisy) multiple reads $\mathbf{y}_1, \dots, \mathbf{y}_M$ of the DNA sequence \mathbf{x} to compute (approximate) a posteriori probabilities (APPs) for the symbols in \mathbf{w} . These APPs are then fed to the outer decoder, which decides on the decoded sequence $\hat{\mathbf{u}}$. Furthermore, we can also iterate between the inner and outer decoder, exchanging extrinsic information between them, which is referred to as *turbo decoding* in the literature.

III. BOUND ON THE FINITE BLOCKLENGTH PERFORMANCE

In this section, we provide an upper bound to the frame error probability, denoted by $P_f(e)$, achievable over the DNA storage channel in the finite blocklength regime. In particular, we consider the DT bound [7]. The bound we provide is tailored to concatenated coding schemes with an inner synchronization code and depends on the inner code. Hence, it can be used to guide its choice and serves as a benchmark to compare coding schemes.

The DT bound for the combination of the inner code and the DNA storage channel is given by

$$P_f(e) \leq \mathbb{E} \left[2^{-\left(i(\mathbf{w}; \mathbf{y}) - \log_2 \frac{q_0^{N_o} - 1}{2} \right)^+} \right], \quad (1)$$

where $(x)^+ \triangleq \max(x, 0)$, $\mathbb{E}[\cdot]$ denotes expectation, $\mathbf{y} = (\mathbf{y}_1, \dots, \mathbf{y}_M)$ for the multiple sequences case, and

$$i(\mathbf{w}; \mathbf{y}) \triangleq \log_2 \frac{p(\mathbf{y}|\mathbf{w})}{p(\mathbf{y})}$$

is the so-called *information density* with expected value equal to the mutual information between \mathbf{w} and \mathbf{y} .² The distribution of the information density $i(\mathbf{w}; \mathbf{y})$ is not known in closed form for the DNA storage channel. However, the right-hand-side of (1) can be accurately estimated using the Monte-Carlo approach proposed in [10], [11], which exploits concentration properties of Markov chains to estimate the mutual information between an input process \mathbf{w} and an output process \mathbf{y} via trellis-based simulations. We can then approximate (1) by

$$P_f(e) \lesssim \frac{1}{V} \sum_{(\mathbf{w}, \mathbf{y})} 2^{-(i(\mathbf{w}; \mathbf{y}) - (N_o \log_2 q_o - 1))^+}, \quad (2)$$

where V is the number of pairs (\mathbf{w}, \mathbf{y}) considered in the computation.

In the following, we show how to efficiently compute $i(\mathbf{w}; \mathbf{y})$ for fixed \mathbf{w} and \mathbf{y} . We stress that the values of $i(\mathbf{w}; \mathbf{y})$ and, hence, their distribution depends on the choice of the inner code. The information density can be written as

$$i(\mathbf{w}; \mathbf{y}) = -\log_2 p(\mathbf{w}) - \log_2 p(\mathbf{y}) + \log_2 p(\mathbf{w}, \mathbf{y}), \quad (3)$$

where the probabilities $p(\mathbf{w})$, $p(\mathbf{y})$, and $p(\mathbf{w}, \mathbf{y})$ can be computed using the forward recursion of the symbol-wise maximum a posteriori decoding algorithm on the trellis describing the combination of the inner code and the DNA storage channel [5] (hereafter in this paragraph referred to as simply the inner code for the sake of simplicity). For simplicity, we consider the case of a single sequence, i.e., $M = 1$. However, the approach below can be generalized to $M > 1$ straightforwardly. For $M = 1$, the APP of the outer code symbol w_i can be computed as $p(w_i|\mathbf{y}) = \frac{p(w_i, \mathbf{y})}{p(\mathbf{y})}$. The joint probability $p(w_i, \mathbf{y})$ can be computed by marginalizing the trellis states of the inner code that correspond to symbol w_i . Introducing the joint state variable $\sigma_i = (s_i, d_i)$, where s_i denotes the memory state variables of the convolutional code, we obtain

$$p(w_i, \mathbf{y}) = \sum_{(\sigma, \sigma'): w_i} p(\mathbf{y}, \sigma, \sigma'),$$

where σ and σ' denote realizations of the random variables σ_{i-1} and σ_i , respectively. The summation is over all the inner code memory states that correspond to information symbol w_i . Introducing a drift random variable retains the Markov property of the hidden Markov model (HMM) that was lost due to the insertions and deletions [4]. In this new HMM, a transition from time $i-1$ to time i corresponds to a transmission of a vector of symbols $\mathbf{x}_{(i-1)n+1}^{in}$, where $\mathbf{x}_a^b = (x_a, x_{a+1}, \dots, x_b)$. Further, when transitioning from state d_{i-1} to d_i , the HMM emits $n + d_i - d_{i-1}$ output symbols depending on both the previous and the new drift. As a result, using the Markov property of the underlying trellis, we can

²In order to distinguish between random variables and their realizations, \mathbf{w} and \mathbf{y} denote the random variables corresponding to \mathbf{w} and \mathbf{y} , respectively.

TABLE I
INNER SYNCHRONIZATION CODE SCHEME SELECTION

| Scheme | Inner code | Gen. polynomial | Alt. pattern | Rate |
|--------|---|-----------------------|--------------|------|
| CC | (1, 1, 2) ₄ Conv. code with RS | [5, 7] _{oct} | - | 0.98 |
| WM | [4, 4, 1] ₄ Watermark code | - | - | 1.0 |
| TVC-1 | [4, 4, 4] ₄ TVC | - | Random* | 1.0 |
| TVC-2 | [4, 4, 4] ₄ TVC with RS | - | CB1 to CB4* | 1.0 |

*The alternating pattern of the TVC-1 scheme is done by choosing randomly the 4 codebooks, denoted by CB1-CB4, from [5, Tab. I] and avoiding consecutive codebooks. For the TVC-2 scheme, it is simply done by repeating CB1 to CB4 in a round Robin fashion. RS is shorthand for random sequence.

factor the joint probability $p(\mathbf{y}, \sigma, \sigma')$ into three terms as

$$p(\mathbf{y}, \sigma, \sigma') = p(\mathbf{y}_1^{(i-1)n+d}, \sigma) p(\mathbf{y}_{(i-1)n+d+1}^{in+d}, \sigma' | \sigma) p(\mathbf{y}_{in+d'+1}^{N'} | \sigma').$$

Abbreviating the above terms by $\alpha_{i-1}(\sigma)$, $\gamma_i(\sigma, \sigma')$, and $\beta_i(\sigma')$ in order of appearance, one can deduce the forward and backward recursions

$$\alpha_i(\sigma') = \sum_{\sigma} \alpha_{i-1}(\sigma) \gamma_i(\sigma, \sigma'), \quad (4)$$

$$\beta_{i-1}(\sigma) = \sum_{\sigma'} \beta_i(\sigma') \gamma_i(\sigma, \sigma'), \quad (5)$$

where $\gamma_i(\sigma, \sigma') = p(w_i) p(\mathbf{y}_{(i-1)n+d+1}^{in+d}, d' | d, s, s')$ can be efficiently computed using a lattice implementation [12].

Now, $\log_2 p(\mathbf{y})$ and $\log_2 p(\mathbf{w}, \mathbf{y})$ in (3) can be computed based on the forward recursion in (4). In particular,

$$p(\mathbf{y}) = \sum_{\sigma} p(\mathbf{y}_1^{(N_o+m)n+d}, \sigma) \stackrel{(a)}{=} \sum_{\sigma} \alpha_{N_o+m}(\sigma),$$

where (a) follows since $\alpha_i(\sigma) = p(\mathbf{y}_1^{in+d}, \sigma)$. The quantity $\log_2 p(\mathbf{w}, \mathbf{y})$ can be computed in a similar manner by restricting the summation in (4) to be over all states σ with an outgoing edge to σ' labeled with the input sequence symbol w_i at time i . Since we consider an input sequence of independent and uniformly distributed symbols, the first term $\log_2 p(\mathbf{w})$ in (3) is equal to $N_o \log_2 q_o$. Note that the backward recursion in (5) is not required for the computation of the information density, but only for the calculation of the APP $p(w_i|\mathbf{y})$ in decoding.

To obtain an estimate of the right-hand-side of (1), we randomly generate \mathbf{w} and encode it using the inner code to obtain \mathbf{x} . Then, we pass \mathbf{x} through the DNA storage channel to obtain \mathbf{y} . For each tuple (\mathbf{w}, \mathbf{y}) , we evaluate $i(\mathbf{w}; \mathbf{y})$ using the defined recursions and the corresponding summand in (2). We repeat this procedure V times, each time creating a new random \mathbf{w} , and average over the outcomes according to (2).

IV. CONCATENATED CODING SCHEME DESIGN

A. Inner Code

We consider four different inner codes: the watermark code introduced in [4], a convolutional code [13], and two time-varying codes (TVCs) recently introduced in [5]. The watermark code is an $[n, k, 1]_q$ block code to which a random sequence is added, which can also be thought of as a TVC with $t = 1$. We will use the TVCs from [5, Tab. I] with $t = 4$ and a minimum Levenshtein distance of 4. The inner coding schemes that we consider are summarized in Table I.

B. Outer Code

We use protograph-based LDPC codes for the outer code. Formally, the protograph of an LDPC code is a multi-edge-type graph with n_p variable-node (VN) types and r_p check-node (CN) types. A protograph can be represented by a base matrix

$$\mathbf{B} = \begin{pmatrix} b_{0,0} & b_{0,1} & \dots & b_{0,n_p-1} \\ b_{1,0} & b_{1,1} & \dots & b_{1,n_p-1} \\ \vdots & \vdots & \dots & \vdots \\ b_{r_p-1,0} & b_{r_p-1,1} & \dots & b_{r_p-1,n_p-1} \end{pmatrix},$$

where $b_{i,j}$ is an integer representing the number of edge connections between a type- i VN and a type- j CN. A parity-check matrix \mathbf{H} of an LDPC code can be constructed from a protograph by lifting the base matrix \mathbf{B} . Lifting is the procedure of replacing each nonzero (zero) $b_{i,j}$ with a $Q_p \times Q_p$ permutation (zero) matrix with row and column weight equal to $b_{i,j}$. The LDPC code resulting from the lifting procedure has length $Q_p n_p$ and dimension at least $Q_p(n_p - r_p)$. To construct a nonbinary code from the lifted matrix, we randomly assign nonzero entries from \mathbb{F}_{q_o} to the edges of the corresponding Tanner graph.

In this work, we optimize the protograph \mathbf{B} using EXIT charts, extended to the DNA storage channel. Particularly, we optimize the protograph for the case of iterations between the decoder of the LDPC code and the decoder of the combination of the inner code and the DNA storage channel. We limit our search to protographs of dimensions 3×6 (larger protographs may lead to better performance). The choice of the protograph is done by considering both the iterative decoding threshold from the EXIT chart, for $p_1 = p_D$ and $p_5 = 0$, and the frame error rate (FER) performance of the corresponding code ensemble (i.e., by using random permutation matrices for the protograph liftings). More precisely, we sort the protographs from highest to lowest decoding threshold, and then we pick the first protograph (starting from the top of the list) that shows no sign of an error floor above a FER of 10^{-3} . The best protographs from this list are

$$\mathbf{B}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 3 \\ 0 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \mathbf{B}_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix} \quad (6)$$

for the CC and WM, and TVC-1 and TVC-2 inner coding schemes, respectively. We remark that the search provided protographs with a better threshold, but they all showed a higher error floor than \mathbf{B}_1 and \mathbf{B}_2 . All protographs were optimized for the case of $M = 1$ and over \mathbb{F}_{16} , except for the CC inner coding scheme for which \mathbb{F}_2 was used.

V. NUMERICAL RESULTS

In this section, we evaluate the DT bound (2) with the inner synchronization codes listed in Table I.

A. Simulation Parameters

We perform our simulations over the DNA alphabet $\{A, C, G, T\}$, which corresponds to $q = 4$. We consider the DNA storage channel in Figs. 1 and 2 with $p_5 = 0$ and

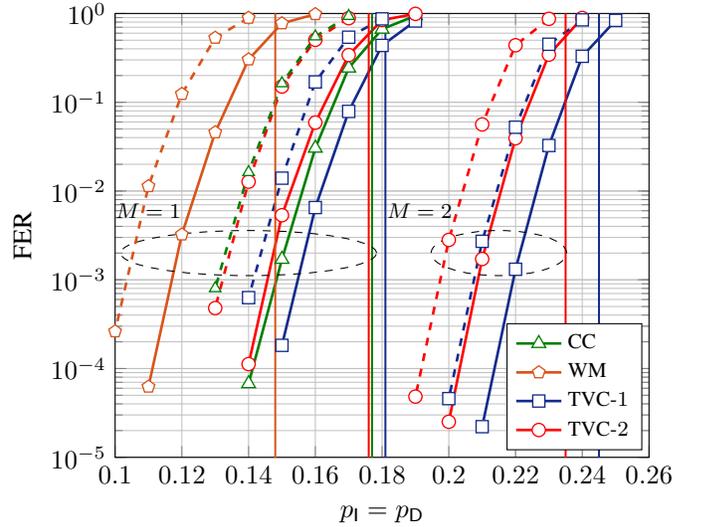


Fig. 3. DT bounds (solid lines with markers) for different inner synchronization codes, $N = 960$ DNA symbols, and $M = 1$ and $M = 2$. The simulated FER performance (dashed lines with markers) are for a concatenated code with an optimized outer LDPC code of rate $R = 1/2$.

$p_1 = p_D$ so that the drift random variable has zero mean (however, we remark that similar results are observed for other values and $p_5 \neq 0$). To limit the complexity of the decoder of the combination of the inner code and the DNA storage channel, we set the maximum number of consecutive insertions considered by the decoder to 2. Furthermore, we set the limit of the drift random variable to five times the standard deviation of the final drift at position N , i.e., to $5\sqrt{N \frac{p_D}{1-p_D}}$. Note, however, that the simulated channel may introduce more than two consecutive insertions and lead to a larger drift. The outer LDPC code is decoded with belief propagation with a maximum number of 100 iterations, and the maximum number of turbo iterations is set to 100.

We compute the DT bound for two code lengths, $N = 960$ and $N = 128$ DNA symbols, corresponding to a short and a medium-length sequence, respectively, and for $M = 1$ and $M = 2$ reads. The choice of these lengths is motivated by the current DNA sequencing technologies. All inner codes are of rate (or close to) $R_i = 1$ (in bits per DNA symbol) and all outer codes are of rate $R_o = 1/2$.

B. Discussion

In Figs. 3 and 4, we plot the DT bound (solid lines with markers) for the DNA storage channel with the inner synchronization codes in Table I for $N = 960$ and $N = 128$, respectively. The bound for $M = 2$ is obtained by considering the *joint decoding* algorithm proposed in [5]. Furthermore, in the figures we plot the asymptotic achievable information rates (vertical lines) computed in [5] for each inner coding scheme.

The TVC-1 scheme yields the best bound for both code lengths and values of M , and the watermark code gives the worst bound. Interestingly, the hierarchy of the bounds coincides with the hierarchy of the asymptotic achievable information rates.

In the figures, we also plot the FER performance (dashed lines with markers) for a concatenated code with an outer

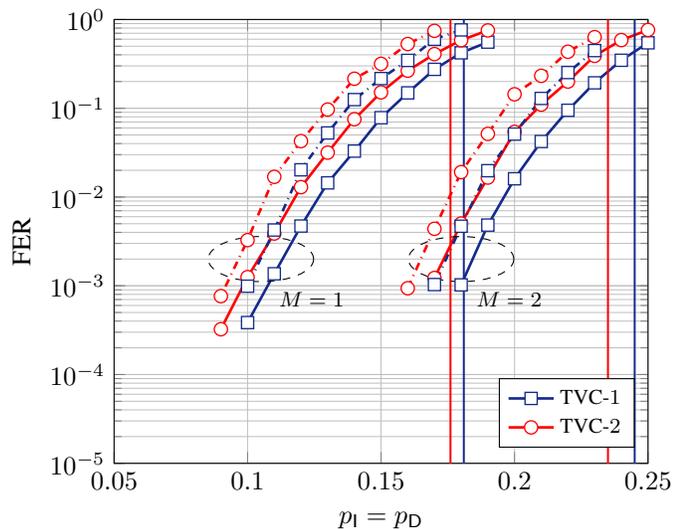


Fig. 4. DT bounds (solid lines with markers) for the TVC-1 and TVC-2 inner coding schemes, $N = 128$ DNA symbols, and $M = 1$ and $M = 2$. The simulated FER performance (dashed lines with markers) are for a concatenated code with an optimized outer LDPC code of rate $R = 1/2$.

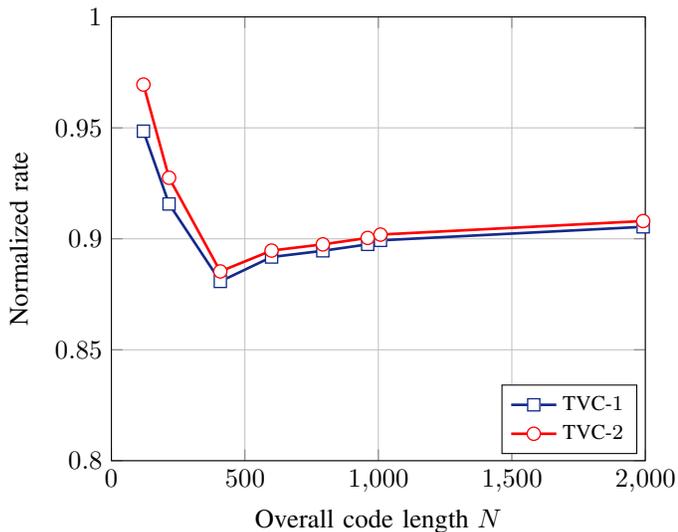


Fig. 5. Normalized rate for a concatenated coding scheme with an optimized outer LDPC code constructed from the protograph \mathbf{B}_2 in (6) and the TVC-1 and TVC-2 inner coding schemes as a function of the code length N . The overall code rate is $R = 1/2$ and the target FER is 10^{-3} .

LDPC code built from the optimized protographs in (6) and the inner coding schemes from Table I. In contrast to the optimization, circulant matrices for the protograph liftings, built using the progressive edge-growth algorithm [14], are used. The slope of the FER curves is similar to the slope of the corresponding DT bounds and a similar gap to the bounds is observed for the simulated FER curves. Notably, the proposed concatenated schemes perform close to the DT bounds.

To gain more insight on the performance of the proposed concatenated schemes to the DT bound, in Fig. 5 we plot the normalized rate [7] as a function of the code length N for the concatenated code with the TVC-1 and TVC-2 inner coding schemes. The normalized rate is computed as the fraction between the rate of the concatenated code and the

maximum rate provided by the DT bound so that decoding with a probability of error below a given value is possible. In other words, we want a normalized rate close to one and a normalized rate of one means that the code achieves the DT bound. In the plot, we consider a FER of 10^{-3} .

For both TVC-1 and TVC-2, the normalized rate is within 87% to 97% for a code length up to $N = 2000$ DNA symbols. These values are similar to those for state-of-the-art codes over memoryless channels [7, Fig. 15], indicating that the proposed concatenated codes yield excellent performance on the DNA storage channel.

VI. CONCLUSION

We provided an upper bound to the performance of random coding schemes on a DNA storage channel with insertions, deletions, and substitutions in the practical short-to-medium blocklength regime. The bound, which is based on the dependency testing bound yields an achievability result and is particularly useful to capture the performance of concatenated coding schemes with an inner synchronization code as it depends on the inner code. Hence, it is a handy tool to guide the choice of the inner synchronization code and provides a reference to benchmark the performance of coding schemes.

REFERENCES

- [1] S. M. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Sci. Rep.*, vol. 7, no. 5011, Jul. 2017.
- [2] L. Organick *et al.*, "Random access in large-scale DNA data storage," *Nature Biotechnol.*, vol. 36, no. 3, pp. 242–248, Mar. 2018.
- [3] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, no. 6102, p. 1628, Aug. 2012.
- [4] M. C. Davey and D. J. C. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.
- [5] I. Maarouf, A. Lenz, L. Welter, A. Wachter-Zeh, E. Rosnes, and A. Graell i Amat, "Concatenated codes for multiple reads of a DNA sequence," *IEEE Trans. Inf. Theory*, vol. 69, no. 2, pp. 910–927, Feb. 2023.
- [6] S. R. Srinivasavaradhan, S. Gopi, H. D. Pfister, and S. Yekhanin, "Trellis BMA: Coded trace reconstruction on IDS channels for DNA storage," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, VIC, Australia, Jul. 2021, pp. 2453–2458.
- [7] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
- [8] J. A. Briffa, H. G. Schaathun, and S. Wesemeyer, "An improved decoding algorithm for the Davey-MacKay construction," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Cape Town, South Africa, May 2010.
- [9] V. Buttigieg and J. A. Briffa, "Codebook and marker sequence design for synchronization-correcting codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 1579–1583.
- [10] D. M. Arnold, H.-A. Loeliger, P. O. Vontobel, A. Kavčić, and W. Zeng, "Simulation-based computation of information rates for channels with memory," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3498–3508, Aug. 2006.
- [11] H. D. Pfister, J. B. Soriaga, and P. H. Siegel, "On the achievable information rates of finite state ISI channels," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, San Antonio, TX, USA, Nov. 2001, pp. 2992–2996.
- [12] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Inf. Theory*, vol. 21, no. 4, pp. 404–411, Jul. 1975.
- [13] M. F. Mansour and A. H. Tewfik, "Convolutional decoding in the presence of synchronization errors," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 218–227, Feb. 2010.
- [14] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Progressive edge-growth Tanner graphs," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, San Antonio, TX, USA, Nov. 2001, pp. 995–1001.