Coding for Optimized Writing Rate in DNA Storage

Siddharth Jain*, Farzad Farnoud†, Moshe Schwartz‡, and Jehoshua Bruck*
*Electrical Engineering, California Institute of Technology, U.S.A., {sidjain,bruck}@caltech.edu
†Electrical & Computer Engineering, University of Virginia, U.S.A., farzad@virginia.edu
‡Electrical and Computer Engineering, Ben-Gurion University of the Negev, Israel, schwartz@ee.bgu.ac.il

Abstract—A method for encoding information in DNA sequences is described. The method is based on the precision-resolution framework, and is aimed to work in conjunction with a recently suggested terminator-free template independent DNA synthesis method. The suggested method optimizes the amount of information bits per synthesis time unit, namely, the writing rate. Additionally, the encoding scheme studied here takes into account the existence of multiple copies of the DNA sequence, which are independently distorted. Finally, quantizers for various run-length distributions are designed.

I. INTRODUCTION

DNA is believed to be a compelling medium for storing information due to its high density, energy efficiency, stability and longevity. Studies in the past decade have strengthened this belief by providing evidence that data on DNA can be written, stored and read both outside and inside living organisms [4], [7]–[9], [17], [19], [20]. DNA has been synthesized by a powerful method of phosphoramidite chemistry. However, the quantity and quality of the synthesized DNA using this method is limited by cost [4], [7] and chemistry limitations [6], [11], [13].

Recently, a new inexpensive enzymatic method of DNA synthesis was proposed in [10]. Unlike other synthesis methods [1], [3] that focus on the synthesis of a precise DNA sequence, in this method information storage does not rely on precision at a single base level. The method relies on the synthesis of runs of homopolymeric bases, the length of which may vary. Moreover, the length of each run is dependent on the nucleotide appearing before the current run, which makes the run length distribution context dependent or in other words have "memory". The distribution of run lengths also depends on the synthesis time for each run. One can expect reading long strands as an outcome of this synthesis method which is not a limitation now due to the advance long-read sequencing technologies like Oxford nanopore and PacBio [2], [15].

In [10], a coding approach was proposed that encodes information in the transitions between non-identical nucleotides. Hence, any message could be encoded using 3 symbols (trits as mentioned in the paper). Further, the synthesis method was found to incur deletion, insertion and substitution of nucleotides out of which the deletion error was the most prominent. These errors were tackled by synthesizing multiple strands with the introduction of synchronization bases interspersed between information carrying nucleotides and then using multiple sequence alignment for the reconstruction.

The main drawback in the suggested coding approach in [10] is that it does not make use of the information in the run-length distribution. Thus, the rate of information in [10] is inherently upper bounded by $\log_2 3$ bits per run.

In this paper, we suggest a method for encoding information that works in tandem with this new synthesis method. As a figure of merit, we study the number of information bits encoded per synthesis time unit, namely, we optimize for the *writing rate*. This differs from previous methods, that optimize the number of bits per channel use. The method we suggest is based on the Precision-Resolution (PR) scheme that was introduced in [16] to improve Run-Length Limitation (RLL) constraints. The PR framework provided an optimal set of run lengths when the clock frequency on the transmitter and receiver side was not the same. For the DNA synthesis method described in [10], the PR framework cannot be applied in its current form [16] because of the memory constraint, quantization error of run lengths and the availability of multiple copies of the same message on the receiver end. We extend the PR framework to include those constraints and design encoder and decoder utilizing the variations in run-lengths to store information in DNA.

The rest of the paper is organized as follows. In section II we describe constrained systems [14] and the terminator-free template independent DNA synthesis method introduced in [10]. In section III, we describe the generalized Precision Resolution scheme and provide examples of encoder and decoder when the run length distributions are Binomial and Poisson. In section IV, we conclude the paper and provide directions for future work.

II. PRELIMINARIES

A. Constrained Systems

Throughout the paper, let Σ denote some fixed finite alphabet. We shall eventually take $\Sigma \triangleq \{A, C, G, T\}$ to stand for the four bases found in DNA sequences. We use Σ^* to denote the set of all finite strings over Σ, and $\Sigma^+ \triangleq \Sigma^* \setminus \{\epsilon\}$, where ϵ denotes the unique empty string. We denote the length of $w \in \Sigma^*$ by $|w|$. The concatenation of $w, w' \in \Sigma^*$ is denoted by ww', and w^i denotes a concatenation of i copies of w.

Assume $G = (V, E, L)$ is a finite directed and labeled graph, that is V is a finite set of vertices, $E \subseteq V \times V$ is a finite *multiset* of edges (since we allow parallel edges), and $L : E \to \Sigma^+$ are the labels on the edges. We say the *length* of an edge $e \in E$ is $\ell(e) \triangleq |L(e)|$. If all the edge lengths are 1, we say the graph is *ordinary*.

Let $\gamma = e_1 e_2 \ldots e_m$ be a path in $G = (V, E, L)$, $e_i \in E$. We say the path generates the word $L(\gamma) \triangleq L(e_1)L(e_2)\ldots L(e_m)$. The empty path generates the empty word ϵ. The graph G is called *lossless* if given vertices $v_1, v_2 \in V$ (perhaps the same vertex) and a string $w \in \Sigma^*$, there is at most one path γ from v_1 to v_2 such that $L(\gamma) = w$.

A *constrained system* is represented by a finite directed and labeled graph $G = (V, E, L)$ as above. The language of the constrained system is defined as all the words generated by finite paths in G, namely,

$$S(G) \triangleq \{L(\gamma) \,|\, \gamma \text{ is a finite path in } G\}.$$

The *capacity* of a constrained system $S(G)$ is defined as

$$\mathsf{cap}(S(G)) \triangleq \limsup_{n \to \infty} \frac{1}{n} \log_2 |S(G) \cap \Sigma^n|.$$

This intuitively measures the average numbers of bits of information per string symbol in long strings. It is well known that there exist encoders with rates below the capacity, and no encoders with rate higher than the capacity. If G is ordinary and lossless, then it is also known that

$$\mathsf{cap}(S(G)) = \log_2 \lambda(A_G),$$

where A_G is the adjacency matrix of G, and $\lambda(A_G)$ is the spectral radius of A_G, which by Perron-Frobenius theory is the largest positive eigenvalue of A_G. We note that if G is not ordinary, we may easily construct an equivalent ordinary graph G' by converting each edge $v \xrightarrow{w_1 \ldots w_\ell} u$ (where $w_i \in \Sigma$) into a path $v \xrightarrow{w_1} v_1 \xrightarrow{w_2} \ldots \xrightarrow{w_{\ell-1}} v_{\ell-1} \xrightarrow{w_\ell} u$ by inserting auxiliary vertices $v_1, \ldots, v_{\ell-1}$.

B. Terminator-Free DNA Synthesis

In [10], a method for DNA synthesis is described which is faster and cheaper than known methods. It does, however, suffer from distinct noise characteristics. We describe here the relevant coding-theoretic aspects of this method.

The alphabet is, of course, $\Sigma \triangleq \{A, C, G, T\}$. The synthesis process proceeds in rounds. Assume at the beginning of the round, the current string is $w \in \Sigma^*$. A letter $a \in \Sigma$ is chosen, which differs from the last letter of w, denoted $b \in \Sigma$. A chemical reaction is then allowed to occur for a duration of $t \in \mathbb{N}$ time units. The resulting string at the end of the round is wa^m, where m is a random variable with distribution $D_{b \to a}(t)$. We emphasize the fact that the distribution of the new run lengths depends on the new letter being appended, the last letter of the string at the beginning of the round, and the duration of the chemical reaction.

III. GENERALIZED PRECISION-RESOLUTION SCHEMES

In this section we generalize the precision-resolution (PR) framework to be able to handle the DNA synthesis method described in Section II.

The PR framework was introduced in [16] as a way to improve RLL constraints. In essence, the RLL constraint was originally devised to solve the problem of clocking differences between the transmitter and receiver. A binary sequence may

be thought of as runs of zeros separated by ones, where the information is encoded in the lengths of the runs of zeros. However, if the clock frequency in the transmitter and receiver is not exactly the same, a transmitted run length may be measured differently by the receiver than originally intended by the transmitter. Thus, the constrained code employed should make sure run lengths may be unambiguously recoverable at the receiver side. It was shown in [16] that the choice of allowable run lengths in the RLL constraint is sub-optimal, and that the PR framework provides an optimal set of run-lengths, thus increasing the system's capacity.

The PR framework, however, cannot be applied "as-is" in the case at hand. This is due to three main differences:

1) **Memory** – The encoders described in [16] are *block encoders*, namely, encoders with a single state only. In the case of DNA synthesis as described in Section II, we must have memory since the behavior of each writing phase depends on the previous run that was written.
2) **Quantization error** – In [16] it is assumed that a transmitted run length may change, and that the distribution of the noisy run length has a finite support. Thus, [16] is able to partition the set of all possible run lengths into disjoint intervals, and the receiver may quantize a received run length to an interval unambiguously. This does not seem to be the case described in [10], and we must assume the quantizer at the receiver may err.
3) **Multiple copies** – The setting described in [16] allows the encoder to create a single output that is transmitted over the channel, and decoded at the receiver. Here, however, multiple copies of the DNA string may be created simultaneously. All of these copies are created from the same user information, but the synthesis process may result in different DNA strings. We may use these extra copies to improve our overall scheme.

We therefore describe a new framework that builds upon the work of [16] and generalizes it.

A. Framework Description

Fix the alphabet $\Sigma \triangleq \{A, C, G, T\}$. Assume a synthesis round of $t \in \mathbb{N}$ time units, attempting to write a letter $a \in \Sigma$ following a run of $b \in \Sigma$, $b \neq a$. The actual resulting run length is a random variable $r \sim D_{b \to a}(t)$, where we note that the distribution depends on the current run being written, a, the previous written run, b, and the duration of the synthesis round t.

Next, we observe that the synthesis method lends itself naturally to writing several copies of the same string (see [10]). Let $N \in \mathbb{N}$ denote the number of copies we write. Thus, we have N i.i.d. random variables $r_1, \ldots, r_N$ representing the actual run length resulting in each of the N copies.

Following the general PR framework of [16], we now do the following:

- We fix a maximal run-length decoding error, $\delta > 0$.
- We fix a maximal synthesis time for all runs, M. This is a desirable feature since long runs may affect the DNA molecule's stability.

- For each $a, b \in \Sigma$, $a \neq b$, a sequence of allowable synthesis-round times, $1 \leqslant t_{b \to a}^{(1)} < t_{b \to a}^{(2)} < \cdots < t_{b \to a}^{(\ell)} \leqslant M$, to be used when writing a run of a's following a run of b's. Thus, ℓ is the number of possible synthesis times for a run . (We note that for later convenience, ℓ does not depend on a and b.)
- Finally, we fix a quantizing function, $\mathcal{Q}_{b \to a} : \mathbb{N}^N \to [\ell]$. This function receives $(r_1, \ldots, r_N)$, a vector of N i.i.d. random variables that are distributed according to $D_{b \to a}(t_{b \to a}^{(i)})$ for some i. The function attempts to find i, returning $\hat{i} \in [\ell]$. We require $\Pr(\hat{i} = i | i) \geqslant 1 - \delta$.

We point out here, in the setup described above, we only consider the run-length errors and ignore other types of noise, such as insertion, deletion, and substitution errors. We also observe that this setting may cause a designed run to not appear at all, and then perhaps causing its two adjacent runs to be merged. However, by synthesizing several copies, the probability that the same run is deleted from *all* copies, is exponentially small in the number of copies. A similar strategy was used by a previous work, and run deletions may then be handled by an alignment algorithm [10].

We now construct the following graph $G = (V, E, L)$. We set $V \triangleq \Sigma$. For all $a, b \in \Sigma$, $a \neq b$, we add ℓ parallel edges directed from b to a with labels $a^{t_{b \to a}^{(i)}}$, for $i \in [\ell]$.

At this point we make an important observation: the system $S(G)$ does not describe the DNA strings we write. Instead, the strings in $S(G)$ may be thought of as a description of the synthesis algorithm. An edge with a label a^t describes running a synthesis round for the letter a for a duration of t time units. This is in particular important since the capacity, $\mathsf{cap}(S(G))$ measures the average bits per time unit of synthesis rounds. While this may seem odd at first, we emphasize the following two points. First, the main cost in the transmission process described in this paper is *not* the length of the DNA strings. Instead, the main cost stems from the time it takes to synthesize the DNA molecules. Second, even in a standard use of constrained systems, the length of the transmitted string differs from the length of the received string (for example, due to a difference in clock frequencies). Here, we may think of the transmitter as attempting to write the string describing the synthesis rounds, whereas the receiver reads the resulting DNA string.

Lemma 1. *The graph G constructed in this section is lossless. If G' is the equivalent ordinary graph of G (created by inserting dummy vertices) then*

$$\mathsf{cap}(S(G)) = \log_2 \lambda(A_{G'}).$$

Proof: Assume to the contrary there exist two distinct paths, γ and γ', from $u \in V$ to $v \in V$ that generate the same word. Without loss of generality we also assume the two paths disagree on the first edge taken.

Let a^ℓ be the label of the first edge in γ. We must have that the first edge of γ' is labeled by $a^{\ell'}$, where $\ell \neq \ell'$. Again, without loss of generality, assume $\ell > \ell'$. We must have that

γ' has a second edge, but by construction, it is labeled by b^m with $b \neq a$. It follows that $L(\gamma) \neq L(\gamma')$, a contradiction. Hence, G is lossless.

The proof for G' follows along the same lines after noting that two hypothetical distinct paths that generate the same word and differ in their first edge, must start in a vertex originally from G (and not in an auxiliary vertex).

Finally, the fact that

$$\mathsf{cap}(S(G)) = \mathsf{cap}(S(G')) = \log_2 \lambda(A_{G'}),$$

is well known (e.g., see [12]). ∎

Using standard techniques, e.g., the state-splitting algorithm (see [12], [14]), we can build encoders $\mathcal{E}$ for $S(G)$ with rate that asymptotically approaches $\mathsf{cap}(S(G))$. At the receiver side, the run lengths are qunatized using the quantizing functions $\mathcal{Q}_{b \to a}$, and a decoder, $\mathcal{D}$, is employed to reverse the encoding process of $\mathcal{E}$.

B. Fixing Quantizer Errors

We now turn to discuss errors due to the quantizers. Intuitively, at the receiver side, the receiver is well aware of the fact that the previous run was a run of b's, whereas the current one is a run of a's, for some $a, b \in \Sigma$, $a \neq b$. It is, therefore, the quantizer's goal to determine $t_{b \to a}^{(i)}$ from the received run length of a's. By construction, the probability of error is at most δ.

Assume a total of s runs was written, where the jth run was of designed length $t_{a_{i-1} \to a_i}^{(i_j)}$. It is now our goal to protect the sequence $i_1, i_2, \ldots, i_s \in [\ell]^s$ from a typical δ-fraction of errors. To that end, we may use any error-correcting code capable of correcting a typical δ-fraction of errors. Such codes exist with asymptotic redundancy of no more than $s' \triangleq s(\frac{1}{C_{\delta, \ell}} - 1)$ [5], where

$$C_{\delta, \ell} \triangleq 1 + \delta \log_\ell \frac{\delta}{\ell - 1} + (1 - \delta) \log_\ell (1 - \delta).$$

Using such a linear systematic code, we map $(i_1, i_2, \ldots, i_s) \mapsto (i_1, \ldots, i_s, i_{s+1}, \ldots, i_{s+s'})$, where $i_j \in [\ell]$ for all j.

Since $i_1, \ldots, i_s$ are already represented by the sequence synthesized thus far, we only need to append DNA bases to represent the redundancy of the code, namely the sequence $(i_{s+1}, \ldots, i_{s+s'}) \in [\ell]^{s'}$. We do so by first identifying our original alphabet Σ with $\mathbb{Z}_q$, where $q \triangleq |\Sigma|$ (where in the case of DNA bases, we get $q = 4$). We then transform the sequence of redundancy symbols $(i_{s+1}, \ldots, i_{s+s'}) \in [\ell]^{s'}$ to a sequence over the alphabet $[q - 1] = \{1, 2, \ldots, q - 1\}$, $(i_{s+1}, \ldots, i_{s+s'}) \mapsto (\bar{i}_1, \ldots, \bar{i}_{s''})$, where $\bar{i}_j \in [q - 1]$ for all j, and $s'' \triangleq \lceil s' \cdot \log_{q-1}(\ell) \rceil$.

Finally, to write the redundancy we use runs of designed length 1, and a $\mathbb{Z}_q$-differential encoding. In detail, if the first s runs were of the letters $a_1, \ldots, a_s$ respectively, the following s'' runs are of the letters

$$a_{s+1} = a_s + \bar{i}_1, a_{s+2} = a_{s+1} + \bar{i}_2, \ldots, a_{s+s''} = a_{s+s''-1} + \bar{i}_{s''}$$

where addition is done in $\mathbb{Z}_q$.

Overall, upon receiving a transmission, the receiver extracts the sequence of run lengths, $\hat{i}_1, \ldots, \hat{i}_s$. For the redundancy runs, the run lengths are of no importance, and the letters of the runs determine $\bar{i}_1, \ldots, \bar{i}_{s''}$, which may be used to find $i_{s+1}, \ldots, i_{s+s'}$. The decoder then uses the ℓ-ary error-correcting code to obtain $i_1, \ldots, i_s$ with high probability. It then feeds the corrected sequence to the decoder $\mathcal{D}$ to obtain the user information. We therefore have:

Theorem 2. *Using the setting described in this section, for all large enough k, any k user bits may be encoded into a sequence using at most*

$$k \cdot \frac{1}{\mathsf{cap}(S(G))} \cdot \left(1 + \left(\frac{1}{C_{\delta,\ell}} - 1\right) \log_{q-1}(\ell)\right)$$

synthesis time, and be decoded correctly with high probability.

Proof sketch: Take the k bits and feed them to the encoder $\mathcal{E}$ resulting in a DNA synthesis process of time $k \cdot \frac{1}{\mathsf{cap}(S(G))}$ asymptotically. In the worst case, this sequence is composed of synthesis rounds of length 1 only. Protect these $k \cdot \frac{1}{\mathsf{cap}(S(G))}$ rounds using an error-correcting code with rate (asymptotically) $C_{\delta,\ell}$, resulting in $k \cdot \frac{1}{\mathsf{cap}(S(G))}\left(\frac{1}{C_{\delta,\ell}} - 1\right)\log_{q-1}(\ell)$ more rounds of length 1 each that need to be written. ∎

Finally, this last theorem may be further improved by assuming the user input is uniform i.i.d. bits and replacing the worst-case assumption of all rounds of length 1, with the expected number of rounds. This may be done by a Markov-chain analysis of the encoder $\mathcal{E}$.

Theorem 3. *Assume the setting described in this section, and let G' be the ordinary version of G. Further assume the k user information bits are i.i.d. uniform random bits. Then for all large enough k, the user information bits may be encoded into a sequence using at most*

$$k \cdot \frac{1}{\mathsf{cap}(S(G))} \cdot \left(1 + \alpha\left(\frac{1}{C_{\delta,\ell}} - 1\right) \log_{q-1}(\ell)\right)$$

synthesis time, and be decoded correctly with high probability. Here α is the sum of probabilities of non-auxiliary vertices in the stationary distribution of the max-entropic Markov chain over G'.

Proof sketch: The proof is similar to that of Theorem 2, with the difference being that the expected number of synthesis rounds (after encoding the user information bits) is $k \cdot \frac{1}{\mathsf{cap}(S(G))} \cdot \alpha$. This is due to the fact that a new round starts every time we reach a non-auxiliary vertex in G'. ∎

C. Quantizers for Binomial Run Lengths

We proceed in this section and the following one, to study some special cases of quantizers, depending on the distribution of run lengths. In this section we assume that the run lengths have binomial distribution. Further, we assume that $D_{b \to a}(t)$ are independent of the choice of a and b, i.e. $D_{b \to a}(t) = D(t)$.

We construct a maximum-likelihood quantizer to find the correct run-length index. For a given run, let $(r_i)_{1 \leqslant i \leqslant N}$

denote the run lengths received at the output from N copies that were synthesized. Assuming r_i is i.i.d and distributed according to Binomial(t, p), the received run length sum $r(t) = \Sigma_{i=1}^{N} r_i$ is distributed according to Binomial(Nt, p), where $t \in \{t^{(1)}, t^{(2)}, \cdots, t^{(\ell)}\}$, i.e.

$$\Pr(r(t) = k) = \binom{Nt}{k} p^k (1-p)^{Nt-k}.$$

The log-likelihood is given by

$$\log \Pr(r|t^{(j)}) = \log\binom{Nt^{(j)}}{r} + r\log\frac{p}{1-p} + Nt^{(j)}\log(1-p).$$

The index j^* is determined by

$$j^* = \arg\max_j \log\binom{Nt^{(j)}}{r} + r\log\frac{p}{1-p} + Nt^{(j)}\log(1-p).$$

Let the true index be denoted by i. Let τ_h, $h = 0, \ldots, \ell$, be such that the range for which the decision is correct, i.e., $j^* = i$, is given by

$$\tau_{i-1} < r \leqslant \tau_i.$$

The probability of error is

$$\Pr(j^* \neq i | i) = \Pr(r > \tau_i) + \Pr(r \leqslant \tau_{i-1}) \leqslant \delta.$$

Below, we describe the steps to design an encoder for this setup for different values of p and δ.

- *Step 1:* Set $\tau_0 = 0$ and

$$t^{(1)} = \min\{t : \Pr(r(t) > 0) \geqslant 1 - \delta\}$$
$$\tau_1 = \min\{x : \Pr(1 \leqslant r(t^{(1)}) \leqslant x) \geqslant 1 - \delta\}$$

- *Step 2:* For each $2 \leqslant i \leqslant \ell$, $t^{(i)}$ is calculated by finding the minimum $t > t^{(i-1)}$ such that

$$\log\frac{(Nt^{(i-1)} + 1)\cdots Nt}{(Nt^{(i-1)} - \tau_{i-1} + 1)\cdots(Nt - \tau_{i-1})}$$
$$+ (Nt - Nt^{(i-1)})\log(1-p) \leqslant 0.$$

and $\tau_i = \min\{x : \Pr(\tau_{i-1} < r(t^{(i)}) \leqslant x) \geqslant 1 - \delta\}$.
Note that ℓ is the index where $t^{(\ell)} \leqslant M$ and $t^{(\ell+1)} > M$.

Figures 1(a) and 1(b) show the achievable rates for the PR system described above for $N = 1$ and $N = 5$, respectively, for different choices of p, δ, and $M = 10$. Figure 1(c) shows the achievable rates for fixed parameters, except for varying values of N.

D. Quantizers for Poisson Run Lengths

Suppose that the run lengths have a Poisson distribution with a parameter that can be set by the encoder. The parameter is chosen from the set $\{\lambda_1, \lambda_2, \ldots\}$, $\lambda_j < \lambda_{j+1}$. The decoder decides λ_{j^*} if $\tau_{j^*-1} < \bar{r} \leqslant \tau_{j^*}$ where $\bar{r} = \frac{1}{N}\Sigma_{i=1}^{N} r_i$, and $\{\tau_0, \tau_1, \cdots\}$ are designed such that

$$\Pr(\bar{r} \leqslant \tau_{j^*-1}|\lambda_{j^*}) \leqslant \frac{\delta}{2} \quad \text{and} \quad \Pr(\bar{r} > \tau_{j^*}|\lambda_{j^*}) \leqslant \frac{\delta}{2}.$$
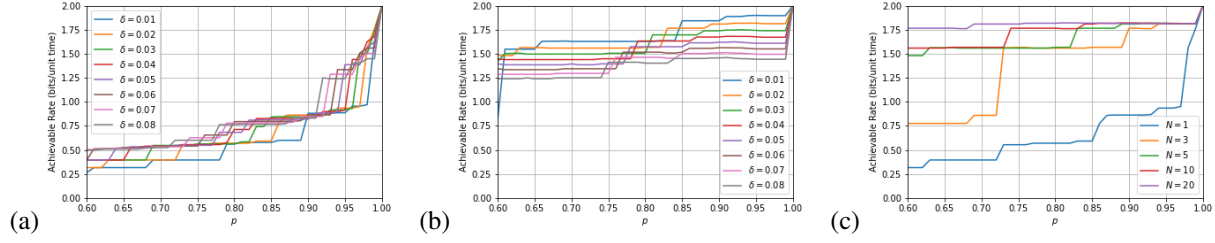
Fig. 1. (a) Achievable rate for Binomial(Nt, p), $N = 1$ and $M = 10$ for different choices of δ using Theorem 3. (b) Achievable rate for Binomial(Nt, p), $N = 5$ and $M = 10$ for different choices of δ using Theorem 3. (c) Comparison of Achievable rates for Binomial(Nt, p), $\delta = 0.02$ and $M = 10$ for different choices of N using Theorem 3.
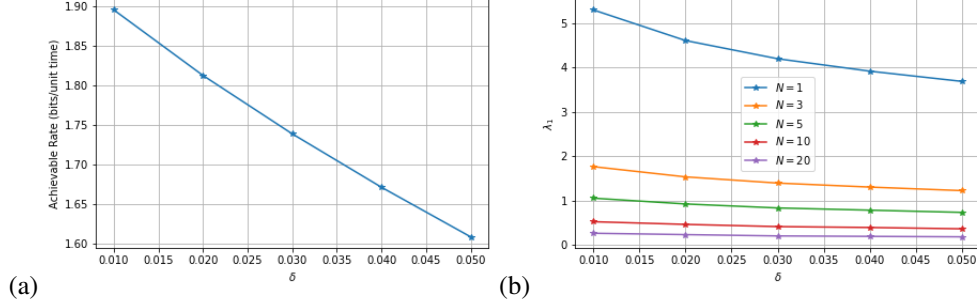

Fig. 2. (a) Achievable rates for Poisson Run lengths for different values of δ using Theorem 3 with $\lambda_j = \lambda_1 (t^{(j)})^2$ where $t^{(1)} = 1$. Here λ_1 (see Figure 2(b)) is a function of N and δ. (b) Required λ_1 for different values of N and δ.

Let the true index be denoted by i. The range for which the decision is correct is given by

$$\tau_{i-1} < \bar{r} \leqslant \tau_i$$

The probability of error is given by:

$$\Pr(\bar{r} > \tau_i) + \Pr(\bar{r} \leqslant \tau_{i-1}) \leqslant \delta$$

for $N\bar{r} \sim Poisson(N\lambda_i)$.

Below, we describe the design an encoder for this setup.

- *Step 1:* Set $\tau_0 = 0$ and $\lambda_1 = \frac{1}{N} \log(\frac{2}{\delta})$.
- *Step 2:* For each $1 \leqslant i \leqslant \ell - 1$, τ_i and λ_{i+1} are calculated as follows:

$$\tau_i = \min\{x : \Pr(\bar{r} > x) \leqslant \frac{\delta}{2}\}$$

for $N\bar{r} \sim \text{Poisson}(N\lambda_i)$.

$$\lambda_{i+1} = \min\{\lambda > \lambda_i : \Pr(\bar{r} \leqslant \tau_i) \leqslant \frac{\delta}{2}\}$$

for $N\bar{r} \sim \text{Poisson}(N\lambda)$.

Figure 2(a) shows the achievable rate for different values of δ for Poisson run length based encoder and decoder obtained using the steps described above with $\ell = 10$. Figure 2(b) shows the λ_1 required for different values of N to achieve the rate curve given in Figure 2(a).

We point out here that the steps presented for choosing $t^{(1)}, t^{(2)}, \cdots, t^{(\ell)}$ in sections III-C and III-D are based on heuristics and may not be optimal. Nevertheless, the results show that the achievable rate is well beyond the maximum rate of $\log_2 3$ of [10].

IV. CONCLUSION

We described in this paper a method for encoding information in DNA sequences, that is based on the precision-resolution framework [16], in conjunction with the terminator-free synthesis method [10]. This method takes into account several possible run lengths for each run of the synthesized sequence, thus increasing the rate beyond the fundamental upper bound of $\log_2 3$ bits per symbol. Additionally, the method accounts for quantizer error during the reading process, and it utilizes the fact that the receiver is in possession of multiple copies (independently distorted) of the synthesized sequence. We make note that the synthesis process may delete entire runs, perhaps also causing neighboring runs to merge. However, as in a previous work [10], such deletions may be handled by the alignment algorithm.

We also studied Binomial and Poisson distributions for synthesized run lengths, and provided methods for designing quantizers for these distributions. A point for future research is to conduct empirical studies of the synthesis process of [10] in order to find the actual parameters of run length distributions.

Finally, we believe this method should eventually be used in conjunction with addressless information encoding, such as coding of sliced information [18]. We leave this for future research.

REFERENCES

[1] S. Beaucage and M. Caruthers, "Deoxynucleoside phosphoramidites—a new class of key intermediates for deoxypolynucleotide synthesis," *Tetrahedron Letters*, vol. 22, no. 20, pp. 1859 – 1862, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0040403901904617

[2] C. Bleidorn, "Third generation sequencing: technology and its potential impact on evolutionary biodiversity research," *Systematics and Biodiversity*, vol. 14, no. 1, pp. 1–8, 2016. [Online]. Available: https://doi.org/10.1080/14772000.2015.1099575

[3] M. H. Caruthers, "A brief review of DNA and RNA chemical synthesis," *Biochemical Society Transactions*, vol. 39, no. 2, pp. 575–580, 03 2011. [Online]. Available: https://doi.org/10.1042/BST0390575

[4] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in DNA," *Science*, vol. 337, p. 1628, 2012.

[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[6] P. Gaytán, "Chemical synthesis of oligonucleotides using acetone as a washing solvent," *Biotechniques*, vol. 47, pp. 701–702, 2009.

[7] 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.

[8] 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," in *Angew*. Engl. 54, 2015, pp. 2552–2555.

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

[10] 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. 2383, pp. 1–12, 2019.

[11] E. M. LeProust, B. J. Peck, K. Spirin, H. B. McCuen, B. Moore, E. Namsaraev, and M. H. Caruthers, "Synthesis of high-quality libraries of long (150mer) oligonucleotides by a novel depurination controlled process," *Nucleic Acids Research*, vol. 38, no. 8, pp. 2522–2540, 03 2010. [Online]. Available: https://doi.org/10.1093/nar/gkq163

[12] D. Lind and B. H. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1985.

[13] D. Lowe, *The Great Acetonitrile Shortage. In the Pipeline*, 2009. [Online]. Available: http://blogs.sciencemag.org/pipeline/archives/2009/01/22/the_great_acetonitrile_shortage

[14] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Constrained Systems and Coding for Recording Channels*. V. S. Pless and W. C. Huffman (Editors), Elsevier, Amsterdam, 1998.

[15] M. O. Pollard, D. Gurdasani, A. J. Mentzer, T. Porter, and M. S. Sandhu, "Long reads: their purpose and place," *Human Molecular Genetics*, vol. 27, no. R2, pp. R234–R241, 05 2018. [Online]. Available: https://doi.org/10.1093/hmg/ddy177

[16] M. Schwartz and J. Bruck, "On the capacity of the precision-resolution system," *IEEE Trans. Inform. Theory*, vol. 56, no. 3, pp. 1028–1037, 2010.

[17] S. L. Shipman, J. Nivala, J. D. Macklis, and G. M. Church, "CRISPR-Cas encoding of digital movie into the genomes of a population of living bacteria," *Nature*, vol. 547, pp. 345–349, Jul. 2017.

[18] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," in *Proceedings of the 2019 IEEE International Symposium on Information Theory (ISIT2019), Paris, France*, Jul. 2019, pp. 767–771.

[19] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific reports*, vol. 7, no. 1, 2017.

[20] S. H. T. Yazdi, Y. Yuan, J. Ma, H. Zhao, and O. Milenkovic, "A rewritable, random-access DNA-based storage system," *Scientific reports*, vol. 5, 2015.