Sequence Reconstruction Under Stutter Noise in Enzymatic DNA Synthesis

Roy Shafir[*], **Omer Sabary**[‡], **Leon Anavy**[*†] **Eitan Yaakobi**[*], and **Zohar Yakhini**[*†]
[*] Taub Faculty of Computer Science, Technion, Haifa, 3200003, Israel.
[†] School of Computer Science, Herzliya IDC, Herzliya, 4610101, Israel.
[‡] Electrical and Computer Engineering Department, UCSD, La Jolla, CA 92093-0407 USA
roy.shafir@campus.technion.ac.il, osabary@ucsd.edu, yaakobi@cs.technion.ac.il

Abstract—**Synthetic DNA is an attractive alternative for data storage media due to its high information density, low energy usage, and exceptional robustness. *Enzymatic DNA synthesis* was recently introduced to allow cost effective synthesis of longer DNA molecules for data storage. This method is characterized by *stutter errors* which are sticky insertions so that every base in the designed sequence may be synthesized more than once. In this work, we study the problem of reconstructing the original sequence from a set of noisy reads originating from the stuttering enzymatic synthesis. We present different reconstruction algorithms and analyze their expected success probability and error rate for three different scenarios that depend on the information which is known about the stutter errors. We evaluate algorithmic performance analytically as well as by using simulations. We are especially interested in characterizing the performance as a function of the read depth. Our findings can be used to evaluate the trade-offs between synthesis quality indicators and the sequencing depth required for reconstruction with high probability. In principle, the probability of reconstruction failure exponentially decays with the sequencing depth, as demonstrated in the study. We also analyze the use of error-correcting codes to improve the error performance.**

I. Introduction

Synthetic DNA is an attractive alternative for data storage media. With an information density orders of magnitude better than that of magnetic media and due to its exceptional resilience DNA can potentially efficiently store data for centuries and in dynamic conditions [13, 18]. This was demonstrated in studies covering a variety of encoding schemes, sequencing technologies, and data access capabilities [1, 4, 6, 8, 12, 13, 29, 30].

One of the main limitations of DNA based storage systems is the DNA synthesis technology. Current synthesis technology is based on iterative phosphoramidite chemistry which limits the length of the synthesized DNA molecules. Different approaches to overcome this limitation have been proposed over the years.

Two recent studies suggested the use of customized enzymatic reaction for de-novo synthesis of DNA molecules [19, 23]. Lee et al. [19] suggested an enzymatic DNA synthesis system that uses a combination of enzymes to create a desired DNA sequence. Briefly, template independent DNA polymerase is used to elongate the synthesized molecules with a desired base as substrate while apyrase is used to degrade the substrate bases simultaneously. The result is a DNA molecule which consists of a sequence of homopolymer stretches (runs), however with *variable* lengths. The order of the different runs is determined according to the designed sequence while their length varies and depends on the chemical dynamics of the two enzymes. The resulting molecules may be referred to as a stuttering representation of the original sequence where each letter in the design appears one or more times consecutively. To overcome the ambiguity of the variable run-lengths, Lee et al. suggested using a ternary encoding so that no two consecutive letters will be identical (thus avoiding the synthesis of long "merged runs"). This encoding simplifies the decoding and reduces the error rates. However, this also significantly reduces the density of the DNA based storage so that every letter encodes no more than $\log_2(3) = 1.58$ bits. Later, a graph representation of this synthesis model was studied in [17], with the purpose of optimizing the synthesis time. Yet another strongly related work in [22] explores a similar error model that was motivated by Nanopore sequencers. Detailed descriptions of [17] and [22] can be found in Section III.

The problem of decoding from multiple noisy copies of the information falls under the general framework of the *reconstruction problem* such as the *trace reconstruction* and the *sequence reconstruction* problems. The latter one was introduced by Levenshtein [20, 21], where a sequence is transmitted over several noisy channels and the goal is to study the minimum number of channels that guarantee recovery in the *worst case*. This problem has been extensively studied for several error channels, such as substitutions, insertions, deletions, and more; see e.g. [11, 14, 15, 24, 25, 26, 28]. In the trace reconstruction problem [2], a sequence x is transmitted over several deletion channels, where each symbol is deleted with probability p. This transmission creates multiple noisy copies of x (also called traces), and the goal is to determine the minimum number of i.i.d traces in order to reconstruct x with *high probability*. A comprehensive survey of the reconstruction problems and their application to computational biology can be found in [3].

This work is focused on stutter noise that can only increase the length of each homopolymer (run). We present decoding algorithms that minimize errors with no constraints on the encoded data. Our algorithms decode the sequence from a set of noisy reads generated through the stuttering synthesis process. First, we represent each read as a sequence of runs corresponding to the sequence of runs in the designed sequence. Next we estimate the designed length of each run from the set of observed lengths. Our algorithms differ in this estimation step. We show that a maximum-likelihood estimation outperforms other approaches. We analyze the different algorithms both analytically and by simulations. Similarly to previous works, we first examine the case in which the stutter noise of the synthesis process is well characterized and this information is available to the decoding algorithms. However, as opposed to previous works, we also investigate the more realistic case in which the information on the stuttering noise is not available to the decoding algorithms. Finally, we evaluate the use of error-correcting codes with our algorithms to improve the system's reliability. Due to lack of space some of the proofs are omitted and appear in the full version of this paper in [27].

II. Problem Definition and Notations

A communication channel is said to generate traces of a fixed message if it produces several perturbed or noisy copies of the message. A *stutter channel* $\mathcal{C}$ with M outputs produces M traces from the designed sequence s; see Fig. 1. In our context these traces are equivalent to reads sampled from a population of synthesized molecules, all nominally the same but with stutter synthesis noise. Stutter enzymatic synthesis noise is predominantly due to the same base (nucleotide) being synthesized in multiplicities, i.e., it is assumed to have *sticky insertions* [7, 16].

Let Σ be the alphabet of the sequences, which is typically Σ={A,C,G,T} and let $\boldsymbol{s} = s_1^{k_1} s_2^{k_2} \cdots s_\ell^{k_\ell}$ be the designed sequence which consists of ℓ homopolymers (runs) where the j-th homopolymer, $s_j^{k_j}$, corresponds to k_j intended repetitions of $s_j \in \Sigma$ and for $1 \leq j < \ell, s_j \neq s_{j+1}$. Let $V_M = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M\}$ be the multiset of M traces received at the M outputs from the stutter channel $\mathcal{C}$. Note that these traces represent M sequencing reads from the synthesized molecules. The stutter channel $\mathcal{C}$ is characterized by a conditional probability Pr, which is defined by $Pr\{\boldsymbol{v} \text{ rec.} | \boldsymbol{s} \text{ trans.}\}$ for every pair $(\boldsymbol{v}, \boldsymbol{s}) \in (\Sigma^*)^2$. Under the assumption of stutter noise, the probability is positive only for pairs $(\boldsymbol{v}, \boldsymbol{s}) \in (\Sigma^*)^2$ for which $\boldsymbol{v}$ can be received by sticky insertions in $\boldsymbol{s}$. Hence, the sequence $\boldsymbol{v}$ has to be of the form $\boldsymbol{v} = s_1^{n_1} s_2^{n_2} \cdots s_\ell^{n_\ell}$, where $n_j \geq k_j$ for $1 \leq j \leq \ell$. Furthermore, the conditional probability Pr can be succinctly described by the probability over the values of n_j given the ones of k_j, that is, $Pr\{N_j = n_j | k_j\}$, where N_j is a random variable which indicates the length of the j-th homopolymer. Note that since the bases are synthesized sequentially, one at a time, these lengths are i.i.d. and also the distribution of each length N_j can be further simplified. That is, $N_j = \sum_{i=1}^{k} T_i$ where T_i is a random variable indicating the number of bases the channel outputs when a single base is synthesized. Hence, the stutter channel $\mathcal{C}$ can be described by the probability $Pr\{T = t\}$, or simply by T. For the rest of the paper, the stutter channel $\mathcal{C}$ will be simply described by the probability distribution of the random variable T or simply by the random variable T and is denoted by $\mathcal{C}(T)$. Note that $Pr\{N_j = n_j | k_j = 1\} = Pr\{T = n_j\}$. Hence, the relation between $Pr\{N = n|k\}$ and $Pr\{T = t\}$ is given by

$$Pr\{N = n|k\} = \prod_{i_1, \ldots, i_k : \sum_{j=1}^{k} i_j = n} Pr\{T = i_j\}.$$

That is, given the value of k_j, the random variable N_j is equivalent to the summation of k_j independent instances of the random variable T.

Assume the random variable T in the synthesis channel has a geometric distribution given by the parameter p, i.e., $T \sim Geom(p)$ and $Pr(T = t) = (1-p)^{t-1}p$. Then, the random variable N_j has a negative binomial distribution since

$$Pr\{N = n|k\} = \prod_{i_1, \ldots, i_k : \sum_{j=1}^{k} i_j = n} Pr\{T = i_j\}$$

$$= \prod_{i_1, \ldots, i_k : \sum_{j=1}^{k} i_j = n} (1-p)^{i_j - 1} p = \binom{n-1}{k-1} p^k (1-p)^{n-k}.$$

This case where the stutter channel has a geometric distribution will receive close attention in the paper since we believe it best describes the error behavior of this synthesis method [19].

Note that the probability $Pr\{T = t\}$ may depend on several more parameters such as the base of the homopolymer, its location in the sequence, and the previous homopolymer. However, since for stutter noise only sticky insertions occur, every homopolymer is solved independently and as such these parameters are taken into account in the random variable T that might differ in its parameters between different homopolymers. This observation motivates us to define the reconstruction algorithms that will be studied in this paper.

For a single homopolymer s^k, a *reconstruction algorithm* $\mathcal{A}$ takes as an input the values $n_1, \ldots, n_M$ observed in V_M and outputs an estimate $\widehat{k}$ of the value of k. Note that there is no need to get $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M$, since the symbol of the homopolymer is known and therefore it suffices for the algorithms to get just the homopolymer's observed lengths. We consider the following two performance indicators for the algorithms. They are naturally extended to the entire sequence in Section V.

1) $P_{fail}(\mathcal{A}; k, M, \mathcal{C}(T))$ is the *failure probability* of $\mathcal{A}$, i.e.,
$$P_{fail}(\mathcal{A}; k, M, \mathcal{C}(T)) = P\{\mathcal{A}(n_1, \ldots, n_M) \neq k\}.$$
2) $E_{err}(\mathcal{A}; k, M, \mathcal{C}(T))$ is the *expected deviation error rate* of $\mathcal{A}$, in short *error rate*, i.e. the expectation of the normalized absolute difference between $\mathcal{A}(n_1, \ldots, n_M)$ and k,
$$E_{err}(\mathcal{A}; k, M, \mathcal{C}(T)) = E\left[\frac{|\mathcal{A}(n_1, \ldots, n_M) - k|}{k}\right].$$

We distinguish three different scenarios. In the first one, no knowledge on the stutter channel is assumed with respect to the distribution of the random variable T. The second one provides the type of the random variable T, but does not specify its parameters. For example, we can assume that $T \sim Geom(p)$ but the value of p is unknown. Lastly, the third provides both the type of the random variable as well as its parameters. The goal of this work is to study several reconstruction algorithms for the stutter channel, with its three scenarios, and analyze their failure probability and error rate. Previous works only addressed special cases of the third scenarios.

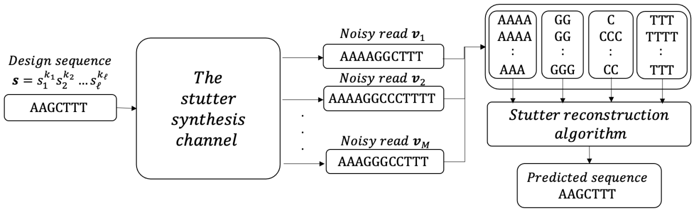

Design sequence
$\boldsymbol{s} = s_1^{k_1} s_2^{k_2} \ldots s_\ell^{k_\ell}$
AAGCTTT
The stutter synthesis channel
Noisy read $\boldsymbol{v}_1$
AAAAGGCTTT
Noisy read $\boldsymbol{v}_2$
AAAAGGCCCTTTT
Noisy read $\boldsymbol{v}_M$
AAAGGGCCTTT
AAAA AAAA : AAA
GG GG : GGG
C CCC : CC
TTT TTTT : TTT
Stutter reconstruction algorithm
Predicted sequence
AAGCTTT

Figure 1: The communication channel of the stutter synthesis method and a description of the stutter reconstruction process.

III. Related Work

The stutter synthesis method was first introduced by Lee et al. [19] as a new low cost DNA synthesis technology. This synthesis method uses enzymes to synthesize sequences according to a given design sequence. This process results with several copies of the sequence, while each can be erroneous and the dominant errors are reflected in variations in the length of each homopolymer in the sequence. Lee et al. also described in their work an encoding scheme with the following two main ideas. The first one imposes the length of each homopolymer to be 1, which limits the information rate to be at most $\log_2 3$. The second one uses synchronization nucleotides, which are designated homopolymers within the design sequence that are used to eliminate the less likely sequences in the decoding.

Jain et al. [17] studied the capacity of stutter synthesis with the intention of optimizing the synthesis time. They allowed homopolymers of length greater than one, and thereby the achievable information rate could be larger than $\log_2 3$. They modeled the possible errors of the synthesized homopolymers according to some error distribution (Binomial or Poisson). Then, they translated the error distribution into a constrained system and created a directed graph that represents all possible sequences that can be synthesized using the stutter synthesis method. The constrained system graph was used in order to determine and optimize the number of synthesis cycles and thus to calculate the capacity of this synthesis method. Additionally, they explained how error-correcting codes can be used in their coding scheme in order to correct run-length errors of the homopolymers. Our work is focused on the other side of the coin when no coding is used, and presents practical algorithms for efficient reconstruction of DNA strands that were synthesized with this enzymatic method.

Magner et al. [22] studied the reconstruction problem from Nanapore sequencers. In their model they assumed "sticky" insertions and deletions, in the sense that the length of any homopolymer in the sequence can increase or decrease but cannot be deleted completely. They proved that, under some assumptions on the error distributions, the necessary number of reads needed to reconstruct a sequence with high probability is $\theta(\log \ell)$ where ℓ is the number of the homopolymers in the sequence. Their reconstruction algorithm assumes knowledge on the expected length of the read homopolymers, as a function of their original length. Accordingly, their decoding algorithm calculates the average length of the read homopolymers and uses it to estimate its original length. They proved that $C \log_2^{2\gamma+1} \ell$ reads, for some constants C and γ, are sufficient to recover the sequence with high probability. In this case, γ depends on the variance of the error distribution of the homopolymers. Even though this work is strongly related to ours, there are three major differences. First, our model is motivated by the enzymatic DNA synthesis (as opposed to the Nanopore sequencers) and hence we assume error distributions where the length of any homopolymer can only increase. This can be easily achieved by tuning the ratio between the two enzymes to ensure elimination of deletion errors. Second, in our problem the number of reads is given and can not be controlled. Hence, an adaptation of the algorithm presented in [22] to our model, referred in this paper as the mean-driven algorithm, is described in Section IV-B and is also compared to our other algorithmic approaches. Third, we also consider the more realistic model in which the exact stutter noise behavior is not known during decoding.

IV. Reconstruction Algorithms

In this section we describe our reconstruction algorithmic approaches. Let $T \sim Dist(\Theta)$ where $Dist$ is the stutter distribution and Θ describes the parameters defining the distribution $Dist$. As mentioned before, there are several distinct scenarios regarding our assumed knowledge of $Dist$ and/or Θ. Hence, we present in this section several reconstruction algorithms that can be used under these different scenarios. Clearly, any information about T can only improve the reconstruction performance. Reasonable estimates of $Dist$ and Θ can be obtained when the biological processes involved become mature and when inspection can be repeatedly performed on many production cycles. In Section V we investigate the performance of relevant approaches and compare them. We also discuss the advantage of knowing $Dist$ and/or Θ. We note that said advantage diminishes as the read depth, and therefore M increases.

Throughout this paper we use the following two additional abbreviated notations. The condidtional probability $Pr\{N_i = n|k\}$ defined in Section II is denoted by $p_k(n)$. Additionally, $CP_k(m)$ denotes the cumulative probability of m under p_k. That is, $CP_k(m) = \sum_{h=k}^{m} p_k(h)$. Note that P_{fail} and E_{err} are functions of p_k and CP_k.

A. The Min-Driven Algorithm

Our first and naive approach to estimate the length k, of a homopolymer s^k, assumes no knowledge on T. This approach is based on the fact that for all $i \in [M], n_i \geq k$. I.e., homopolymers can only be prolonged by the stutter synthesis channel. Hence, the output of our first algorithm, which is referred to as the *min-driven algorithm*, is simply $\mathcal{A}_{\min}(n_1, \ldots, n_M) = \min_{i=1}^{M}\{n_i\}$. Next, we analyze the failure probability and the error rate of the min-driven algorithm $\mathcal{A}_{\min}$.

Theorem 1 *It holds that*

1) $P_{fail}(\mathcal{A}_{\min}; k, M, \mathcal{C}(T)) = (1 - p_k(k))^M$.

2) $E_{err}(\mathcal{A}_{\min}; k, M, \mathcal{C}(T)) = \frac{1}{k}\sum_{h=k}^{\infty}(1 - CP_k(h))^M$.

Proof: Denote $\widehat{k} = \mathcal{A}_{\min}(n_1, \ldots, n_N)$. First, note that as $\mathcal{A}_{\min}$ outputs the shortest length of the reads, the value of $\widehat{k}$ is at least k.

1) $P_{fail}(\mathcal{A}_{\min}; k, p) = Prob(\min_{i=1}^{N}\{X_i\} > k) = \prod_{i=1}^{N} Prob(X_i > k)^N = (1 - p_k(k))^N = (1 - p^k)^N$.

2) $E_{err}(\mathcal{A}; k, p) = \frac{E[|\widehat{k} - k|]}{k} = \frac{E(\widehat{k}) - k}{k} \overset{(*)}{=} \frac{1}{k}\sum_{h=k}^{\infty}(1 - CP_k(h))^N$, where (*) holds since $E(\widehat{k}) = E(\min_{i=1}^{N}\{X_i\}) = \sum_{h=1}^{\infty} Prob(\min_{i=1}^{N}\{X_i\} \geq h) = \sum_{h=0}^{\infty}(1 - CP_k(h))^N = k + \sum_{h=k}^{\infty}(1 - CP_k(h))^N$.

■

Note that the failure probability of this simple algorithm exponentially decays with M and that for the case $T \sim Geom(p)$, it holds that $P_{fail}(\mathcal{A}_{\min}; k, M, \mathcal{C}(T)) = (1 - p^k)^M$.

B. The Mean-Driven Algorithm

Next, the case where the expectation of the random variable T, denoted by μ_T, is known. That is, it is assumed that the distribution is not necessarily known but the expected stutter length on a single base is known. Inspired by the decoding algorithm presented in [22], we develop an algorithm for assessing the value of k, using the average length of the respective homopolymers within the reads in V_M. This algorithm, referred to as the *mean-driven algorithm* and denoted $\mathcal{A}_{\text{mean}}$, leverages the fact that $T \sim Dist(\Theta)$. The total observed length of the homopolymer is then the sum of k independent instances, $N = \sum_{i=1}^{k} T_i$, where T_i is the random variable counting the number of synthesized bases for the i-th base in the homopolymer. Recall that $E[N] = k\mu_T$. Hence, if the length of the designed homopolymer is k then the average length of the observed homopolymers should be roughly $k\mu_T$, especially when M is large enough.

Let $\bar{n} = \sum_{i=1}^{M} n_i/M$ be the average value of the lengths n_i of the observed reads. Then, the output of the mean-driven algorithm is $\mathcal{A}_{\text{mean}}(n_1, \ldots, n_M) = \widehat{k} = [\bar{n}/\mu_T]$, where $[x]$ is the closest integer to x. In words, the algorithm predicts $\widehat{k}$ to be the integer yielding an expectation of the random variable that is closest to the observed average of the length of the reads (In case there are two closest integers we choose the larger one). Another improvement which we used in our simulations in Section V takes advantage of the fact that the possible values of k are necessarily between 1 and $\min_{i=1}^{M}\{n_i\}$. We did not consider this possible improvement in the next theorem in order to simplify the analysis. Note that the expressions in Theorem 2 upper bound the failure probability and error rate of this algorithm. Denote by $P_{out}\{\mathcal{A}_{\text{mean}}$ outputs $r; k, M, \mathcal{C}(T)\}$, or shortly $P_{out}(r; k)$, the probability that the algorithm's output is r, given M, T, and that the designed homopolymer is of length k.

Theorem 2 *It holds that*

1)$P_{out}(r; k) =$

$$CP_{Mk}(\lceil M(r + 0.5)\mu_T \rceil - 1) - CP_{Mk}(\lceil M(r - 0.5)\mu_T \rceil - 1).$$

2) $P_{fail}(\mathcal{A}_{mean}; k, M, \mathcal{C}(T)) = 1 - P_{out}(k; k)$.

3) $E_{err}(\mathcal{A}_{mean}; k, M, \mathcal{C}(T)) = \frac{1}{k}\sum_{\widehat{k}=1}^{\infty} |\widehat{k} - k| P_{out}(\hat{k}; k)$.

Proof: Let $N = \sum_{i=1}^{M} N_i$ and $\bar{N} = N/M$ the total and average length respectively. Note that N is a sum of M independent random variables N_i, $N = \sum_{i=1}^{M}\sum_{j=1}^{k} T_{i,j}$. The cumulative distribution of N is therefore CP_{Mk}. Let $\widehat{k} = \mathcal{A}_{\text{mean}}(n_1, \ldots, n_M) = [\bar{n}/\mu_T]$. Clearly, the estimation is correct when $k - 0.5 \leq \bar{n}/\mu_T < k + 0.5$, that is, for any $\bar{n}$ such that $(k-0.5)\mu_T \le \bar{n} < (k+0.5)\mu_T$. Therefore, $P_{out}(r;k) = \Pr\{r-0.5 \le \bar{N}/\mu_T < r+0.5\} \overset{(*)}{=} CP_{Mk}(\lceil M\mu_T(r+0.5)\rceil - 1) - CP_{Mk}(\lceil M\mu_T(r-0.5)\rceil - 1)$ where (*) holds as for $z \in \mathbb{R}^+$, $Pr\{\bar{N} < z\} = CP_{Mk}(\lceil Mz\rceil - 1)$.

■

C. The Maximum Likelihood Algorithm

The maximum likelihood (ML) approach is to find the most likely $\widehat{k}$ given the observed data. That is, our estimator $\widehat{k}$ is the one that defines the model that maximizes the (posterior) probability of observing the data V_M. This algorithm can be used in two scenarios regarding our knowledge of T. First, when we have all the knowledge of T (both $Dist$ and Θ) and the second when only $Dist$ is known but the exact parameters Θ are not known. Furthermore, as for all $i \in [M]$, $n_i \ge k$ we specifically have $n_{\min} = \min_{i=1}^{M}\{n_i\} \ge k$. Let K be a random variable that governs the **designed** length of the homopolymer. That is, $Pr\{K = k\}$ is the prior probability for the designed length. Let $\mathcal{N}$ be a random variable which indicates the multiset of the observed lengths of the homopolymer in the reads.

Additionally, we define $Pr\{\mathcal{N} = \{n_1, \ldots, n_M\}|K = k\}$ as the conditional probability of the observed lengths given the designed length k and under the stutter noise T. Since the stutter noise in the different reads is assumed to be independent, it holds that $Pr\{\mathcal{N} = \{n_1, \ldots, n_M\}|K = k\} = \prod_{i=1}^{M} p_k(n_i)$.

The output of the ML decoder is

$$\begin{aligned} &\mathcal{A}_{\text{ML}}(n_1, \ldots, n_M) \qquad (1)\\ &= \underset{k' \in [n_{\min}]}{\arg\max}\{Pr\{K = k'\}Pr\{\mathcal{N} = \{n_1, \ldots, n_M\}|K = k\}\}\\ &= \underset{k' \in [n_{\min}]}{\arg\max}\{Pr\{K = k'\}\prod_{i=1}^{M} p_{k'}(n_i)\}. \end{aligned}$$

As $\mathcal{A}_{\text{ML}}$ assumes that $p_k(n)$ can be calculated, we first explain briefly how $p_k(n)$ can be calculated assuming the distribution of T is known. Let U, V be independent random variables over $\mathbb{N}^+$ with mass functions f, g respectively. The convolution of f, g is defined as $Conv(f,g)(n) = \sum_{i=0}^{\infty} f(i)g(n-i)$. Note that $Conv(f,g)(n)$ is the mass function of the random variable $Z = U + V$. In the case of a single probability mass function f, representing a random variable U, we define recursively the ℓ-th convolution of f as follows. $Conv(f;\ell) = Conv(f, Conv(f;\ell - 1))$ and $Conv(f,2) = Conv(f,f)$. $Conv(f;\ell)$ represents the probability mass function of the sum of ℓ collectively independent copies of U. Denote by $f_T(x)$ the probability mass function of T which defines the stutter channel $\mathcal{C}$. Then, the probability mass function of $N = \sum_{i=1}^{k} T_i$, which we denote by $p_k(n) = Pr\{N = n\} = Conv(f_T;k)$. Note that in many cases, given a specific distribution of T, a simpler calculation of $p_k(n)$ (or $Conv(f_T;k)$) is possible. For example, if $T \sim Geom(p)$ we get that $p_k(n) = \binom{n-1}{k-1}p^k(1-p)^{n-k}$.

Four variants of $\mathcal{A}_{\text{ML}}$ are defined by two properties:

1) **Θ is known or Θ is unknown:** this property affects the way we calculate the probability $Pr\{\mathcal{N}|K = k\}$.
   Θ is known: in this case the calculation of $Pr\{\mathcal{N}|K = k\}$ is straightforward as we are conditioned by k. Therefore, we have all the information required to calculate $p_k(n_i)$ using convolutions and it is possible to calculate and find the ML decoder's output as expressed in (1).
   Θ is unknown: first note that in this case the expression in (1) cannot be calculated directly since the probability $p_{k'}(n_i)$ depends on the value of Θ. Hence, for a given value of k', it is necessary to first find an estimator for Θ, and for ML decoding we find the one that maximizes the probability $\prod_{i=1}^{M} p_{k'}(n_i)$. If the expression $\prod_{i=1}^{M} p_{k'}(n_i)$ can be differentiated with respect to Θ, then we set the maximum-likelihood estimator $\widehat{\Theta}$ by finding $\widehat{\Theta} = \widehat{\Theta}(k')$ that nullifies the derivative of $\prod_{i=1}^{M} p_{k'}(n_i)$ and maximizes $\prod_{i=1}^{M} p_{k'}(n_i)$. For example, for the specific case of the geometric distribution we get that given k', the maximum likelihood for $\widehat{p}$ is $\widehat{p} = (Mk')/(\sum_{i=1}^{M} n_i)$.
2) **With or without prior:** the prior refers to $Pr\{K = k'\}$, the distribution function of the homopolymer lengths, which follows from the coding of the binary message into DNA.
   Without prior: $Pr\{K = k'\}$ is ignored.
   With prior: this case fits better a real life scenario. In most of the cases we assume that all the possible designed sequences (messages) are equiprobable. This follows from the fact that the original binary message is assumed to be compressed. That is, all $|\Sigma|^m$ possible sequences of length m are equiprobable. Therefore for any letter ℓ_i at index $1 \le i \le m$ and $\sigma \in \Sigma$ it holds that $Pr\{\ell_i = \sigma\} = 1/|\Sigma|$. It follows that the homopolymer length is distributed geometrically, that is, $K \sim Geom(1 - 1/|\Sigma|)$, which means that $\{Pr\{K = k'\} = (1/|\Sigma|)^{k-1}(1 - 1/|\Sigma|)$.

Any of the 4 variants of $\mathcal{A}_{\text{ML}}$ can be used depending on whether Θ is known and on whether prior knowledge on the distribution of the designed sequences is available. Note that in stutter synthesis the typical case Θ would be unknown (synthesis not well characterized) and a prior on K is known (the binary message is compressed).

V. Performance Evaluation

This section evaluates the effectiveness of the different algorithms by simulations in the case where the stutter channel has geometric distribution. For given values of k and given that $T \sim Geom(p)$ we generate a set $n_1, \ldots, n_N$ of observed lengths, simulated as instances of a sum of k random variables (one per symbol in the designed homopolymer), that is, $\sum_{i=1}^{k} T_i$.

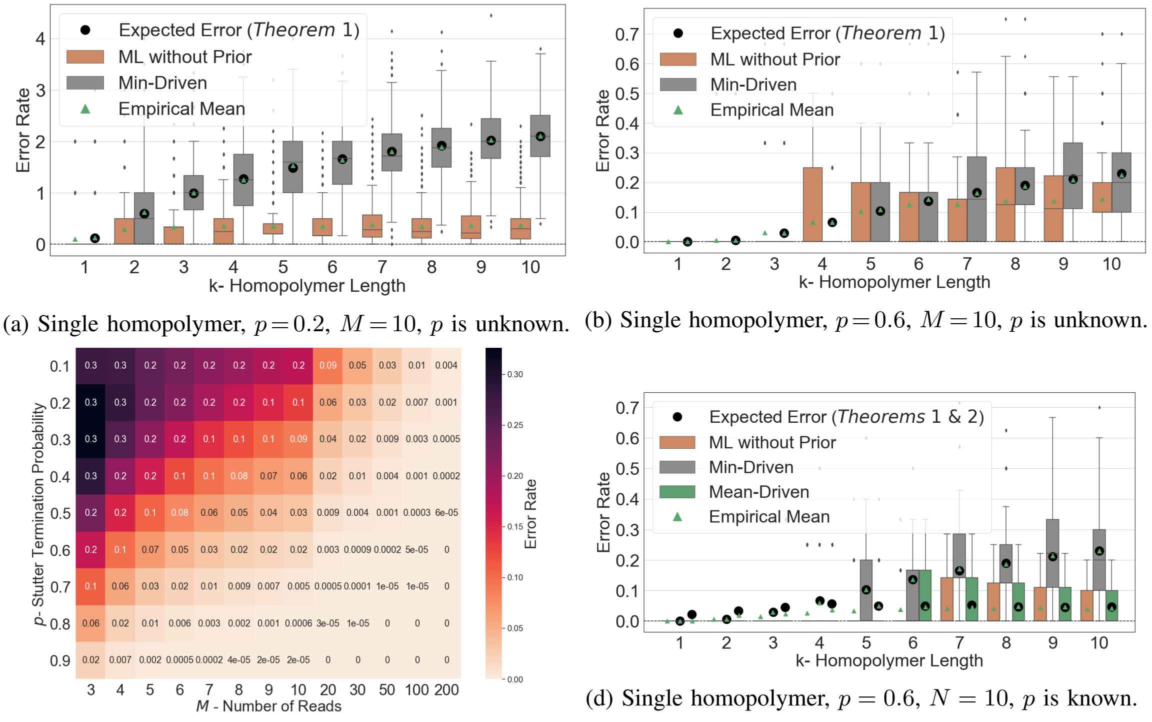
For the single homopolymer evaluations we compared $\mathcal{A}_{\min}, \mathcal{A}_{\text{mean}}$, and $\mathcal{A}_{\text{ML}}$ and calculated P_{fail} and E_{err} as defined in Section II. Note that when using $\mathcal{A}_{\text{ML}}$ on a single homopolymer, we worked without the prior probability. To simulate a full sequence, representing an encoded message, a random sequence $\boldsymbol{s}$ of length $|\boldsymbol{s}| = 150$ bases is created, where in each position the symbol is selected uniformly from Σ. Based on the designed sequence $\boldsymbol{s}$ we simulated a set of M reads $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_M$, such that each read is a noisy copy of $\boldsymbol{s}$. Each simulated read is created by simulating every homopolymer of length k_j as described above. We then applied $\mathcal{A}_{\min}, \mathcal{A}_{\text{mean}}$ and $\mathcal{A}_{\text{ML}}$ on every homopolymer lengths set from the simulated reads. We extended P_{fail} and E_{err} for the case of the full sequence so that P_{fail} is the *failure probability* of the entire sequence $\boldsymbol{s} = s_1^{k_1}s_2^{k_2}\cdots s_\ell^{k_\ell}$ and E_{err} is the average error per base.

$$\begin{aligned} &P_{fail}(\mathcal{A};\boldsymbol{s},M,\mathcal{C}(T)) = Pr\{\exists 1 \le j \le \ell; \mathcal{A}(n_{1,j}, \ldots, n_{M,j}) \ne k_j\}\\ &= 1 - \prod_{j=1}^{\ell}(1 - P_{fail}(\mathcal{A};k_j,M,\mathcal{C}(T))),\\ &E_{err}(\mathcal{A};\boldsymbol{s},M,\mathcal{C}(T)) = E\left[\frac{\sum_{j=1}^{\ell}|\mathcal{A}(n_{1,j}, \ldots, n_{M,j}) - k_j|}{\sum_{j=1}^{\ell} k_j}\right]. \end{aligned}$$

Due to lack of space, some of the simulation results were eliminated from this version of the paper and they appear in the full version of the paper [27].

A. Known p

We first tested the case where p is known to the algorithms. Figure 2d depicts the effect of the designed length k on E_{err} when tested on a single homopolymer. We observed that larger values of k yield larger errors for $\mathcal{A}_{\min}$ while $\mathcal{A}_{\text{mean}}$ and $\mathcal{A}_{\text{ML}}$ perform better.



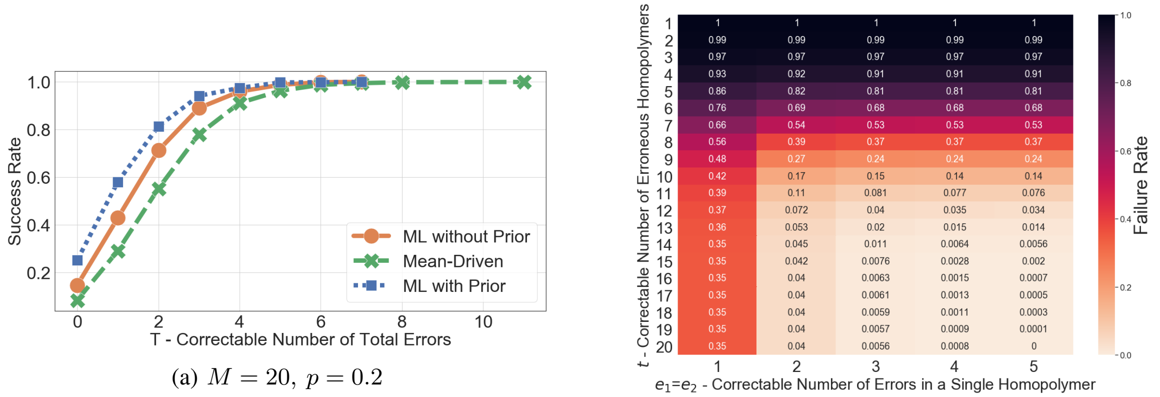
(a) Single homopolymer, $p = 0.2$, $M = 10$, p is unknown.

(b) Single homopolymer, $p = 0.6$, $M = 10$, p is unknown.

(c) ML with prior. Average error rate, p is unknown.

(d) Single homopolymer, $p = 0.6$, $N = 10$, p is known.

Figure 2: Performance evaluation. Single homopolymer (a, b, and d), and full sequence (c).



(a) $M = 20$, $p = 0.2$

(b) Failure rate as a function of the error correcting capability. $p = 0.2$, $M = 20$.

Figure 3: Expected failure/success rates as a function of the reconstruction algorithm and the applied error-correcting code.

B. Unknown p

Next, we tested the case where p is unknown which is the more realistic case for enzymatic DNA synthesis. Figure 2a and 2b depict the effect of the designed length k on E_{err} when tested on a single homopolymer. As expected $\mathcal{A}_{\text{min}}$ performs the same (as the knowledge of p is not used), and $\mathcal{A}_{\text{ML}}$ has higher error rates but still preforms better, especially for lower p and/or large k. In the next figure we analyze the performance on a full sequence. On Figure 2c we look into $\mathcal{A}_{\text{ML}}$ with prior performance and observe how p and M affect E_{err}. Clearly, as M and/or p increases the error rate decreases.

C. Error-Correcting Codes for Improved Reliability

In this section we evaluate how the data-reliability of the stutter synthesis can be improved when using error-correcting codes. Since in the stutter synthesis the dominant errors are run-length errors, when a length of homopolymer is changed, we evaluate two families of error-correcting codes. The first family, which corresponds to error-correcting codes in the Lee or the Manhattan metrics [9, 10], can correct a *total* of T indel errors, where T corresponds to the sum of the errors in all the homopolymers in the sequence. The second family, typically referred by *limited-magnitude error-correcting codes* [5], can correct up to a given e_1 insertion and e_2 deletion errors in at most some t homopolymers. Note that from the practical point of view of constructing error-correcting codes it is recommended that the number of homopolymers in the sequences will be fixed and hence a few changes might be needed in the sequence design. Figure 3a present the expected success rates of $\mathcal{A}_{\text{mean}}$ and $\mathcal{A}_{\text{ML}}$ for different error-correcting coding schemes when $p = 0.2$ is known. The X-axis represents the value of T in the first family of codes, while the Y-axis represents the observed success rate when using this scheme for $M = 20$. It can be seen that the ML algorithm improves the reliability of the data, especially when the goal is to successfully retrieve the whole data. For example, for $p = 0.2$ using the $\mathcal{A}_{\text{mean}}$ algorithm requires the code to correct 11 errors to achieve a success rate of 1, while using the $\mathcal{A}_{\text{ML}}$

algorithm the required correctable number of errors by the code is 7. Hence, less redundancy is needed when using $\mathcal{A}_{\mathrm{ML}}$ as the reconstruction algorithm and thus the information rate increases. Lastly, Figure 3b presents, for the second family of codes, the expected failure rate of $\mathcal{A}_{\mathrm{ML}}$ as a function of the error-correction capability when $p = 0.2$ and $N = 20$. The X-axis represents the maximal number of errors in a single homopolymer, where we use $e_1 = e_2$, while the Y-axis represents the correctable number of homopolymers. As expected, the failure rate decreases when the error-correction capability increases.

References

[1] 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.

[2] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 910–918. Society for Industrial and Applied Mathematics, 2004.

[3] V. Bhardwaj, A. P. Pevzner, C. Rashtchian, Y. Safonova, "Trace reconstruction problems in computational biology," *arXiv preprint* arXiv:2010.06083, 2020.

[4] M. Blawat, K. Gaedke, I. Hutter, 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.

[5] Y. Cassuto, M. Schwartz, V. Bohossian, and J. Bruck, "Codes for asymmetric limited-magnitude errors with applications to multilevel flash memories," *IEEE Trans. on Inform. Theory*, vol. 56, no. 4, pp. 1582-1595, 2010.

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

[7] E. Drinea and M. Mitzenmacher, "Improved lower bounds for the capacity of iid deletion and duplication channels," *IEEE Trans. on Inform. Theory*, vol. 53, no. 8, pp. 2693–2714, 2007.

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

[9] T. Etzion, "Product Constructions for Perfect Lee Codes," *IEEE Trans. on Inform. Theory*, vol. 57, no. 11, pp. 7473-7481, 2011.

[10] T. Etzion, A. Vardy, and E. Yaakobi, "Coding for the Lee and Manhattan metrics with weighing matrices," *IEEE Trans. on Inform. Theory*, vol. 59, no. 10, pp. 6712-6723, 2013.

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

[12] 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, 2013.

[13] 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.

[14] N. Holden, R. Pemantle, and Y. Peres, "Subpolynomial trace reconstruction for random strings and arbitrary deletion probability," *arXiv preprint*, arXiv:1801.04783, 2018.

[15] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," *The nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 389–398. 2008.

[16] A.R. Iyengar, P.H. Siegel, and J.K. Wolf, "On the capacity of channels with timing synchronization errors," *IEEE Trans. on Inform. Theory*, vol. 62, no. 2, pp. 793–810, 2016.

[17] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Coding for Optimized Writing Rate in DNA Storage," *IEEE International Symposium on Information Theory*, 2020.

[18] J. Koch, S. Gantenbein, K. Masania, W. J. Stark, Y. Erlich, and R. N. Grass. A DNA-of-things storage architecture to create materials with embedded memory. Nature Biotechnology, 38(1), 39-43, 2020.

[19] 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.

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

[21] 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.

[22] A. Magner, J. Duda, W. Szpankowski, and A. Grama, "Fundamental bounds for sequence recon- struction from Nanopore sequences," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 92–106, 2016.

[23] S. Palluk, D.H. Arlow, T. de Rond, S. Barthel, J.S. Kang, R. Bector, H.M. Baghdassarian, A.N. Truong, P.W. Kim, A.K. Singh, N.J. Hillson, J.D. Keasling, "De novo DNA synthesis using polymerase-nucleotide conjugates," *Nature Biotechnology*, vol. 36, pp. 645–650, 2018.

[24] Y. Peres and A. Zhai, "Average-case reconstruction for the deletion channel: subpolynomially many traces suffice," *IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 228–239, 2017.

[25] O. Sabary, A. Yucovich, G. Shapira, E. Yaakobi, "Reconstruction Algorithms for DNA-Storage Systems," bioRxiv 2020.09.16.300186; doi: https://doi.org/10.1101/2020.09.16.300186, 2020.

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

[27] R. Shafir, O. Sabary, L. Anavy, E. Yaakobi, and Z. Yakhini, "Sequence Reconstruction Under Stutter Noise in Enzymatic DNA Synthesis," , 2021.

[28] M. Abu Sini and E. Yaakobi, "Reconstruction of sequences in DNA storage," *IEEE International Symposium on Information Theory*, pp. 290-294. 2019.

[29] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Ka- math, P. Gopalan, B. Nguyen, C. N. Takahashi, S. Newman, H.-Y. Parker, C. Rashtchian, K. Stewart, G. Gupta, R. Carlson, J. Mulligan, D. Carmean, G. Seelig, L. Ceze, and K. Strauss, "Random access in large-scale DNA data storage," *Nature Biotechnology*, vol. 36, pp. 242. 2018.

[30] S. H. T. Yazdi, R. Gabrys, and O. Milenkovic, "Portable and error-free DNA-based data storage," *Scientific Reports*, vol. 7, no.1, pp. 5011, 2017.