Batch Optimization for DNA Synthesis

Konstantin Makarychev*, Miklós Z. Rácz†, Cyrus Rashtchian‡, and Sergey Yekhanin§
*Northwestern University, Evanston, IL 60208, USA, konstantin@northwestern.edu
†Princeton University, Princeton, NJ 08544, USA, mracz@princeton.edu
‡University of California, San Diego, La Jolla, CA 92093, USA, crashtchian@eng.ucsd.edu
§Microsoft Research, Redmond, WA 98052, USA, yekhanin@microsoft.com

***Abstract*—Large pools of synthetic DNA molecules have been recently used to reliably store significant volumes of digital data. While DNA as a storage medium has enormous potential because of its high storage density, its practical use is currently severely limited because of the high cost and low throughput of available DNA synthesis technologies.**

We study the role of batch optimization in reducing the cost of large scale DNA synthesis, which translates to the following algorithmic task. Given a large pool $\mathcal{S}$ of random quaternary strings of fixed length, partition $\mathcal{S}$ into batches in a way that minimizes the sum of the lengths of the shortest common supersequences across batches.

We introduce two ideas for batch optimization that both improve (in different ways) upon a naive baseline: (1) using both $(ACGT)^*$ and its reverse $(TGCA)^*$ as reference strands, and batching appropriately, and (2) batching via the quantiles of an appropriate ordering of the strands. We also prove asymptotically matching lower bounds on the cost of DNA synthesis, showing that one cannot improve upon these two ideas. Our results uncover a surprising separation between two cases that naturally arise in the context of DNA data storage: the asymptotic cost savings of batch optimization are significantly greater in the case where strings in $\mathcal{S}$ do not contain repeats of the same character (homopolymers), as compared to the case where strings in $\mathcal{S}$ are unconstrained.

A full version of this paper is accessible at:
https://arxiv.org/abs/2011.14532

I. Introduction

Storing digital data in synthetic DNA molecules has received much attention in the past decade [1]–[11]. DNA data storage offers several orders of magnitude higher information density compared to conventional storage media, as well as the potential to store data reliably for hundreds or thousands of years. However, the prohibitively high cost and low throughput of modern DNA synthesis technologies present a key barrier that needs to be addressed in order to make DNA data storage a commonplace technology.

For the purposes of the current paper we can think of a DNA molecule as a string (strand) in the quaternary alphabet $\{A, C, G, T\}$. Today the dominant method for producing large quantities of DNA molecules is array-based DNA synthesis [12], [13]. With this technology the DNA synthesizer creates a large number of DNA strands in parallel, where each strand is grown by one DNA base (character) at a time. To append bases to strands, the synthesis machine follows a fixed supersequence of bases, called a reference strand. As the machine iterates through this supersequence, the next base is added to a select subset of the DNA strands. This process continues until the machine reaches the end of the supersequence. In particular, each synthesized DNA strand must be a subsequence of the reference strand. The cost of DNA synthesis is proportional to the length of the reference strand.

In applications to DNA data storage one typically needs to synthesize very large quantities of DNA molecules, significantly exceeding the capacity of any single DNA synthesizer. Therefore the pool of strands that one aims to synthesize needs to be partitioned into batches, where the size of each batch corresponds to the maximum load of the synthesizer. In this setting the total cost of DNA synthesis is proportional to the sum of the lengths of the shortest common supersequences of each batch. The focus of this paper is the algorithmic task of *batch optimization*, where the goal is to partition the strands into batches and assign every batch a reference strand in a way that minimizes this cost.

The encoding process that generates the list of DNA strands that need to be synthesized to store a given digital file varies with the specific system [2], [5], [9], [14] and is usually quite complex. The encoder adds redundancy to the data to allow for the correction of various types of errors that occur during DNA synthesis, storage, and sequencing, including insertions, deletions, and substitutions of individual bases, as well as missing DNA strands.

We now describe two aspects of encoding of digital data in DNA that are relevant to our work. Commonly, input digital data is randomized [9] using a seeded pseudorandom number generator or compressed and encrypted [14]; this is done in order to reduce the frequency of undesirable patterns that may occur in strands that are used to represent the data, for instance, patterns likely to cause the presence of DNA secondary structure [15]. Ensuring that strands look random also facilitates certain tasks that may be a part of the decoding process such as clustering [9], [16] and trace reconstruction [9], [17]–[21]. Another important aspect is as follows. Algorithms that encode digital data in DNA [3], [9] often ensure that the resulting strands do not contain long runs of the same character (i.e., homopolymers), since such runs are known to cause errors during the DNA sequencing stage. The length of the longest allowed homopolymer run may be as low as one—that is, not allowing homopolymers—or unconstrained, depending on the scenario.

Motivated by the above considerations, in the current paper we model pools $\mathcal{S}$ of DNA strands that we aim to synthesise

as large collections of random quaternary strings. We consider two key representative cases: the case where strings in $\mathcal{S}$ are unconstrained and the case where strings in $\mathcal{S}$ do not contain repeats of the same character.

II. Problem statement

Fix a strand length n, and consider two different choices for the strand universe $\mathcal{U}$.

1) *Unconstrained strands:* $\mathcal{U} = \{A, C, G, T\}^n$.
2) *Strands without homopolymers:* $\mathcal{U}$ is the subset of $\{A, C, G, T\}^n$ that contains all strands with no consecutively repeated characters.

Let $\mathcal{S}$ be a subset of elements of $\mathcal{U}$, with $M := |\mathcal{S}|$; this is the pool of strands we wish to synthesize. Let k be an integer that divides M, and let π be a partition of $\mathcal{S}$ into k subsets (which we refer to as *batches*) $\mathcal{B}_1, \ldots, \mathcal{B}_k$ of size M/k.[1] We define $\mathsf{cost}(\mathcal{B}_i)$, the cost of synthesizing elements of the batch $\mathcal{B}_i$, as the length of the shortest common supersequence of all strands in $\mathcal{B}_i$. Using this notation, we define the cost of synthesizing the whole pool $\mathcal{S}$ as:

$$\mathsf{cost}(\mathcal{S}) := \min_{\pi} \sum_{i=1}^{k} \mathsf{cost}(\mathcal{B}_i). \tag{1}$$

We assume that elements of $\mathcal{S}$ are selected i.i.d. from $\mathcal{U}$ uniformly at random, and we are interested in upper and lower bounds for $\mathsf{cost}(\mathcal{S})$ that hold with high probability.

While in the practice of DNA synthesis the parameters n, M, and k are concrete numbers, to facilitate the asymptotic study of the problem we focus on the following relevant scenario: n is growing, M is significantly larger than but polynomial in n, and k is either a constant or a slowly growing function of n. For simplicity, we often state results in terms of universal constants; these can be made explicit.

Example 2.1: Consider the setting of strands with no homopolymers. Let $n = 4$ and $M = 4$. Let $\mathcal{S} = \{AGCT, GCAT, CAGA, GAGC\}$. Assume that $k = 2$, that is, there are two batches and each batch contains two strands.

We can partition $\mathcal{S}$ into $\mathcal{B}_1 = \{AGCT, GCAT\}$ and $\mathcal{B}_2 = \{CAGA, GAGC\}$. The DNA synthesizer (printer) first prints $\mathcal{B}_1$. It starts with two empty strings $(\emptyset, \emptyset)$. Then, it appends A to the first strand and obtains strands $(A, \emptyset)$. It appends G to both strands and obtains (AG, G). Then, it appends the letters C, A, and T as follows:

$$(\emptyset, \emptyset) \xrightarrow{A} (A, \emptyset) \xrightarrow{G} (AG, G) \xrightarrow{C} (AGC, GC) \xrightarrow{A} (AGC, GCA) \xrightarrow{T} (AGCT, GCAT).$$

After the last step, we get the set $\mathcal{B}_1 = \{AGCT, GCAT\}$. The printer prints $\mathcal{B}_2$ as follows:

$$(\emptyset, \emptyset) \xrightarrow{C} (C, \emptyset) \xrightarrow{G} (C, G) \xrightarrow{A} (CA, GA) \xrightarrow{G} (CAG, GAG) \xrightarrow{A} (CAGA, GAG) \xrightarrow{C} (CAGA, GAGC).$$

[1]The assumption that the batches are of equal size is made for simplicity. Indeed, our techniques extend to a more general setting where the batches are roughly the same size (e.g., up to a constant factor), and several results are phrased in this more general setting.

In this example, we used the reference strand $AGCAT$ to print the set $\mathcal{B}_1$ in five steps and the reference strand $CGAGAC$ to print the set $\mathcal{B}_2$ in six steps. Therefore $\mathsf{cost}(\mathcal{S}) \leq 11$.

III. Main results for multiple batches

Before describing our main results for multiple batches, we briefly and informally discuss the setting of a single batch—formal statements and proofs are in the full version of the paper [22]. A natural reference strand to use to print a pool of strands $\mathcal{S}$ is the periodic strand $(ACGT)^*$, where $ACGT$ repeats indefinitely. Our results can also be generalized to periodic sequences, such as $(ATATCG)^*$ or $(ACGACGT)^*$, leading to a cost analysis for other supersequences.

For the reference strand $(ACGT)^*$, we can write the cost of printing a random strand as $\sum_{i=1}^{n} X_i$, where $\{X_i\}_{i=1}^{n}$ are i.i.d. uniformly random on $\{1, 2, 3, 4\}$ in the case of unconstrained strands; in the case of strands without homopolymers, $\{X_i\}_{i=1}^{n}$ are independent, with X_1 uniformly random on $\{1, 2, 3, 4\}$ and X_i uniformly random on $\{1, 2, 3\}$ for $i \geq 2$. By a standard concentration inequality, we obtain the upper bounds $\mathsf{cost}\,(\mathcal{S}) \leq 2.5n + 3\sqrt{n \log M}$ for unconstrained strands and $\mathsf{cost}\,(\mathcal{S}) \leq 2n + 3\sqrt{n \log M}$ for strands without homopolymers, with both bounds holding with probability $1 - o(1)$. Combining this with an appropriate stochastic domination argument that compares random walks, we also obtain matching lower bounds, for both choices of the strand universe $\mathcal{U}$. This shows that for a single batch no reference strand can do asymptotically better than the periodic strand $(ACGT)^*$.

The setting of multiple batches, which is the focus of our work, presents interesting challenges. As a simple baseline, we could consider randomly partitioning $\mathcal{S}$ into k batches. A direct application of the single batch upper bound would provide a cost of roughly $2.5nk + O(k\sqrt{n \log(M/k)})$ for unconstrained strands and $2nk + O(k\sqrt{n \log(M/k)})$ for strands without homopolymers. We provide improvements in both cases by using a slightly more sophisticated batching method.

We first observe a symmetry property: For any strand without homopolymers the cost of printing it using $(ACGT)^*$ and the cost of printing it using $(TGCA)^*$ add up to $4n + 1$, so the better choice of reference strand results in a cost of at most $2n$. This idea can be extended to a large set of strands, by choosing for each strand the better reference strand out of $(ACGT)^*$ and its reverse $(TGCA)^*$. We further improve upon the cost by leveraging a second idea. After partitioning strands based on which of the two reference strands is better, we then sort the strands based on their cost (with respect to the chosen reference strand). We then use a quantile-based batching process to group the first M/k lowest cost strands, then the next M/k, etc. Combining these two ideas reduces the total cost to $2nk - \Theta(k\sqrt{n})$ for $k \geq 3$ batches.

In the case of unrestricted strands, such an improvement is not possible, although we are able to show that with k batches a similar partitioning strategy, based on appropriately ordering the strands and using quantiles, enables us to save a factor of k in the deviation term and obtain a total cost of $2.5nk + O(\sqrt{n \log M})$. We now formally state our results.

Theorem 3.1 (Upper bounds): Let $\mathcal{S}$ be a set of M random strands in $\{A, C, G, T\}^n$, and let k be an integer satisfying $3 \leq k \leq \frac{1}{4}\sqrt{\frac{M}{\log M}}$. There exist absolute constants $C_1 > 0$ and $C_2 < \infty$ such that the following hold.

1) ***(Strands without homopolymers)*** There exists a way to efficiently partition $\mathcal{S}$ into k equal size batches $\mathcal{B}_1, \ldots, \mathcal{B}_k$ such that with probability at least $1 - 1/M$ we have that
$$\sum_{i=1}^{k} \mathsf{cost}(\mathcal{B}_i) \leq 2nk - C_1 k\sqrt{n}.$$
2) ***(Unconstrained strands)*** There exists a way to efficiently partition $\mathcal{S}$ into k equal size batches $\mathcal{B}_1, \ldots, \mathcal{B}_k$ such that with probability at least $1 - 1/M$ we have that
$$\sum_{i=1}^{k} \mathsf{cost}(\mathcal{B}_i) \leq 2.5nk + C_2\sqrt{n \log M}.$$

We complement these results with almost tight lower bounds. Proving the following theorem is the most technically challenging part of our work.

Theorem 3.2 (Lower bounds): Let $\mathcal{S}$ be a set of $M \geq 10n^2 \log n$ random strands in $\{A, C, G, T\}^n$, and let k be a positive integer satisfying $k \leq \frac{1}{10}\sqrt{\log M / \log\log M}$.

1) ***(Strands without homopolymers)*** There exists an absolute constant $c_1 < \infty$ such that the following holds with probability at least $1 - c_1/M$. For any partition of $\mathcal{S}$ into k equal size batches $\mathcal{B}_1, \ldots, \mathcal{B}_k$, we have that
$$\sum_{i=1}^{k} \mathsf{cost}(\mathcal{B}_i) \geq 2nk - c_1 k\sqrt{n \log k}.$$
2) ***(Unconstrained strands)*** Suppose that $M \leq \exp(n)$. There exists an absolute constant $c_2 > 0$ such that the following holds with probability at least $1 - c_2^{-1}/M$. For any partition of $\mathcal{S}$ into k equal size batches $\mathcal{B}_1, \ldots, \mathcal{B}_k$, we have that
$$\sum_{i=1}^{k} \mathsf{cost}(\mathcal{B}_i) \geq 2.5nk + c_2\sqrt{n \log M}.$$

Comparing Theorems 3.1 and 3.2, we see that the upper and lower bounds match up to the absolute constants in the deviation term when k is small enough. As a consequence, this provides evidence that our batching method is nearly optimal, perhaps surprisingly.

Furthermore, Theorems 3.1 and 3.2 provide a clear separation between the two representative strand universes. On the one hand, for unconstrained strands we have, with probability $1 - o(1)$, that $\mathsf{cost}(\mathcal{S}) = 2.5nk + \Theta\left(\sqrt{n \log M}\right)$; that is, the cost *exceeds* the main term $2.5nk$ by the deviation term. On the other hand, for strands without homopolymers we have, with probability $1 - o(1)$, that $2nk - c_1 k\sqrt{n \log k} \leq \mathsf{cost}(\mathcal{S}) \leq 2nk - C_1 k\sqrt{n}$; that is, the cost is *smaller than* the main term $2nk$ by the deviation term.

IV. Related work

For an overview of the biochemical DNA synthesis process, we refer the interested reader to the surveys [10], [12]. Our work is motivated by several experimental papers that address the challenge of reducing the synthesis cost in both single and multi-batch settings [23]–[32]. Variants of the problem have also been studied that incorporate certain quality control measures [33]–[36]. Much of this previous work considers the $(ACGT)^*$ supersequence when analyzing the synthesis cost. Rahmann first observed that in this case the single batch cost of uniformly random strings is approximately Gaussian, but he did not provide a formal analysis nor any asymptotic or finite-size bounds [25]. In the multi-batch setting, previous work uses the same cost function as we do, namely the sum of the shortest common supersequence (SCS) lengths for each batch [27], [31]. In general, a wide array of algorithms have been proposed and empirically evaluated for selecting a short reference string given the set of DNA strands to synthesize. However, these heuristics do not come with guarantees, and many of them implicitly solve the SCS problem, which is known to be NP-hard for a collection of strings [37], [38].

From a theoretical point of view, a few recent works have considered minimizing the synthesis cost through coding-based approaches. Lenz et al. study reference strings that have a large number of subsequences, and they consider mappings to encode data by a set of strings while minimizing the single-batch synthesis cost [39]. This coded synthesis approach maximizes information density for fixed synthesis cost. However, the strands will then have additional structure (e.g., contained in a small deletion ball of $(ACGT)^*$ with many pairs close in edit distance). On the other hand, using random strings is known to be easier for clustering as the strings are far apart in edit distance [16] and for string reconstruction [21]. One avenue for future work could be to optimize for many parts of the DNA storage pipeline through codes. A different synthesis model has also been considered, storing information based on run-length patterns in the strings [40]–[42].

There is also a large body of prior work on the longest common subsequence (LCS) of random strings [43]–[49]. The expected LCS length of two random length n strings is known to be $(\gamma + o(1))n$ for a value $\gamma > 0$ called the Chvátal-Sankoff constant. Despite decades of effort, the exact value of γ remains unknown for constant alphabet sizes. For two length n strings, the LCS and SCS are related via the equality $\mathsf{SCS}(S_1, S_2) = 2n - \mathsf{LCS}(S_1, S_2)$, but for larger sets, no analogous relationship is known. In particular, our results show that the average SCS length for a large collection of strings behaves very differently than for a pair of strings. While we are not aware of prior results on the SCS for multiple batches, our single batch results improve an existing bound on the expected SCS length in the special case of $M = n$ strings (see Remark 3.4 in the full version of this paper [22]).

V. Proof Overview

In this section we give an overview of our results and the associated proofs; full proofs are in the full version [22].

Suppose we want to synthesize a DNA strand S using a reference strand R. Denote the length of the prefix of R which we use for synthesis by $\mathsf{cost}_R(S)$. Then, the cost of printing a batch of strands $\mathcal{B}$ using R equals the maximum cost of printing S for $S \in \mathcal{B}$:

$$\mathsf{cost}_R(\mathcal{B}) = \max_{S \in \mathcal{B}} \mathsf{cost}_R(S).$$

We observe that the cost of printing any strand of length n using the periodic reference strand $(ACGT)^*$ is at most $4n$, since the i-th base of S can be printed using the corresponding base in the i-th quadruple of $(ACGT)^*$. Hence, the cost of synthesizing any batch of strands of length n is bounded from above by $4n$. As we discuss later, the cost of every strand without homopolymers with respect to the reference strand $(ACGT)^*$ is at most $3n + 1$. So the cost of any batch of strands without homopolymers is also at most $3n + 1$.

Since the cost of synthesizing every batch of strands is upper bounded by $4n$, we do not need to consider reference strands of length more than $4n$. However, for the sake of analysis, we shall assume that all reference strands R have an infinite length. The first $4n$ bases of these strands are arbitrary, while the remaining infinite suffix is a repetition of the pattern $ACGT$. We denote the set of all such strands by $\mathcal{R}^*$. Observe that every strand $\mathcal{S}$ can be synthesized using every $R \in \mathcal{R}^*$ because R contains the substring $(ACGT)^*$. Note that when we synthesize a batch $\mathcal{B}$ using a reference strand $R \in \mathcal{R}^*$, we truncate R after $\mathsf{cost}_R(\mathcal{B})$ bases, so effectively we use a reference strand of length $\mathsf{cost}_R(\mathcal{B})$.

A. Cost of a Single Batch

We first show how to estimate the cost of synthesizing a single batch of DNA strands. We prove that for a random strand S of length n and reference strand $\widetilde{R} = (ACGT)^*$, the expected $\mathsf{cost}_{\widetilde{R}}(S)$ equals $2.5n$. We then use concentration inequalities to argue that the maximum cost of strands in $\mathcal{B}$ is upper bounded by $2.5n + O(\sqrt{n \log M})$ with high probability, where M is the batch size. Similarly, we show that for every fixed strand R, we have that $\mathbb{E}[\mathsf{cost}_R(S)] \geq 2.5n$. Hence, for every fixed R the cost of $\mathcal{B}$ is also lower bounded by $2.5n + \Omega(\sqrt{n \log M})$ with high probability. We obtain a lower bound on the cost of a batch by taking the union bound over all $R \in \mathcal{R}^*$. Similarly, we get lower and upper bounds of $2n + \Omega(\sqrt{n \log M})$ and $2n + O(\sqrt{n \log M})$ for random strands without homopolymers.

We now discuss how to compute $\mathbb{E}[\mathsf{cost}_R(S)]$ for a given reference strand R and random S. Let $\tau_i(S, R)$ be the cost of the prefix $S_1, \ldots, S_i$. In other words, $\tau_i(S, R)$ is the index of the base in R that is used for synthesizing the i-th base in S. We let $\tau_0(S, R) = 0$. Observe that $\{\tau_i(S, R)\}_{i \geq 0}$ is a Markov chain: the value of $\tau_{i+1}(S, R)$ depends only on the current state $\tau_i(S, R)$ and the random value of S_{i+1}. We denote the increments of $\tau_i(S, R)$ by $X_i(S, R)$: for $i \in \{1, \ldots, n\}$, let

$$X_i(S, R) := \tau_i(S, R) - \tau_{i-1}(S, R).$$

Then, $\mathsf{cost}_R(S) = \tau_n(S, R) = \sum_{i=1}^{n} X_i(S, R)$. For the reference strand $\widetilde{R} = (ACGT)^*$, each increment $X_i(S, \widetilde{R})$ is a random variable uniformly distributed in $\{1, 2, 3, 4\}$, and all $X_i(S, \widetilde{R})$ are mutually independent. Consequently, $\mathbb{E}[X_i(S, \widetilde{R})] = 2.5$ for all i and thus $\mathbb{E}[\mathsf{cost}_{\widetilde{R}}(S)] = 2.5n$. Furthermore, by the central limit theorem, the deviation of the cost from its expectation, $\mathsf{cost}_{\widetilde{R}}(S) - 2.5n$, is approximately Gaussian with mean 0 and variance $1.25n$. Thus, we can use Hoeffding's inequality and other concentration inequalities to obtain upper and lower bounds of on $\mathsf{cost}_{\widetilde{R}}(S)$. These bounds imply that the cost of a single batch of M strands equals $2.5n + \Theta(\sqrt{n \log M})$.

To show that $\mathbb{E}[X_i(S, R)] \geq 2.5$ for every $R \in \mathcal{R}^*$ and not only for $R = \widetilde{R}$, we observe that the sequence $X_1(S, R), \ldots, X_n(S, R)$ *stochastically dominates* a sequence of i.i.d random variables $Y_1, \ldots, Y_n$, where each Y_i is uniformly distributed in $\{1, 2, 3, 4\}$. Hence,

$$\mathbb{E}[X_1(S, R) + \cdots + X_n(S, R)] \geq \mathbb{E}[Y_1 + \cdots + Y_n] = 2.5n.$$

For random strands without homopolymers, each jump $X_i(S, \widetilde{R})$ is uniformly distributed in $\{1, 2, 3\}$ for $i > 1$; and $X_1(S, \widetilde{R})$ is uniformly distributed in $\{1, 2, 3, 4\}$. Hence, the expected cost $\mathsf{cost}_{\widetilde{R}}(S)$ is $2n + 1/2$. Also, note that the maximum possible value of $X_i(S, \widetilde{R})$ is 3 (for $i > 1$). Hence, the cost of every strand is upper bounded by $3n + 1$.

B. Upper Bounds for Multiple Batches

We use the same reference strand $\widetilde{R} = (ACGT)^*$ for synthesizing all batches of unconstrained strands. For synthesizing batches of strands without homopolymers, we use two different reference strands, $\widetilde{R} = (ACGT)^*$ and its reverse $\overline{R} = (TGCA)^*$,

Naïve Approach. Suppose we assign strands randomly to k batches. Then, each batch consists of M/k random strands sampled uniformly from $\{A, C, G, T\}^n$. Hence, the cost of every batch is $2.5n + \Theta(\sqrt{n \log M})$. Consequently, the total cost of synthesising k batches is $2.5nk + k \cdot \Theta(\sqrt{n \log M})$. We now show that by carefully assigning strands to batches we can improve this cost to $2.5nk + \Theta(\sqrt{n \log M})$ for unconstrained strands. Similarly, we show how to improve a naïve solution of cost $2nk + k \cdot \Theta(\sqrt{n \log M})$ for strands without homopolymers to a solution of cost $2nk - \Omega(k\sqrt{n})$.

Unconstrained Strands. Our strategy for splitting the set of unconstrained strands $\mathcal{S}$ into k batches is quite simple. For every strand S in $\mathcal{S}$, we compute $\mathsf{cost}_{\widetilde{R}}(S)$ and then sort strands by this cost. We put the first M/k strands in the first batch, the second M/k strands in the second batch, and so on. Then, the cost of the i-th batch is equal to the empirical i/k-th quantile of $\left\{\mathsf{cost}_{\widetilde{R}}(S)\right\}_{S \in \mathcal{S}}$ (see the full version [22] for formal definitions). In [22] we also show that, with high probability, empirical quantiles of $\left\{\mathsf{cost}_{\widetilde{R}}(S)\right\}_{S \in \mathcal{S}}$ are very close to the corresponding quantiles of the distribution of the random variable $\mathsf{cost}_{\widetilde{R}}(S)$, where S is randomly and uniformly drawn from $\{A, C, G, T\}^n$. The only exception is the empirical 1-quantile of the sample $\mathcal{S}$ which corresponds to the cost of the most expensive strand in $\mathcal{S}$. This cost is approximately equal

to the $(1-1/M)$-quantile of the distribution of $\mathsf{cost}_{\widetilde{R}}(S)$, where M is the size of $\mathcal{S}$.

As we discussed above, $\mathsf{cost}_{\widetilde{R}}(S)$ can be approximated by the random variable $2.5n+g$, where g is a Gaussian random variable with mean 0 and variance $1.25n$. The sum of the $1/k, 2/k, \ldots, (k-1)/k$ quantiles of a symmetric Gaussian distribution equals 0, since the quantiles i/k and $(k-i)/k$ are symmetric around 0. However, the $(1-1/M)$-quantile of the distribution of g is relatively large and approximately equals $c\sqrt{n \log M}$. Hence, the total cost of synthesizing k batches approximately equals

$$2.5nk + c\sqrt{n \log M}.$$

Strands without Homopolymers. If we use the same batching strategy as we discussed above for strands without homopolymers, we obtain a solution of cost $2nk + c\sqrt{n \log M}$ with high probability. However, somewhat surprisingly, we can do better by utilizing two reference strands, $\widetilde{R} = (ACGT)^*$, and its reverse, $\overline{R} = (TGCA)^*$, instead of just the single strand $\widetilde{R}$. We show that the random variables $\mathsf{cost}_{\widetilde{R}}(S)$ and $\mathsf{cost}_{\overline{R}}(S)$ are anticorrelated. Specifically, for every strand S without homopolymers, we (deterministically) have

$$\mathsf{cost}_{\widetilde{R}}(S) + \mathsf{cost}_{\overline{R}}(S) = 4n + 1. \tag{2}$$

Also, there is a bijection $\varphi : \{A,C,G,T\}^n \to \{A,C,G,T\}^n$ such that for every strand $S \in \{A,C,G,T\}^n$ we have that

$$\mathsf{cost}_{\widetilde{R}}(S) + \mathsf{cost}_{\widetilde{R}}(\varphi(S)) = 4n + 1. \tag{3}$$

To see this, let $\tau_i(S, \widetilde{R})$ and $\tau_i(S, \overline{R})$ be the time that the i^{th} character of S is printed using $\widetilde{R}$ or $\overline{R}$, respectively. Consider the per-character costs $X_i(S, \widetilde{R}) = \tau_i(S, \widetilde{R}) - \tau_{i-1}(S, \widetilde{R})$ and $X_i(S, \overline{R}) = \tau_i(S, \overline{R}) - \tau_{i-1}(S, \overline{R})$. Observe that $X_i(S, \widetilde{R}) + X_i(S, \overline{R}) = 4$ for $i > 1$ and $X_1(S, \widetilde{R}) + X_1(S, \overline{R}) = 5$. Hence,

$$\mathsf{cost}_{\widetilde{R}}(S) + \mathsf{cost}_{\overline{R}}(S) = \sum_{i=1}^{n} (X_i(S, \widetilde{R}) + X_i(S, \overline{R})) = 4n + 1.$$

We now map every strand S to its compliment by replacing each base A with T, C with G, G with C, and T with A. Observe that if we renamed each base as above both in S and the reference strand $\widetilde{R}$, then the cost would not change. That is, $\mathsf{cost}_{\widetilde{R}}(S) = \mathsf{cost}_{\overline{R}}(\varphi(S))$. Using (2) we thus obtain (3).

These observations suggest the following strategy: We first sort all strands $\mathcal{S}$ by their cost when printed with $\widetilde{R}$. For the first $\lceil k/2 \rceil$ batches, we print them with $\widetilde{R}$, and we print the remaining batches with $\overline{R}$. Overall, we will argue that this batching process results in $k-2$ batches having a cost of at most $2n$, and a constant fraction of these batches having an additional savings of $\Omega(\sqrt{n})$, which results in the ultimate savings of $\Omega(k\sqrt{n})$. The only challenging batches are the "middle" two. We handle these by arguing that their costs are coupled so that together they do not exceed $4n+1$. We next explain the intuition behind the main savings.

Since $(X_i(S, \widetilde{R}) + X_i(S, \overline{R}))/2 = 2$ for all $i > 1$ and S does not have homopolymers, the random variables $\mathsf{cost}_{\widetilde{R}}(S)$ and $\mathsf{cost}_{\overline{R}}(S)$ can be approximated by *correlated* random variables $2n-g$ and $2n+g$, where g is a Gaussian random variable with mean 0 and variance $2/3\, n$. The cost of every strand is thus approximately equal to $\min\{2n-g, 2n+g\} = 2n-|g|$, and the total cost of k batches is approximately equal to the sum of the i/k-quantiles of the random variable $2n-|g|$ for $i = 1, \ldots, k$. For sufficiently large k, this sum is approximately equal to

$$k \cdot (\mathbb{E}[2n - |g|]) = k \cdot (2n - \mathbb{E}[|g|]) = 2nk - k\sqrt{\frac{4}{3\pi}\, n}.$$

For small $k > 2$, the sum is upper bounded by $2nk - \Omega(k\sqrt{n})$.

C. Lower Bounds for Multiple Batches

We now discuss how to obtain lower bounds on the cost of batch synthesis. We start with lower bounds that are based on the following observation: Every batch $\mathcal{B}$ must contain a $1/k$ fraction of all strands in $\mathcal{S}$. Consequently, its cost is lower bounded by the empirical $1/k$-quantile of $\{\mathsf{cost}_R(S)\}_{S \in \mathcal{S}}$, which, in turn, approximately equals the $1/k$-quantile of the distribution of the random variable $\mathsf{cost}_R(S)$, where S is a random strand. Here R is the reference strand used for synthesising $\mathcal{B}$. Using the notation (see full version [22]) for empirical q-quantiles $\widetilde{Q}_{q,R}(\mathcal{S})$ and q-quantiles $Q_{q,R}(D)$ of a distribution D, we can lower bound the cost of $\mathcal{B}$ as follows:

$$\mathsf{cost}(\mathcal{B}) \geq \min_{R \in \mathcal{R}^*} \widetilde{Q}_{1/k,R}(\mathcal{S}) \gtrsim \min_{R \in \mathcal{R}^*} Q_{1/k,R}(D_{1/4}),$$

where $D_{1/4}$ is the uniform distribution of strands of length n. Using Hoeffding's inequality for $\mathsf{cost}_R(S)$ along with bounds on $\widetilde{Q}_{1/k,R}(\mathcal{S})$ and $Q_{1/k,R}(D_{1/4})$, we then show that $Q_{1/k,R}(D_{1/4}) \geq 2.5n - O(\sqrt{n \log k})$ which yields a lower bound of $k \cdot (2.5n - O(\sqrt{n \log k}))$ on the total cost of synthesizing k batches. For strands without homopolymers, the same argument gives a bound of $k \cdot (2n - O(\sqrt{n \log k}))$.

Improved Lower Bound for Unconstrained Strands. We then improve the lower bound on the cost of batch synthesis of unconstrained strands by showing that while the cost of all batches are lower bounded by $2.5n - O(\sqrt{n \log k})$, the cost of the most expensive batch is at least $2.5n + \Omega(\sqrt{n \log M})$. Note that a similar statement does not hold for strands without homopolymers. To prove that the cost of the most expensive batch is $2.5n + \Omega(\sqrt{n \log M})$, we consider a subset $\mathcal{S}''$ of strands that have disproportionately many (roughly, $n/4 + c\sqrt{n \log M}$) repeated bases. We show that a random set $\mathcal{S}$ contains many such strands (approximately $\sqrt{M}$) and then prove that for random strands S from $\mathcal{S}''$, the expected cost $\mathsf{cost}_R(S)$ is at least $2.5n + c\sqrt{n \log M}$. This gives us a lower bound of $2.5nk + c\sqrt{n \log M} - O(k\sqrt{n \log k})$ on the total cost of synthesising k batches (note, typically $M \gg k$).

ACKNOWLEDGMENTS

M.Z.R. was supported in part by NSF grant DMS 1811724 and by a Princeton SEAS Innovation Award. We would like to thank Karin Strauss for helpful discussions about this work.

REFERENCES

[1] M. Neiman, "On the molecular memory systems and the directed mutations," *Radiotekhnika*, pp. 1–8, 1965.

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

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

[4] S. H. T. Yazdi, H. M. Kiah, E. Garcia-Ruiz, J. Ma, H. Zhao, and O. Milenkovic, "DNA-based storage: trends and methods," *IEEE Trans. Mol. Biol. Multi-Scale Comm.*, vol. 1, no. 3, pp. 230–248, 2015.

[5] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-based archival storage system," in *Proceedings of ASPLOS*, 2016, pp. 637–649.

[6] S. L. Shipman, J. Nivala, J. D. Macklis, and G. M. Church, "CRISPR–Cas encoding of a digital movie into the genomes of a population of living bacteria," *Nature*, vol. 547, no. 7663, pp. 345–349, 2017.

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

[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] L. Organick, S. D. Ang, Y.-J. Chen, R. Lopez, S. Yekhanin, K. Makarychev, M. Z. Racz, G. Kamath, 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, no. 3, p. 242, 2018.

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

[11] L. C. Meiser, P. L. Antkowiak, J. Koch, W. D. Chen, A. X. Kohll, W. J. Stark, R. Heckel, and R. N. Grass, "Reading and writing digital data in DNA," *Nature Protocols*, vol. 15, no. 1, pp. 86–101, 2020.

[12] S. Kosuri and G. Church, "Large-scale de novo DNA synthesis: technologies and applications," *Nature Methods*, pp. 499–507, 2014.

[13] H. Lee, D. Wiegand, K. Griswold, S. Punthambaker, H. Chun, R. Kohman, and G. Church, "Photon-directed multiplexed enzymatic DNA synthesis for molecular digital data storage," Feb. 2020, available at https://www.biorxiv.org/content/10.1101/2020.02.19.956888v1.

[14] S. Chandak, J. Neu, K. Tatwawadi, J. Mardia, B. Lau, M. Kubit, R. Hulett, P. Griffin, M. Wootters, T. Weissman, and H. Ji, "Overcoming high nanopore basecaller error rates for DNA storage via basecaller-decoder integration and convolutional codes," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

[15] M. Bochman, K. Paeschke, and V. Zaikan, "DNA secondary structures: stability and function of G-quadruplex structures," *Nature Reviews Genetics*, vol. 13, pp. 770–780, 2012.

[16] C. Rashtchian, K. Makarychev, M. Rácz, S. D. Ang, D. Jevdjic, S. Yekhanin, L. Ceze, and K. Strauss, "Clustering Billions of Reads for DNA Data Storage," in *Advances in Neural Information Processing Systems*, 2017, pp. 3360–3371.

[17] T. Batu, S. Kannan, S. Khanna, and A. McGregor, "Reconstructing strings from random traces," in *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004, pp. 910–918.

[18] T. Holenstein, M. Mitzenmacher, R. Panigrahy, and U. Wieder, "Trace reconstruction with constant deletion probability and related results," in *SODA*, 2008, pp. 389–398.

[19] K. Viswanathan and R. Swaminathan, "Improved string reconstruction over insertion-deletion channels," in *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008, pp. 399–408.

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

[21] N. Holden, R. Pemantle, and Y. Peres, "Subpolynomial trace reconstruction for random strings and arbitrary deletion probability," in *Conference On Learning Theory (COLT)*, 2018, pp. 1799–1840.

[22] K. Makarychev, M. Z. Rácz, C. Rashtchian, and S. Yekhanin, "Batch Optimization for DNA Synthesis," 2020, preprint available at https://arxiv.org/abs/2011.14532.

[23] S. Hannenhalli, E. Hubbell, R. Lipshutz, and P. A. Pevzner, *Chip Technology*. Springer, 2002, vol. 77, ch. Combinatorial Algorithms for Design of DNA Arrays, pp. 1–19.

[24] A. B. Kahng, I. Măndoiu, P. A. Pevzner, S. Reda, and A. Zelikovsky, "Border length minimization in DNA array design," in *Int. Workshop on Algorithms in Bioinformatics*. Springer, 2002, pp. 435–448.

[25] S. Rahmann, "The shortest common supersequence problem in a microarray production setting," *Bioinformatics*, vol. 19, pp. 156–161, 2003.

[26] A. B. Kahng, I. I. Măndoiu, P. A. Pevzner, S. Reda, and A. Z. Zelikovsky, "Scalable heuristics for design of DNA probe arrays," *Journal of Computational Biology*, vol. 11, no. 2-3, pp. 429–447, 2004.

[27] K. Ning and H. W. Leong, "The distribution and deposition algorithm for multiple oligo nucleotide arrays," *Genome Informatics*, vol. 17, no. 2, pp. 89–99, 2006.

[28] S. Rahmann, "Subsequence combinatorics and applications to microarray production, DNA sequencing and chaining algorithms," in *Ann. Symp. on Combinatorial Pattern Matching*, 2006, pp. 153–164.

[29] A. Kumar, M. Cho, and D. Z. Pan, "DNA Microarray placement for improved performance and reliability," in *Proc. of Int. Symp. on VLSI Design, Automation and Test*. IEEE, 2010, pp. 275–278.

[30] D. Trinca and S. Rajasekaran, "Fast Local-Search-based Parallel Algorithms for DNA Probe Placement on Small Oligonucleotide Arrays," *Advanced Modeling and Optimization*, vol. 12, no. 1, pp. 45–55, 2010.

[31] K. Ning and H. W. Leong, "The multiple sequence sets: problem and heuristic algorithms," *J. Comb. Opt.*, vol. 22, no. 4, pp. 778–796, 2011.

[32] S. Srinivasan, V. Kamakoti, and A. Bhattacharya, "A Review of Algorithms for Border Length Minimization Problem," *IETE Technical Review*, vol. 31, no. 5, pp. 369–382, 2014.

[33] E. Hubbell and P. A. Pevzner, "Fidelity Probes for DNA Arrays," in *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, 1999, pp. 113–117.

[34] C. J. Colbourn, A. C. Ling, and M. Tompa, "Construction of optimal quality control for oligo arrays," *Bioinformatics*, vol. 18, no. 4, pp. 529–535, 2002.

[35] R. Sengupta and M. Tompa, "Quality control in manufacturing oligo arrays: A combinatorial design approach," *Journal of Computational Biology*, vol. 9, no. 1, pp. 1–22, 2002.

[36] O. Milenkovic, "Error and quality control coding for DNA microarrays," in *Inaugural Workshop of the Center for Information Theory and Application, San Diego*, 2006.

[37] K.-J. Räihä and E. Ukkonen, "The shortest common supersequence problem over binary alphabet is NP-complete," *Theoretical Computer Science*, vol. 16, no. 2, pp. 187–198, 1981.

[38] T. Jiang and M. Li, "On the approximation of shortest common supersequences and longest common subsequences," *SIAM Journal on Computing*, vol. 24, no. 5, pp. 1122–1139, 1995.

[39] A. Lenz, Y. Liu, C. Rashtchian, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Coding for Efficient DNA Synthesis," in *IEEE International Symposium on Information Theory (ISIT)*, 2020.

[40] 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.

[41] H. H. Lee, R. Kalhor, N. Goela, J. Bolot, and G. M. Church, "Terminator-free template-independent enzymatic DNA synthesis for digital information storage," *Nature Communications*, vol. 10, no. 1, pp. 1–12, 2019.

[42] S. Jain, F. Farnoud, M. Schwartz, and J. Bruck, "Coding for Optimized Writing Rate in DNA Storage," 2020, preprint available at https://arxiv.org/abs/2005.03248.

[43] V. Chvatal and D. Sankoff, "Longest common subsequences of two random sequences," *J. Applied Prob.*, vol. 12, pp. 306–315, 1975.

[44] V. Dančík and M. Paterson, "Upper bounds for the expected length of a longest common subsequence of two binary sequences," *Random Structures & Algorithms*, vol. 6, no. 4, pp. 449–458, 1995.

[45] G. Navarro, "A Guided Tour to Approximate String Matching," *ACM Computing Surveys*, vol. 33, no. 1, pp. 31–88, 2001.

[46] M. Kiwi, M. Loebl, and J. Matoušek, "Expected length of the longest common subsequence for large alphabets," *Advances in Mathematics*, vol. 197, no. 2, pp. 480–498, 2005.

[47] G. S. Lueker, "Improved bounds on the average length of longest common subsequences," *J. ACM*, vol. 56, no. 3, pp. 1–38, 2009.

[48] C. Houdré and H. Matzinger, "Closeness to the diagonal for longest common subsequences in random words," *Electronic Communications in Probability*, vol. 21, no. 36, pp. 1–19, 2016.

[49] B. Bukh and C. Cox, "Periodic words, common subsequences and frogs," 2019, preprint available at https://arxiv.org/abs/1912.03510.