Sketching and Sequence Alignment: A Rate-Distortion Perspective

Ilan Shomorony
University of Illinois at Urbana-Champaign
Urbana, IL, USA
ilans@illinois.edu

Govinda M. Kamath
Microsoft Research New England
Cambridge, MA, USA
gokamath@microsoft.com

***Abstract*—Pairwise alignment of DNA sequencing data is a ubiquitous task in bioinformatics and typically represents a heavy computational burden. A standard approach to speed up this task is to compute “sketches” of the DNA reads (typically via hashing-based techniques) that allow the efficient computation of pairwise alignment scores. We propose a rate-distortion framework to study the problem of computing sketches that achieve the optimal tradeoff between sketch size and alignment estimation distortion. We consider the simple setting of i.i.d. error-free sources of length n and introduce a new sketching algorithm called “locational hashing.” While standard approaches in the literature based on min-hashes require $B = (1/D) \cdot O(\log n)$ bits to achieve a distortion D, our proposed approach only requires $B = \log^2(1/D) \cdot O(1)$ bits. This can lead to significant computational savings in pairwise alignment estimation.**

I. Introduction

A key step in many bioinformatics analysis pipelines is the identification of regions of similarity between pairs of sequences (typically DNA sequencing reads). This task, known as (pairwise) sequence alignment, is a heavy computational burden, particularly in the context of sequencing data from long-read sequencing technologies such as PacBio and Oxford Nanopore. These technologies typically produce DNA reads of lengths in the tens of thousands, but at the expense of significantly higher error rates (∼10%) [1].

The optimal alignment between two sequences of length n, illustrated in Figure 1, can be found via dynamic programming in $O(n^2)$ time [2]. Hence, the time complexity for the pairwise alignment of m reads of length n is $O(m^2 n^2)$, which is impractical even for bacterial genome datasets (for which we may have $m \sim 10^5$ long reads of length $n \sim 10^4$). This challenge is commonly addressed by first converting each long DNA sequence into a short *sketch* of length s. The sketch can be thought of as low-dimensional representation of the original sequence, and is typically constructed using hashing techniques. The sketching procedure is designed in a way that estimating the alignment score between two sequences from their sketches is straightforward and can be performed in $O(s)$ time. Based on this idea, most practical sequence alignment pipelines are based on a two-step approach: they first use pairwise sketch comparisons (which take $O(m^2 s)$ time), and then perform more careful alignment only on a small set of selected pairs that are “likely” to have a significant alignment [3–8]. This approach can be cast under the more general framework of sketching algorithms [9], which are becoming increasingly popular in genomic data science [10–13].

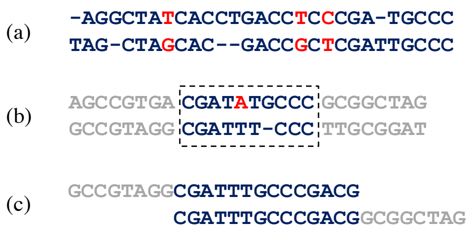

(a)
-AGGCTATCACCTGACCTCCCGA-TGCCC
TAG-CTAGCAC--GACCGCTCGATTGCCC
(b)
AGCCGTGA CGATATGCCC GCGGCTAG
GCCGTAGG CGATTT-CCC TTGCGGAT
(c)
GCCGTAGGCGATTTGCCCGACG
CGATTTGCCCGACGGCGGCTAG

Fig. 1. (a) Optimal *global* alignment between two sequences. Dashes represent indels while red symbols represent mismatches. The optimal alignment maximizes a score of the form $M\times$matches$-D\times$dashes$-S\times$mismatches [2]. (b) In a *local* alignment, only an interior segment of the sequences is required to be aligned. (c) An (error-free) overlap is a special kind of alignment where a suffix of one sequence should be aligned to a prefix of the other.

At a high level, the goal of sketching-based sequence alignment is to individually *compress* sequences down to (typically sublinear-size) sketches, from which pairwise alignments can be accurately estimated. This suggests a natural tradeoff between compression level and alignment accuracy. Motivated by this, we propose a rate-distortion framework to study this fundamental tradeoff and characterize optimal sketching.

We study the problem under a simplified setting where we have two binary length-n sequences X_1 and X_2. Moreover, we focus on a specific type of alignment – an error-free *overlap* (see Figure 1(c)) – since overlaps are the main type of alignment one is interested in in the context of genome assembly [14]. We assume that X_1 and X_2 have an overlap of size θn and consider a distributed source coding formulation, where X_1 and X_2 are compressed down to B bits each, as illustrated in Figure 2. From the B-bit sketches, we would like to estimate θ under a quadratic distortion constraint

$$E[(\theta - \hat{\theta})^2] \le D. \qquad (1)$$

We are interested in characterizing the fundamental tradeoff between the pairs $(B, D) \in \mathbb{R}_+^2$ that can be achieved in the asymptotic regime $n \to \infty$.

As we discuss in Section III, existing sketching algorithms based on min-hashes [15] can achieve a bounded distortion D with sketches of size $B = (1/D) \cdot O(\log n)$ bits. Our main result surprisingly shows that it is possible to improve over that and achieve a bounded distortion D with sketches of a

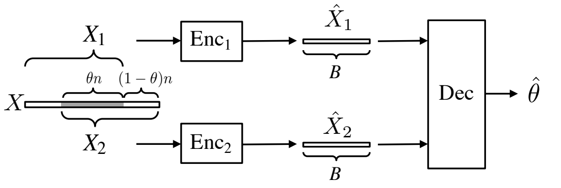


Fig. 2. The source sequences X_1 and X_2 are individually encoded into B-bit sketches, from which the overlap θ must be estimated.

constant (in n) length $B = \log^2(1/D) \cdot \Theta(1)$. To this end, we introduce a new sketching algorithm, which we term *locational hashing*. In addition, we show that a lower bound on the sketch size needed is $B = \Omega(\log(1/D))$, implying that the locational hashing scheme is not far from optimum.

Finally, we notice that, from a practical standpoint, sketching procedures with linear-time encoding and decoding are desirable. Otherwise, the two-step approach to alignment outlined above is ineffective. While all schemes discussed in this paper do have the property of linear-time encoding and decoding, we do not formally place this as a constraint for the source coding problem shown in Figure 2.

II. Problem Setting

The general distributed source coding problem [16–19] is concerned with the compression of correlated sources by encoders that do not communicate with each other. In our setting, instead of being correlated, the two length-n source sequences X_1 and X_2 have an overlap of size θn. We assume that the unknown overlap parameter satisfies $\theta \in \{0\} \cup [\theta_0, 1]$ for some fixed $\theta_0 > 0$. This is because typically in practice one is not interested in accurately estimating small overlaps, only no-overlaps or moderately sized overlaps [13].

Our goal is to design an encoder (i.e., a sketching procedure) with small encoding rates and a decoder that takes the sketches and attempts to estimate the (normalized) overlap size θ. We focus on a quadratic distortion metric, thus requiring the decoder to produce an estimator $\hat{\theta}$ satisfying (1).

For a sequence length n, the joint distribution (X_1, X_2) is defined as follows. First, a single i.i.d. $\text{Ber}(1/2)$ sequence X of length $(2-\theta)n$ is generated. In the context of DNA sequencing, X can be thought of as the underlying genome sequence. We then define $X_1 = X[1:n]$ (the first n bits of X) and $X_2 = X[(1-\theta)n+1 : (2-\theta)n]$ (the last n bits of X). Notice that the length-θn suffix of X_1 matches the length-θn prefix of X_2. In this paper, we assume that X_1 and X_2 are not corrupted by noise. Extensions to the more realistic noisy setting are briefly discussed in Section VI.

We will also define the *overlap fraction* α to be the ratio between the overlap size θn and the length of X; i.e.,

$$\alpha \triangleq \frac{\theta n}{(2-\theta)n} = \frac{\theta}{2-\theta}. \tag{2}$$

Notice that $\alpha \in (0,1)$ and θ are in one-to-one correspondence.

Unlike in standard source coding, where a length-n source X_i is compressed into nR bits, for a positive rate R, the regime of interest for our problem requires sketches of constant size in n. Our main results show that this is in fact feasible.

Definition 1. *A sketching procedure with sketch length B is a mapping* $\text{Sk} : \{0,1\}^n \to \{0,1\}^B$.

The sketching procedure is independently applied to each source, producing $\hat{X}_i = \text{Sk}(X_i)$. The decoder's task is then to estimate θ from $\hat{X}_1$ and $\hat{X}_2$.

Definition 2. *A decoder is any function* $\text{Dec} : \{0,1\}^B \times \{0,1\}^B \to (0,1)$.

The decoder is used to estimate $\hat{\theta} = \text{Dec}(\hat{X}_1, \hat{X}_2)$. We say that a rate-distortion pair (B, D) is achievable if, for sufficiently large n_0, there exists a sketching procedure for any $n \geq n_0$ with sketch length B satisfying

$$\max_{\theta} E[(\theta - \hat{\theta})^2] \leq D. \tag{3}$$

The rate-distortion function $B(D)$ (i.e., the minimum number of sketching bits needed to guarantee distortion D) is defined as $B(D) = \inf\{B : (B, D) \text{ is achievable}\}$.

Notation: Throughout we assume that $\log(\cdot)$ refers to the base-2 logarithm. For $x \in \mathbb{R}$, we let $(x)^+ = \max(x, 0)$. For a positive integer m we let $[1:m] = \{1, \ldots, m\}$. For a real number $x \in (0,1)$, we let $x = \sum_{i=1}^{\infty} x_i 2^{-i}$, $x_i \in \{0,1\}$, be its (minimal) binary expansion and for some $b \in \mathbb{Z}_+$, we let

$$[x]_b \triangleq \sum_{i=1}^{b} x_i 2^{-i}$$

be a b-bit truncation of x.

III. Min-Hash-based Sketching

As a concrete example motivated by practice, we consider the min-hash-based sketching procedure from the MHAP algorithm [4], illustrated in Figure 3. For a parameter k, we let $\Gamma(X_i)$ be the set of all k-mers of X_i; i.e., the set of all $n-k+1$ length-k substrings of X_i. For a hash function $h : \{0,1\}^k \to \mathbb{Z}$, we define the *min-hash* of X_i to be

$$h^{\min}(X_i) \triangleq \operatorname*{argmin}_{x \in \Gamma(X_i)} h(x). \tag{4}$$

A sketch can then be defined by taking m distinct hash functions $h_1, \ldots, h_m$ and concatenating the min-hashes as

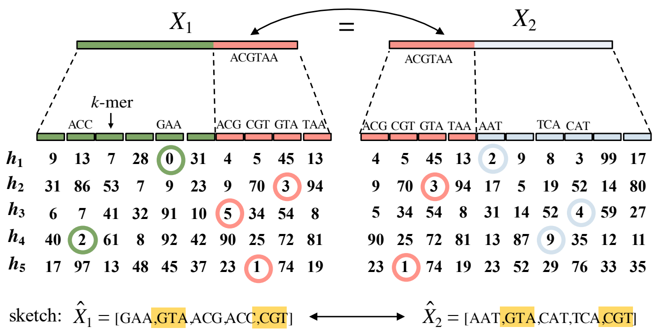
$$\text{Sk}(X_i) = [h_1^{\min}(X_i), \ldots, h_m^{\min}(X_i)]. \tag{5}$$

The key motivation behind this sketching procedure is that for a randomly selected hash function h (so that all k-mers are equally likely to be the minimizer of h), it can be shown (see, for example, [20, Chapter 3]) that

$$\Pr\left[h^{\min}(X_1) = h^{\min}(X_2)\right] = \frac{|\Gamma(X_1) \cap \Gamma(X_2)|}{|\Gamma(X_1) \cup \Gamma(X_2)|}. \tag{6}$$

The right-hand side of (6) is known as the *Jaccard similarity* between the sets $\Gamma(X_1)$ and $\Gamma(X_2)$.

Given the i.i.d. source setting from Section II, we would like to choose k so that k-mers in X are unique, so that the event



X_1
=
X_2
ACGTAA
ACGTAA
k-mer
ACC
GAA
ACG CGT GTA TAA
ACG CGT GTA TAA
AAT
TCA CAT
h_1 9 13 7 28 0 31 4 5 45 13 4 5 45 13 2 9 8 3 99 17
h_2 31 86 53 7 9 23 9 70 3 94 9 70 3 94 17 5 19 52 14 80
h_3 6 7 41 32 91 10 5 34 54 8 5 34 54 8 31 14 52 4 59 27
h_4 40 2 61 8 92 42 90 25 72 81 90 25 72 81 13 87 9 35 12 11
h_5 17 97 13 48 45 37 23 1 74 19 23 1 74 19 23 52 29 76 33 35
sketch: $\hat{X}_1$ = [GAA,GTA,ACG,ACC,CGT]
$\hat{X}_2$ = [AAT,GTA,CAT,TCA,CGT]

Fig. 3. Estimating the overlap θ between two reads using min-hash-based sketches [4] with $k = 3$. Since there are two min-hash matches, we have estimates $\hat{\alpha} = 2/5$ and $\hat{\theta} = (4/5)/(1 + 2/5) = 4/7$.

$\{h^{\min}(X_1) = h^{\min}(X_2)\}$ must be caused by a k-mer that occurs in the overlap between X_1 and X_2. It is straightforward to show (see (26)) that, if $k \geq 3 \log n$, then the probability that X contains two identical k-mers vanishes as $n \to \infty$. In fact, if we choose $k = 3 \log n$, with probability at least $1 - 1/n$,

$$\frac{|\Gamma(X_1) \cap \Gamma(X_2)|}{|\Gamma(X_1) \cup \Gamma(X_2)|} = \frac{\theta n - k + 1}{(2 - \theta)n - k + 1} \xrightarrow{n \to \infty} \alpha. \tag{7}$$

This means that by comparing the sketches $\text{Sk}(X_1)$ and $\text{Sk}(X_2)$ from (5) one entry at a time, we obtain m independent Bernoulli random variables

$$Z_j = \mathbb{1}\{h_j^{\min}(X_1) = h_j^{\min}(X_2)\}$$

for $j = 1, \ldots, m$, each having $\Pr(Z_j = 1) \to \alpha$ as $n \to \infty$. Therefore, a natural estimator for α is

$$\hat{\alpha} = \frac{1}{m} \sum_{j=1}^{m} Z_j. \tag{8}$$

An estimator for θ can also be obtained by setting $\hat{\theta}(\alpha) = \frac{2\alpha}{1+\alpha}$. From the Delta method [21, Chapter 3], we have that, as $m \to \infty$, $\text{Var}(\hat{\theta}(\hat{\alpha})) = (\hat{\theta}'(\hat{\alpha}))^2 \text{Var}(\hat{\alpha}) \leq \text{Var}(\hat{\alpha})[\sup_{\alpha \in [0,1]} (\hat{\theta}'(\alpha))^2] = 4\text{Var}(\hat{\alpha})$. Since $\text{Var}(\hat{\alpha}) = \frac{\alpha(1-\alpha)}{m} \leq \frac{1}{4m}$, we have that, for large m, $D \approx \text{Var}(\hat{\theta}) \leq \frac{1}{m}$. We conclude that, to achieve a distortion D using the sketching procedure in (5), we need

$$B = m(2 \log n) \leq \frac{2 \log n}{D} \tag{9}$$

bits. Hence, while this scheme achieves a finite distortion (which can be made arbitrarily small by increasing B), the sketch size B is logarithmic in n.

We conclude this section by pointing out that one could in principle modify the scheme described above to work with $k = O(1)$, but it is not straightforward to analyze the scheme in the presence of repeated k-mers. Moreover, we point out that the min-hash defined in (4) is more commonly defined as returning the value of the minimum hash itself instead of the minimizing k-mers (i.e., $h(x)$ instead of x). However, in order to guarantee that all k-mers have a different hash value, the number of bits required per min-hash is at least k, so this alternative definition does not affect the analysis above.

IV. Main Results

Our main contribution is a new sketching procedure that significantly improves over the required number of bits B in (9). Combined with a simple lower bound on the minimum sketch size, we have the following main result.

Theorem 1. *The minimum sketch size B required to achieve achieve distortion D satisfies*

$$\frac{1}{8} \log\left(\frac{1 - \theta_0}{D}\right) \leq B(D) \leq \frac{24^2}{\alpha_0^2} \ln^2\left(\frac{3}{D}\right) \tag{10}$$

for any D small enough.

We first introduce the new sketching technique and then analyze its performance.

A. Locational Hashing

Our sketching procedure is motivated by the min-hash approach from Section III. But instead of recording the minimizer k-mers of certain hash functions, we record the approximate *location* of special substrings in the sequence X_i.

We start by introducing a simple construction that stores only one locational hash, and then improve upon it. Suppose we have a sequence X_1 for which we want to compute a sketch $\hat{X}_1$. We start by lexicographically sorting the set

$$\mathcal{S} = \{X_1[m : n] \cdot \$: 1 \leq m \leq n\}$$

of all suffixes of X_1. To make the sorting operation well-defined, we append to each suffix $X_1[m : n]$ a character \$, which is assumed to be the last character in the lexicographic order of the alphabet. We then take the lexicographically first string from $\mathcal{S}$, and its corresponding location m_1. Notice that, for the case of binary strings, this essentially corresponds to taking the starting point of the longest run of zeros in X_1 (but the lexicographic approach extends to larger alphabets).

Our sketch will be constructed by storing the first B bits of the binary expansion of m_1/n; i.e.,

$$\hat{X}_1 = [m_1/n]_B.$$

Notice that the difference between $[m_1/n]_B$ and m_1/n is at most 2^{-B}. Let m_2 and $\hat{X}_2$ be the counterparts of m_1 and $\hat{X}_1$ for X_2. The decoder can then estimate the overlap θ from the sketches $\hat{X}_1$ and $\hat{X}_2$ by simply setting

$$\hat{\theta} = 1 - (\hat{X}_1 - \hat{X}_2)^+. \tag{11}$$

To analyze the performance of this estimator, consider the original length-$(2 - \theta)n$ sequence X, from which X_1 and X_2 were generated. Notice that, for large n, the location of the lexicographically first suffix of X is essentially uniformly distributed over all possible starting locations. Hence the probability that the lexicographically first suffix of X falls in the overlap between X_1 and X_2 approaches

$$\frac{\theta n}{(2 - \theta)n} = \alpha$$

as $n \to \infty$. This implies that, with probability approaching α, the lexicographically first suffix of X_1 and X_2 corresponds to the same location in X, and we have

$$m_1 = m_2 + (1 - \theta)n. \tag{12}$$

Thus, with a probability approaching α, the estimator satisfies

$$|\theta - \hat{\theta}| \le 2 \cdot 2^{-B} = 2^{-(B-1)}.$$

Whenever (12) is not satisfied (which occurs with probability approaching $1 - \alpha$ as $n \to \infty$), the absolute error of the estimator is at most 1, which implies that

$$E[(\theta - \hat{\theta})^2] \le \alpha 2^{-2(B-1)} + (1 - \alpha). \tag{13}$$

We see that the locational hashing approach to sketching allows us to obtain a term in the distortion bound (13) of the form 2^{-B}. However, because the estimator only "works" with probability α, our achieved distortion cannot go below $1-\alpha$. In the next subsection, we fix this issue by repeating the locational hashing procedure for distinct lexicographic orders.

B. Improved Locational Hashing

In order to improve on the performance of the sketching procedure in Section IV-A, we consider using different lexicographic orders to compare strings (while maintaining \$ as the last character in the lexicographic order). We do this by picking an ordering of 0 and 1 at each position uniformly at random, independent across positions. In other words, for position i of the suffix, $1 > 0 > \$$ with probability 0.5 and $0 > 1 > \$$ with probability 0.5. Using multiple lexicographic orderings thus generated, we order the suffixes obtained from X_1 and X_2, as shown in Figure 4. We note that with any ordering generated, the probability that the lexicographically first suffixes of X_1 and X_2 occur in their overlap is the same and approaches α as $n \to \infty$.

In order to build the improved sketch, we consider two parameters, u and v. We then generate u different lexicographic orderings to compute u different starting locations $m_{i,1}, \ldots, m_{i,u}$ for the lexicographically first suffix of X_i. We build our sketch of X_i by taking the v-bit truncation of $m_{i,j}/n$, for $j = 1, \ldots, u$; i.e.,

$$\text{Sk}(X_i) = \left([m_{i,1}/n]_v, \ldots, [m_{i,u}/n]_v\right). \tag{14}$$

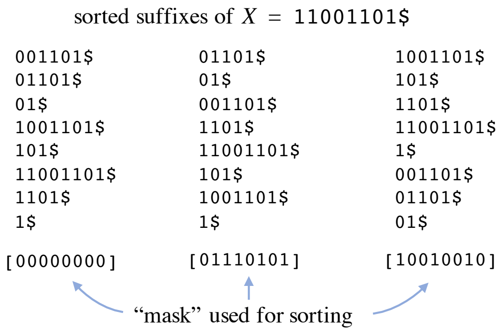


Fig. 4. Lexicographic sorting of the suffixes of a length-8 sequence X for three different lexicographic orderings. The mask is a binary sequence whose ith entry indicates whether, at the ith entry, 0 or 1 should be considered to be lexicographically first. \$ is always the lexicographically last character.

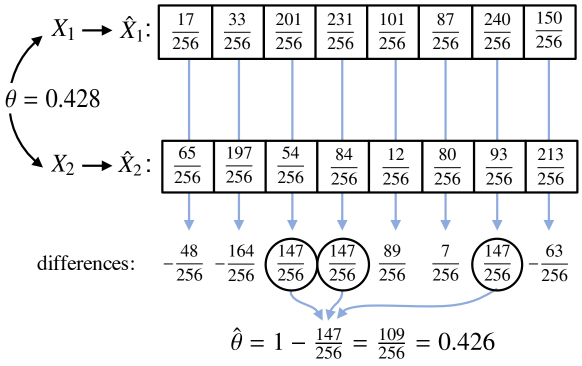

$X_1 \to \hat{X}_1$:
$\frac{17}{256}$ $\frac{33}{256}$ $\frac{201}{256}$ $\frac{231}{256}$ $\frac{101}{256}$ $\frac{87}{256}$ $\frac{240}{256}$ $\frac{150}{256}$
$\theta = 0.428$
$X_2 \to \hat{X}_2$:
$\frac{65}{256}$ $\frac{197}{256}$ $\frac{54}{256}$ $\frac{84}{256}$ $\frac{12}{256}$ $\frac{80}{256}$ $\frac{93}{256}$ $\frac{213}{256}$
differences:
$-\frac{48}{256}$ $-\frac{164}{256}$ $\frac{147}{256}$ $\frac{147}{256}$ $\frac{89}{256}$ $\frac{7}{256}$ $\frac{147}{256}$ $-\frac{63}{256}$
$\hat{\theta} = 1 - \frac{147}{256} = \frac{109}{256} = 0.426$

Fig. 5. Illustration of sketching and decoding scheme for $u = v = 8$. Each sketch $\hat{X}_i$ comprises 8 numbers $[m_{i,j}]_8$, $j = 1, ..., u$, each with 8-bit precision. Each of these numbers corresponds to the location of the lexicographically first suffix of X_i, each using a different lexicographic ordering. The decoder takes the difference between pairs of corresponding entries of $\hat{X}_1$ and $\hat{X}_2$, and then finds the mode of this set (147/256). The estimate of θ is then $\hat{\theta} = 1 - 147/256$. The sketch size is $B = 64$ bits.

This is illustrated in Figure 5. Notice that this sketching procedure requires $B = uv$ bits. For $j \in [1 : u]$,

$$1 - \left([m_{1,j}/n]_v - [m_{2,j}/n]_v\right)^+$$

is an estimator for θ with the same performance as (13). Moreover, in expectation, αu of them will satisfy

$$m_{1,j} - m_{2,j} = (1 - \theta)n \tag{15}$$

and thus

$$[m_{1,j}/n]_v - [m_{2,j}/n]_v = (1 - \theta) + \delta_j, \tag{16}$$

where δ_j is an error term due to the finite-bit representation, and satisfies $|\delta_j| \le 2 \cdot 2^{-v}$.

Notice that, intuitively, for the remaining $(1-\alpha)u$ indices j, the value of $(m_{1,j} - m_{2,j})/n$ will be distributed in $(0, 1)$ and is not expected to concentrate on any particular value. Hence, an intuitive estimator for θ is obtained by letting a be the most frequent value (i.e., the mode) in the multiset

$$\mathcal{A} = \{[m_{1,j}/n]_v - [m_{2,j}/n]_v : 1 \le j \le u\},$$

and then setting $\hat{\theta} = 1 - a$ if $a \ge 0$ and $\hat{\theta} = 0$ if $a < 0$. This procedure is illustrated in Figure 5. We make one final modification to this estimator. If the mode of $\mathcal{A}$ appears less than $\alpha_0 u/6$ times, where $\alpha_0 \triangleq \theta_0/(2 - \theta_0)$, then we simply output $\hat{\theta} = 0$. In the next section, we show that the resulting distortion satisfies

$$E[(\theta - \hat{\theta})^2] \le 2^{-2(v-1)} + (u + 1)e^{-2u(\frac{\alpha_0}{12} - 2^{-(v-1)})^2}. \tag{17}$$

It is easy to see that the expected distortion achieved with sketches of length $B = uv$ is significantly better than what is achieved with a single locational hash (13) and with the min-hash-based approach (9).

V. Proof of Main Result

In this section we provide the key steps to prove the achievability of Theorem 1. Proofs of lemmas and the proof of the converse result are relegated to a longer version of this manuscript [22]. The key step in the analysis of our sketching procedure and its decoder is obtaining, for the jth lexicographic ordering, $j \in [1:u]$, bounds for the pmf

$$p_j(k) \triangleq \Pr\left([m_{1,j}/n]_v - [m_{2,j}/n]_v = k\,2^{-v}\right).$$

Intuitively, this pmf has a probability mass of α near $k \approx (1-\theta)2^v$, and no other probability mass anywhere. This is made precise in the following lemma.

Lemma 1. *If $\mathcal{K} = \{k : \ |k\,2^{-v} - (1-\theta)| \le 2^{-(v-1)}\}$, then*

$$\sum_{k \in \mathcal{K}} p_j(k) \ge \alpha - \epsilon_n, \tag{18}$$

where $\lim_{n\to\infty} \epsilon_n = 0$. Moreover,

$$\max\{p_j(k) : k \notin \mathcal{K}\} \le 2^{-(v-1)}. \tag{19}$$

Given the lemma, we can bound the expected distortion as follows. First, we note that the set $\mathcal{K}$ from Lemma 1 has cardinality at most 5 (as we work with a resolution of 2^{-v} and are considering an interval of lenght 4×2^{-v}). Hence, for n large enough, there is at least one value of $k \in \mathcal{K}$ with $p_j(k) \ge \alpha/6$. Notice that, since the u different lexicographic orderings are chosen independently at random, $(m_{1,j}, m_{2,j})$ is independent of $(m_{1,\ell}, m_{2,\ell})$ for $j \ne \ell$. Hence, the random variables $[m_{1,j}/n]_v - [m_{2,j}/n]_v$, for $1 \le j \le u$, are independent (and identically distributed).

We bound the distortion by considering two separate cases. If $\alpha = 0$ (which is equivalent to $\theta = 0$), then the estimator will be $\hat{\theta} = 0$ as long as no element of $\mathcal{A}$ appears more than $\alpha_0 u/6$ times, and the distortion will be zero. If this condition is violated for some element $k2^{-v}$ with $k \in \mathcal{K}$, then we incur a distortion of at most $(2^{-(v-1)})^2$. If this condition is violated for some $k2^{-v}$ with $k \notin \mathcal{K}$, then the distortion can be at most 1, but this happens with low probability, as shown in the following lemma (which holds regardless of whether $\alpha = 0$).

Lemma 2. *The probability that there is a value $k2^{-v}$ for $k \notin \mathcal{K}$ that appears more than $\alpha_0 u/12$ times in $\mathcal{A}$ is at most*

$$\exp\left[-2u(\alpha_0/12 - u2^{-(v-1)})^2\right], \tag{20}$$

provided that $\alpha_0/12 > u2^{-(v-1)}$.

Hence, when $\alpha = 0$ the expected distortion satisfies

$$E[(\theta - \hat{\theta})^2] \le 2^{-2(v-1)} + e^{-2u(\frac{\alpha_0}{12} - u2^{-(v-1)})^2}. \tag{21}$$

Next we consider the case $\alpha \ge \alpha_0$. Notice that in this case Lemma 2 still holds. Moreover, the probability that, for no $k \in \mathcal{K}$, $k2^{-v}$ appears at least $(\alpha_0/12)u$ times in $\mathcal{A}$ is at most

$$\exp\left(-2u(\alpha/6 - \alpha_0/12)^2\right) \le \exp\left(-2u(\alpha_0/12)^2\right). \tag{22}$$

We conclude that, if $\alpha_0/12 > u2^{-(v-1)}$, the expected distortion in the case $\alpha \ge \alpha_0$ can bounded as

$$\begin{aligned} E[(\theta - \hat{\theta})^2] &\le 2^{-2(v-1)} + e^{-2u(\frac{\alpha_0}{12} - u2^{-(v-1)})^2} + e^{-2u(\frac{\alpha_0}{12})^2} \\ &\le 2^{-2(v-1)} + 2e^{-2u(\frac{\alpha_0}{12} - u2^{-(v-1)})^2}. \end{aligned} \tag{23}$$

Since the right-hand side of (23) is greater than the right-hand side of (21), we can take (23) to be a bound for both cases.

Finally, by setting $v = \alpha_0\sqrt{B}/24$ $u = B/v$, we obtain

$$E[(\theta - \hat{\theta})^2] \le 3\exp\left(-\sqrt{B}\alpha_0/24\right), \tag{24}$$

which holds for B large enough. This implies that $\ln(3/D) \ge \alpha_0\sqrt{B}/24$, which yields the bound in Theorem 1. Since this bound only holds for B large enough, our bound on $B(D)$ only holds for small enough D.

VI. Discussion

We introduced a distributed source coding framework to study the problem of sketching sequences for pairwise alignment. This problem is practically relevant as sketching can lead to significant computational savings in the pairwise alignment of a large number of sequences.

As a starting point, we considered the case where we have two binary sequences with an error-free overlap. We showed that, for a desired distortion bound D, it is possible to compress the strings into B-bit sketches with $B = \log^2(1/D)O(1)$. Since our encoder produces a sketch with a time complexity that is linear in n, and our decoder produces an estimator $\hat{\theta}$ with a time complexity that is linear in the sketch size, the pairwise overlapping of a large number m of length-n sequences can be performed in time $O(mn) + O(m^2)\log^2(1/D)$, while existing methods would require $O(mn) + O(m^2)(1/D)$ for the same distortion.

However, it is important to point out that the absence of noise in our model makes the approach and analysis here presented not directly practical. In the context of genomics, all DNA sequencing technologies have some error rate. If the lexicographically first suffix of X occurs in the overlap between X_1 and X_2, but happens to be corrupted by noise in X_1 (but not in X_2) it is likely to no longer be the lexicographically first suffix in X_1. Notice that the lexicographically first suffix is expected to be determined by the first $\approx 2\log n$ bits of each suffix. Hence, as $n \to \infty$, the scheme we presented is only expected to work if the error rate p decays as $p = O(1/\log n)$. In the presence of a constant error rate $p = O(1)$, even min-hash-based schemes cannot provide any meaningful distortion guarantees. Characterizing the rate-distortion function $B(D)$ in this case is an interesting direction for future work.

We also point out that, depending on the application, one may be interested in identifying local alignments beyond just overlaps (e.g., Figure 1(b)). In these situations, the scheme we presented produces an estimate of the "shift" between the two sequences that leads to the largest match, rather than an estimate of the size of the match. Modifying our scheme for the case of general overlaps is another follow-up direction.

References

[1] J. L. Weirather, M. de Cesare, Y. Wang, P. Piazza, V. Sebastiano, X.-J. Wang, D. Buck, and K. F. Au, "Comprehensive comparison of pacific biosciences and oxford nanopore technologies and their applications to transcriptome analysis," *F1000Research*, vol. 6, 2017.

[2] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[3] G. Myers, "Efficient local alignment discovery amongst noisy long reads," in *International Workshop on Algorithms in Bioinformatics*, pp. 52–67, Springer, 2014.

[4] K. Berlin, S. Koren, C.-S. Chin, J. P. Drake, J. M. Landolin, and A. M. Phillippy, "Assembling large genomes with single-molecule sequencing and locality-sensitive hashing," *Nature biotechnology*, vol. 33, no. 6, p. 623, 2015.

[5] H. Li, "Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences," *Bioinformatics*, vol. 32, no. 14, pp. 2103–2110, 2016.

[6] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 34, no. 18, pp. 3094–3100, 2018.

[7] B. D. Ondov, T. J. Treangen, P. Melsted, A. B. Mallonee, N. H. Bergman, S. Koren, and A. M. Phillippy, "Mash: fast genome and metagenome distance estimation using minhash," *Genome biology*, vol. 17, no. 1, pp. 1–14, 2016.

[8] C. Jain, A. Dilthey, S. Koren, S. Aluru, and A. M. Phillippy, "A fast approximate algorithm for mapping long reads to large reference databases," in *International Conference on Research in Computational Molecular Biology*, pp. 66–81, Springer, 2017.

[9] P. Indyk, "Sketching, streaming and sublinear-space algorithms," *Graduate course notes, available at*, vol. 33, p. 617, 2007.

[10] W. P. Rowe, "When the levee breaks: a practical guide to sketching algorithms for processing the flood of genomic data," *Genome biology*, vol. 20, no. 1, p. 199, 2019.

[11] G. Marçais, B. Solomon, R. Patro, and C. Kingsford, "Sketching and sublinear data structures in genomics," *Annual Review of Biomedical Data Science*, 2019.

[12] T. Baharav, G. M. Kamath, D. N. Tse, and I. Shomorony, "Spectral Jaccard similarity: A new approach to estimating pairwise sequence alignments," *Patterns*, vol. 1, no. 6, 2020.

[13] G. M. Kamath, T. Z. Baharav, and I. Shomorony, "Adaptive learning of rank-one models for efficient pairwise sequence alignment," *NeurIPS*, 2020.

[14] G. M. Kamath, I. Shomorony, F. Xia, T. A. Courtade, and D. N. Tse, "Hinge: long-read assembly achieves optimal repeat resolution," *Genome research*, vol. 27, no. 5, pp. 747–756, 2017.

[15] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig, "Syntactic clustering of the web," *Computer networks and ISDN systems*, vol. 29, no. 8-13, pp. 1157–1166, 1997.

[16] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[17] R. Ahlswede and J. Körner, "Source coding with side information and a converse for degraded broadcast channels," *IEEE Trans. on Information Theory*, vol. 21, pp. 629–637, Nov. 1975.

[18] A. Wyner, "On source coding with side information at the decoder," *IEEE Trans. on Information Theory*, vol. 21, pp. 294–300, May 1975.

[19] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Information Theory*, vol. 22, pp. 1–10, Jan. 1976.

[20] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge university press, 2020.

[21] A. W. Van der Vaart, *Asymptotic statistics*, vol. 3. Cambridge university press, 2000.

[22] I. Shomorony and G. Kamath, "Sketching and sequence alignment: A rate-distortion perspective," *available online: ilanshom.github.io/papers/sketching.pdf*, 2021.

Appendix

A. Proof of Lemma 1

To prove (18), we follow similar steps to those in Section IV-A, but more formally. Let $\mathcal{R}$ be the event that the underlying sequence X has a repeat of length $k = 3\log n$, and $\bar{\mathcal{R}}$ its complement. Then $\Pr(\mathcal{R})$ is at most

$$\sum_{i,j\in[1:n-k+1]} \Pr\left(X[i:i+k-1] = X[j:j+k-1]\right) \tag{25}$$

$$\leq n^2\, 2^{-k} = 1/n. \tag{26}$$

Hence, with probability $1-1/n$, the lexicographically first suffix of X will be determined by the first $3\log n$ bits of all suffixes. Hence, if the lexicographically first suffix of X starts inside $X[(1-\theta)n+1 : n-3\log n]$, it will be completely contained in the overlap between X_1 and X_2, and (15) will hold. Therefore, as $n\to\infty$, Therefore, with probability approaching α, (16) holds, which in turn implies that

$$[m_{1,j}/n]_v - [m_{2,j}/n]_v = k2^{-v} \tag{27}$$

for some $k\in\mathcal{K}$. This proves (18).

To prove (19), we define three segments of X, $S_1 = [1:(1-\theta)n - 3\log n]$, $S_{12} = [(1-\theta)n : n-3\log n]$, and $S_2 = [n:(2-\theta)n - 3\log n]$. Notice that, as $n\to\infty$, we must have $m_{1,j}\in S_1\cup S_{12}$ and $m_{2,j}\in S_2\cup S_{12}$ with probability tending to 1. If $m_{1,j}$ and $m_{2,j}$ are both in S_{12}, they must correspond to the same suffix of X, and (27) will hold for $k\in\mathcal{K}$. Conditioned on $\mathcal{E}\triangleq\bar{\mathcal{R}}\cap\{m_{1,j}\in S_1\}\cap\{m_{2,j}\in S_2\}$, $m_{1,j}$ and $m_{2,j}$ are uniformly distributed in their intervals. Hence,

$$\Pr\left([m_{1,j}/n]_v - [m_{2,j}/n]_v = k2^{-v}\middle|\mathcal{E}\right)$$
$$\leq \max_{\ell}\Pr\left([m_{1,j}/n]_v = \ell 2^{-v}\middle|\mathcal{E}\right) \leq \frac{2^{-v}}{1-\theta}.$$

Similarly, conditioning on the events $\mathcal{E}'\triangleq\bar{\mathcal{R}}\cap\{m_{1,j}\in S_1\}\cap\{m_{2,j}\in S_{12}\}$ and $\mathcal{E}''\triangleq\bar{\mathcal{R}}\cap\{m_{1,j}\in S_{12}\}\cap\{m_{2,j}\in S_2\}$, we see that the conditional pmf is always upper bounded by $2^{-v}/(1-\theta)$. Since $\Pr(\mathcal{E}\cup\mathcal{E}'\cup\mathcal{E}'')\to(1-\alpha)$ as $n\to\infty$, we conclude that

$$\max\{p_j(k) : k\notin\mathcal{K}\} \leq \left(\frac{1-\alpha}{1-\theta}\right)2^{-v} = \left(\frac{2}{2-\theta}\right)2^{-v},$$

and since $\theta\in(0,1)$, (19) is proved.

B. Lower Bound on Sketch Size

In this section we provide the converse result to establish the lower bound in Theorem 1; i.e., $B\geq\frac{1}{8}\log\left(\frac{1-\theta_0}{D}\right)$. First we observe that, by combining both sketches we have a total of $2B$ bits, which implies that the output of any decoder function $\mathrm{Dec}(\cdot)$ can take at most 2^{2B} values in $(0,1)$. Hence, for any decoder, there exists $\theta\in\{0\}\cup[\theta_0,1]$ that is at a distance $(1-\theta_0)2^{-4B}$ from every value in the range of $\mathrm{Dec}(\cdot)$. Hence,

$$\max_\theta E[(\theta-\hat\theta)^2] \geq (1-\theta_0)2^{-8B}.$$

This, in turn, implies that $D\geq(1-\theta_0)2^{-8B}$.

C. Proof of Lemma 2

For simplicity we consider two strings X_1 and X_2 with $\alpha=0$. We go through the u length sketch vectors $\mathrm{Sk}(X_1)-\mathrm{Sk}(X_2)$ in sequence.

We note that $\mathrm{Sk}(X_1)-\mathrm{Sk}(X_2)\in\{-2^v,\cdots 0,\cdots 2^v-1\}$ Further we define $\mathcal{U}_i$ to be set of the most seen values in $\{0,\cdots 2^v-1\}$ in the first i components of $\mathrm{Sk}(X_1)-\mathrm{Sk}(X_2)$. In case of ties, we pick the numerically smaller value in $\mathcal{U}_i$. Similarly, we define $\mathcal{U}_0$.

We next define random variables Γ_i, $1\leq i\leq u$ as follows

$$\Gamma_i = \begin{cases} 1 & (\mathrm{Sk}(X_1)-\mathrm{Sk}(X_2))_i \in U_{i-1}, \\ 0 & \text{otherwise}\end{cases} \tag{28}$$

We define $\Gamma_0=0$.

We clearly have that $\mathbb{E}[\Gamma_i|\Gamma_0,\cdots\Gamma_{i-1}] = u2^{-v}$, for $i\in\{1,\cdots,n\}$

We define $\tilde{\Gamma}_i$ as,

$$\tilde{\Gamma}_0 = 0 \text{ w.p. } 1, \tag{29}$$

$$\tilde{\Gamma}_i = \Gamma_i - u2^{-v},\ 1 \leq i \leq u. \tag{30}$$

Clearly $\tilde{\Gamma}_i$ is a martingale, with $|\tilde{\Gamma}_i| < 1$, allowing us to use Azuma-Hoeffding's inequality.

Further we have that $(\mathrm{Sk}(X_1) - \mathrm{Sk}(X_2))_i = j$ for some j, then $j \in \mathcal{U}_k, k \geq i$, because we at most see u distinct items in the difference of sketches. Let F be the maximum number of times we see an item in $\mathrm{Sk}(X_1) - \mathrm{Sk}(X_2)$. We thus have that $\sum_{i=1}^{u} \Gamma_i \geq F - 1$.

Hence we have that,

$$\Pr\left[F \geq \frac{\alpha_0 u}{12}\right] \leq \Pr\left[\sum_{i=1}^{u} \Gamma_i \geq \frac{\alpha_0 u}{12} + 1\right], \tag{31}$$

$$\leq \Pr\left[\sum_{i=1}^{u} \Gamma_i \geq \frac{\alpha_0 u}{12}\right], \tag{32}$$

$$= \Pr\left[\sum_{i=1}^{u} \Gamma_i - u2^{-v} \geq \frac{\alpha_0 u}{12} - u^2 2^{-v}\right], \tag{33}$$

$$= \Pr\left[\sum_{i=1}^{u} \tilde{\Gamma}_i \geq \frac{\alpha_0 u}{12} - u^2 2^{-v}\right], \tag{34}$$

$$= \Pr\left[\frac{1}{u}\sum_{i=1}^{u} \tilde{\Gamma}_i \geq \frac{\alpha_0}{12} - u2^{-v}\right], \tag{35}$$

$$\overset{(i)}{\leq} \exp\left(-2u\left(\frac{\alpha_0}{12} - u2^{-v}\right)^2\right), \tag{36}$$

$$\tag{37}$$

where (i) follows from Azuma-Hoeffding's inequality.