Partial DNA Assembly: A Rate-Distortion Perspective

Ilan Shomorony[1], Govinda M. Kamath[2], Fei Xia[3], Thomas A. Courtade[1], and David N. Tse[2]
[1] University of California, Berkeley, USA, [2] Stanford University, Stanford, USA, [3] Tsinghua University, China.
Email: ilan.shomorony@berkeley.edu, gkamath@stanford.edu, xf12@mails.tsinghua.edu.cn,
courtade@berkeley.edu, dntse@stanford.edu.

Abstract—**Earlier formulations of the DNA assembly problem were all in the context of perfect assembly; i.e., given a set of reads from a long genome sequence, is it possible to perfectly reconstruct the original sequence? In practice, however, it is very often the case that the read data is not sufficiently rich to permit unambiguous reconstruction of the original sequence. While a natural generalization of the perfect assembly formulation to these cases would be to consider a rate-distortion framework, partial assemblies are usually represented in terms of an *assembly graph*, making the definition of a distortion measure challenging. In this work, we introduce a distortion function for assembly graphs that can be understood as the logarithm of the number of Eulerian cycles in the assembly graph, each of which correspond to a candidate assembly that could have generated the observed reads. We also introduce an algorithm for the construction of an assembly graph and analyze its performance on real genomes.**

I. Introduction

The cost of DNA sequencing has been falling at a rate exceeding Moore's law. The dominant technology, called *shotgun sequencing* involves obtaining a large number of fragments called reads from random locations on the DNA sequence. This technology comes in two flavors:

(a) *Short-read technologies*, which generate reads typically shorter than 200 base pairs (bp), with error rates around 1%, with substitutions being the primary form of errors.

(b) *Long-read technologies*, which generate reads of length around $10,000$ bp, with error rates around 15%, with insertions and deletions being the primary form of errors.

The reads obtained from either technology are then merged to each other based on regions of overlap using an *assembly algorithm* to obtain an estimate of the DNA sequence. During the last decade, several such algorithms were developed first aimed at short-read sequencing technologies and, more recently, focused on long-read technologies. While these approaches attained varied degrees of success in the assembly of many genomes, very few of them are known to provide any kind of performance guarantee.

A theoretical framework to assess the performance of various algorithms relative to the *fundamental limits* for DNA assembly was proposed in [1]. However, this framework focuses on *perfect assembly*; i.e., when the goal is to reconstruct the whole genome perfectly. In particular, a critical read length ℓ_{crit}, defined as a function of the repeat patterns of a given genome [1, 2], is proved to be a fundamental lower bound for perfect assembly, and shown to be achievable by their proposed algorithm. Nonetheless, for real genomes, ℓ_{crit} can be very large, and read data for many practical DNA sequencing projects does not meet the information-theoretic lower bounds from [1, 2], rendering the task of perfect assembly fundamentally impossible. This makes the derivation of a theoretic framework to compare algorithms in terms of partial assembly of paramount importance.

The first step in constructing a theoretical framework for partial assembly is to select an appropriate measure of the quality of a partial assembly. Measuring the quality of partial assemblies is a challenging problem. In practice, a metric that is often used is the N50. To describe N50, recall that a *contig* is an unambiguous sequence in a genome that an assembly returns. The N50 of an assembly is then defined as the largest length ℓ such that the sum of the lengths of contigs at least ℓ long accounts for at least half of the sum of the lengths of all contigs returned. This is practically a very popular metric, because it does not require knowledge of the ground truth genome (which is usually not known in practice) to compute. However, the fact that N50 does not depend upon the ground truth genome makes it mathematically unsatisfying. For example, an algorithm could just output a random string over $\Sigma = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ of length 1 trillion and obtain an N50 of 1 trillion, despite the fact that the output is unrelated to the target genome. Another discouraging aspect of N50 is that it cannot capture what is known about the relative position between the contigs. In particular, contigs are typically extracted from an *assembly graph*, which is basically a graph with the contigs as vertices, and an edge from contig u to contig v if contig v comes after one copy of contig u. N50 does not account for any of the structural information contained in such a graph.

In this manuscript, we introduce a distortion metric that attempts to capture how good an assembly graph is. Roughly speaking, this measure coincides with the logarithm of the number of Eulerian cycles in the assembly graph. Intuitively, every Eulerian cycle corresponds to a distinct assembly and all of them explain the data equally well; as such, the distortion represents the missing information still needed for perfect assembly. To the best of our knowledge, this is the first work in this direction.

With the yardstick defined, we then seek an assembly algorithm whose performance can be characterized in terms of the proposed distortion measure. While *de Bruijn* graph-based algorithms [3] are better understood from a theoretical standpoint [1, 3], and would constitute better candidates for the distortion analysis, they are not very relevant in the context of assembly from long-read technologies. This is due to their high sensitivity to read errors, which prevents them from leveraging the potential of long-read technologies (all of which have error rates above 10% and will continue to have for the foreseeable future [4, Sections $1, 3$]). In contrast, overlap-based assembly approaches (e.g., string graphs [5]) are better suited to long-read, high-error sequencing. This class of algorithms relies on the identification of long overlaps between reads, which is inherently robust to errors [4], and has been recently shown to attain the same theoretical performance as de Bruijn graph-based approaches in terms of perfect assembly [6]. In this

978-1-5090-1805-5/16/$31.00 ©2016 IEEE

work, we propose a new overlap-based assembly algorithm and introduce techniques to provide theoretical guarantees in terms of the new distortion measure.

II. Problem Setting

Let $\mathbf{x}$ be a string of ℓ symbols from the alphabet $\Sigma = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$. We let $|\mathbf{x}| = n$ be the length of the string, and $\mathbf{x}[i]$ be its ith symbol. A substring of $\mathbf{x}$ is a contiguous interval of the symbols in $\mathbf{x}$, and is denoted as $\mathbf{x}[i:j] \triangleq (\mathbf{x}[i], \mathbf{x}[i+1], ..., \mathbf{x}[j])$. A substring of the form $\mathbf{x}[1:\ell]$ is called a prefix (or an ℓ-prefix) of $\mathbf{x}$. Similarly, a substring of the form $\mathbf{x}[|\mathbf{x}| - \ell + 1 : |\mathbf{x}|]$ is a suffix (or an ℓ-suffix) of $\mathbf{x}$. We say that strings $\mathbf{x}$ and $\mathbf{y}$ have an *overlap* of length ℓ if the ℓ-suffix of $\mathbf{x}$ and the ℓ-prefix of $\mathbf{y}$ are equal. We let $\mathbf{x} \oplus \mathbf{y}$ denote the concatenation of $\mathbf{x}$ and $\mathbf{y}$.

We assume that there exists an unknown target DNA sequence $\mathbf{s}$ of length $|\mathbf{s}| = G$ which we wish to assemble from a set of N reads $\mathcal{R}$. Throughout the paper, we will make two simplifying assumptions about the set of reads:

(A1) All reads in $\mathcal{R}$ have length L.
(A2) The reads in $\mathcal{R}$ are error-free.

The first assumption is made to simplify the exposition of the results. The second assumption is motivated by the existence of overlapping tools (such as DAligner [4]), which can efficiently identify significant matches between reads at error rates of over 15%. The algorithm described in this manuscript can be adapted to work with the approximate matches found by such tools, essentially by treating them as exact matches.

For ease of exposition, we will assume that $\mathbf{s}$ is a circular sequence of length G; i.e., $\mathbf{s}[t+G] = \mathbf{s}[t]$ for any t. This way we will avoid edge effects and a read $\mathbf{x} \in \mathcal{R}$ can correspond to any substring $\mathbf{s}[t : t+L-1]$, for $t = 1, ..., G$. We will use the standard Poisson sampling model for shotgun sequencing. This means that each of the N reads is drawn independently and uniformly at random from the set of length-L substrings of $\mathbf{s}$, $\{\mathbf{s}[t : t+L-1] : t = 1, ..., G\}$.

A. Repeats and Bridging

A *repeat* of length ℓ in $\mathbf{s}$ is a substring $\mathbf{x} \in \Sigma^\ell$ appearing at distinct positions t_1 and t_2 in $\mathbf{s}$; i.e., $\mathbf{s}[t_1 : t_1 + \ell - 1] = \mathbf{s}[t_2 : t_2 + \ell - 1] = \mathbf{x}$, that is maximal; i.e., $\mathbf{s}[t_1 - 1] \neq \mathbf{s}[t_2 - 1]$ and $\mathbf{s}[t_1 + \ell] \neq \mathbf{s}[t_2 + \ell]$. A repeat is bridged if there is a read that extends beyond one copy of the repeat in both directions, as shown in Fig. 1. A repeat is doubly-bridged if both copies are bridged. Similarly, a triple repeat of length ℓ is a substring $\mathbf{x}$ that appears at three distinct locations in $\mathbf{s}$ (possibly overlapping); i.e., $\mathbf{s}[t_1 : t_1 + \ell - 1] = \mathbf{s}[t_2 : t_2 + \ell - 1] = \mathbf{s}[t_3 : t_3 + \ell - 1] = \mathbf{x}$ for distinct t_1, t_2 and t_3 (modulo G, given the circular DNA assumption), and is maximal (that is, all three copies can not be extended in a direction). A triple repeat is said to be bridged if at least one of its copies is bridged. It is said to be *all-bridged* if all of its copies are bridged, as illustrated in Fig. 1(b), and *all-unbridged* if none of its copies are bridged.

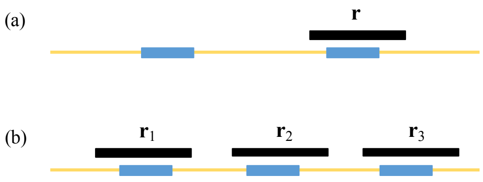

Fig. 1. (a) Illustration of a bridged repeat in $\mathbf{s}$; (b) Illustration of a triple repeat all-bridged by reads $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$.

III. A Distortion Metric for Assembly Graphs

To motivate our notion of distortion for assembly graphs, let us first consider an idealized setting in which *all* reads of length L from $\mathbf{s}$ are given. We call this ensemble of reads the L-mer composition of $\mathbf{s}$, defined as the multiset

$$\mathcal{C}_L(\mathbf{s}) = \{\mathbf{s}[i : i+L-1] : 1 \leq i \leq G\}. \tag{1}$$

We will let $\overline{\mathcal{C}}_L(\mathbf{s})$ represent the support of $\mathcal{C}_L(\mathbf{s})$; i.e., $\mathcal{C}_L(\mathbf{s})$ without the copy counts. The L-mer composition $\mathcal{C}_L(\mathbf{s})$ does not, in general, determine $\mathbf{s}$ unambiguously. In particular, any sequence $\mathbf{x}$ with $\mathcal{C}_L(\mathbf{x}) = \mathcal{C}_L(\mathbf{s})$ is indistinguishable from $\mathbf{s}$ when only $\mathcal{C}_L(\mathbf{s})$ is observed. Thus, one requires at least

$$\log |\{\mathbf{x} : \mathcal{C}_L(\mathbf{x}) = \mathcal{C}_L(\mathbf{s})\}| \tag{2}$$

additional bits to determine $\mathbf{s}$ unambiguously from all other sequences having the same L-mer composition. This uncertainty characterization can be translated to the language of sequence graphs through the notion of a k-mer graph[1] of $\mathbf{s}$, $B_k(\mathbf{s})$ [3].

Definition 1. *The multigraph $B_k(\mathbf{s})$ has $\overline{\mathcal{C}}_{k-1}(\mathbf{s})$ as its node set and, for $\mathbf{x}, \mathbf{y} \in \overline{\mathcal{C}}_{k-1}(\mathbf{s})$, we place m edges from $\mathbf{x}$ to $\mathbf{y}$ if the k-mer $\mathbf{x} \oplus \mathbf{y}[k-1]$ has multiplicity m in $\mathcal{C}_k(\mathbf{s})$.*

It is easy to see that $B_k(\mathbf{s})$ is an Eulerian graph for any k, and every $\mathbf{x}$ with $\mathcal{C}_k(\mathbf{x}) = \mathcal{C}_k(\mathbf{s})$ corresponds to a distinct Eulerian cycle in $B_k(\mathbf{s})$. Furthermore, two Eulerian cycles that are distinct up to edge multiplicities correspond to distinct sequences. Therefore, a natural measure of the "distortion" of $B_k(\mathbf{s})$ as an assembly graph of $\mathbf{s}$ would be

$$D_k(\mathbf{s}) \triangleq \log \mathrm{ec}(B_k(\mathbf{s})), \tag{3}$$

where $\mathrm{ec}(G)$ is the number of Eulerian cycles in G, distinct up to edge multiplicity. We point out that the number of Eulerian cycles in a k-mer graph has been previously used in the related context of DNA-based storage channels [7].

As illustrated in Fig. 2, $D_k(\mathbf{s})$ can be computed for real genomes, and can be interpreted as a lower bound on how good an assembly from reads of length k can be.

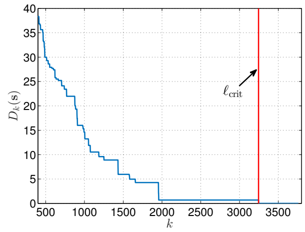

Fig. 2. $D_k(\mathbf{s})$ as a function of k, when $\mathbf{s}$ is the genome of *E. coli 536*. Notice that $D_k(\mathbf{s})$ reaches zero when $k = \ell_{\rm crit}(\mathbf{s})$ [1].

In the actual setting for the assembly problem, however, one does not have access to the entire L-mer composition of $\mathbf{s}$, nor can be expected to perfectly construct $B_k(\mathbf{s})$ for some $k < L$ (other than for small values of k). Hence, when defining a distortion metric for assembly graphs, one must consider a larger class of graphs than $B_k(\mathbf{s})$. In this work, we will consider the following:

[1] In the assembly literature, such graphs are sometimes referred to as de Bruijn graphs. However, since our proposed algorithm is not a de Bruijn graph based algorithm in the usual sense of [3], we avoid the terminology.

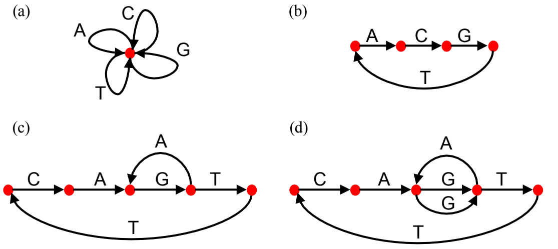

Fig. 3. (a) The trivial sequence graph G_0 is always sufficient. (b,c,d) An example of the distortion computed for the assembly of a cyclic sequence $\mathbf{s} = \mathsf{CAGAGTT}$ is shown. If the graph in (b) is returned by an assembly algorithm, then as the graph is not a sufficient sequence graph with respect to $\mathbf{s}$, the distortion is computed to be $\log \left\lceil \frac{1}{7}\binom{7}{2,1,2,2} \right\rceil + 1 = 2.95$. If the sequence graph G of order $k = 1$ in (c) is returned, then $G[\mathbf{s}]$ is as shown in (d). The distortion is $D(G, \mathbf{s}) = 0$ as there is exactly one Eulerian cycle in $G[\mathbf{s}]$ (modulo differences in traversing edges between the same two vertices).

Definition 2. *A sequence graph $G = (V, E, \phi)$ of order k is a directed multigraph where each edge $e \in E$ is labeled with a k-mer $\phi(e) \in \Sigma^k$, and each node $v \in V$ is labeled with a $(k-1)$-mer $\phi(v) \in \Sigma^{k-1}$ satisfying the property that if $\phi(u, v) = \mathbf{x}$, then $\phi(u) = \mathbf{x}[1 : k-1]$ and $\phi(v) = \mathbf{x}[2 : k]$.*

Notice that any path $p = (v_1, ..., v_\ell)$ on a sequence graph of order k naturally defines a length-$(\ell - 2 + k)$ string

$$\mathrm{st}(p) \triangleq \phi(v_1, v_2)[1] \oplus ... \oplus \phi(v_{\ell-2}, v_{\ell-1})[1] \oplus \phi(v_{\ell-1}, v_\ell).$$

If a path $p = (v_1, ..., v_\ell)$ ends in a node with out-degree zero, it will be called a graph suffix, and if it starts in a node with in-degree zero, it will be called a graph prefix.

Definition 3. *A Chinese Postman cycle in a sequence graph G, is a cycle that traverses every edge at least once.*

A natural formulation for the genome assembly problem is to identify a Chinese Postman cycle in the constructed sequence graph which corresponds to the true sequence [8, 9].

Definition 4. *A sequence graph G is said to be sufficient (for the assembly of $\mathbf{s}$) if it contains a Chinese Postman cycle $c_{\mathbf{s}}$ such that $\mathrm{st}(c_{\mathbf{s}}) = \mathbf{s}$ (up to cyclic shifts).*

While it is natural to define the goal of the partial assembly problem to be the construction of a sufficient sequence graph, it is typically unreasonable to expect the assembly algorithm to correctly estimate the multiplicities of all the edges; i.e., the number of times $c_{\mathbf{s}}$ traverses each edge. The reason is that the length of the genome is not known in advance, and hence neither is the coverage depth (i.e., the average number of reads covering a given position). Other practical issues like uneven coverage and sequence specific biases only add to the difficulty there. Thus our distortion metric should not penalize incorrect multiplicities, and thus not require the produced graph to be Eulerian. To define our distortion metric, we will consider an "Eulerian version" of the constructed sequence graph. More precisely, if $G = (V, E, \phi)$ is a sufficient sequence graph and $c_{\mathbf{s}}$ is a Chinese Postman cycle in G corresponding to the sequence $\mathbf{s}$, we will let $G[\mathbf{s}]$ be the multigraph obtained by setting the multiplicity of edge e to be the number of times $c_{\mathbf{s}}$ traverses e.

Definition 5. *The distortion of a sequence graph G is*

$$D(G, \mathbf{s}) \triangleq \begin{cases} \log \mathrm{ec}(G[\mathbf{s}]) & \text{if } G \text{ is a sufficient sequence graph for } \mathbf{s} \\ D_1(\mathbf{s}) + 1 & \text{otherwise} \end{cases} \qquad (4)$$

where $\mathrm{ec}(G)$ is the number of Eulerian cycles in G that are distinct up to edge multiplicities, and $D_1(\mathbf{s})$ is the distortion achieved by the 1-mer graph of $\mathbf{s}$, $B_1(\mathbf{s})$.

We note that if $\mathbf{s}$ contains all of $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, then $D_1(\mathbf{s})$ would be the distortion achieved by the graph G_0, shown in Figure 3(a). It is not difficult to see that the distortion of any sufficient sequence graph is at most $D_1(\mathbf{s})$. Fig. 3 shows the computation of this distortion in a toy example.

IV. A Greedy Algorithm for Partial Assembly

In this section we describe an algorithm to assemble a sequence graph. We then analyze its performance in terms of its ability to produce a sufficient sequence graph and the resulting distortion.

The algorithm can be seen as a generalization of the greedy algorithm for sequence assembly [10]. In the standard greedy algorithm, prefixes and suffixes of reads are iteratively merged in order to produce a single sequence. However, when an incorrect merging occurs, it has no way of detecting and fixing it at later iterations. Our algorithm overcomes this issue by allowing a read prefix/suffix to be merged to the *interior* of another read, or to a previously merged prefix/suffix, as illustrated in Fig. 4(a). As we will show, this additional flexibility is helpful in constructing a sufficient sequence graph in the sense of Definition 4, making the algorithm robust from the point of view of partial assembly.

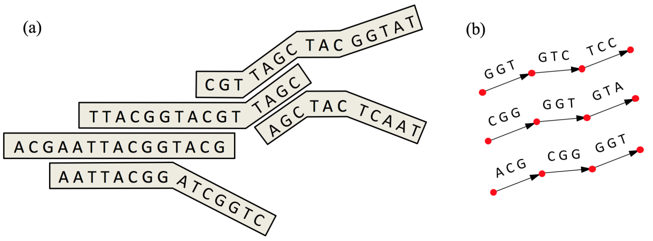

Fig. 4. (a) In the greedy merging algorithm, we allow matches between a prefix/suffix of a read and the interior of another read, producing a graph that is not a line, as is the case with the standard greedy algorithm [10]; (b) Initial sequence graph for reads GGTCC, CGGTA, and ACGGT for $k = 3$. Notice that a match of ℓ symbols corresponds to a path of $\ell - k + 1$ edges.

Our algorithm will maintain at all times a sequence graph in the sense of Definition 2, where each read $\mathbf{r}_i \in \mathcal{R}$ corresponds to a path p_i with $L - k + 2$ nodes and $L - k + 1$ edges, which correspond to the $L - k + 1$ consecutive k-mers of $\mathbf{r}_i$. Initially, all N paths will be disjoint components of the graph, as illustrated in Fig. 4(b).The algorithm then proceeds by finding matches between a previously unused prefix or suffix and any part of another read, and merging the corresponding paths. The algorithm is termed greedy since it searches for matches in decreasing order of length.

Algorithm 1 Greedy merging algorithm

```
Input: Initial sequence graph (Fig. 4(b)), and parameter $k$
for $\ell = L, L-1, L-2, ..., k$ do
    $X \leftarrow \{\mathbf{x} \in \Sigma^\ell$ : $\mathbf{x}$ is a current graph prefix or suffix that
            appears in more than one read$\}$
    for $\mathbf{x} \in X$ do
        Merge the path corresponding to $\mathbf{x}$ from all reads that
        contain the substring $\mathbf{x}$
Output: Resulting sequence graph of order $k$
```

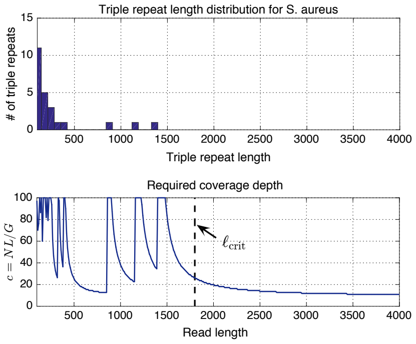

Fig. 5. Distribution of triple repeat lengths on *S. aureus*, and coverage depth required for the conditions in Theorem 1 to be achieved with probability 0.99.

The parameter k should be chosen as the minimum overlap we expect adjacent reads to have, and can be made large for sequencing experiments with high coverage depth. For instance, when assembling long reads (10, 000 bp) with high error rates, a typical choice for the minimum match length k is 1000 [4].

We proceed to analyze the distortion achieved by the sequence graph that Algorithm 1 outputs in two steps. We first obtain conditions for the sequence graph to be sufficient, and then characterize conditions under which the resulting distortion can be upper bounded by $D_k(\mathbf{s})$ for some $k > 1$.

Definition 6. *We say that $\mathcal{R}$ k-covers the sequence* $\mathbf{s}$ *if there is a read starting in every k-length substring of* $\mathbf{s}$.

Theorem 1. *Algorithm 1 constructs a sufficient sequence graph of order k if the set of reads $\mathcal{R}$ k-covers the sequence* $\mathbf{s}$ *and every triple repeat is either unbridged or all-bridged.*

As described in Section V, given the conditions in Theorem 1, one can bound the probability that the graph produced by Algorithm 1 is not sufficient. This bound can then be translated into a value of coverage depth $c = NL/G$ for which the resulting sequence graph is sufficient with a desired probability $1-\epsilon$. This is illustrated in Fig. 5 for the *S. aureus* genome from the GAGE dataset [11]. We notice that for values of L that are far from the length of some triple repeat, a small coverage depth suffices.

We remark that an interesting open question is to determine if the non-monotonicity caused by the peaks in required coverage near triple repeat lengths (as shown in Fig. 5) represents a fundamental barrier or a limitation of the algorithm. Most existing algorithms face challenges when there are triple repeats of lengths that are close to the read length. In fact, overlap-based algorithms also suffer from similar problems when there are double repeats of lengths close to L.

In addition to the sufficiency property guaranteed by Theorem 1, we need a way to characterize the distortion achieved by the resulting graph. To do so, we will bound the distortion achieved by assembling reads of length L by the quantity $D_q(\mathbf{s})$, defined in (3), for some $q < L$. We begin with a definition.

Definition 7. *Two repeats* $\mathbf{s}[a_1 : a_1 + \ell]$, $\mathbf{s}[a_2 : a_2 + \ell]$ *and* $\mathbf{s}[b_1 : b_1 + m]$, $\mathbf{s}[b_2 : b_2 + m]$ *are said to be linked if* $a_2 < b_1 \leq a_2 + \ell + 1$. *We call* $a_2 + \ell + 1 - b_1$ *the link length.*

As illustrated in Fig. 6, linked repeats are potential causes of ambiguity in the sequence graph.

Theorem 2. *Suppose that the set of reads $\mathcal{R}$ from the sequence* $\mathbf{s}$ *satisfies the following conditions:*

(a) each triple repeat is either all-bridged or all-unbridged,

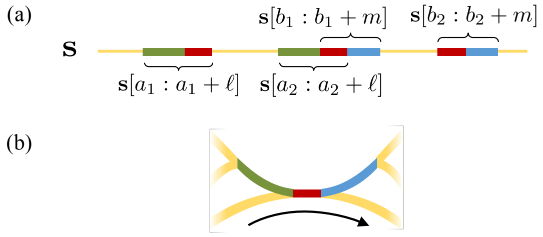
(a)
$\mathbf{s}[b_1 : b_1 + m]$
$\mathbf{s}[b_2 : b_2 + m]$
$\mathbf{s}$
$\mathbf{s}[a_1 : a_1 + \ell]$
$\mathbf{s}[a_2 : a_2 + \ell]$
(b)

Fig. 6. (a) Illustration of linked repeats with link length $a_2 + \ell + 1 - b_1$. (b) If we merge both repeats in the sequence graph, the link (red segment) creates a path that is not in the true sequence $\mathbf{s}$.

(b) all repeats of length $\leq q$ are doubly-bridged,

(c) for all pairs of linked repeats with link length ℓ satisfying $k - 1 \leq \ell \leq q$, at least one is doubly-bridged.

Then the sufficient sequence graph G produced by Algorithm 1 has a distortion satisfying $D(G, \mathbf{s}) \leq D_q(\mathbf{s})$.

Notice that if all repeats in $\mathbf{s}$ are doubly bridged, the conditions in Theorem 2 are satisfied for any q, implying that $D(G, \mathbf{s}) = 0$. The standard greedy algorithm [10], on the other hand, achieves perfect assembly when all repeats are bridged, not necessarily doubly bridged [1]. Intuitively, the more stringent requirement of double bridging is the price paid to obtain guarantees in a range of L where the genome is much more repetitive.

V. Distortion on a Real Genome

Clearly in practice we cannot verify whether the conditions in Theorems 1 and 2 are satisfied, as we do not have access to the genome being sequenced. The purpose of these results is to allow us to compute the rate-distortion tradeoff achieved by Algorithm 1 on previously assembled genomes. This provides a framework to analyze the algorithm's performance and compare it to the fundamental lower bound (or to other algorithms).

For an organism whose whole genome $\mathbf{s}$ is known, we can compute repeat statistics, which can then be used to numerically compute the number of reads N required to guarantee that the condition in Theorem 1 holds with probability at least $1 - \epsilon$, for some target error probability $\epsilon > 0$ (see [12] for details). This yields the curve in Fig. 5(b).

Similarly, by identifying the distribution of repeat lengths and characterizing which pairs of repeats are linked, one can compute the probability that conditions (b) and (c) in Theorem 2 are not satisfied for a given q. By computing $D_q(\mathbf{s})$

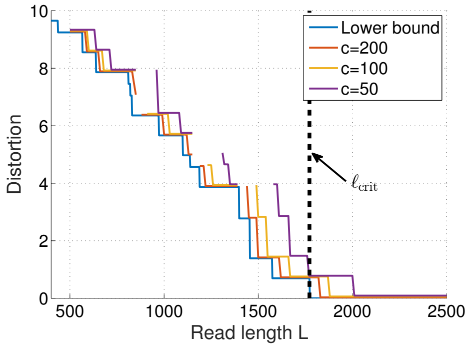

Fig. 7. Distortion achieved by Algorithm 1 with $k = 300$ on *S. aureus* with probability 0.99 for different coverage depths $c = NL/G$, compared to the lower bound $D_L(\mathbf{s})$. Gaps indicate that the probability of the conditions of Theorem 2 not being satisfied is at least 0.01.

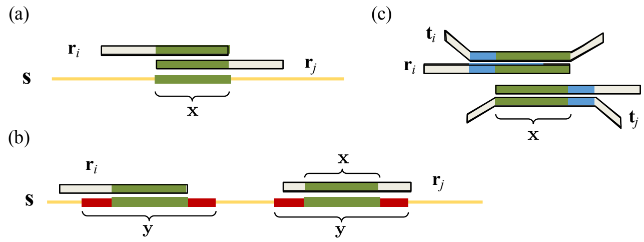

Fig. 8. Substrings $\mathbf{x}$ of $\mathbf{r}_i$ and $\mathbf{r}_j$ mapping to: (a) same segment in $\mathbf{s}$; (b) distinct segments in $\mathbf{s}$ corresponding to an unbridged repeat. (c) If at iteration $L-\ell$ the ℓ-suffix of $\mathbf{r}_i$ and the ℓ-prefix of $\mathbf{r}_j$ are not merged, they must have been merged to some reads $\mathbf{t}_i$ and $\mathbf{t}_j$ in previous iterations.

for a range of values of q, which can be done using the well-known BEST Theorem, we can upper bound the distortion achieved by Algorithm 1 with a desired probability $1-\epsilon$. Notice that $D_L(\mathbf{s})$ is also the minimum distortion that can be achieved with reads of length L, which provides a lower bound to the distortion that can be achieved by any algorithm. In Fig. 7 we show these curves computed for *S. aureus* for different values of the coverage depth $c = NL/G$. We notice that the upper bound curves follow the lower bound closely but have gaps in them, representing the ranges of L where the conditions of Theorem 1 are not satisfied with the desired probability, and the achieved distortion jumps to $D_1(\mathbf{s})$.

VI. Proofs of Main Results

Here, we provide proof sketches for the main results in the paper. The longer version of this manuscript [12] has full proofs.

A. Sufficiency of sequence graph (Theorem 1)

Note that each $\mathbf{r} \in \mathcal{R}$ induces a mapping between its symbols and a segment of length L in $\mathbf{s}$, from where $\mathbf{r}$ was sampled.

Lemma 1. *Suppose that reads $\mathbf{r}_i$ and $\mathbf{r}_j$ share substrings $\mathbf{x}$ that*

(i) map to the same string $\mathbf{x}$ in $\mathbf{s}$, as in Fig. 8(a), or

(ii) map to two different strings $\mathbf{x}$ in $\mathbf{s}$, which are part of an unbridged repeat, as shown in Fig. 8(b).

At the end of iteration $L-\ell$ of Algorithm 1, the paths corresponding to $\mathbf{x}$ in $\mathbf{r}_i$ and $\mathbf{r}_j$ are merged.

Given Lemma 1, Theorem 1 follows immediately because, at the end of the algorithm, the overlapping part of any two consecutive reads $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ (which must be of length at least k when $\mathcal{R}$ k-covers $\mathbf{s}$) must be merged in the sequence graph.

Lemma 1 is proved by induction on $|\mathbf{x}| = L-1, L-2, ..., k$. The base case follows since, if reads $\mathbf{r}_i$ and $\mathbf{r}_j$ have a matching substring of size $L-1$, they must correspond to a graph suffix or prefix in the beginning of the algorithm and will thus be merged, so assume the lemma holds up to $|\mathbf{x}| = \ell+1$. Suppose $\mathbf{r}_i$ and $\mathbf{r}_j$ share a substring $\mathbf{x}$ with $|\mathbf{x}| = \ell$ and we are in case (i). If Algorithm 1 does not merge the paths corresponding to $\mathbf{x}$ in $\mathbf{r}_i$ and $\mathbf{r}_j$ at iteration $L-\ell$, it must be the case that a longer suffix of $\mathbf{r}_i$ and a longer prefix of $\mathbf{r}_j$ were merged to other reads, say $\mathbf{t}_i$ and $\mathbf{t}_j$ in previous iterations, as illustrated in Fig. 8(c). Now if the substring $\mathbf{x}$ in $\mathbf{t}_i$ (or $\mathbf{t}_j$) maps to the same segment of $\mathbf{s}$ as $\mathbf{r}_i$ and $\mathbf{r}_j$, it has an overlap greater than ℓ with both $\mathbf{r}_i$ and $\mathbf{r}_j$ and, by the induction hypothesis, it is merged to both, causing the path $\mathbf{x}$ in $\mathbf{r}_i$ and $\mathbf{r}_j$ to be merged. Similarly, if the substrings $\mathbf{x}$ in $\mathbf{t}_i$ and $\mathbf{t}_j$ map to the same segment of $\mathbf{x}$, by the induction hypothesis, they have been previously merged, causing $\mathbf{r}_i$ and $\mathbf{r}_j$ to be merged as well. Finally, if the substrings $\mathbf{x}$ in $\mathbf{t}_i$ and $\mathbf{t}_j$ map to two other segments $\mathbf{x}$ in $\mathbf{s}$, $\mathbf{x}$ is a triple repeat. Since $\mathbf{x}$ is unbridged at the intersection of $\mathbf{r}_i$ and $\mathbf{r}_j$, by the assumption in Theorem 1, it must be all-unbridged. Hence, $\mathbf{t}_i$ and $\mathbf{t}_j$ are part of an unbridged repeat, and by the induction hypothesis have been previously merged to each other. Case (ii) follows similarly.

B. Distortion Bound (Theorem 2)

First we show that when the conditions in Theorem 2 are satisfied, Algorithm 1 can only merge two paths if they either correspond to the same segment in $\mathbf{s}$, or they correspond to an unbridged repeat (or a substring of it) in $\mathbf{s}$. A simple proof by contradiction is used to show this.

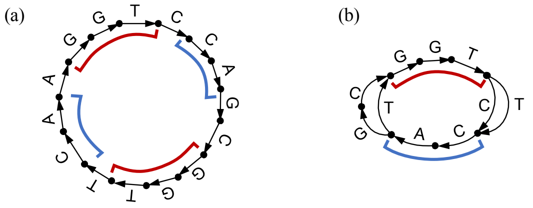

Fig. 9. (a) Cycle sequence graph $\mathcal{G}$ (of order $k = 1$) for the sequence GGTCCAGTCGGTTCAA; (b) Contracted graph $\mathcal{G}_{\mathcal{U}(\mathcal{R})}$ where $\mathcal{U}(\mathcal{R})$ corresponds to the two pairs of repeats shown in red and blue.

We then consider the set of repeats in $\mathbf{s}$ that are not doubly bridged by $\mathcal{R}$, $\mathcal{U}(\mathcal{R})$, and define a sufficient sequence graph $\mathcal{G}_{\mathcal{U}(\mathcal{R})}$ by taking a cycle graph representation of $\mathbf{s}$ and merging the repeats in $\mathcal{U}(\mathcal{R})$, as illustrated in Fig. 9. Using the previous result, we show that G, the graph returned by Algorithm 1, can be converted to $\mathcal{G}_{\mathcal{U}(\mathcal{R})}$ via node contractions. This implies that there is a surjection between the set of Eulerian cycles in $\mathcal{G}_{\mathcal{U}(\mathcal{R})}[\mathbf{s}]$ to the set of Eulerian cycles in $G[\mathbf{s}]$. This thus gives us that, $D(G, \mathbf{s}) \le D(G_{\mathcal{U}(\mathcal{R})}, \mathbf{s})$.

Finally we show that, when the conditions of Theorem 2 are met, any path of $q-k$ edges in $\mathcal{G}_{\mathcal{U}(\mathcal{R})}$ corresponds to a q-mer from $\mathcal{C}_q(\mathbf{s})$, the q-mer composition of $\mathbf{s}$. This directly gives us that the sequence corresponding to any Eulerian cycle in $\mathcal{G}_{\mathcal{U}(\mathcal{R})}[\mathbf{s}]$ has q-mer composition $\mathcal{C}_q(\mathbf{s})$. This implies that $D(\mathcal{G}_{\mathcal{U}(\mathcal{R})}, \mathbf{s}) \le D_q(\mathbf{s})$, which is the desired result.

References

[1] G. Bresler, M. Bresler, and D. Tse, “Optimal Assembly for High Throughput Shotgun Sequencing,” *BMC Bioinformatics*, 2013.

[2] E. Ukkonen, “Approximate String Matching with q-grams and maximal matches,” *Theoretical Computer Science*, vol. 92, no. 1, 1992.

[3] P. A. Pevzner, H. Tang, and M. S. Waterman, “An Eulerian path approach to DNA fragment assembly,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 17, pp. 9748–9753, 2001.

[4] E. W. Myers, “Efficient local alignment discovery amongst noisy long reads,” in *Algorithms in Bioinformatics*. Springer, 2014, pp. 52–67.

[5] ——, “The fragment assembly string graph,” *Bioinformatics*, vol. 21, pp. 79–85, 2005.

[6] I. Shomorony, S. Kim, T. Courtade, and D. Tse. Optimal Assembly via Sparse Read-Overlap Graphs. [Online]. Available: http://www.eecs.berkeley.edu/~courtade/pdfs/NSG.pdf

[7] H. M. Kiah, G. J. Puleo, and O. Milenkovic, “Codes for DNA sequence profiles,” *arXiv:1502.00517*, 2015.

[8] N. Nagarajan and M. Pop, “Parametric complexity of sequence assembly: theory and applications to next generation sequencing,” *Journal of computational biology*, vol. 16, no. 7, pp. 897–908, 2009.

[9] P. Medvedev, K. Georgiou, G. Myers, and M. Brudno, “Computability of models for sequence assembly,” in *Algorithms in Bioinformatics*. Springer, 2007, pp. 289–301.

[10] J. Tarhio and E. Ukkonen, “A greedy approximation algorithm for constructing shortest common superstrings,” *Theoret. Comput. Science*, vol. 57, pp. 131–145, 1988.

[11] [Online]. Available: http://gage.cbcb.umd.edu/

[12] I. Shomorony, G. M. Kamath, F. Xia, T. Courtade, and D. N. C. Tse, “Partial DNA Assembly: A Rate-Distortion Perspective,” *arXiv:1605.01941*, 2016.