JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 28, Number 4, 2021
Mary Ann Liebert, Inc.
Pp. 395–409
DOI: 10.1089/cmb.2020.0432

Lower Density Selection Schemes via Small Universal Hitting Sets with Short Remaining Path Length

HONGYU ZHENG, CARL KINGSFORD, and GUILLAUME MARÇAIS

ABSTRACT

Universal hitting sets (UHS) are sets of words that are unavoidable: every long enough sequence is hit by the set (i.e., it contains a word from the set). There is a tight relationship between UHS and minimizer schemes, where minimizer schemes with low density (i.e., efficient schemes) correspond to UHS of small size. Local schemes are a generalization of minimizer schemes that can be used as replacement for minimizer scheme with the possibility of being much more efficient. We establish the link between efficient local schemes and the minimum length of a string that must be hit by a UHS. We give bounds for the remaining path length of the Mykkeltveit UHS. In addition, we create a local scheme with the lowest known density that is only a log factor away from the theoretical lower bound.

Keywords: de Bruijn graph, depathing set, minimizers, sequence sketch, universal hitting set.

1. INTRODUCTION

We study the problem of finding *Universal Hitting Sets* (UHS) (Orenstein et al., 2016). A UHS is a set of words, each of length k, such that every long enough string (say of length L or longer) contains as a substring element from the set. We call such a set a UHS for parameters k and L. They are sets of unavoidable words, that is, words that must be contained in any long strings, and we are interested in the relationship between the size of these sets and the length L.

More precisely, we say that a k-mer a (a string of length k) *hits* a string S if a appears as a substring of S. A set A of k-mers hits S if at least one k-mer of A hits S. A UHS for length L is a set of k-mers that hits every string of length L. Equivalently, the *remaining path length* of a universal set is the length of the longest string that is not hit by the set ($L-1$ here).

The study of UHS is motivated, in part, by the link between UHS and the common method of *minimizers* (Schleimer et al., 2003; Roberts et al., 2004a,b). The minimizer method is a way to sample a string for representative k-mers in a deterministic way by breaking a string into windows, each window containing w k-mers, and selecting in each window a particular k-mer (the ''minimum k-mer,'' as defined by a preset order on the k-mers). This method is used in many bioinformatic software programs (Ye et al., 2012; Grabowski and Raniszewski, 2013; Chikhi et al., 2015; Deorowicz et al., 2015; Jain et al., 2017) to reduce the amount of computation and improve run time (see Marçais et al., 2019 for usage examples). The

Computational Biology Department, Carnegie Mellon University, Pittsburgh Pennsylvania, USA.

© Hongyu Zheng, et al., 2020. Published by Mary Ann Liebert, Inc. This Open Access article is distributed under the terms of the Creative Commons License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

minimizer method is a family of methods parameterized by the order on the k-mers used to find the minimum. The *density* is defined as the expected number of sampled k-mers per unit length of sequence. Depending on the order used, the density varies.

In general, a lower density (i.e., fewer sampled k-mers) leads to greater computational improvements and is therefore desirable. For example, a read aligner such as Minimap2 (Li and Birol, 2018) stores all the locations of minimizers in the reference sequence in a database. It then finds all the minimizers in a read and searches in the database for these minimizers. The locations of these minimizers are used as seeds for the alignment. Using a minimizer scheme with a reduced density leads to a smaller database and fewer locations to consider, hence an increased efficiency, while preserving the accuracy.

There is a two-way correspondence between minimizer methods and UHS: each minimizer method has a corresponding UHS, and a UHS defines a family of *compatible* minimizer methods (Marçais et al., 2017, 2018). This correspondence also links the remaining path length of a UHS and the window size of a compatible minimizer scheme: the remaining path length of the UHS is upper bounded by the number of bases in each window in the minimizer scheme ($L \leq w+k-1$).

Moreover, the relative size of the UHS, defined as the size of UHS over the number of possible k-mers, provides an upper bound on the density of the corresponding minimizer methods: the density is no more than the relative size of the UHS. Precisely, $\frac{1}{w} \leq d \leq \frac{|U|}{\sigma^k}$, where d is the density, U is the UHS, σ^k is the total number of k-mers on an alphabet of size σ, and w is the window length. In other words, the study of UHS with small size leads to the creation of minimizer methods with provably low density.

Local schemes (Mykkeltveit, 1972) and forward schemes are generalizations of minimizer schemes. These extensions are of interest because they can be used in place of minimizer schemes while sampling k-mers with lower density. In particular, minimizer schemes cannot have density close to the theoretical lower bound of $1/w$ when w becomes large, while local and forward schemes do not suffer from this limitation (Marçais et al., 2018). Understanding how to design local and forward schemes with low density will allow us to further improve the computation efficiency of many bioinformatic algorithms.

The previously known link between minimizer schemes and UHS relied on the definition of an ordering between k-mers, and therefore is not valid for local and forward schemes that are not based on any ordering. Nevertheless, UHS play a central role in understanding the density of local and forward schemes.

Our first contribution is to describe the connection between UHS, local and forward schemes. More precisely, there are two connections: first, between the density of the schemes and the relative size of the UHS, and second, between the window size w of the scheme and the *remaining path length* of the UHS (i.e., the maximum length L of a string that does not contain a word from the UHS). This motivates our study of the relationship between the size of a UHS U and the remaining path length of U.

There is a rich literature on unavoidable word sets (Lothaire, 2002). The setting for UHS is slightly different for two reasons. First, we impose that all the words in the set U have the same length k, as a k-mer is a natural unit in bioinformatic applications. Second, the set U must hit any string of a given finite length L, rather than being unavoidable only by infinitely long strings.

Mykkeltveit (1972) answered the question of what is the size of a minimum unavoidable set with k-mers by giving an explicit construction for such a set. The k-mers in the Mykkeltveit set are guaranteed to be present in any infinitely long sequence, and the size of the Mykkeltveit set is minimum in the sense that for any set $\mathcal{S}$ with fewer k-mers, there is an infinitely long sequence that avoids $\mathcal{S}$. On the contrary, the construction gives no indication on the remaining path length.

The DOCKS (Orenstein et al., 2016) and ReMuVal (DeBlasio et al., 2019) algorithms are heuristics to generate unavoidable sets for parameters k and L. Both of these algorithms use the Mykkeltveit set as a starting point. In many practical cases, the longest sequence that does not contain any k-mer from the Mykkeltveit set is much larger than the parameter L of interest (which for a compatible minimizer scheme corresponds to the window length). Therefore, the two heuristics extend the Mykkeltveit set to cover every L-long sequence. These greedy heuristics do not provide any guarantee on the size of the unavoidable set generated compared with the theoretical minimum size and are only computationally tractable for limited ranges of k and L.

Our second contribution is to give upper and lower bounds on the remaining path length of the Mykkeltveit sets. These are the first bounds on the remaining path length for minimum size sets of unavoidable k-mers.

Defining local or forward schemes with a density of $O(1/w)$ (i.e., within a constant factor of the theoretical lower bound) is not only of practical interest to improve the efficiency of existing algorithms, but it is also interesting for a historical reason. Both Roberts et al. (2004a) and Schleimer et al. (2003) used a probabilistic model to suggest that minimizer schemes have an expected density of $2/w$. Unfortunately, this simple probabilistic model does not correctly model the minimizer schemes outside of a small range of values for parameters k and w, and minimizers do not have an $O(1/w)$ density in general. Although the general question of whether a local scheme with $O(1/w)$ exists is still open, our third contribution is an almost-optimal forward scheme with density of $O(\ln(w)/w)$ density. This is the lowest known density for a forward scheme, beating the previous best density of $O(\sqrt{w}/w)$ (Marçais et al., 2018), and hinting that $O(1/w)$ might be achievable.

Understanding the properties of UHS and their many interactions with selection schemes (minimizer and forward and local schemes) is a crucial step toward designing schemes with lower density and improving the many algorithms using these schemes. In Section 2, we give an overview of the results, and in Section 3, we give detailed proofs. Further research directions are discussed in Section 4.

2. RESULTS

2.1. Notation

2.1.1. Universal hitting sets. Consider a finite alphabet $\Sigma = \{0, \ldots, \sigma - 1\}$ with $\sigma \geq 2$ elements. If $a \in \Sigma$, a^k denotes the letter a repeated k times. We use Σ^k to denote the set of strings of length k on alphabet Σ, and call them k-mers. If S is a string, $S[n, l]$ denotes the substring starting at position n and of length l. For a k-mer $a \in \Sigma^k$ and an l-long string $S \in \Sigma^l$, we say "a hits S" if a appears as substring of S [$a = S[i, k]$ for some i]. For a set of k-mers $A \subseteq \Sigma^k$ and $S \in \Sigma^l$, we say "A hits S" if there exists at least one k-mer in A that hits S. A set $A \subseteq \Sigma^k$ is a UHS for length L if A hits every string of length L.

2.1.2. de Bruijn graphs. Many questions regarding strings have an equivalent formulation with graph terminology using *de Bruijn graphs*. The de Bruijn graph $B_{\Sigma,k}$ on alphabet Σ and of order k has a node for every k-mer, and an edge (u, v) for every string of length $k+1$ with a prefix u and the suffix is v. There are σ^k vertices and σ^{k+1} edges in the de Bruijn graph of order k.

There is a one-to-one correspondence between strings and paths in $B_{\Sigma,k}$: a path with w nodes corresponds to a string of $L = w + k - 1$ characters. A UHS A corresponds to a *depathing set* of the de Bruijn graph: a UHS for k and L intersects with every path in the de Bruijn graph with $w = L - k + 1$ vertices. We say "A is a (α, l)-UHS" if A is a set of k-mers that is a UHS, with relative size $\alpha = |A|/\sigma^k$ and hits every walk of l vertices (and therefore every string of length $L = l + k - 1$).

A *de Bruijn sequence* is a particular sequence of length $\sigma^k + k - 1$ that contains every possible k-mer once and only once. Every de Bruijn graph is Hamiltonian and the sequence spelled out by a Hamiltonian tour is a de Bruijn sequence.

2.1.3. Selection schemes. A *local scheme* (Schleimer et al., 2003) is a method to select positions in a string. A local scheme is parameterized by a *selection function f*. It works by looking at every w-mer of the input sequence S: $S[0, w]$, $S[1, w]$, ..., and selecting in each window a position according to the selection function f. The selection function selects a position in a window of length w, that is, it is a function $f : \Sigma^w \to [0 : w - 1]$. The output of a forward scheme is a set of selected positions: $\{i + f(S[i, w]) | 0 \leq i < |S| - w\}$.

A *forward scheme* is a local scheme with a selection function such that the selected positions form a nondecreasing sequence. That is, if ω_1 and ω_2 are two consecutive windows in a sequence S, then $f(\omega_2) \geq f(\omega_1) - 1$.

A *minimizer scheme* is a scheme where the selection function takes in the sequence of w consecutive k-mers and returns the "minimum" k-mer in the window (hence the name minimizers). The minimum is defined by a predefined order on the k-mers (e.g., lexicographic order) and the selection function is $f : \Sigma^{w+k-1} \to [0 : w - 1]$.

See Figure 1 for examples of all three schemes. The local scheme concept is the most general as it imposes no constraint on the selection function, while a forward scheme must select positions in a nondecreasing way. A minimizer scheme is the least general and also selects positions in a nondecreasing way.

Local and forward schemes were originally defined with a function defined on a window of w k-mers, $f : \Sigma^{w+k-1} \to [0 : w - 1]$, similarly to minimizers. Selection schemes are schemes with $k = 1$, and have a single parameter w as the word length. While the notion of k-mer is central to the definition of the

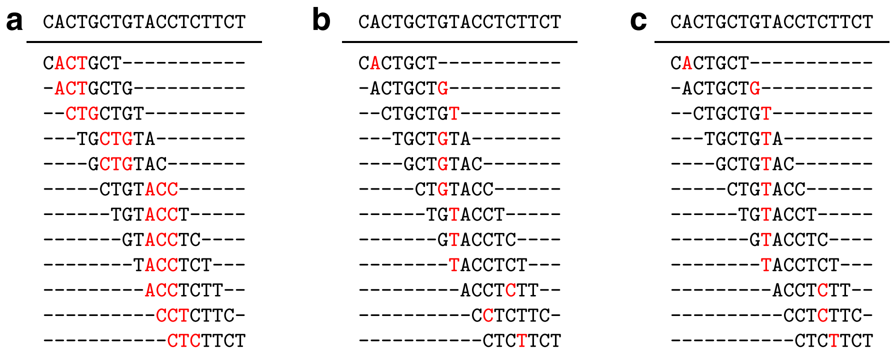

FIG. 1. **(a)** Example of selecting minimizers with $k=3$, $w=5$, and the lexicographic order (i.e., $AAA < AAC < AAG < \ldots < TTT$). The top line is the input sequence, each subsequent line is a 7-bases long window (the number of bases in a window is $w+k-1=7$) with the minimum 3-mer highlighted. The positions {1, 2, 5, 9, 10, 11} are selected for a density $d=6/(18-3+1)=0.375$. **(b)** On the same sequence, an example of a selection scheme for $w=7$ (and $k=1$ because it is a selection scheme, hence the number of bases in a window is also w). The set of positions selected is {1, 6, 7, 8, 11, 13, 14}. This is not a forward scheme as the sequence of selected position is not decreasing. **(c)** A forward selection scheme for $w=7$ with selected positions {1, 7, 8, 12, 13}. Like the minimizer scheme, the sequence of selected positions is nondecreasing.

minimizer schemes, it has no particular meaning for a local or forward scheme: these schemes select positions within each window of a string S, and the sequence of the k-mers at these positions is no more relevant than a sequence elsewhere in the window to the selection function.

There are multiple reasons to consider selection schemes. First, they are slightly simpler as they have only one parameter, namely the window length w. Second, in our analysis, we consider the case where w is asymptotically large, therefore $w \gg k$ and the setting is similar to having $k=1$. Finally, this simplified problem still provides information about the general problem of local schemes. Suppose that f is the selection function of a selection scheme, for any $k > 1$ we can define $g_k : \Sigma^{w+k-1} \rightarrow [0,\ w-1]$ as $g_k(\omega)=f(\omega\ [0,\ w])$. That is, g_k is defined from the function f by ignoring the last $k-1$ characters in a window. The functions g_k define proper selection functions for local schemes with parameters w and k, and because exactly the same positions are selected, the density of g_k is equal to the density of f. In the following sections, unless noted otherwise, we use forward and local schemes to denote forward and local selection schemes.

2.1.4. Density. Because a local scheme on string S may pick the same location in two different windows, the number of selected positions is usually less than $|S|-w+1$. The *particular density* of a scheme is defined as the number of distinct selected positions divided by $|S|-w+1$ (Fig. 1). The *expected density*, or simply the *density*, of a scheme is the expected density on an infinitely long random sequence. Alternatively, the expected density is computed exactly by computing the particular density on any de Bruijn sequence of order $\geq 2w-1$. In other words, a de Bruijn sequence of large enough order ‘‘looks like’’ a random infinite sequence with respect to a local scheme (see Marçais et al., 2017 and Section 3.1).

2.2. Main results

The density of a local scheme is in the range $[1/w,\ 1]$, as $1/w$ corresponds to selecting exactly one position per window, and 1 corresponds to selecting every position. Therefore, the density goes from a low value with a constant number of positions per window [density is $O(1/w)$, which goes to 0 when w gets large], to a high with constant value [density is $\Omega(1)$] where the number of positions per window is proportional to w. When the minimizers and winnowing schemes were introduced, both articles used a simple probabilistic model to estimate the expected density to $2/(w+1)$, or about 2 positions per window. Under this model, this estimate is within a constant factor of the optimal, $O(1/w)$.

Unfortunately, this simple model properly accounts for the minimizer behavior only when k and w are small. For large k—that is, $k \gg w$—it is possible to create an almost-optimal minimizer scheme with a density $\sim 1/w$. More problematic, for large w—that is, $w \gg k$—and for all minimizer schemes, the density becomes constant [$\Omega(1)$] (Marçais et al., 2018). In other words, minimizer schemes cannot be optimal or within a constant factor of optimal for large w, and the estimate of $2/(w+1)$ is very inaccurate in this regime.

This motivates the study of forward schemes and local schemes. It is known that there exist forward schemes with a density of $O(1/\sqrt{w})$ (Marçais et al., 2018). This density is not within a constant factor of the optimal density but at least shows that forward and local schemes do not have constant density such as minimizer schemes for large w and that they can have much lower density.

2.2.1. Connection between UHS and selection schemes. In the study of selection schemes, as for minimizer schemes, UHS play a central role. We describe the link between selection schemes and UHS, and show that the existence of a selection scheme with low density implies the existence of a UHS with a small relative size.

Theorem 1. *Given a local scheme f on w-mers with density d_f, we can construct a $(d_f, w)-$UHS on $(2w - 1)$-mers. If f is a forward scheme, we can construct a $(d_f, w)-$UHS on $(w+1)$-mers.*

2.2.2. Almost-optimal relative size UHS for linear path length. Conversely, because of their link to forward and local selection schemes, we are interested in UHS with remaining path length $O(w)$. Necessarily a universal hitting hits any infinitely long sequences. On de Bruijn graphs, a set hitting every infinitely long sequence is a *decycling set*: a set that intersects with every cycle in the graph. In particular, a decycling set must contain an element in each of the cycles obtained by the rotation of the w-mers (e.g., cycle of the type 001 → 010 → 100 → 001). The number of these rotation cycles is known as the "necklace number" $N_{\sigma,w} = \frac{1}{n}\sum_{d|w} \varphi(d)\,\sigma^{w/d} = O(\sigma^w/w)$ (Golomb, 2014), where $\varphi(d)$ is the Euler's totient function.

Consequently, the relative size of a UHS, which contains at least one element from each of these cycles, is lower bounded by $O(1/w)$. The smallest previously known UHS with $O(w)$ remaining path length has a relative size of $O(\sqrt{w}/w)$ (Marçais et al., 2018). We construct a smaller UHS with relative size $O(\ln(w)/w)$:

Theorem 2. *For every sufficiently large w, there is a forward scheme with density of $O(\ln(w)/w)$ and a corresponding $(O(\ln(w)/w), w)$-UHS.*

2.2.3. Remaining path length bounds for the Mykkeltveit sets. Mykkeltveit (1972) gave an explicit construction for a decycling set with exactly one element from each of the rotation cycles, and thereby proved a long-standing conjecture (Golomb, 2014) that the minimal size of decycling sets is equal to the necklace number. Under the UHS framework, it is natural to ask what the remaining path length for Mykkeltveit sets is. Given that the de Bruijn graph is Hamiltonian, there exist paths of length exponential in w: the Hamiltonian tours have σ^w vertices. Nevertheless, we show that the remaining path length for Mykkeltveit sets is upper and lower bounded by polynomials of w:

Theorem 3. *For sufficiently large w, the Mykkeltveit set is a $(N_{\sigma,w}/\sigma^w, g(w))$-UHS, having the same size as minimal decycling sets, while $c_1 w^2 \leq g(w) \leq c_2 w^3$ for some constants c_1 and c_2.*

3. METHODS AND PROOFS

For length reasons, several parts of the proof are found in the Supplementary Material.

3.1. UHS from selection schemes

3.1.1. Contexts and densities of selection schemes. We derive another way of calculating densities of selection schemes based on the idea of *contexts*.

Recall a local scheme is defined as a function $f : \Sigma^w \to [0,\ w-1]$. For any sequence S and scheme f, the set of selected locations is $\{f(S[i,\ w]) + i\}$ and the density of f on the sequence is the number of selected locations divided by $|S| - w + 1$. Counting the number of distinct selected locations is the same as counting the number of w-mers $S[i, w]$ such that f picks a new location from all previous w-mers. f can pick identical locations on two w-mers only if they overlap, so intuitively, we only need to look back $(w-1)$ windows to check if the position is already picked. Formally, f picks a new position in window $S\ [i, w]$ if and only if $f(S\ [i,\ w])+i \neq f(S\ [i-d,\ w])+(i-d)$ for all $1 \leq d \leq w-1$.

For a location i in sequence S, the context at this location is defined as $c_i = S\ [i-w+1,\ 2w-1]$, a $(2w\ -\ 1)$-mer whose last w-mer starts at i. Whether f picks a new position in $S\ [i, w]$ is entirely determined by its context, as the conditions only involve w-mers as far back as $S\ [i-w+1,\ w]$, which are all included in the context. This means that instead of counting selected positions in S, we can count the contexts c satisfying $f(c\ [w-1,\ w])+w-1 \neq f(c\ [j,\ w])+j$ for all $0 \leq j \leq w-2$, which are the contexts such that f on the last w-mer of c picks a new location. We denote by $\mathcal{C}_f \subset \Sigma^{2w-1}$ the set of contexts that satisfy this condition.

Definition 1. For given w and local selection scheme $f : \Sigma^w \to [0,\ w-1]$, $\mathcal{C}_f = \{c \in \Sigma^{2w-1} | \forall 0 \leq i \leq w-2,\ f(c\ [w-1,\ w])+(w-1) \neq f(c\ [i, w])+i\}$ is a subset of Σ^{2w-1}.

The expected density of f is computed as the number of selected positions over the length of the sequence for a random sequence, as the sequence becomes infinitely long. For a sufficiently long random sequence $(|S|\ \gg w)$, the distribution of its contexts converges to a uniform random distribution over $(2w-1)$-mers. Because the distribution of these contexts is exactly equal to the uniform distribution on a circular de Bruijn S sequence of order at least $2w-1$, we can calculate the expected density of f as the density of f on S, or as $|\mathcal{C}_f|/\sigma^{2w-1}$.

3.1.2. UHS from selection schemes. The set $\mathcal{C}_f$ over $(2w-1)$-mers is the UHS needed for Theorem 1. Intuitively, it is a UHS with remaining path length of at most $w-1$, because one location must be picked every w window, meaning there is a window that picked a new location. The context that is prefix of this window is in $\mathcal{C}_f$ by definition.

Lemma 1. *$\mathcal{C}_f$ is a UHS with remaining path length of at most $w-1$.*

Proof. By contradiction, assume there is a path of length w in the de Bruijn graph of order $(2w-1)$, say $\{c_0, c_1, \cdots, c_{w-1}\}$, that avoids $\mathcal{C}$. We construct the sequence S' corresponding to the path: $S' \in \Sigma^{3w-2}$ such that $S'[i,\ 2w-1] = c_i$.

Since $c_{w-1} \notin \mathcal{C}$ and S' include c_{w-1}, it means f on the last w-mer of c_{w-1} (which is $S'[2w\ -\ 2,\ w]$) picks a location that has been picked before on S'. The coordinate l of this selection in S' satisfies $l\ \geq\ 2w-2$. As $0 \leq f(x) \leq w-1$, the first w-mer $S'[m, w]$ in S' such that f picks $S'[l]$ (i.e., $m+f(S'\ [m,\ w])=l$) satisfies $m \geq w-1$. The context $c_{m-w+1} = S'\ [m-(w-1),\ 2w-1]$ then satisfies that a new location l is picked when f is applied to its last w-mer, and by definition $c_{m-w+1} \in \mathcal{C}$, contradiction. □

This results is also a direct consequence of the definition of $\mathcal{C}$. An alternative direct proof is available in Supplementary Section S1.

When f is a forward scheme, to determine if a new location is picked in a window, looking back one window is sufficient. This is because if we do not pick a new location, we have to pick the same location as in the last window. This means the context with two w-mers, or as a $(w+1)$-mer, is sufficient, and our other arguments involving contexts still hold. Combining the pieces, we prove the following theorem:

Theorem 4. *Given a local scheme f on w-mers with density d_f, we can construct a $(d_f,\ w)-$ UHS on $(2w-1)$-mers. If f is a forward scheme, we can construct a $(d_f,\ w)-$ UHS on $(w+1)$-mers.*

3.2. Forbidden word depathing set

3.2.1. Construction and path length. In this section, we construct a set that is $(O(\ln(w)/w), w)-$ UHS.

Definition 2 (Forbidden Word UHS). Let $d = \lfloor \log_\sigma(w/\ln(w)) \rfloor - 1$. Define $\mathcal{F}_{\sigma, w}$ as the set of w-mers that satisfies either of the following clauses: (1) 0^d is the prefix of x (2) 0^d is not a substring of x.

We assume that w is sufficiently large such that $d \geq 1$.

Lemma 2. *The longest remaining path in the de Bruijn graph of order w after removing $\mathcal{F}_{\sigma,w}$ is $w-d$.*

Proof. Let $\{x_0, x_1, \cdots, x_{w-d}\}$ be a path of length $w-d+1$ in the de Bruijn graph. If x_0 does not have a substring equal to 0^d, it is in $\mathcal{F}_{\sigma,w}$. Otherwise, let c be the index such that $x_0[c, d] = 0^d$. Since $c \le w-d$, $x_c[0, d] = 0^d$ and x_c is in $\mathcal{F}_{\sigma,w}$.

On the contrary, let $S = 1^{w-d}0^d1^{w-d-1} \in \Sigma^{2w-d-1}$ and $x_i = S[i, w]$ for $0 \le i < w-d$. None of $\{x_i\}$ is in $\mathcal{F}_{\sigma,w}$, meaning there is a path of length $w-d$ in the remaining graph. □

The number of w-mer satisfying clause 1 is $\sigma^{w-d} = O(\ln(w)\sigma^w/w)$. For the rest of this section, we focus on counting w-mers satisfying clause 2 in Definition 2, that is, the number of w-mers not containing 0^d.

3.2.2. Number of w*-mers not containing 0*$^{\mathrm{d}}$. We construct a finite state machine (FSM) that recognizes 0^d as follows. The FSM consists of $d+1$ states labeled "0" to "d," where "0" is the initial state and "d" is the terminal state. The state "i" with $0 \le i \le d-1$ means that the last i characters were *0* and $d-i$ more zeroes are needed to match 0^d. The terminal state "d" means that we have seen a substring of d consecutive zeroes. If the machine is at nonterminal state "i" and receives the character 0, it moves to state "$i+1$," otherwise it moves to state "0"; once the machine reaches state "d," it remains in that state forever.

Now, assume we feed a random w-mer to the FSM. The probability that the machine does not reach state "d" for the input w-mer is the relative size of the set of w-mer satisfying clause 2. Denote $p_k \in \mathbb{R}^d$ such that $p_k(j)$ is the probability of feeding a random k-mer to the machine and ending up in state "j," for $0 \le j < d$ (note that the vector does not contain the probability for the terminal state "d"). The answer to our problem is then $\|p_w\|_1 = \sum_{i=0}^{d-1} p_w(i)$, that is, the sum of the probabilities of ending at a nonterminal state.

Define $\mu = 1/\sigma$. Given that a randomly chosen w-mer is fed into the FSM, that is, each base is chosen independently and uniformly from Σ, the probabilities of transition in the FSM are: "i" → "i + 1" with probability μ, "i" → "0" with probability $1-\mu$. The probability matrix to not recognize 0^d is a $d\times d$ matrix, as we discard the row and column associated with terminal state "d":

$$A_d = \begin{bmatrix} 1-\mu & 1-\mu & \dots & 1-\mu & 1-\mu \\ \mu & 0 & \dots & 0 & 0 \\ 0 & \mu & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \mu & 0 \end{bmatrix}_{d\times d} = \begin{bmatrix} (1-\mu)\mathbf{1}_{d-1}^T & 1-\mu \\ \mu \mathrm{I}_{d-1} & \mathbf{0}_{d-1} \end{bmatrix}$$

Starting with $p_0 = (1, 0, \dots, 0) \in \mathbb{R}^d$ as initially no sequence has been parsed and the machine is at state "0" with probability 1, we can compute the probability vector p_w as $p_w = A_d p_{w-1} = A_d^w p_0$.

3.2.3. Bounding $\|p_w\|_1$. We start by deriving the characteristic polynomial $p_{A_d}(\lambda)$ of A_d and its roots, which are the eigenvalues of A_d:

Lemma 3.
$$p_{A_d}(\lambda) = \det(A_d - \lambda I) = \begin{cases} (-1)^d \frac{\lambda^{d+1}-\lambda^d-\mu^{d+1}+\mu^d}{\lambda-\mu} & \lambda \ne \mu \\ (-\mu)^{d-1}((1-\mu)d-\mu) & \lambda = \mu \end{cases}$$

Proof. The characteristic polynomial of A_d satisfies the following recursive formula, obtained by expanding the determinant over the first column and using the linearity of the determinant:

$$\det(A_d - \lambda I_d) = \begin{vmatrix} 1-\mu-\lambda & 1-\mu & 1-\mu & \cdots & 1-\mu \\ \mu & -\lambda & 0 & \cdots & 0 \\ 0 & \mu & -\lambda & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda \end{vmatrix}_{d\times d}$$
$$= (1-\lambda)(-\lambda)^{d-1} - \mu p_{A_{d-1}}(\lambda).$$

For $d=1$, we have $p_{A_1}(\lambda)=1-\mu-\lambda$. Assuming $\lambda \neq \mu$ for now, we repeatedly expand the recursive formula to obtain a closed-form formula for $p_{A_d}(\lambda)$:

$$\begin{aligned} p_{A_d}(\lambda) &= (1-\lambda)\left[(-\lambda)^{d-1}+(-\mu)^1(-\lambda)^{d-2}+\cdots+(-\mu)^{d-2}(-\lambda)^1+(-\mu)^{d-1}\right] \\ &\quad +(-\mu)^d \qquad (*) \\ &= (-1)^d\left[(\lambda-1)\frac{\lambda^d-\mu^d}{\lambda-\mu}+\mu^d\right] \\ &= (-1)^d\frac{\lambda^{d+1}-\lambda^d-\mu^{d+1}+\mu^d}{\lambda-\mu} \end{aligned}$$

The value for the characteristic polynomial when $\lambda = \mu$ can be derived by plugging $\lambda = \mu$ in the line marked with (*) to obtain $p_{A_d}(\lambda)=(1-\mu)d\mu^{d-1}+(-\mu)^d$. □

Now we fix d and focus on the polynomial $f_d(\lambda)=\lambda^{d+1}-\lambda^d-\mu^{d+1}+\mu^d$. Since this is a polynomial of degree $d+1$, it has $d+1$ roots and except for μ, which is a root of f_d but not of p_{A_d}, f_d and p_{A_d} have the same roots.

Lemma 4. *For sufficiently large d, $f_d(\lambda)$ has a real root λ_0 satisfying $1-\mu^d < \lambda_0 < 1-\mu^{d+1}$.*

Proof. We show f_d has opposite signs on the lower and upper bound of this inequality for sufficiently large d.

$$\begin{aligned} f_d(1-\mu^d) &= (1-\mu^d)^{d+1}-(1-\mu^d)^d-\mu^{d+1}+\mu^d \\ &= 1-(d+1)\mu^d+O(\mu^{2d})-1+d\mu^d-O(\mu^{2d})-\mu^{d+1}+\mu^d \\ &= -\mu^{d+1}+O(\mu^{2d}) < 0 \\ f_d(1-\mu^{d+1}) &= (1-\mu^{d+1})^{d+1}-(1-\mu^{d+1})^d-\mu^{d+1}+\mu^d \\ &= 1-(d+1)\mu^{d+1}+C(d+1,2)\mu^{2d+2}+O(\mu^{3d+6}) \\ &\quad -1+d\mu^{d+1}-C(d,2)\mu^{2d+2}-O(\mu^{3d+6})-\mu^{d+1}+\mu^d \\ &= -2\mu^{d+1}+\mu^d+d\mu^{2d+2}+O(\mu^{3d+6}) > 0 \end{aligned}$$

For the last line, if $\sigma = 2$ the first two terms cancel out and $d\mu^{2d+2}$ becomes dominant and positive, otherwise $\mu^d=\sigma\mu^{d+1} > 2\mu^{d+1}$. Since f_d is polynomial, f_d is continuous and thus has a root between $1-\mu^d$ and $1-\mu^{d+1}$. □

Lemma 5. *Let $s=\mu/\lambda_0$. $\nu_0=(1, s, s^2, \cdots, s^{d-1})$ is the right eigenvector of A_d corresponding to eigenvalue λ_0, and $\|\nu_0\|_1 < 3$ for sufficiently large d.*

Proof. For the first part, we need to verify $A_d\,\nu_0=\lambda_0\,\nu_0$. For indices $1 \leq i < d$, $(A_d\,\nu_0)_i=\mu(\nu_0)_{i-1}=\mu s^{i-1}=\lambda_0 s^i=(\lambda_0\,\nu_0)_i$. For the first element in the vector, we have:

$$\begin{aligned} (A_d\,\nu_0)_0-(\lambda_0\,\nu_0)_0 &= (1-\mu)(1+s+\cdots s^{d-1})-\lambda_0 \\ &= \frac{(1-\mu)(s^d-1)-\lambda_0(s-1)}{s-1} \\ &= \frac{\lambda_0^d(\mu^d-\lambda_0^d-\mu^{d+1}+\lambda_0^{d+1})}{s-1} \\ &= \frac{\lambda_0^d f_d(\lambda_0)}{s-1}=0. \end{aligned}$$

This verifies $A_d\nu_0=\lambda_0\nu_0$. For the second part, note that for sufficiently large d we have $\lambda_0 > 1-\mu^d > 0.9$ and since $\mu \leq 0.5$, we have $s=\mu/\lambda_0 < 2/3$. Every element of ν_0 is positive, so $\|\nu_0\|_1=\sum_{i=0}^{d-1} s^i < \sum_{i=0}^{\infty} s^i=1/(1-s) < 3$. □

Lemma 6. $\|p_w\|_1=\left\|A_d^w p_0\right\|_1=O(1/w)$.

Proof. Let $\eta_0 = \nu_0 - p_0 = (0,\ s,\ s^2,\ \cdots, s^{d-1})$, where $s = \mu/\lambda_0$ from last lemma. Because $\lambda_0 > 0$, the elements of η_0 and A_d are all non-negative, then the elements of $A_d^w \eta_0$ and $\lambda_0 \eta_0$ are also non-negative. Now, recall that $d = \lfloor \log_\sigma (w/\ln(w)) \rfloor - 1$, which implies that $\mu^{d+1} \geq \ln(w)/w$.

$$\begin{aligned}
\|p_w\|_1 &= \left\|A_d^w p_0\right\|_1 \\
&= \left\|A_d^w(\nu_0 - \eta_0)\right\|_1 \\
&= \left\|\lambda_0^w \nu_0 - A_d^w \eta_0\right\|_1 && (\nu_0 \text{ is a eigenvector of } A_d) \\
&\leq \lambda_0^w \|\nu_0\|_1 && (\text{non} - \text{negative elements}) \\
&< 3(1 - \mu^{d+1})^w && (\text{by Lemmas 4 and 5}) \\
&\leq 3(1 - \ln(w)/w)^w && (\text{by definition of } d) \\
&\leq 3\exp(-\ln(w)) && (1 - x \leq e^{-x}) \\
&= O(1/w).
\end{aligned}$$

□

This lemma implies that the relative size for the set $\mathcal{F}_{\sigma,w}$ is dominated by the w-mers satisfying clause 1 of Definition 2 and $\mathcal{F}_{\sigma,w}$ is of relative size $O(\ln(w)/w)$. This completes the proof that $\mathcal{F}_{\sigma,w}$ is $(O(\ln(w)/w), w) - \text{UHS}$.

3.3. Construction of the Mykkeltveit sets

In this section, we construct the Mykkeltveit set $\mathcal{M}_{\sigma,w}$ and prove some important properties of the set. We start with the definition of the Mykkeltveit embedding of the de Bruijn graph.

Definition 3 (Modified Mykkeltveit Embedding). For a w-mer x, its embedding in the complex plane is defined as $P(x) = \sum_{i=0}^{w-1} x_i r_w^{i+1}$, where r_w is a wth root of unity, $r_w = e^{2\pi i/w}$.

Intuitively, the position of a w-mer x is defined as the following center of mass. The w roots of unity form a circle in the complex plane, and a weight equal to the value of the base x_i is set at the root r_w^{i+1}. The position of x is the center of mass of these w points and associated weights. Originally, Mykkeltveit defined the embedding with weight r_w^i (Mykkeltveit, 1972). This extra factor of r_w in our modified embedding rotates the coordinate and is instrumental in the proof.

Define the *successor function* $S_a(x) = x_1 x_2 \cdots x_{w-1} a$, where $a \in \Sigma$. The successor function gives all the neighbors of x in the de Bruijn graph. A *pure rotation* of x is the particular neighbor $R(x) = S_{x_0}(x)$, that is, the sequence of $R(x)$ is a left rotation of x.

We focus on a particular kind of cycle in the de Bruijn graph. A *pure cycle* in the de Bruijn graph, also known as *conjugacy class*, is the sequence of w-mers obtained by repeated rotation: $(x,\ R(x),\ R^2(x),\ \ldots)$. Each pure cycle consists of w distinct w-mer s, unless $x_0 x_1 \cdots x_{w-1}$ is periodic, and in this case, the size of the cycle is equal to its shortest period.

The embeddings from pure rotations satisfy a curious property:

Lemma 7 (Rotations and Embeddings). *$P(R(x))$ on the complex plane is $P(x)$ rotated clockwise around origin by $2\pi/w$. $P(S_a(x))$ is $P(R(x))$ shifted by $\delta = a - x_0$ on the real part, with the imaginary part unchanged.*

Proof. By Definition 3 and the definition of successor function $S_a(x)$:

$$\begin{aligned}
P(S_a(x)) &= \sum\nolimits_{i=0}^{w-1} (S_a(x))_i r_w^{i+1} \\
&= \sum\nolimits_{i=0}^{w-2} x_{i+1} r_w^{i+1} + a r_w^{w-1+1} \\
&= r_w^{-1} \sum\nolimits_{i=0}^{w-1} x_i r_w^{i+1} + (a - x_0) \\
&= r_w^{-1} P(x) + \delta
\end{aligned}$$

Note that for pure rotations $\delta = 0$, and $r_w^{-1}P(x)$ is exactly $P(x)$ rotated clockwise by $2\pi/w$. □

The range for δ is $[-\sigma+1, \sigma-1]$. In particular, δ can be negative. In a pure cycle, either all w-mers satisfy $P(x)=0$, or they lie equidistant on a circle centered at origin. Figure 2a shows the embeddings and pure cycles of 5-mers. It is known that we can partition the set of all w-mers into $N_{\sigma,k}$ disjoint pure cycles. This means any decycling set that breaks every cycle of the de Bruijn graph will be at least this large. We now construct our proposed depathing set with this idea in mind.

Definition 4 (Mykkeltveit set). We construct the Mykkeltveit set $\mathcal{M}_{\sigma,w}$ as follows. Consider each conjugacy class, we will pick one w-mer from each of them by the following rule:

1. If every w-mer in the class embeds to the origin, pick an arbitrary one.
2. If there is one w-mer x in the class such that $\mathrm{Re}(P(x)) <0$ and $\mathrm{Im}(p(x)=0)$ (on the negative real axis), pick that one.
3. Otherwise, pick the unique w-mer x such that $\mathrm{Im}(p(x)<0)$ and $\mathrm{Im}(P(\mathrm{R}(x))) >0$. *Intuitively, this is the w-mer in the cycle right below the negative real axis.*

This set breaks every pure cycle in the de Bruijn graph by its construction, with an interesting property:

Lemma 8. *Let* $\{x_i\}$ *be a path on the de Bruijn graph that avoids* $\mathcal{M}_{\sigma,w}$. *If* $\mathrm{Im}(P(x_i)) \leq 0$, *then for all* $j \geq 1$, $\mathrm{Im}(P(x_j)) \leq 0$.

Proof. It suffices to show that in the remaining de Bruijn graph after removing $\mathcal{M}_{\sigma,w}$, there are no edges $x \to y$ such that $\mathrm{Im}(P(x)) \leq 0$ and $\mathrm{Im}(P(y)) >0$. The edge $x \to y$ means that $y=S_a(x)$ for some a. By Lemma 7, $\mathrm{Im}(P(R(x)))=\mathrm{Im}(P(S_a(x)))=\mathrm{Im}(P(y)) >0$.

- If we have $\mathrm{Im}(P(x)) <0$, by clause 3 of Definition 4, $x \in \mathcal{M}_{\sigma,w}$.
- If we have $\mathrm{Im}(P(x))=0$ and $\mathrm{Re}(P(x)) <0$, by clause 2 of Definition 4, we have $x \in \mathcal{M}_{\sigma,w}$.
- If we have $\mathrm{Im}(P(x))=\mathrm{Re}(P(x))=0$, we would have $\mathrm{Im}(P(y))=\mathrm{Im}(P(R(x)))=0$, a contradiction.
- If we have $\mathrm{Im}(P(x))=0$ and $\mathrm{Re}(P(x)) >0$, $P(x)$ lies on positive half of the real axis, so rotating it clockwise by $2\pi/w$ degrees we would have $\mathrm{Im}(P(y))=\mathrm{Im}(P(R(x)))=0$, a contradiction. □

3.4. Upper bounding the remaining path length in Mykkeltveit sets

In this section, we show that the remaining path after removing $\mathcal{M}_{\sigma,w}$ is at most $O(w^3)$ long. This polynomial bound is a stark contrast to the number of remaining vertices after removing the Mykkeltveit set—that is, $\sigma^w - N_{\sigma,w} \sim (1-\frac{1}{w})\sigma^w$, which is exponential in w. Our main argument involves embedding a w-mer to point in the complex plane, similar to Mykkeltveit's construction.

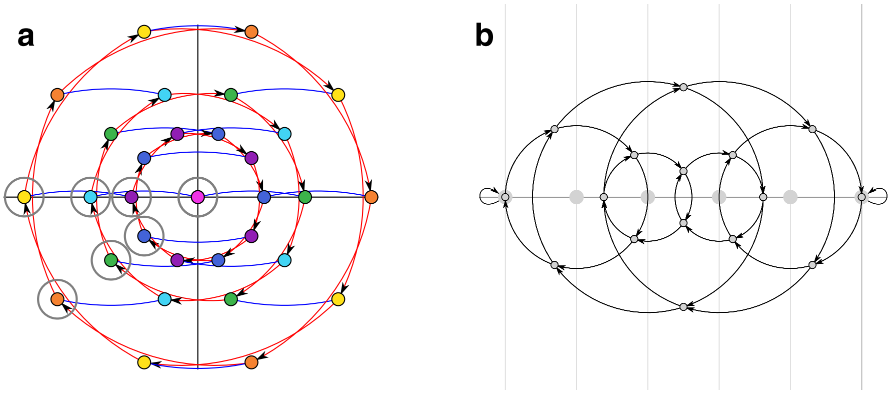

FIG. 2. **(a)** Mykkeltveit embedding of the de Bruijn graph of order 5 on the binary alphabet. The nodes of a conjugacy class have the same color and form a circle (there is more than one class per circle). The pure rotations are represented by the red edges. A nonpure rotation $S_a(x)$ is a red edge followed by a horizontal shift (blue edge). The set of nodes circled in gray is the Mykkeltveit set. **(b)** Weight-in embedding of the same graph. Multiple w-mers map to the same position in this embedding and each circle represents a conjugacy class. The gray dots on the horizontal axis are the w centers of rotations and the vertical gray lines going through the centers separate the space in subregions of interest.

3.4.1. From w*-mers to embeddings.* In this section, we formulate a relaxation that converts paths of w-mers to trajectories in a geometric space. Precisely, we model S_a in Lemma 7 as a rotation operating on a complex embedding with attached weights, where the weights restrict possible moves.

Formally, given a pair (z, t) where z is a complex number and t an integer, define the family of operations $Z_\delta(z, t) = (r_w^{-1}z + \delta, t + \delta)$. When $z = P(x)$ is the position of a w-mer x, $t = W(x) = \sum_{i=0}^{w-1} x_i$ is its weight, and when $0 \leq \delta + x_0 < \sigma$, $Z_\delta(P(x), W(x)) = (P(S_{\delta+x_0}(x)), \ W(S_{\delta+x_0}(x)))$. This means Z_δ is equivalent to finding the position and weight of the successor $S_{\delta+x_0}$.

We are now looking for the length of the longest path by repeated application of Z_δ that satisfies $0 \leq t \leq W_{\max}$, where $W_{\max} = (\sigma - 1)\,w$ is the maximum weight of any w-mer. This is a relaxation of the original problem of finding a longest path as some choices of δ and some pairs $(z, \ t)$ on these paths may not correspond to the actual transition or w-mer in the de Bruijn graph (when $\delta + x_0$ is negative or greater than $\sigma - 1$, then it is not a valid transition). In some sense, the pair $(z, \ t)$ is a loose representation of a w-mer where the precise sequence of the w-mer is ignored and only its weight is considered. On the contrary, every valid path in the de Bruijn graph corresponds to a path in this relaxation, and an upper bound on the relaxed problem is an upper bound of the original problem.

3.4.2. Weight-in embedding and relaxation. The *weight-in embedding* maps the pair $x = (z, \ w)$ to the complex plane. This transforms the original longest remaining path problem into a geometric problem of bounding the length in the complex plane under some operation S_δ.

Definition 5 (Weight-In Embedding). The weight-in embedding of $x = (z, \ t)$ is $Q(x) = z - t$. Accordingly, for a w-mer x, its embedding is $Q(x) = Q(P(x), \ W(x)) = P(x) - W(x)$.

The Z_δ operations in this embedding correspond to a rotation, and, maybe surprisingly, this rotation is independent of the value δ.

Lemma 9. *Let $x = (z, \ t)$. For all δ, the point $Q(Z_\delta(x))$ is the point $Q(x)$ rotated clockwise $2\pi/w$ around the point $(-t, \ 0)$.*

Proof. By definition of weight-in embedding and the operation Z_δ:

$$Q(Z_\delta(z, \ t)) = r_w^{-1}z + \delta - (t + \delta) = r_w^{-1}(Q(z, \ t) + t) - t$$

In the complex plane, the rotation formula around center c and of angle θ is $c + e^{i\theta}(z - c)$. Therefore, the operations Z_δ is a rotation around $c = (-t, \ 0)$ of angle $\theta = -2\pi/w$. □

Figure 2b shows the weight-in embedding of a de Bruijn graph. The set $\mathcal{C}_{\sigma,w} = \{(-j, \ 0) | 0 \leq j \leq W_{\max}\}$ is the set of all the possible centers of rotation, and is shown by large gray dots on the x-axis of Figure 2b. Because all the w-mers in a given conjugacy class have the same weight, say t_0, the conjugacy classes form a circle around a particular center $(-t_0, \ 0)$. The image after application of S_δ is independent of the parameter δ, but dependent on the weight t of the underlying pair $(z, \ t)$.

Multiple pairs of $x = (z, \ t)$ can share the same weight-in embedding $Q(x)$. As seen in Figure 2b, every node belongs to two circles with different centers, meaning there are two embeddings with the same $Q(x)$ but different t.

Lemma 8 naturally divides any path in the de Bruijn graph avoiding $\mathcal{M}_{\sigma,w}$ into two parts, the first part with $\mathrm{Im}(P(x)) > 0$, and the second part with $\mathrm{Im}(P(x)) \geq 0$. Thanks to the symmetry of the problems, we focus on the upper half-plane, defined as the region with $\mathrm{Im}(P(x)) \geq 0$. With the weight-in embedding, as long as the path is contained in the upper half-plane, it is always traveling to the right (toward large real value) or stays unmoved, as stated below:

Lemma 10 (Monotonicity of Re(Q($\cdot$))). *Assume $Q(x)$ and $Q(Z_\delta(x))$ are both in the upper half-plane. If $Q(x)$ does not coincide with its associated rotation center $(-t, \ 0)$, then* $\mathrm{Re}(Q(Z_\delta(x))) > \mathrm{Re}(Q(x))$, *otherwise $Q(Z_\delta(x)) = Q(x)$.*

Proof. The operation is a clockwise rotation where the rotation center is on the x-axis and the two points are on the non-negative half-plane. Necessarily, the real part increased, unless the point is on the fix point of the rotation [which is when $Q(x) = (-t, \ 0)$]. □

We further relax the problem by allowing rotations from any of the centers in $\mathcal{C}_{\sigma,w}$, not just from some $(-t,\ 0)$ corresponding to the weight in the weight-in embedding. Lemma 10 still applies in this case and the points in the upper half-plane move from left to right. We are now left with a purely geometric problem involving no w-mers or weights to track:

> What is the longest path $\{z_i\}$ possible where z_{i+1} is obtained from z_i by a rotation of $2\pi/w$ clockwise around a center from $\mathcal{C}_{\sigma,w}$, while staying in the upper halfplane at all times ($Im(z_i) \geq 0, \forall i$)?

We now break the problem into smaller stages as the weight-in embedding pass through rotation centers, defined as $\mathcal{C}_{\sigma,w} = \{(-j, 0) \mid 0 \leq j \leq W_{\max}\}$, the set of points that $Q(x)$ could possibly rotate around regardless of t. As there are $W_{\max}+1$ rotation centers and the maximum $\text{Re}(Q(x)) = \text{Re}(P(x))$ for any w-mer is also $W_{\max}$, we define $2W_{\max}$ subregions, two between any adjacent pair. Formally:

Definition 6 (Half Subregions). A subregion is defined as the area $[-j,\ -j+0.5) \times [0,\ W_{\max}]$ called a left subregion or $[-j\ +\ 0.5,\ \ -j\ +\ 1) \times [0,\ W_{\max}]$ called a right subregion, for $0 < j \leq W_{\max}$.

We now define the problem of finding the longest path, localized to one left subregion, as follows:

Definition 7 (Longest Local Trajectory Problem). Define the feasible region $(0,\ 0.5) \times [0,\ W_{\max}]$, and relaxed rotation centers $\mathcal{C}' = \{(j, 0) \mid -W_{\max} \leq j \leq W_{\max}\}$. A feasible trajectory is a list of points $\{z_i\}$ such that each point is in the feasible region, and z_i can be obtained by rotating z_{i-1} around $c \in \mathcal{C}'$ clockwise by $2\pi/w$ degrees. The solution is the longest feasible trajectory.

Again, note that this new definition is a purely geometric problem involving no w-mers and no weights $W(x)$ to track. z_i might stagnate if it coincides with one of the rotation centers, so we do not allow $\text{Re}(z_i) = -j$ in this geometric problem. Still, it suffices to solve this simpler problem, as indicated by the following lemma:

Lemma 11. *For fixed w and σ, if the solution to the problem in Definition 7 is L, the longest path in the de Bruijn graph avoiding* $\mathcal{M}_{\sigma,w}$ *is upper bounded by* $4W_{\max}L + O(w^2) = O(wL + w^2)$.

We prove this lemma in Supplementary Section S2.

3.4.3. Backtracking, heights, and local potentials.

In this section, we prove $L = O(w^2)$. We frequently switch between polar and Cartesian coordinates in this section and the next section. For simplicity, let $r(z)$ and $\phi(z)$ denote the radius and the polar angle of z written in polar coordinate.

Lemma 12. *Any feasible trajectory within the region* $(0,\ d] \times [0,\ W_{\max}]$ *for* $d \leq 0.5$ *is at most* $O(dw^3)$ *long.*

Proof. The key observation is if a rotation is not around the origin, $\text{Re}(Q(x))$ increases by $\Omega(1/w^2)$.

To see this, assume $(d,\ \theta)$ is the polar coordinate of $\text{Re}(Q(x))$ with respect to the rotation center. The polar coordinate for $\text{Re}(Q(S_a(x)))$ is then $(d,\ \theta - 2\pi/w)$. We note that $d \geq 0.5$ as $Q(x)$ satisfies $0 < \text{Re}(Q(x)) < 0.5$ and is at least 0.5 away from any other rotation centers. The difference in real coordinate is $d(\cos(\theta - 2\pi/w) - \cos(\theta)) = 2d\sin(\theta - \pi/w)\sin(\pi/w)$. Now, we require $\theta \in [0,\ \pi]$ and $\theta - 2\pi/w \in [0,\ \pi]$, so $\sin(\theta - \pi/w) \geq \sin(\pi/w)$ and the whole term is lower bounded by $2d\sin^2(\pi/w) = \Omega(d/w^2) = \Omega(1/w^2)$.

Only $O(dw^2)$ rotations not around origin are possible in the defined region, otherwise $\text{Re}(Q(x))$ would increase by $\Omega(dw^2)\Omega(1/w^2) = \Omega(d)$ already. Between two rotations not around the origin, only $w/2$ rotations around the origin can happen, or the point would have rotated π degrees and cannot stay in the upper half-plane. This means the possible number of pure rotations is $O(dw^3)$, which is also the asymptotic upper bound of path length. □

This lemma is sufficient to prove $L = O(w^3)$ and a total number of steps of $O(w^4)$. To obtain $L = O(w^2)$, we need a potential-based argument. Define $u = 1 - r_w^{-1}$, and let $s(z)$ be the lowest point above the real axis of form $z + ju$ where $j \in \mathbb{Z}$. We can show that a potential function of form $E(z) = -wr(s(z))/\pi + \phi(s(z))$ is guaranteed to decrease by at least $2\pi/w$ every rotation, and it can only decrease by $O(w)$ total inside the feasible region, which would complete the proof. This proof can be found in Supplementary Section S3.

3.5. Lower bounding the remaining path length in Mykkeltveit sets

We provide here a constructive proof of the existence of a $\Omega(w^2)$ long path in the de Bruijn graph after removing $\mathcal{M}_{\sigma,w}$. Since all w-mers in $\mathcal{M}_{\sigma,w}$ satisfy $\mathrm{Im}(P(x)) \leq 0$, a path satisfying $\mathrm{Im}(P(x)) > 0$ at every step is guaranteed to avoid $\mathcal{M}_{\sigma,w}$ and our construction will satisfy this criterion. It suffices to prove the theorem for binary alphabet as the path constructed will also be a valid path in a graph with a larger alphabet. We present here the constructions for even values of w.

We need an alternative view of w-mers in this section, close to a shift register. Imagine a paper ring with w slots, labeled tag 0 to tag $w-1$ with content $y = y_0 y_1 \cdots y_{w-1}$, and a pointer initially at 0. The w-mer from the ring is $y_j y_{j+1} \cdots y_{w-1} y_0 \cdots y_{j-1} = y[j,\ w-j] \cdot y[0,\ j]$, assuming pointer is at tag j. A pure rotation $R(x)$ on the ring is simply moving the pointer one base forward, and an impure one $S_a(x)$ is to write a to y_j before moving the pointer forward.

Let $w = 2m$. We create $\lceil w/8 \rceil$ ordered quadruples of tags taken modulo w: $Q_j = \{a-j,\ a+j,\ b-j,\ b+j\}$ where $j \in [1,\ \lceil w/8 \rceil]$, $a = m-1$, and $b = w-1$. In each quadruple Q_j, the set of associated root of unity r_w^{i+1} for the four tags is of form $\{-e^{-i\theta},\ -e^{i\theta},\ e^{-i\theta},\ e^{i\theta}\}$, adding up to 0. Consequently, changing y_k for each k in Q_j from 1 to 0 does not change the resulting embedding. The strategy consists of creating "pseudo-loops": start from a w-mer, rotate it a certain number of times, and switch the bit of the w-mer corresponding to the index in a quadruple to 0 to return to almost the starting position (the same position in the plane but a different w-mer with lower weight).

More precisely, the initial w-mer x is all ones but x_{w-1} set to zero, with paper ring content $y = x$ and pointer at tag 0. The resulting w-mer satisfies $P(x) = -1$. The sequence of operations is as follows. First, do a pure rotation on x. Then, for each quadruple Q_j from $j = 1$ to $j = \lceil w/8 \rceil$, we perform the following actions on x: pure rotations until the pointer is at tag $a-j$, impure rotation S_0, pure rotations until the pointer is at tag $a+j$, impure rotation S_0, pure rotations until pointer is at tag $b-j$, impure S_0, pure rotations until pointer is at tag $b+j$, impure S_0.

Each round involves exactly $w+1$ rotations since the last step is to an impure rotation S_0 at tag $b+j$, which increases by one between quadruples Q_j and Q_{j+1}. The total length of the path over all Q_i is at least cw^2 for some constant c. Figure 3 shows an example of quadruples and a generated long path that fits in the upper half-plane.

The correctness proof for the construction is presented in Supplementary Section S4 and the construction for odd w is presented in Supplementary Section S5.

4. DISCUSSION

4.1. Relationship between UHS and selection schemes

Our construction of a UHS of relative size $O(\ln(w)/w)$ and remaining path length w also implies the existence of a forward selection scheme with density $O(\ln(w)/w)$, only a $\ln(w)$ factor away from the lower bound on density achievable by forward and local schemes.

Unfortunately, this construction does not apply for an arbitrary UHS. In general, given a UHS with relative size d and remaining path length w, it is still unknown how to construct a forward or a local scheme with density $O(d)$. As described in Section 3.1, we can construct a UHS from a scheme by taking the set $\mathcal{C}_f$

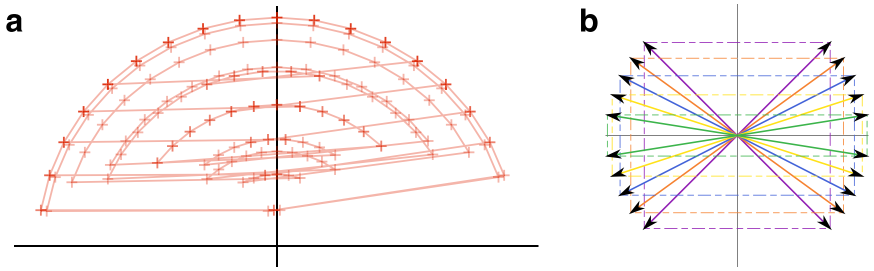

FIG. 3. **(a)** For $w = 40$, each set of four arrows of the same color represents a quadruple set of root of unity. There are a total of five sets. They were crafted so that the four vectors in each set cancel out. **(b)** The path generated by these quadruple sets. The top circle of radius 1 is traveled many times (between tags r_1 and r_2 in each quadruple), as after setting the 4 bits to 0, the w-mer has the same norm as the starting point.

of contexts yielding new selections. However, it is not always possible to go the other way: there are UHS that are not equal to a set of contexts $\mathcal{C}_f$ for any function f.

We are thus interested in the following questions. Given a UHS U with relative size d, is it possible to create another UHS U' from U that has the same relative size d and corresponds to a local scheme (i.e., there exists f such that $U' = \mathcal{C}_f$)? If not, what is the smallest price to pay (extra density compared to relative size of the UHS) to derive a local scheme from UHS U?

4.2. Existence of ''perfect'' selection schemes

One of the goals in this research is to confirm or deny the existence of asymptotically ''perfect'' selection schemes with a density of $1/w$, or at least $O(1/w)$. A study of UHS might shed light on this problem. If such a perfect selection scheme exists, asymptotic perfect UHS defined as $(O(1/w), w)$-UHS would exist. On the contrary, if we denied the existence of an asymptotic perfect UHS, this would imply nonexistence of a ''perfect'' forward selection scheme with density $O(1/w)$.

4.3. Asymptotic results and practical uses of minimizer schemes

This line of research places more focus on asymptotic densities of minimizers (and naturally, asymptotic densities of local schemes, and asymptotic relative proportion and path length of UHS). That is, we focus on characterization of these quantities in the limit where w and k, the window length (number of k-mers in a window) and the length of a k-mer, go to infinity. On the contrary, for current practices, the values of w and k are relatively small, with w below 100 and k below 30 for the vast majority of use cases. While our results are not immediately useful to analyze these practical scenarios as we do not attempt to determine the constants behind the big-O notation, we believe that further refinement of our approaches can close the gap between theory and practice, and make rigorous analyses possible for practical minimizer and/or local schemes.

4.4. Remaining path length of minimum decycling sets

There is more than one decycling set of minimum size (MDS) for given w. The Mykkeltveit set (Mykkeltveit, 1972) is one possible construction, and a construction based on very different ideas is given in Champarnaud et al. (2004). The number of MDSs is much larger than the two sets obtained by these two methods. Empirically, for small values of w, we can exhaustively search all the MDSs on the binary alphabet: for $2 \leq w \leq 7$ the number of MDSs is, respectively, 2, 4, 30, 28, 68 288, and 18 432.

While experiments suggest that the longest remaining path in a Mykkeltveit depathing set defined in the original article is around $\Theta(w^3)$, matching our upper bound, we do not know if such bound is tight across all possible minimal decycling sets. The Champarnaud set seems to have a longer remaining path than the Mykkeltveit set, although it is unknown if it is within a constant factor, bounded by a polynomial of w of different degree, or is exponential. More generally, we would like to know what is the range of possible remaining path lengths as a function of w over the set of all MDSs.

AUTHOR DISCLOSURE STATEMENT

H.Z. declares no competing financial interests. C.K. is a cofounder of Ocean Genomics, Inc. G.M. is V.P. of software development at Ocean Genomics, Inc.

FUNDING INFORMATION

This work was partially supported, in part, by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through grant GBMF4554 to Carl Kingsford, by the U.S. National Science Foundation (CCF-1256087, CCF-1319998), and by the U.S. National Institutes of Health (R01GM122935).

SUPPLEMENTARY MATERIAL

Supplementary Material

REFERENCES

Champarnaud, J.-M., Hansel, G., and Perrin, D. 2004. Unavoidable sets of constant length. *Int. J. Algebra Comput.* 14, 241–251.

Chikhi, R., Limasset, A., and Medvedev, P. 2015. Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics* 32, i201–i208.

DeBlasio, D., Gbosibo, F., Kingsford, C., et al. 2019. Practical universal K-mer sets for minimizer schemes, 167–176. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB'19. ACM, Niagara Falls, NY, New York, NY.

Deorowicz, S., Kokot, M., Grabowski, S., et al. 2015. KMC 2: Fast and resource-Frugal k-Mer counting. *Bioinformatics* 31, 1569–1576.

Golomb, S.W. 2014. Nonlinear shift register sequences, 110–168. *In Shift Register Sequences*. https://www.worldscientific.com/worldscibooks/10.1142/936. World Scientific. Singapore.

Grabowski, S., and Raniszewski, M. 2013. Sampling the suffix array with minimizers, 287–298. *In* Iliopoulos, C., Puglisi, S., and Yilmaz, E, eds. *String Processing and Information Retrieval*. *Lecture Notes in Computer Science 9309.* Springer International Publishing. Switzerland.

Jain, C., Dilthey, A., Koren, S., et al. 2017. A fast approximate algorithm for mapping long reads to large reference databases, 66–81. *In* Sahinalp, S.C., ed. *Research in Computational Molecular Biology*. *Lecture Notes in Computer Science*. Springer International Publishing. Switzerland.

Li, H., and Birol, I. 2018. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* 34, 3094–3100.

Lothaire, M. 2002. *Algebraic Combinatorics on Words,* vol. 90. Cambridge University Press. Cambridge, United Kingdom.

Marçais, G., DeBlasio, D., and Kingsford, C. 2018. Asymptotically optimal minimizers schemes. *Bioinformatics* 34, i13–i22.

Marçais, G., Pellow, D., Bork, D., et al. 2017. Improving the performance of minimizers and winnowing schemes. *Bioinformatics* 33, i110–i117.

Marçais, G., Solomon, B., Patro, R., et al. 2019. Sketching and sublinear data structures in genomics. *Annu. Rev. Biomed. Data Sci.* 2, 93–118.

Mykkeltveit, J. 1972. A proof of Golomb's conjecture for the de Bruijn graph. *J. Comb. Theory Ser. B* 13, 40–45.

Orenstein, Y., Pellow, D., Marçais, G., et al. 2016. Compact universal K-mer hitting sets, 257–268. In *Algorithms in Bioinformatics*. *Lecture Notes in Computer Science.* Cham, Springer.

Roberts, M., Hayes, W., Hunt, B.R., et al. 2004a. Reducing storage requirements for biological sequence comparison. *Bioinformatics* 20, 3363–3369.

Roberts, M., Hunt, B.R., Yorke, J.A., et al. 2004b. A preprocessor for shotgun assembly of large genomes. *J. Comput. Biol.* 11, 734–752.

Schleimer, S., Wilkerson, D.S., and Aiken, A. 2003. Winnowing: local algorithms for document fingerprinting, 76–85. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. SIGMOD'03*. ACM. New York, New York, USA.

Ye, C., Ma, Z.S., Cannon, C.H., et al. 2012. Exploiting sparseness in de novo genome assembly. *BMC Bioinformatics* 13, S1.

Address correspondence to:
Dr. Guillaume Marçais
Computational Biology Department
Carnegie Mellon University
Pittsburgh, PA 15213
USA

E-mail: gmarcais@cs.cmu.edu