Theory and Practice of Low-Density Minimizer Sketches

Hongyu Zheng

CMU-CB-22-100
February 2022

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Carl Kingsford, Chair
Guillaume Marçais
Hosein Mohimani
Maria Chikina, University of Pittsburgh
Ron Shamir, Tel Aviv University

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

This material is based upon research supported by the National Institute of General Medical Sciences of the National Institutes of Health under award number R01GM122935, The Shurl and Kay Curci Foundation, and a gift from Schmidt Sciences.

The department specifically disclaims responsibility for any analyses, interpretations, or conclusions. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, donors or the U.S. Government.

Keywords: Sequence Sketch, Sequence Analysis, Combintorial Algorithms, String Algorithms

For this cruel world.

Abstract

Sequence sketch methods generate compact fingerprints of large string sets for efficient indexing and searching. Minimizers are one of such sketching methods, sampling k-mers from a string by selecting the minimal k-mer from each sliding window with a predetermined ordering. Minimizers sketches preserve information to detect sufficiently long substring matches. This favorable property of minimizers, and its ease of implementation, lead to wide adoption in a large number of software and pipelines, including read mappers, genome assemblers, sequence databases and more. Despite the method's popularity, many theoretical and practical questions regarding its performance remain open. This is especially true regarding the density of minimizers, a simple yet powerful metric for minimizer performance. We have neither understanding of density growth under various conditions, nor principled approaches to construct minimizers with provably low density. This dissertation attacks the lack of knowledge in minimizer sketches from multiple fronts.

In the first part, we are primarily concerned with asymptotic behavior for the density of minimizer sketches. Minimizers are parameterized by the k-mer length k, a window length w and an order on the k-mers. Using a number of new techniques, we are able to provide a complete picture of how the density of optimal minimizer grows with its parameters w and k. We also derive structural lemmas for universal hitting sets and local schemes, two highly related concepts for minimizer design. Together, these results serve as building blocks for future theoretical advances.

The next two parts are focused on constructing low-density minimizers in two scenarios. In the second part, we propose the Miniception, the first construction of minimizers that provably achieves better densities than a random minimizer in practical configurations of w and k. This method is also simple to implement and requires minimal overhead compared to existing implementations of random minimizers. In the third and final part, we consider optimization of minimizer density on a given reference sequence. The most common use case is when the given sequence is the reference genome or transcriptome. We propose the concept of polar sets, which itself is a complementary concept of universal hitting sets and similarly can be used to construct minimizers. Using polar sets, we propose efficient algorithms to directly optimize the density of minimizers on a reference sequence up to some error. Experiments show that both the Miniception and the polar set algorithms can reliably outperform existing methods for constructing low-density minimizers, without a reference and with a reference, respectively.

Acknowledgments

"My mom always said life was like a box of chocolates. You never know what you're gonna get." opens the legendary 1994 film *Forrest Gump*. To me, a box of *Bertie Bott's Every Flavour Beans* from the Harry Potter universe is probably a more accurate description of my Ph.D. journey. There are high highs, and low lows, just like there are strawberry flavor beans (my personal favorite flavor for a lot of things), and there are vomit flavor beans. I would not be able to get through such a rollercoaster without help from many others.

I'd like to thank my advisor, Prof. Carl Kingsford, first and foremost. He has been instrumental in my development as a Ph.D. candidate, leading me in development of research plans while leaving me with sufficient freedom to come up with novel ideas. I also appreciate Carl's unique vision of the research field; My bad habit of tunnel visioning in research hampers progress from time to time, and Carl has always been able to pull me out of it and show me the big picture. He is also a well-rounded researcher with deep expertise in coding, presenting (including giving elevator pitches) and paper writing. All in all, I cannot thank Carl enough for everything he has done in the past four years. I'm also eternally grateful to the other members of the thesis committee: Prof. Hosein Mohimani, Dr. Guillaume Marçais, Prof. Maria Chikina, Prof. Ron Shamir, for their service and guidance in the process.

Next up the list, I'd like to thank my co-authors: Dr. Guillaume Marçais (who also kindly agreed to serve on the thesis committee), Dr. Cong Ma, and Minh Hoang. Thank you all for the the high-quality collaboration, I really enjoy the whole process. Also kudos to Guillaume for keeping the Kingsford group server in a good shape. I'd also like to thank the remaining past and present members of the Kingsford group: Dr. Heewook Lee, Dr. Dan DeBlasio, Dr. Mingfu Shao, Dr. Prashant Pandey, Dr. Hongyi Xin, Dr. Brad Solomon, Dr. Natalie Sauerwald, Laura Tung, Yihang Shen, Yutong Qiu, Mohsen Ferdosi, Gweneth Ge, Daniel Bork, Raehash Shah, Shawn Baker and Yinjie Gao. I always have fun chatting all kinds of stuff and engage in research discussions in the weekly group meetings; I surely will miss it. I thank Cathy Su and Amir Alavi for being decent officemates for the past 2.5 years (even though the office is vacant most of the time due to COVID-19), as well as other people in CPCB and CMU CBD for contributing to a well-run Ph.D. program and an amazing academic department.

Outside of academics, my parents have always supported me emotionally and financially during the journey. I cannot make it through the Ph.D. years without having comfort in their unwavering support. And, thank you to my two long-time meal pals, Ruochi Zhang and Xiangrui Zeng, for introducing me to various eateries in Pittsburgh, gossiping together, and keeping me from going insane during the long and brutal COVID lockdown (and getting me to play Genshin Impact I guess). I wish you all a bright and fruitful future. Lastly, thanks to my long-time roommate Ruosong Wang for splitting the rent and utility bills throughout the Pittsburgh years, among other things.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Era of Big Data Genomics is Here

Computational Genomics is the research field where advanced statistical and algorithmic methods are used to analyze genomic sequence and associated data, to gain insight into how DNA and RNA sequences control the biology. This research field has seen a new wave of rapid growth in the past few years, thanks to a multitude of factors:

- Rapid advance of genomic sequencing technologies, including short read sequencers (second-gen or next-gen like Illumina) and long read sequencers (so-called "third-gen" such as Oxford Nanopore MinION and PacBio SMRT [47, 102, 121, 123]) results in exploding throughput of sequences.
- Emergence of single-cell RNA-sequencing [92, 104, 107, 114, 118, 120] and spatial transcriptomics [18, 29, 50, 68, 69, 78, 98, 105, 106, 113, 122] for more fine-grained detection of heterogeneity between cell populations, and better utilization of the advanced sequencing technologies as such graunlarity would not be possible without sufficient sequencing thoroughput.
- Democratization of computing and storage due to falling costs and increased efficiency of computing equipment, which in turn fosters large-scale (sometimes consortium-style) collaborative efforts for a diverse set of topics.

All of these advancements indicate computational genomics has entered the phase where sequencing data is abundant. As a simple indicator of the transformation undergoing this field, the SRA (Sequence Read Archive), one of the largest public repositories for sequencing datasets, is still on a healthy pace of exponential growth in both total sequence length and number of datasets even up to the year of 2021 [33], while the Moore's law (a similar thesis for computing power) has been considered failing [108] or dead [119]. Thus, arguably even more so than the computing field, we are undeniably at the big data era for computational genomics. This brings along a number of interesting research lines, for which access to big data is essential.

Pan-Genomics [109]. The classical approach to human genomics is to first establish a single reference genome, then for each new individuals, compare the new genome to the reference and resolve differences. Such method has historically worked well, because the variation between human genomes is typically small compared to its size – 0.1 percent is a common estimate. However, as we expand to non-modal species (although for this discussion we will limit ourselves to human genomics) and sequence more human genomes from diverse populations, the shortcomings of such approaches have emerged. For example, studies on different populations [4, 22, 27, 38, 42, 51] reveal that diverging proportion of genomes are missing from the commonly used human reference, which impose limitations on how to use these genomes. Thus, studies on a large collection of genomes, without a central notion of "reference", is very important. While large-scale projects like 1KGP (1000 Genomes Project [115]) and SGDP (Simons Genome Diversity Project [71]) are more focused on catalogue of genetic variants, efforts to construct a pan-genome for humans include national efforts for Icelandic [51], Netherland [12, 34] and African and Asian [65] populations. Pan-genomes are usually represented as genome graphs [36, 44, 93, 94] so the shared part of different genomes need not be stored multiple times. While pan-genome studies are more common in other species with smaller genomes now, for human pan-genome, analyses has revealed hidden parts of the the genome that are associated to certain diseases [51, 112].

Cancer Genomics [70]. Early efforts in cancer genomics rely on RNA microarrays, which while helped expanding our understanding in cancer subtyping [23, 91], are severely limited in that it cannot provide genome-wide base-pair information. The emergence of second-generation sequencing technologies enabled better profiling of cancer genomes, and international consortium like The Cancer Genome Atlas (TCGA) lead the effort to sequence multiple types of cancers. The TCGA has since grown to include more cancer types, with different types of sequencing data (including marker studies [39, 60, 61, 81] and more). Access to these comprehensive atlases of cancer genomes has allowed "pan-cancer-genomics" type of studies [2, 15, 48, 97, 117] where cancer induced genomic changes of different types can be jointly studied to reveal common biological pathways and processes.

1.2 Sequence Sketches are Essential Infrastructure

Underlying the big data paradigm of computational genomics, we face the fundamental problem of storing, indexing and comparing large genomic datasets. Without space and/or time-efficient representations of the dataset, these operations on large datasets are computationally infeasible both in time and in (disk or memory) space. Sequence sketching is a broad class of algorithms that specifically tackles this challenge by generating compact, (usually) lossy representations of large sequencing datasets. These methods constitute essential infrastructure underlying modern computational genomics.

The focus for this dissertation is the minimizer sketch, which is arguably one of the most widely-used sketching methods in modern computational genomics. To put minimizer sketches into context, in this section, we briefly discuss several other representative types of sequence sketches and how they are used in practice for computational genomics.

Sequence Indexes: Suffix Trees and FM-indexes. Suffix trees, suffix arrays [1, 37, 124] and FM-indexes [32] are data structures that allow for quick exact substring search, while potentially smaller in size compared to the underlying sequence. They are very popular for underlying sequence alignment and read mapping, a pair of closely related fundamental problem in computational genomics. A number of existing methods, including MUMmer3 [56], its successor MUMmer4 [76], Bowtie [58], Bowtie 2 [57], SOAP2 [64], and many others, rely on these data structures to quickly index massive genomes.

Locality-Sensitive Hashing and Set Sketching. Locality-sensitive hashes [41] are families of hash functions where the probability of hash collision $P(h(a) = h(b))$ equals a similarity measure $S(a, b)$ between elements. This can be useful for quick approximate neighbour queries, for which sequence alignment can be seen as a special instance. MHAP [8] and MashMap [43] follow this approach.

Methods like MinHash sketch [13], where the sketch of a given set is the minimum hash value from a random hash function, also fall into this category because the method itself can be seen as LSH for Jaccard distance. Ordered MinHash [77] refines the skeching method to obtain a gapped locality-sensitive hash (a relaxation of LSH definition) for edit distance. Mash [84] and sourmash [14] use LSH to perform quick metagenomic distance estimation, which is used for clustering and estimating number of species in metagenomic sequence mixtures.

Approximate Membership Queries. AMQ data structures, as its name suggests, supports lossy query of membership, the most famous one being the Bloom Filter [11], named after its inventor. Counting bloom filters [31], cascading bloom filters [55, 99] and quotient filters [7, 87] are improvements to bloom filters that support more sophisticated queries (most notably counting occurrences instead of presence) with better performance both space-wise and time-wise. These data structures also support a number of important primitives in computational genomics, such as k-mer counting. BFCounter [79] uses bloom filters as the underlying data structure. They also see some use in genome assembly algorithms [19, 55] and de Bruijn graph construction algorithms [86] to improve speed and sensitivity.

Sequence Bloom Trees [110] (and its descendants [111, 116]) and SeqOthello [125] extend the concept of AMQ to perform approximate k-mer presence queries among a large collection of sequencing libraries. Mantis [88] also falls into this category, although the algorithm allows exact queries.

1.3 Minimizers as Sequence Sketches

In this section, we discuss minimizer sketches as useful primitives for computational genomics. We present the definition of minimizer sketches, its important properties, and various applications that use minimizer sketches at their core.

1.3.1 Definition of Minimizer Sketches

Minimizer sketches [95, 96] are a class of sequence sketch methods that have seen extensive use. For the purpose of this dissertation, a minimizer sketch (sometimes shorthanded as minimizers) has three parameters: $(w, k, \mathcal{O})$. k is the length of the k-mers of interest while w is the length of

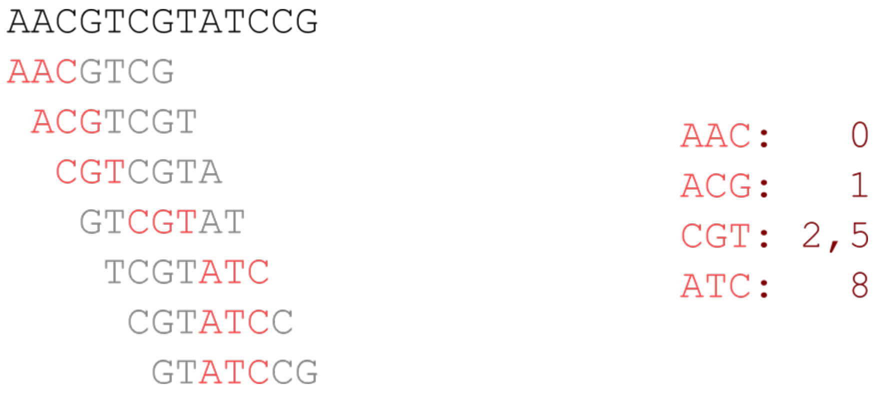

Figure 1.1: Example of a minimizer sketch, with $w = 5$, $k = 3$, and lexicographical orders. Left: The first line is the sequence, every following line is one of the windows, highlighted part is the k-mer selected. Right: The minimizer sketch of this sequence.

the *window*. For minimizers, a least one k-mer must be selected in any window of w consecutive k-mers, or equivalently in any substring of length $w + k - 1$. Finally, $\mathcal{O}$ is a total order (i.e., a permutation) of all the k-mers, and it determines how the k-mers are selected: in each window the minimizer selects the minimum k-mer according to the order $\mathcal{O}$ (hence the name minimizer), and in case of multiple minimum k-mers, the leftmost one is selected. See Figure 1.1 for a concrete example of a minimizer sketch.

1.3.2 Window Guarantee

A key property of minimizer sketches is what we call the *window guarantee*: If two sequences share a sufficiently long substring (as long as a window of the minimizer), their minimizer sketches share a k-mer from the shared part. See Figure 1.2 for a concrete example. It should also be noted that even if two sequences do not share a sufficiently long substring, their sketches might still share a k-mer. Specifically, if two sequences share a substring that are close to a full window's length, there is a good chance they will share a k-mer in the respective minimizer sketches.

1.3.3 Using Minimizers in Computational Genomics

The minimizer sketches are very versatile and have seen extensive use in a diverse range of applications. We discuss a few use cases that are most prominent and relevant to this dissertation.

Sequence Assembly. The minimizer sketches were first introduced to computational genomics [62, 95, 96] as a method to overlap reads in preparation for assembling contigs in the OLC (overlap-layout-consensus) genome assembly paradigm. In short, reads are organized into bins using their minimizers as index, and reads within the same bin are tested for overlapping using more accurate but slower methods.

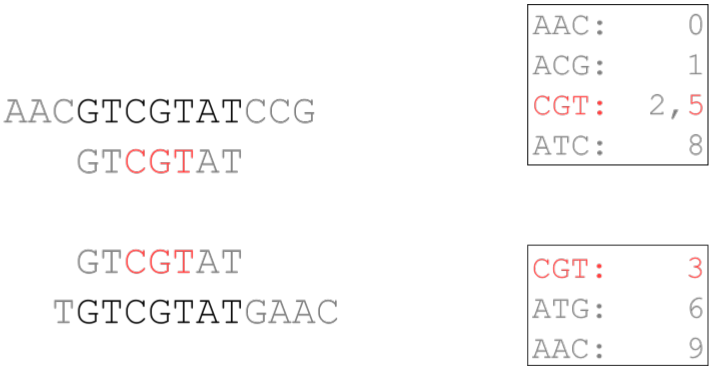

Figure 1.2: Example of window guarantees, with $w = 5$ and $k = 3$. As seen in this figure, the two sequences share a window (substring of length 7), and they are guaranteed to share a k-mer in their respective minimizer sketch. The shared k-mer is highlighted on the right-hand side of this figure.

Read Mapping. Minimap [62] and Minimap2 [63] are modern methods for read mapping that use minimizers to quickly find matches between the reads and the reference genome. Usually, the reference is processed first. The minimizers of the reference enter a hash table with associated locations in the reference. The reads are then processed through the same minimizer sketch. For each minimizer in the read sequence, if that k-mer is also present in the reference as a minimizer, the list of associated locations are used as potential locations for alignment. There are improvements to this approach [45, 46, 127] that also use (modified versions of) minimizers for fast filtering of potential alignment locations.

k-mer Counting. k-mer counting is another important problem in computational genomics, and the first step of these counting methods is usually to split k-mers into buckets then leverage parallel and cloud computing to quickly process each bucket. Minimizers are commonly used to determine the bucketing of k-mers [25, 30, 66, 90] by treating the k-mer as a window of a minimizer. This requires definition of a shorter seed length than k, which we denote as k_0. We thus select k_0-mer in a k-mer, then use the selected k_0-mer as the bucket key. Since minimizer sketches are deterministic, this splitting will never result in same k-mer being dispatched to two different buckets. Additionally, adjacent stretches of k-mers in the same bucket are merged into sequences longer than k called super k-mers. Using super k-mers can greatly decrease space and time cost of k-mer counting algorithms, enabling them to scale to larger sequence collections with longer k-mers.

de Bruijn graph construction. Given a set of sequences and a value k, one can construct the de Bruijn graph of order k, a directed graph where nodes indicate distinct k-mers and edges indicate adjacency between k-mers (more concretely, given an input sequence, for each of the sequence's $(k + 1)$-long substring, there is an edge from its k-long prefix to its k-long suffix). In this graph, it is guaranteed that each input sequence can be reconstructed by walking on the resulting graph. This property makes de Bruijn graphs desirable for genome assemblies [126], as the process of assembling genomes can be seen as spelling out sequences on the de Bruijn graph constructed from input reads. However, construction of de Bruijn graph is not a straightforward matter and can be costly if not implemented carefully. BCALM [20] and its successor, BCALM2 [21] are efficient methods to construct de Bruijn graphs. These methods split k-mers into different buckets with minimizers, similar to how k-mer counters split k-mers, then create maximal contigs from each buckets before merging contigs from different buckets to create a compact representation of the de Bruijn graph.

1.4 Minimizers as Combintorial and Optimization Objects

In this section, we discuss minimizer sketches as combintorial objects that can be optimized. Thus, we present parameterization of minimizer sketches, the objective (density as suggested by disseratation title), and existing methods that perform such optimization.

1.4.1 Optimizing a Minimizer Sketch

A minimizer sketch has three parameters: $\{w, k, \mathcal{O}\}$. We are focused on optimizing $\mathcal{O}$ throughout this dissertation, on a wide range of w and k. It should be noted though that we do not directly select or choose the value of w and k. Selection of these two parameters are highly application-specific and is outside the scope of this dissertation. Recall that $\mathcal{O}$ is a total order (i.e. a permutation) of all k-mers, and it determines how the k-mers are selected. Throughout this dissertation, when we talk about optimization of minimizer sketches, we mean the optimization of this total order with fixed w and k, unless otherwise specified. The search space for this total order is astronomical. There are $O(4^k!)$ possible total orders in a universe of k-mers assuming an alphabet of 4, and brute-force search methods become infeasible even for $k = 2$.

Baseline Methods. When we optimize a minimizer sketch, we need a reference or baseline against which to compare. In this dissertation, the most common baselines we use are *Lexicographical minimizers* and *Random minimizers*. A lexicographical minimizer sketch is simply a minimizer sketch where $\mathcal{O}$ is the lexicographical order; For example, the minimizer as shown in Figure 1.1 is a lexicographical one ($AAA < AAC < AAT \cdots < TTT$). A random minimizer sketch is, strictly speaking, a distribution of minimizer sketches, where each possible ordering gets equal weight. When we measure the performance of a random minimizer sketch, we mean its expected performance over the randomness of the ordering. Random minimizer sketches are widely used, and in practice, they are implemented by first obtaining a pseudo-random hash function with certain guarantees of randomness, then implement $\mathcal{O}$ as the implied total order from hash values. In next section, we will discuss other more sophisticated constructions of minimizer sketches against which we compare our new construction approaches.

1.4.2 Density of Minimizers as Measurement of Efficiency

Sequence-Blind Setting. The core measure of performance we consider for minimizer is the *density*. To calculate density, we first assume a sufficiently long sequence where the character at each location is independently selected from the alphabet with equal probability (for example, using the genomic alphabet, each character has 25 percent chance of being A, C, G or T independent of everything else). The density of the minimizer sketch is the expected number of k-mers in the sequence that is the minimizer of at least one window, divided by the number of k-mers in the string. In the case where the string is sufficiently long, we can also use the length of the string to approximate the number of k-mers. This randomization process ensures that each minimizer will have a unique density value, and a reference sequence is not required for evaluation. The setup where we choose the ordering of k-mers to optimize this density value is called *sequence-blind* throughout this dissertation.

In general, minimizer sketches with lower densities are desirable as fewer selected positions imply reduction in the run time or memory usage of applications using minimizers, while preserving the window guarantee with the same parameter. For example, a long-read aligner like Minimap2 [63] stores the positions of every selected k-mer of a reference genome to find exact match anchors for seed-and-extend alignment. (The parameters k and w are constrained by the required sensitivity of alignment in this case, and we do not tune them.) A minimizer sketch with lower density thus offers both lower storage and memory cost, and faster runtime due to fewer anchors to check for match and fewer pairs of anchors matched. Similar arguments can be made for sequence assemblers where lower density sketches lead to better runtime. These statements can be made explicit under the assumption where the reads are sampled uniformly from the reference sequence. For a k-mer counter, minimizer sketches with lower density naturally imply more even spacing of anchors which implies less space wasted for duplicate anchors, and similarly would lead to improved computational performance.

For given parameters w and k, the choice of the order $\mathcal{O}$ changes the (expected) density. The density of a minimizer is lower bounded by $1/w$, where exactly 1 position per window is selected, and upper bounded by 1, where every position is selected. A minimizer sketch with a density of $1/w$ may not exist for every choice of w and k, and how to find an order $\mathcal{O}$ with the smallest possible density for any w and k efficiently is still an open question due to the astronomical search space of this problem.

Sequence-Specific Setting. Instead of calculating the average density over a random input string, we can also calculate density for one particular string, that is, the number of positions selected divided by the number of k-mers on an input string. Similarly, when the string is long enough, we replace the divisor by the length of the string for simplicity. The resulting density value is called *specific density*, and the setting where we optimize this value (with a fixed input sequence) is called *sequence-specific*.

When applying minimizers in computational genomics, in many scenarios the sequence is known well in advance and it does not change very often. For example, a read aligner may align reads repeatedly against the same reference genome (e.g., the human reference genome). In such cases, optimizing the density on this specific sequence is more meaningful than on a random sequence. Moreover, the human genome has markedly different properties than a random sequence and optimization for the average case may not carry over to this specific sequence. In the read aligner example, a minimizer with lower specific density leads to a smaller index to save on disk and fewer seeds to consider in the seed-and-extend alignment algorithm while preserving the same sensitivity thanks to the window guarantee (as w is unchanged; See Section 1.3.2).

Low specific density similarly provides various benefits to many downstream applications as we have discussed in the previous subsection. It is worth noticing that specific density can be much lower than the (average) density when the minimizer sketch is properly optimized, and optimization of specific density and optimization of (expected) density can be very different problems.

1.4.3 Universal Hitting Sets

Optimization of low-density minimizer sketches, especially in the sequence-specific setup, has long been interwined with optimization of compact universal hitting sets, and in this section we provide a brief overview of the concept to prepare for the upcoming discussion.

A Universal Hitting Set (UHS) is a set of k-mers such that every long enough string (of length $w + k - 1$ or longer) contains an element from the set as a substring. More formally, for any window containing w k-mers, the window would always contain one of the k-mers in the set regardless of the window content. We call such a set a *universal hitting set* for parameters w and k. From the combintorial perspective, these are sets of (constant length) unavoidable words [17], i.e., words that must be contained in any long strings.

Universal hitting sets and unavoidable sets have many interesting properties of their own. However, we are most interested in their interplay with minimizer sketches.

There is a two-way correspondence between minimizer sketches and universal hitting sets: each minimizer sketch has a corresponding UHS, and a UHS defines a family of *compatible* minimizer sketches [73, 74]. A compatible minimizer sketch with a UHS is a sketch with identical k and w to the UHS, and with ordering $\mathcal{O}$ satisfying that any k-mer in the UHS compares less than any k-mer outside the UHS. 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 any compatible minimizer sketch: the density is no more than the relative size (the size of the UHS divided by number of possible k-mers) of the universal hitting set [85].

1.4.4 Early Work on Low-Density Minimizers

As discussed before, random minimizer sketches serve as a reasonable baseline, and indeed it has been widely used in different contexts. Previous work on analysis and construction of low-density minimizers can be roughly split into two groups. The first group of results focus on the asymptotic behaviour of minimizer sketches in sequence-blind scenarios, and the second group of results focus on the practical performance in both sequence-blind and sequence-specific setups (measuring both density and specific density), with slightly more focus on sequence-specific setups.

Winnowing Bounds and Improvements. The original winnowing paper [103] gave two results about the density of minimizer sketches (under the sequence-blind setting): under some simplifying assumptions, (1) the density of a random minimizer sketch, regardless of w and k, is $2/(w+1)$ and (2) the density of any minimizer sketch is lower-bounded by $1.5/(w+1)$. Although these estimates are useful in practice, they are dependent on some hidden assumptions and do not represent the behavior of minimizer sketches in all cases.

These results have since been refined in multiple ways by looking at the asymptotic behavior of minimizers, and by considering the extreme cases with $w \gg k$ (that is, w is much larger than k) and $k \gg w$. First, when w is fixed and $k \gg w$, construction of minimizer sketches with density of $1/w + O(ke^{-\alpha k})$ is possible for some $\alpha > 0$ [74]. Such construction has density arbitrarily close to the optimal $1/w$ for sufficiently large k. The apparent contradiction between this result and the previously stated lower bound of $1.5/(w+1)$ stems from a hidden assumption: k should not be too large compared to w. Second, it is shown that when k is fixed and $w \gg k$, the density of any minimizer is $\Omega(1)$. Hence, a density of $2/(w+1)$, or even $O(1/w)$, as previously stated does not apply when $w \gg k$ [74]. This is similarly because the previous estimate relies on a hidden assumption: k should not be too small compared to w.

Practical Constructions of Minimizer Sketches. Most of the prior methods for constructing practical low-density minimizer sketches take the route of creating UHS-compatible minimizer sketches. That is, they first construct a compact universal hitting set (small cardinality or relative size, implying tight density upper bounds), then use a compatible minimizer sketch for the actual sketching part. The DOCKS paper [85] proposes the idea of using universal hitting sets to construct low-density minimizers, and establishes the connection between the relative size of UHSes and density of compatible minimizer sketches. The DOCKS [85] and ReMuVal [24] algorithms are heuristics to generate compact universal hitting sets (again, these are UHSes with small relative size leading to low density). Both algorithms use the Mykkeltveit set [82], a special subset of k-mers that can be seen as a minimal UHS with sufficiently large window length w, as a starting point. ReMuVal algorithm was proposed to construct UHSes with larger k from smaller k, and it starts with a Mykkeltveit set with a lower order. These methods are later improved by Ekim et al. [28] to enable better scaling. It enables construction of UHSes up to $k = 16$, at which point even storing the universal hitting sets becomes a significant issue.

Depending on the application, some other variants of minimizer sketches have been proposed. These sketches are not explicitly designed with minimizing density in consideration. Frequency-based orders are examples of sequence-specific minimizer sketches [21, 46]. In these constructions, k-mers are ordered by their occurrence frequency, with rarer k-mers placing lower in the minimizer ordering. The intuition is to select rare k-mers as they should be spread apart in the sequence, hence giving a sparse sampling. Winnowmap [46] uses a two-tier classification (very frequent vs. less frequent k-mers), and combined with a weighted hash function to introduce randomness into the ordering, achieves better performance compared to strictly frequency-based orders. Finally, Syncmers [26] are generalizations of minimizer sketches for which the window guarantee does not necessarily hold. They are shown to be effective sketches for sequence identity conservation in several applications.

1.5 Our Contribution

1.5.1 Theoretical Foundation of Low-Density Minimizer Sketches

The prior theoretical work on universal hitting sets and minimizer sketches are severely lacking. We have very limited idea how minimizer sketches perform (measured in density) outside of very extreme cases (except the case where we make some assumptions). Finding *asymptotically optimal* minimizer sketches, that is, minimizer sketches with density $O(1/w)$, has also been long an open problem. Our first major contribution is a complete resolution of this open problem. More specifically, we show that

- A sufficient and necessary condition that asymptotically optimal minimizer sketch exists: When we treat k as a growing function of w, the length of the k-mers needs to grow at least as fast as the logarithm of the window size.
- The lexicographcial minimizer is in fact asymptotically optimal.

The second result is interesting, in the sense that it reveals a sizable gap between the theory and the practice of minimizers. The lexicographic minimizers are optimal for asymptotic density. However, in practice, they are usually considered the worst minimizers and are avoided. Another example of such gap is that prior state-of-the-art methods for practical low-density minimizers (like those generating low-density minimizer sketches with UHSes) only provide weak density guarantees.

We also strengthen the statement in Schleimer et al. [103], which says the density of a random minimizer is $2/(w+1)$ under some assumptions on k-mer distribution. We are able to provide a theorem to the same effect, but with very explicit assumptions on the value of k. Although still in asymptotic small-o notation, this provides assurance that practical random minimizer sketches indeed achieve density close to $2/(w+1)$.

Our contribution to the theoretical front of minimizer research also includes interesting structural lemmas for Mykkeltveit sets and local schemes.

Mykkeltveit sets, as described before, are minimal UHSes when the windows are sufficiently long. More precisely speaking, they are *minimal decycling sets* for the corresponding de Bruijn graph. A natural question is thus how long are the remaining paths in the de Bruijn graph after removing the Mykkeltveit k-mers, and more specifically, if the remaining path length is $O(k)$. If that hypothesis were true, we would be able to construct instances of asymptotically optimal minimizers by simply using minimizer schemes compatible with the original Mykkeltveit set (without any of the modifications that are done in existing methods). We answer this in the negative, and show that the remaining path length is asymptotically bounded between k^2 and k^3. To achieve this, we develop a novel complex plane embedding of the de Bruijn graph, where edges become rotations on the complex plane and a potential function characterizes paths in the de Bruijn graph not touching Mykkeltveit k-mers.

Finally, local schemes are natural extensions of minimizer schemes. They can be seen as the minimal implementation of sequence location selection methods that satisfy the window guarantee, i.e. if two sequences share a window, the same k-mer is present in the respective sketches. More concretely, a local scheme is an arbitrary function from window sequences to a location within the window. We establish several structural lemmas between UHSes, local schemes and minimizers, including an explicit two-way connection between UHSes and local schemes, which provides some hint that optimizing a local scheme might not be significantly harder than optimizing a UHS.

In general, these results give a solid theoretical foundation for better characterization and optimization of low-density minimizers, compact universal hitting sets, and potentially future low-density local schemes.

1.5.2 Sequence-Blind Low-Density Minimizer Sketches

As revealed in the last section, random minimizer sketches have approximately density of $2/(w+1)$ in most practical cases; An intuitive explanation is that a random minimizer selects around two k-mers per window. It is natural to wonder how can we beat this baseline, that is, if we can get a minimizer sketch that selects less than two k-mers per window (preferably down to one k-mer, which is the absolute minimum to still meet the window guarantee).

Earlier analyses [74] proved that when $k \gg w$, constructions of minimizer sketches exist that achieve almost optimal density of $1/w$ (one k-mer per window), but the scenario is unlikely in practice. Another analysis [73] can potentially bound density of universal hitting set derived minimizer sketches away from $2/(w+1)$. However, closer inspection reveals that the approach depends on certain assumptions that are likely not true for practical scenarios (See Section 4.2.1). Earlier methods based on universal hitting sets are able to empirically achieve lower density, but there are no guarantee on the relative size of the universal hitting set, and thus no guarantee on the density. It was an open question whether there exists a minimizer sketch that is strictly better than a random minimizer sketch outside of the restrictive configuration of $k \gg w$.

We answer this open question in the affirmative. We provide a construction of minimizer sketches called the *Miniception*, with expected density on a random string of $1.67/w + o(1/w)$ when $k \approx w$. This is an example of minimizer sketches with guaranteed density $< 2/(w+1)$ that works for infinitely many w and k, not just an ad-hoc example working for one or a small number of parameters w and k. This is also the first example of a family of minimizers with guaranteed expected density $< 2/(w+1)$ that works when $k \approx w$ instead of the less practical case of $k \gg w$.

The construction and analysis of the Miniception critically relies on a concept called *charged windows* with related properties. More specifically, a charged window of a minimizer is a window where the first or the last k-mer is selected as the minimizer. In the first part of the dissertation, we show that the set of charged windows is itself a universal hitting set over $(w+k-1)-$mers. (This is in fact a special case of the local schemes' structural lemma, as noted in the last section.) The name Miniception pays tribute to the 2010 movie *Inception* directed by Christopher Nolan, where the stage is set at dreams that are within other dreams. In constructing the Miniception, we are essentially taking one minimizer sketch and using its set of charged windows to build another minimizer sketch, thus having a minimizer inside a minimizer (and just like the setup in the movie, the process can be repeated if desired).

Moreover, unlike other methods with low density in practice [24, 28, 85], the Miniception does not require the use of expensive heuristics to precompute and store a large set of k-mers. Selecting k-mers with the Miniception is as efficient as a selecting k-mers with a random minimizer using a hash function and does not require any additional storage. It is our belief that the

Miniception is a versatile method and can see more use in practice.

1.5.3 Sequence-Specific Low-Density Minimizer Sketches

Most of prior work on low-density minimizers concentrates on the sequence-blind scenario. The sequence-blind scenario focuses on the "average" scenario where the input sequence for sketching is essentially random. For computational genomic applications though, the more "average" case might be where we are sketching a particular sequence such as the reference genome, which falls into the sequence-specific scenario as we describe before. However, constructions of low-density minimizer sketches have been much less understood in sequence-specific scenario, and there are cases where sequence-specific density can be much lower than the (expected) density for a minimizer sketch. Additionally, the Miniception sees its performance degrade significantly when w becomes larger than k, which is in fact a common configuration in practice. Constructions of low-density minimizers in these configurations, even if only for sequence-specific scenarios, are desirable.

A second motivation for this aim is the prevalence of universal hitting set based approaches. While the framework is elegant, it has critical shortcomings making it infeasible for analysis in many scenarios. In general, prior methods for low-density minimizer schemes start with construction of a compact universal hitting set. While the relative size of the UHS is indeed a upper bound of the resulting density, the bound is one-sided and can in many cases become loose. In such scenarios, the existing methods risk optimizing for a irrelevant objective. Worse, when the window length is somewhat large compared to k-mer length ($w \geq 2k$ approximately), the density bound provided by UHS guarantees are essentially useless. Alternatives to this framework can be helpful for method development of low-density minimizer sketches.

In this part of the dissertation, our contribution is twofold.

First, we propose the concept of *polar sets*. They can be seen as complementary ideas to the universal hitting sets: while a UHS is a set of k-mers that intersects with every w-long window *at least* once, a polar set is a set of k-mers that intersect with any window *at most* once. The name "polar set" is an analogy to a set of polar opposite magnets that cannot be too close to one another. More succinctly speaking, our construction builds upon sets of k-mers that are sparse in the sequence of interest. This property allows us to derive tight bounds on sequence-specific density of compatible minimizer sketches, using simple measurements called the *link energy* over the polar sets. In comparison, while we can simply measure the size of a universal hitting set and use it as a upper bound on the density of compatible minimizer sketches, the bound is not tight on both sides and becomes useless in cases described above.

Second, we show that the problem of finding a polar set with maximum total link energy is, unsurprisingly, NP-hard, and describe heuristics to create polar sets that provably lead to minimizer sketches with low specific density. A generalization called layered polar sets is also needed for performance reasons, although the core concept and theory remains the same. We show that our implementation of these heuristics generates minimizers that have specific density on the human reference genome much lower than any other previous methods, and, for some parameter choices, relatively close to the theoretical minimum. While the polar set method does not universally work for all possible parameter choices of w and k due to its own set of inherent issues, the proposal of polar sets and associated heuristics can serve as important first steps in efficient and practical optimization of low density minimizer sketches.

1.6 Significance

In this dissertation, we focus on minimizer sketches. This sketch has seen extensive use in various aspects in computational genomics, but the research on the sketching method itself is less extensive compared to its usage. Broadly speaking, our work answers questions regarding theoretical guarantees and practical optimization of minimizer sketch performance. Our work settles open conjectures, proposes novel algorithms for optimization, and perform rigorous benchmarking to put the results into practical context. We believe both the results presented in this dissertation, as well as the techniques that lead to the results, are valuable for the research community to further understand and optimize minimizer sketches, and to play a larger role in this era of big data genomics.

Chapter 2

Theoretical Foundation of Low-Density Minimizer Sketches

This chapter describes our work work on the theoretical front of design and analysis of low-density minimizers.

Section 2.1 contains notations and a brief overview of our results. From here, this chapter can be further divided into three smaller parts.

The first part consists of Section 2.2 and Section 2.3. We will start by presenting several theorems establishing stronger connections between minimizers, universal hitting sets and generalizations of minimizers called forward/local schemes in Section 2.2. Underlying the theorems is the core concept of *charged contexts*, which itself is a generalization of charged windows as proposed in Schleimer et al. [103] to local schemes. These results serve as cornerstones of later analyses. Section 2.3 constructs and measures the size of a forbidden word universal hitting set. We present this result because it is the stepping stone to the next set of results.

The second part deals with the Mykkeltveit set and contains Section 2.4, Section 2.5 and Section 2.6. As discussed before, Mykkeltveit sets are minimal decycling sets, and thus minimal universal hitting sets for sufficiently large value of w. However, the window length at which Mykkeltveit sets become universal hitting sets is still unknown. In Section 2.4, we present our slightly modified construction of the Mykkeltveit set and its embedding to the complex plane. In Section 2.5, we upper bound the longest path length for our modified Mykkeltveit set, and in Section 2.6 we present the lower bound.

Finally, in the last part, Section 2.7, we present results on lexicographical and random minimizers, settling lexicographical minimizer sketches as asymptotically optimal, and fixing the densities of random minimizer sketches as $2/(w+1) + o(1/w)$ in certain scenarios.

A version of this chapter except Section 2.7 is published in RECOMB 2020 [130]. A version of Section 2.7 (alongside the next chapter) is published in ISMB 2020 [128].

2.1 Notation and Overview

For the first section, we will use the phrase “minimizer scheme” to refer to minimizer sketches, to keep in line with its extensions which are called “local scheme” and “forward scheme”.

Universal hitting sets. Consider a finite alphabet $\Sigma = \{0, \dots, \sigma - 1\}$ 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 universal hitting set for length L if A hits every string of length L.

The concept of universal hitting sets also appear in classical combintoratics as *unavoidable words* [17, 101] with constant length. Other researches in this line extends the concept to partial k-mers and/or variable length substrings [10, 16, 40].

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 prefix u and 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 universal hitting set A corresponds to a *depathing set* of the de Bruijn graph: a universal hitting set 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 universal hitting set, 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.

Generalization of minimizer schemes. A *local scheme* [103] 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: $S[0, w], S[1, w], \dots$, 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, i.e., it is a function $f : \Sigma^w \to [0 : w - 1]$. The output of a local scheme when applied to sketch a sequence is (similarly to minimizers) the set of selected positions: $\{i + f(S[i, w]) \mid 0 \leq i < |S| - w\}$.

A *forward scheme* is a local scheme with a selection function such that the selected positions form a non-decreasing 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 *minimizers scheme* is 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 2.1 for examples of all 3 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 non-decreasing way. A minimizers scheme is the least general and also selects positions in a non-decreasing way.

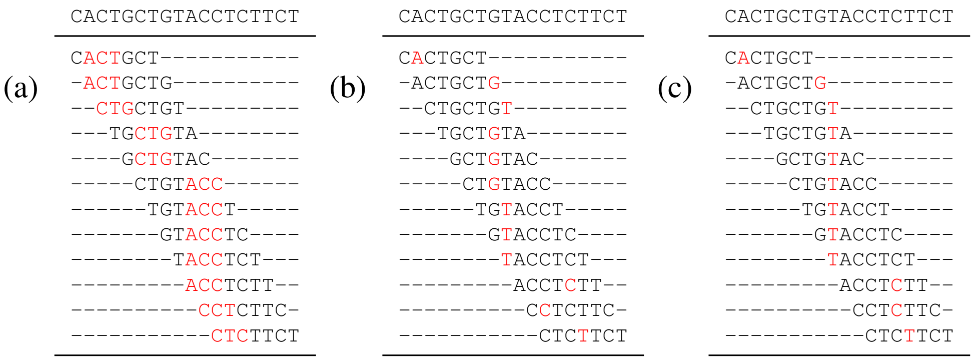

Figure 2.1: Examples of minimizers, local schemes, and forward schemes. (a) Example of selecting minimizers with $k = 3$, $w = 5$ and the lexicographic order (i.e., `AAA` $<$ `AAC` $<$ `AAG` $<$ ... $<$ `TTT`). The top line is the input sequence, each subsequent line is a 7-bases long window (the number 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 non-decreasing. (c) A forward selection scheme for $w = 7$ with selected positions $\{1, 7, 8, 12, 13\}$. Like the minimizers scheme, the sequence of selected positions is non-decreasing.

Selection schemes. A forward or local *selection scheme* is a forward or local scheme with $k = 1$. Selection schemes thus have a single numeric parameter w as the word length. Local and forward schemes were originally defined [74] with a function defined on a window of w k-mers, $f : \Sigma^{w+k-1} \to [0 : w - 1]$, similarly to minimizers. While the notion of k-mer is central to the definition of the minimizers 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 sequence elsewhere in the window to the selection function. Thus, we do not consider minimizer schemes with $k = 1$ for this part.

There are multiple reasons to consider selection schemes. First, they are slightly simpler than general local or forward schemes 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 of optimizing density of selection schemes 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 $g_k : \Sigma^{w+k-1} \to [0, w - 1]$ 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 parameter 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.

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 *specific density* of a scheme is defined as the number of distinct selected positions divided by $|S| - w + 1$ (see Figure 2.1). The *expected density*, or simply the *density*, of a scheme is the expected density on an infinitely long random sequence. Alternatively, the density is computed exactly by computing the specific 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 [73] and Section 2.2).

2.1.1 Overview of Theoretical 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 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 in Roberts et al. [95] and Roberts et al. [96], both papers 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, it is $O(1/w)$.

Unfortunately, this simple model properly accounts for the minimizers behavior only when k and w are small. For large k —i.e., $k \gg w$— it is possible to create almost optimal minimizers scheme with density $\sim 1/w$. More problematically, for large w —i.e., $w \gg k$— and for all minimizer schemes the density becomes constant ($\Omega(1)$) [74]. In other words, minimizers 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 exists forward schemes with density of $O(1/\sqrt{w})$ [74] (this is for a selection scheme, and from our previous argument it works for arbitrary values of k). 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 like minimizers schemes for large w and that they can have much lower density.

Connection between UHS and selection schemes. In the study of selection schemes, as for minimizers schemes, universal hitting sets 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 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.*

Almost-optimal relative size UHS for linear path length. Conversely, because of their link to forward and local selection schemes, we are interested in universal hitting sets 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 sequences 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 $001 \to 010 \to 100 \to 001$). The number of these rotation cycles is known as the "necklace number" $N_{\sigma,w} = \frac{1}{n}\sum_{d|w}\varphi(d)\sigma^{w/d} = O(\sigma^w/w)$ [100], where $\varphi(d)$ is the Euler's totient function.

Consequently, the relative size of a UHS, which need to contain 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 $O(\sqrt{w}/w)$ [74]. We construct a smaller universal hitting set 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.*

This will be improved to $O(1/w)$ later in Section 2.7.

Remaining path length bounds for the Mykkeltveit sets. Mykkeltveit [82] 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 [100] 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 exists 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 $g(w) = O(w^3)$ and $g(w) > cw^2$ for some constant c.*

The proof will be presented in Section 2.5 and Section 2.6.

Asymptotically Optimal Minimizers. We prove that to have asymptotically optimal minimizers (i.e., minimizers with density of $O(1/w)$), the value k (for the k-mers) must be sufficiently large compared to the length w of the windows, and that this condition is necessary and sufficient. To be more precise, we treat k as a function of w, and study the density as w grows to infinity. We show that the lexicographic minimizers are asymptotically optimal provided that k is large enough. Formally:

Theorem 4. *For $k \geq \log_\sigma(w) - c$ with constant c, the lexicographic minimizer achieves density of $O(1/w)$. Otherwise, no minimizer can achieve density of $O(1/w)$.*

This result may be surprising as in practice the lexicographic minimizers have high density ([73, 96] and see our discussion). It also has the effect of improving Theorem 2 from $O(\ln(w)/w)$ to $O(1/w)$, because lexicographical minimizers are also forward schemes. One interpretation of Theorem 4 is that asymptotically, all minimizers behave the same regarding density.

Density of Random Minimizers. The constant hidden in the big-O notation of Theorem 4 may also be too large for practical use. When k is slightly larger than what's in Theorem 4, we can guarantee a density of $2/(w+1) + o(1/w)$:

Theorem 5. *For $k > (3+\epsilon)\log_\sigma(w+1)$, the expected density of a random minimizer is $2/(w+1) + o(1/w)$.*

Interestingly, this leads to the following result for universal hitting sets which might be of independent interest:

Lemma 1. *For sufficiently large k and for arbitrary w, there exists an efficient randomized algorithm to generate a $(2+o(1))$-approximation of a minimum-size UHS over k-mers with path length w.*

2.2 UHS from Selection Schemes

2.2.1 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 are $\{f(S[i,w]) + i\}$ 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 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 $f : \Sigma^w \to [0, w-1]$, $\mathcal{C}_f = \{c \in \Sigma^{2w-1} \mid \forall 0 \leq i \leq w-2, f(c[w-1,w]) + (w-1) \neq f(c[i,w]) + i\}$ 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 $|\mathcal{C}_f|/\sigma^{2w-1}$.

2.2.2 Charged Contexts are Universal Hitting Sets

The set $\mathcal{C}_f$ over $(2w-1)$-mers is the UHS needed for Theorem 1.

Lemma 2. *$\mathcal{C}_f$ 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' includes 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 $S'[l]$ (that is, $m + f(S'[m, w]) = l$) 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 result is a direct consequence of how $\mathcal{C}$ is defined. We can also prove this lemma in a more brute force way as follows:

Proof. (Alternative Proof for Lemma 2.) Assume there exists a path $\{c_0, c_1, \cdots, c_{w-1}\}$ in the de Bruijn graph of order $2w-1$ that avoids $\mathcal{C}$. As $c_{w-1} \notin \mathcal{C}$, there exists some indices v such that $f(c_{w-1}[w-1, w]) + (w-1) = f(c_{w-1}[v, w]) + v$. Let i be the smallest v satisfying this property.

As $c_i \notin \mathcal{C}$, there exists some indices j such that $f(c_i[w-1, w]) + (w-1) = f(c_i[j, w]) + j$. Now, note that $c_i[w-1, w] = c_{w-1}[i, w]$, so they have the same value of f. These two terms are underlined below:

$$\begin{aligned} f(c_{w-1}[w-1,w]) + (w-1) &= \underline{f(c_{w-1}[i,w])} + i \\ \underline{f(c_i[w-1,w])} + (w-1) &= f(c_i[j,w]) + j \\ 2(w-1) - i - j &= f(c_i[j,w]) - f(c_{w-1}[w-1,w]) \leq w-1 \\ i + j - w + 1 &\geq 0 \end{aligned}$$

Let $t = i + j - w + 1$. Since $i, j \in [0, w-2]$, $t = i + j - w + 1 < w - 1$, so it is a valid index between 0 and $w-2$. We now note that:

$$\begin{aligned} f(c_{w-1}[w-1,w]) + (w-1) &= f(c_{w-1}[i,w]) + i \\ &= f(c_i[w-1,w]) + (w-1) + i - w + 1 \\ &= f(c_i[j,w]) + j + i - w + 1 \\ &= f(c_{w-1}[t,w]) + t \end{aligned}$$

Since $j < w-1$, we have $t < i$, contradicting with the fact that i is the smallest index satisfying $f(c_{w-1}[w-1,w]) + (w-1) = f(c_{w-1}[v,w]) + v$. □

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 last window. This means 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 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.*

2.3 Forbidden Word Depathing Set

2.3.1 Construction and Path Length

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

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 3. *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 $c \leq w-d$, $x_c[0,d] = 0^d$ and x_c is in $\mathcal{F}_{\sigma,w}$.

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

Lemma 4. *The relative size of the set of w-mers satisfying clause 1 of Definition 2 is $O(\ln(w)/w)$.*

Proof. 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 as this is the last part required for characterization of the forbidden word set.

2.3.2 Number of w-mers not Containing Consecutive Zeroes

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 $0 \leq i \leq d-1$ means that the last i characters were 0 and $d-i$ more zeroes are expected to match 0^d. The terminal state "d" means that we have seen a substring of d consecutive zeroes. If the machine is at non-terminal 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. See Figure 2.3.2 for an example.

Now, assume we feed a random w-mer to the finite state machine. 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 of Definition 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 $0 \leq j < d$ (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 non-terminal state.

Define $\mu = 1/\sigma$. Given that a randomly chosen w-mer is fed into the FSM, i.e., 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

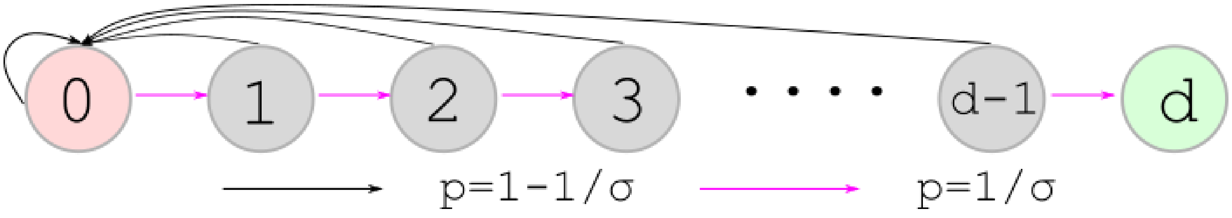

Figure 2.2: Construction of the finite state machine that recognizes 0^d. Different color indicates edge with different transition probability. State 0 is the initial state and state d is the terminal state.

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 $p_w = A_d p_{w-1} = A_d^w p_0$.

2.3.3 Bounding 1-norm of the State Vector

We start by deriving the characteristic polynomial $p_{A_d}(\lambda)$ of A_d and its set of roots (which are the eigenvalues of A_d):

Lemma 5.

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

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 $p_{A_d}(\lambda)$:

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

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. f_d and p_{A_d} have the same roots except μ, which is a root of f_d but not of p_{A_d}.

Lemma 6. *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 7. *Let $s = \mu/\lambda_0$. $\nu_0 = (1, s, s^2, \ldots, s^{d-1})$ 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 $1 \le i < d$, $(A_d\nu_0)_i = \mu(\nu_0)_{i-1} = \mu s^{i-1} = \lambda_0 s^i = (\lambda_0\nu_0)_i$. 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 8. $\|p_w\|_1 = \|A_d^w p_0\|_1 = O(1/w)$.

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

$$\begin{aligned}
\|p_w\|_1 &= \|A_d^w p_0\|_1 \\
&= \|A_d^w(\nu_0 - \eta_0)\|_1 \\
&= \|\lambda_0^w \nu_0 - A_d^w \eta_0\|_1 && (\nu_0 \text{ is a eigenvector of } A_d) \\
&\leq \lambda_0^w \|\nu_0\|_1 && (\text{nonnegative elements}) \\
&< 3(1-\mu^{d+1})^w && (\text{by Lemma 6 \& 7}) \\
&\leq 3(1-\ln(w)/w)^w && (\text{by definition of } d) \\
&\leq 3\exp(-\ln(w)) && (1 - x \leq e^{-x}) \\
&= O(1/w). && \square
\end{aligned}$$

These lemmas imply 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 a $(O(\ln(w)/w), w) - UHS$.

2.4 The Mykkeltveit Set and Complex Embeddings

2.4.1 The Modified Mykkeltveit Embedding

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 [82] 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 w^{th} 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 [82]. This extra factor of r_w in our modified embedding rotates the coordinate and is instrumental for the upcoming analyses.

Define the *successor function* $S_a(x) = x_1x_2\cdots x_{w-1}a$, where $a \in \Sigma$. The successor function can be used to obtain all the successors of x (that x has an outgoing edge pointing to) in the de Bruijn graph. A *pure rotation* of x is the particular successor $R(x) = S_{x_0}(x)$, i.e., 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:

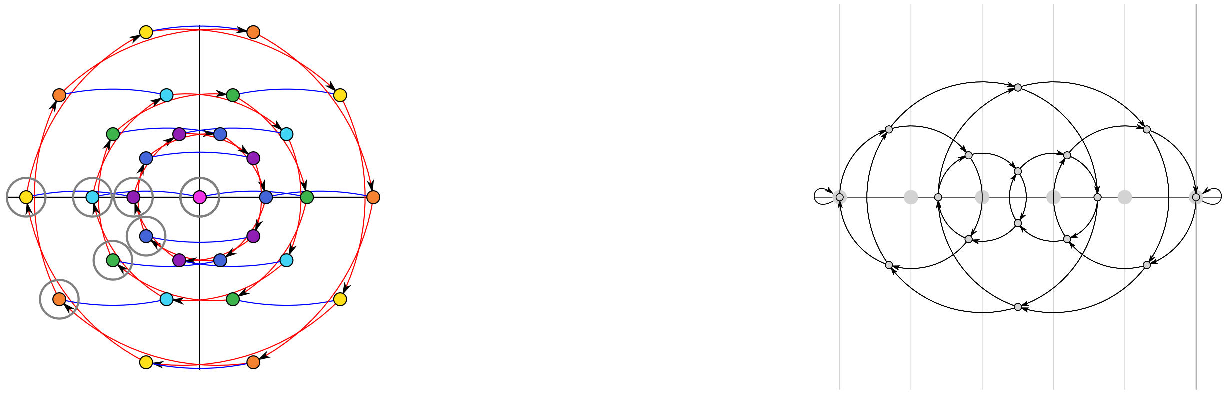

Figure 2.3: Mykkeltveit embedding and Weight-in embedding of order 5 on binary alphabet. (a) Mykkeltveit embedding of the de Bruijn graph of order 5 on the binary alphabet. The nodes of a conjugacy class have the same color and form a circle (there is more than one class per circle). The pure rotations are represented by the red edges. A non-pure 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 represent 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 sub-regions of interest.

$(x, R(x), R^2(x), \ldots)$. Each pure cycle consists of w distinct w-mers, 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 9 (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 the definition of successor function $S_a(x)$:

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

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-mer satisfy $P(x) = 0$, or they lie equidistant on a circle centered at origin. Figure 2.3(a) 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.

2.4.2 Constructing the Mykkeltveit Set

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 $\mathrm{Im}(P(R(x))) > 0$. 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 10. *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 $j \geq i$, $\mathrm{Im}(P(x_j)) \leq 0$.*

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 9, $\mathrm{Im}(P(R(x))) = \mathrm{Im}(P(S_a(x))) = \mathrm{Im}(P(y)) > 0$.

- 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.

This enumeration covers all cases. □

2.5 Upper Bounding the Remaining Path Length in Mykkeltveit Sets

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

2.5.1 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 9 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 $0 \leq t \leq W_{\text{max}}$, where $W_{\text{max}} = (\sigma - 1)w$ 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 actual transitions or w-mers 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 other hand, 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.

2.5.2 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 11. *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 2.3(b) shows the weight-in embedding of a de Bruijn graph. The set $\mathcal{C}_{\sigma,w} = \{(-j, 0) \mid 0 \leq j \leq W_{\text{max}}\}$ is the set of all the possible center of rotations, and is shown by large gray dots on Figure 2.3(b). Because all the w-mer 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 2.3(b), every node belongs to two circles with different centers, meaning there are two embeddings with same $Q(x)$ but different t.

Lemma 10 naturally divides any path in the de Bruijn graph avoiding $\mathcal{M}_{\sigma,w}$ into two parts, the first part in with $\text{Im}(P(x)) > 0$, and the second part with $\text{Im}(P(x)) \leq 0$. Thanks to the symmetry of the problems, we focus on the upper halfplane, defined as the region with $\text{Im}(P(x)) \geq 0$. With the weight-in embedding, as long as the path is contained in the upper halfplane, it is always traveling to the right (towards large real value) or staying unmoved, as stated below:

Lemma 12 (Monotonicity of $\mathrm{Re}(Q(\cdot))$)**.** *Assume $Q(x)$ and $Q(Z_\delta(x))$ are both in the upper halfplane. 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 halfplane. 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 12 still applies in this case and the points in the upper-halfplane 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 ($\mathrm{Im}(z_i) \geq 0, \forall i$)?

We now break the problem into smaller stages as the weight-in embedding pass through rotation centers, defined as $\mathcal{C}_{\sigma,w} = \{(-j, 0) \mid 0 \leq j \leq W_{\text{max}}\}$, the set of points that $Q(x)$ could possibly rotate around regardless of t. As there are $W_{\text{max}} + 1$ rotation centers and the maximum $\mathrm{Re}(Q(x)) = \mathrm{Re}(P(x))$ for any w-mer is also W_{max}, we define $2W_{\text{max}}$ 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 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 $\mathrm{Re}(z_i) = -j$ in this geometric problem. Still, it suffices to solve this simpler problem, as indicated by the following lemma, which will be proved in the next subsection.

Lemma 13. *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 $4W_{max}L + O(w^2) = O(wL + w^2)$.*

2.5.3 Tightness of Local Trajectory Problem

In this section, we prove Lemma 13 by resolving every difference between Definition 7 and the original problem of finding longest path avoiding $\mathcal{M}_{\sigma,w}$. We start from the other type of subregions.

Lemma 14. *For any $0 \leq j < W_{max}$, any path in the de Bruijn graph avoiding $\mathcal{M}_{\sigma,w}$ has at most $2L + 2$ steps satisfying $Q(x) \in (-j - 1, -j) \times [0, W_{max}]$.*

Proof. As the defined region does not contain a rotation center, every move strictly increases $\mathrm{Re}(Q(x))$. We similarly define the left and right subregion as the region with $\mathrm{Re}(Q(x)) < -j - 0.5$ and $\mathrm{Re}(Q(x)) > -j - 0.5$. There is at most one move that goes from the left subregion to the right one, or two moves if there is one w-mer with $\mathrm{Re}(Q(x)) = -j - 0.5$, and all other moves are contained within either subregion.

For the left subregion, the longest path within it is upper bounded by L. Intuitively, we only need to shift the coordinate to coincide with Definition 7. Formally, let $\{z_i\}$ be the weight-in embedding of any path strictly within the left subregion. Then $\{z_i + (j+1)\}$ becomes a feasible trajectory under Definition 7, as all points are within the feasible region $(0, 0.5) \times [0, W_{\text{max}}]$, and each center of rotation, which after shift is $(-W(x) + (j+1), 0) \in \mathcal{C}'$ as $0 \leq j < W(x)$.

For the right subregion, we have the same conclusion using a mirroring argument. To see this, again let $\{z_i\}$ be the weight-in embedding of any path strictly within second subregion. Then $\{j - \overline{z_{-i}}\}$ (z_{-i} is the i^{th} element in z counted backwards, and $\bar{z}$ is the conjugate of z) becomes a feasible trajectory. This is because all points are in the feasible region $(0, 0.5) \times [0, W_{\text{max}}]$, and assuming z_{i+1} is z_i rotated $2\pi/w$ clockwise around $(-t, 0)$, we know $j - \overline{z_i}$ is $j - \overline{z_{i+1}}$ rotated $2\pi/w$ clockwise around $(t - j, 0) \in \mathcal{C}'$. □

Next, we bound the path length outside any subregions.

Lemma 15. *Any path in the de Bruijn graph satisfying* $\mathrm{Re}(Q(x)) > 0$ *and* $\mathrm{Im}(Q(x)) \geq 0$ *has at most* $w/4$ *steps.*

Proof. We let θ denote the polar angle of $Q(x)$. As $Q(x)$ is in first quadrant, $0 \leq \theta < \pi/2$. Next, observe that every rotation around the origin decreases θ by $2\pi/w$, and every rotation not around origin but some $(-i, 0)$ with $i > 0$ will decrease θ by a greater amount. This means the path is at most $w/4$ steps long, because in $w/4$ steps θ would have decreased by at least $\pi/2$, leading to a contradiction. □

We can similarly bound the path length left of all regions by looking at the polar angle of $Q(x)$ when origin is at $(-W_{\text{max}}, 0)$, which leads to the following lemma:

Lemma 16. *Any path in the de Bruijn graph satisfying* $\mathrm{Re}(Q(x)) < -W_{max}$ *and* $\mathrm{Im}(Q(x)) \geq 0$ *has at most* $w/4$ *steps.*

We are now prove the bound over the entire upper halfplane by bounding path length on the boundaries of subregions.

Lemma 17. *Any path in the de Bruijn graph satisfying* $\mathrm{Im}(Q(x)) \geq 0$ *has at most* $2W_{max}L + O(w^2)$ *steps.*

Proof. We again start by taking $\{z_i\}$ to be the weight-in embedding of any path within the upper halfplane. We categorize $\{z_i\}$ using their real coordinates.

- If $\mathrm{Re}(z_i) < -W_{\text{max}}$ or $\mathrm{Re}(z_i) > 0$, by previous two lemmas, we know there are at most $w/2$ of them.
- Else, if $\mathrm{Re}(z_i)$ is not an integer, it falls in one of the regions defined by Lemma 14. As there are W_{max} regions total under that definition, and the point can never re-enter a region, the point belongs to a path contained within the region of at most $2L + 2$ length, and there are at most $W_{\text{max}}(2L + 2)$ points in this category.

- The last category is when $\mathrm{Re}(z_i)$ is an integer. If c_i satisfies $\mathrm{Im}(z_i) > 0$, it will be the only one with this real coordinate as $\mathrm{Re}(z_{i+1}) > \mathrm{Re}(z_i)$. Otherwise, c_i coincides with one of rotation centers $(-j, 0)$. It could stay at the same location by doing a rotation around itself, which corresponds to a pure rotation of a w-mer when that w-mer embeds to origin. By construction of $\mathcal{M}_{\sigma,w}$ (clause 1 of Definition 4), there can only be $w-1$ consecutive moves this way, so at most w elements in $\{z_i\}$ have this real coordinate. There are $W_{\text{max}}+1$ possible real coordinates, and each one of them might contain w points, so total number of points in this category is $(W_{\text{max}}+1)w$.

Summing up between these three categories, we get a bound of $2W_{\text{max}}L + (W_{\text{max}}+1)w + w/2 = 2W_{\text{max}}L + O(w^2)$ for the length of a path in the upper-half plane. □

Finally, we look at the path in the lower halfplane with the concept of w-mer complements:

Definition 8 (Complements of w-mer)**.** *For $a \in \Sigma$, its complement is defined as $\bar{a} = \sigma - a$. For a w-mer x, its complement $\bar{x}$ is the w-mer $\overline{x_0}\overline{x_1}\cdots\overline{x_{w-1}}$. The following property holds:*

- *For any x, $P(x) = -P(\bar{x})$.*
- *For any x and a, $P(S_a(x)) = -P(S_{\bar{a}}(\bar{x}))$.*
- *If there is an edge $x \to y$ in the de Bruijn graph, there is also an edge $\bar{x} \to \bar{y}$ in the de Bruijn graph.*

Lemma 18. *Any path in the de Bruijn graph satisfying $\mathrm{Im}(Q(x)) \le 0$ has at most $2W_{max}L + O(w^2)$ steps.*

Proof. For any path satisfying the condition, the path formed by taking complement of every w-mer is a path satisfying $\mathrm{Im}(Q(x)) \ge 0$. By Lemma 17, the length is also upper bounded by $2W_{\text{max}}L + O(w^2)$. □

We are now ready to prove the original statement as follows:

Lemma 13. *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 $4W_{max}L + O(w^2) = O(wL + w^2)$.*

Proof. As seen in Lemma 10, we can bound the path length in two parts. For the first part with $\mathrm{Im}(P(x)) > 0$, the length is upper bounded by $2W_{\text{max}}L + O(w^2)$ because we prove a strictly stronger statement in Lemma 17 by also allowing points with $\mathrm{Im}(P(x)) = 0$. For the second part with $\mathrm{Im}(P(x)) \le 0$, we also proved a strictly stronger statement in Lemma 18 by also allowing points in $\mathcal{M}_{\sigma,w}$. The path length for the original problem is upper bounded by the sum of two upper bounds, which is $4W_{\text{max}}L + O(w^2)$. □

2.5.4 Backtracking, Heights and Local Potentials

In this section, we prove $L = O(w^2)$. We will frequently switch between polar and Cartesian coordinates in this section and Section 2.6. For simplicity, let $r(z)$ and $\phi(z)$ denote the radius and the polar angle of z written in polar coordinate.

Lemma 19. *Any feasible trajectory within the region $(0, d] \times [0, W_{max}]$ for $d \le 0.5$ is at most $O(dw^3)$ long.*

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

To see this, assume (d, θ) is the polar coordinate of $\mathrm{Re}(Q(x))$ with respect to the rotation center. The polar coordinate for $\mathrm{Re}(Q(S_a(x)))$ is then $(d, \theta - 2\pi/w)$. We note that $d \geq 0.5$ as $Q(x)$ satisfies $0 < \mathrm{Re}(Q(x)) < 0.5$ 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 is possible in the defined region, otherwise $\mathrm{Re}(Q(x))$ 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 can't stay in the upper halfplane. 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 $O(w^4)$. It does not solve the whole problem. We can however set $d = 10/w$ and the above argument shows that the trajectory within the region $(0, 10/w] \times [0, W_{\text{max}}]$ is only $O(w^2)$ long. Thus, we now focus on the trajectory in the region $(10/w, 0.5) \times [0, W_{\text{max}}]$, which we denote as $\mathcal{R}$ for the rest of this section.

We aim to prove $L = O(w^2)$ by proving the near-optimality of a greedy approach: The sequence of rotation such that z_i only rotate around $(1, 0)$ when necessary and otherwise rotate around $(0, 0)$. Intuitively, if z_i is rotated from further rotation centers, it will move a greater distance towards $\mathrm{Re}(z_i) = 0.5$. For trajectories that include moves that deviate from the greedy trajectory, we want to show we can always backtrack to the move, make corrections and yield a longer trajectory. We now introduce the tools to formalize this idea.

Recall the formula $z \leftarrow c + r_w^{-1}(z - c)$ for rotating z around c clockwise by $2\pi/w$ degrees. Note that this is a linear function of c, so if we change c by $(1, 0)$, z will change by $u = 1 - r_w^{-1}$. In other words, if z' is the result from rotating z around some centers, to change the rotation centers retroactively, we can simply move z' by a multiple of u. This leads to the following definition.

Definition 9 (Equivalence Classes and Heights). *Let $u = 1 - r_w^{-1}$ and recall $\mathcal{R} = (10/w, 0.5) \times [0, W_{max}]$. For any point $z \in \mathcal{R}$, its equivalent set is defined as $S(z) = \{z + ju \mid j \in \mathbb{Z}, \mathrm{Im}(z + ju) > 0\}$. The point with smallest j in the set is called representative of the set, which we denote as $s(z)$. The height of a point is defined as the nonnegative integer j such that $s(z) = z - ju$, which is zero if and only if z is a representative itself.*

Now we can define the potential function. Loosely speaking, this potential function measures how many steps are left in the trajectory if we strictly follow the greedy approach, backtracking one step if necessary.

Definition 10 (Local Potential Function). *Let $P(z) = -wr/\pi + \phi$, assuming $z = (r, \phi)$ in polar coordinate. The potential function of a point is $E(z) = P(s(z))$, where $s(z)$ is the representative of z.*

For $z \in \mathcal{R}$, it is not guaranteed the representative $s(z)$ is in the same region. However, we have the following lemma:

Lemma 20. *If $z \in \mathcal{R}$, z' is obtained by rotating z one step according to the longest trajectory problem, then as long as $z' \in \mathcal{R}$, $s(z') \in \mathcal{R}$.*

Proof. By definition of the representative, $\operatorname{Im}(s(z')) \geq 0$. Also by definition of the equivalent set, $s(z')$ is also obtained by rotating z by some points on the real axis, clockwise by $2\pi/w$ degrees. As we have shown before, such move is guaranteed to increase $\operatorname{Re}(z)$, so $\operatorname{Re}(s(z')) > \operatorname{Re}(z) > 10/w$. To show $s(z') \in \mathcal{R}$, we only need $\operatorname{Re}(s(z')) < 0.5$ and $\operatorname{Im}(s(z')) \leq W_{\text{max}}$. However, since $\operatorname{Re}(s(z')) \leq \operatorname{Re}(s(z))$ and $\operatorname{Im}(s(z')) \leq \operatorname{Im}(s(z))$, $s(z') \notin \mathcal{R}$ would imply $z' \notin \mathcal{R}$. □

This means after one rotation in $\mathcal{R}$, or the first step in the trajectory, $s(z)$ is guaranteed to be in the same region and would stay in the region unless z' is already out of the region, indicating end of trajectory. For the rest of our proofs, we assume $s(z) \in \mathcal{R}$ for the whole trajectory.

Our goal from now on is to prove that $E(z)$ reduces by some amount each rotation. Assume a rotation brings z to z'. Note that if we only care about $E(z)$, the rotation center is irrelevant as $s(z')$ is the same, so we can assume every move is a rotation around origin and $z' = r_w^{-1} z$, possibly followed by a shift in multiples of u (which does not change $s(z')$).

If z and z' are both of height 0, $s(z) = z$, $s(z') = z'$ and if $z = (r, \phi)$ in polar coordinate, $z' = (r, \phi - 2\pi/w)$. This means $E(z) - E(z') = 2\pi/w$ and the potential drops by $2\pi/w$, a constant value. If they are both of height j, $s(z) = z - ju$, $s(z') = z' - ju$, and it is no longer clear how much ϕ changes other than that it decreases a bit. However, z' is further toward the origin, and as we prove below, the change in r is enough for our proofs. We prove a stronger lemma:

Lemma 21. *If $z' = e^{-i\theta} z$ is z rotated clockwise by $\theta \leq 2\pi/w$ degrees, and they are of the same height $j \geq 1$, then $E(z) - E(z') \geq 1.9\theta$.*

Proof. We note that the $z_0' = z' - ju$ and $z_0 = z - ju$ are the representatives of z' and z, and $\phi(z_0') < \phi(z_0)$. Let $r' = e^{-i\theta}$.

$$\begin{aligned}
|z_0'| &= |z' - ju| \\
&= |r'z - ju| \\
&= |r'(z_0 + ju) - ju| \\
&= |r'z_0 + j(r' - 1)u| \\
&= |z_0 + jr'^{-1}(r' - 1)u|
\end{aligned}$$

We let $y = r'^{-1}(r' - 1)u$. Written in polar coordinate, $r'^{-1} = (1, \theta)$, $(r' - 1) = (2\sin(\theta/2), 3\pi/2 - \theta/2)$, $u = (2\sin(\pi/w), \pi/2 - \pi/w)$, so $y = (4\sin(\theta/2)\sin(\pi/w), \theta/2 - \pi/w)$. Since $0 < \theta \leq 2\pi/w$, the polar angle of y is between 0 and $-\pi/w$, which becomes 0 as w grows, meaning y is almost parallel to real axis.

We next bound the polar angle of z_0. Since z_0 is in first quadrant, $\phi(z_0) \geq 0$. Next, since $z_0 \in \mathcal{R}$ and it is a representative, $\operatorname{Re}(z_0) \geq 10/w$ and $\operatorname{Im}(z_0) \leq \sin(\pi/w)$ (otherwise, $z_0 - t$ still satisfies $\operatorname{Im}(\cdot) \geq 0$ and would be the representative instead). For sufficiently large w, we have $\operatorname{Im}(z_0) \leq 3/w$ and $\tan(\phi(z_0)) = \operatorname{Im}(z_0)/\operatorname{Re}(z_0) \leq 0.3$, which yields $\phi(z_0) \leq 0.291$.

Let the angle between z_0 and y be ψ, we have $|\psi| \leq (\phi(z_0) + 2\pi/w)$. For large enough w, $|\psi| \leq 0.3$ and $\cos(\psi) \geq 0.955$. Apply the rule of cosines on vector additions:

$$\begin{aligned}|z_0'|^2 &= |z_0 + jy|^2 \\ &= |z_0|^2 + j^2|y|^2 + 2j\cos(\psi)|z_0||y| \\ &\geq (z_0 + 0.955j|y|)^2\end{aligned}$$

So $r(z_0') - r(z_0) \geq 0.955j|y| = 3.82j\sin(\pi/w)\sin(\theta/2) \geq 1.9\pi\theta/w$ for large enough w. We now plug this back to the formula for potential energy:

$$\begin{aligned}E(z) - E(z') &= P(z_0) - P(z_0') \\ &= -w(r(z_0) - r(z_0'))/\pi + (\phi(z_0) - \phi(z_0')) \\ &\geq w(1.9\pi\theta/w)/\pi \\ &= 1.9\theta\end{aligned}$$

This finishes the proof. □

Plugging in $\theta = 2\pi/w$, we have the following:

Lemma 22. *If $z' = r_w^{-1}z$ with same height $h > 0$, $E(z) - E(z') \geq 3.8\pi/w$ for sufficiently large w.*

The last case is when z and z' are of different height. We need to account for the sudden change of height during rotation. Intuitively, changing height from $j+1$ to j while making a small movement costs $2\pi/w$ potential, as follows:

Lemma 23. *For sufficiently large w and a real number $10/w < d < 0.5$, we have $P(d+u) - P(d) = -2\pi/w + O(1/w^2)$.*

Proof. We will calculate the difference in ϕ and r separately. Recall that $d = \Omega(1/w)$. For now, we let $u = a + bi$, that is, $a = 2\sin^2(\pi/w), b = 2\sin(\pi/w)\cos(\pi/w) = \sin(2\pi/w)$. For ϕ, we have:

$$\begin{aligned}\phi(d+u) - \phi(d) =& \text{arctan}(b/(d+a)) \\ =& \text{arctan}((2\pi/w + O(w^{-3}))/(d + O(w^{-2}))) \\ =& \text{arctan}(2\pi/dw + O(w^{-2})) && d = \Omega(w^{-1}) \\ =& 2\pi/dw + O(w^{-2}) && \text{arctan}(x) = x + O(x^{-3})\end{aligned}$$

For r, we have:

$$\begin{aligned}
r(d+u) - r(d) =& \sqrt{(d+a)^2 + b^2} - d \\
&= \sqrt{d^2 + (a+b)^2 + 2ad} - d \\
&= \sqrt{d^2 + 4\sin^2(\pi/w)(1+d)} - d \\
&= \frac{4\sin^2(\pi/w)(1+d)}{\sqrt{d^2 + 4\sin^2(\pi/w)(1+d)} + d} \\
&= \frac{4\pi^2(1+d)/w^2 + O(w^{-4})}{2d + O(w^{-2})} && \sin^2(x) = x^2 + O(x^4) = O(x^2) \\
&= \frac{2\pi^2(1+d)}{dw^2} + O(w^{-3}) && d = \Omega(1/w), 1/(d + O(w^{-2})) = 1/d + O(w^{-1})
\end{aligned}$$

Merging the two terms we have:

$$\begin{aligned}
P(d+u) - P(d) &= -w(r(d+u) - r(d))/\pi + \phi(d+u) - \phi(d) \\
&= -(2\pi/w) - (2\pi/dw) + (2\pi/dw) + O(w^{-2}) \\
&= -2\pi/w + O(w^{-2})
\end{aligned}$$

This finishes the proof. □

Combining previous two lemmas, we can analyze the potential drop for all possible moves and prove the upper bound.

Lemma 24. *If $z' = r_w^{-1}z$ with different height, $E(z) - E(z') \geq 3.9\pi/w$ for sufficiently large w.*

Proof. First of all, the height will only decrease since z' is generated by rotating z around origin in the first quadrant, and $\mathrm{Im}(z') < \mathrm{Im}(z)$. For now, assume the height of z is h and height of z' is $h-1$. We consider the movement of $s(z)$ while rotating from z to z'. There exists one point z'' on the arc from z to z' such that from z to z'' the height of the point is h, and from z'' to z' the height is $h-1$. We can now divide the movement into three parts:

$$E(z) - E(z') = (E(z) - E(z'')) + (E(z'') - E(z'' + \delta)) + (E(z'' + \delta) - E(z'))$$

where δ is an infinitesimal value such that height of z'' is h and height of $z'' + \delta$ is $h-1$. The first and last term correspond to the rotation process with constant height. If the height is 0, the change in potential is exactly the degree rotated. Otherwise, as shown in Lemma 21, the change in potential is at least 1.9 times degree rotated. Since the total rotated degrees is $2\pi/w$, these two terms add up to at least $2\pi/w$. The second term corresponds to the change of height as described in Lemma 23, and for sufficiently large w, it is at least $1.9\pi/w$. Adding both terms up, we get $3.9\pi/w$ as desired. We can use the same technique if the height drops more than 1 and yield at least the same bounds. □

Lemma 25. *$L = O(w^2)$ as in Definition 7.*

Proof. We can divide the trajectory into two parts. The trajectory in the region $(0, 10/w) \times [0, W_{\max}]$ is $O(w^2)$ long as seen in Lemma 19. The potential function $E(z)$ has a maximum value of $\pi/2$ and minimum value of $O(w)$. The minimum holds because $r(s(z)) = \sqrt{\mathrm{Re}(s(z))^2 + \mathrm{Im}(s(z))^2}$ is upper bounded by a constant for $\mathrm{Re}(s(z)) < 0.5$ and $\mathrm{Im}(s(z)) < 1/w$ (otherwise $s(z) - ju$ for $j > 0$ would be the representative). As shown in previous lemmas, each move decreases $E(z)$ by $\Omega(1/w)$, so at most $O(w^2)$ steps are possible in the region $(10/w, 0.5) \times [0, W_{\max}]$. □

With Lemma 13, we conclude $\mathcal{M}_{\sigma,w}$ is a UHS with remaining path length $O(w^3)$.

2.6 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 criteria. It suffices to prove the theorem for binary alphabet as the path constructed will also be a valid path in a graph with larger alphabet. We present the constructions for even w first, then for odd w.

2.6.1 k-mers as Shift Registers

We need an alternative view of w-mers in this section, thinking them as shift registers. Imagine a paper ring with w slots, labelled 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.

2.6.2 Construction and Verification for Even Window Length

First we assume w is even and 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 4 tags are 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).

We now describe the strategy in more detail. 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

Stage	$P(y)$ Composition	$P(y)$ Polar Coordinate	Starting Phase	Ending Phase
Rotation to Tag r_1	-1	$(1, \pi)$	$\pi - \theta$	θ
Rotation to Tag r_2	$-1 - v_1$	$(2\sin(\theta/2), 3\pi/2 - \theta/2)$	$(\pi + \theta)/2$	$(\pi - 3\theta)/2$
Rotation to Tag r_3	$-1 - v_1 - v_2$	$(2\cos(\theta) - 1, 0)$	$\pi - \theta$	θ
Rotation to Tag r_4	$-1 - v_1 - v_2 - v_3$	$(2\sin(\theta/2), \pi/2 + \theta/2)$	$(\pi + 3\theta)/2$	$(\pi - \theta)/2$
End of round	-1	$(1, \pi)$	$\pi - \theta - 2\pi/w$	

Table 2.1: Absolute embedding and polar angle of $P(x)$ throughout a round in the constructive proof for lower bounding path length of Mykkeltveit sets.

tag $a + j$, impure rotation S_0, pure rotations until pointer is at tag $b - j$, impure rotation S_0, pure rotations until pointer is at tag $b + j$, impure rotation 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 quadruple 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 2.4 shows an example of quadruples and a generated long path that fits in the upper halfplane.

The following lemma implies the created path avoids the Mykkeltveit set.

Lemma 26. *The sequence generated from the algorithm satisfies* $\mathrm{Im}(P(x)) > 0$ *at every step.*

Proof. We define the *absolute embedding* as $P(y)$ on the paper ring model. This embedding does not change during pure rotations, and if $P(y) = (r, \phi)$ in polar coordinate, the real embedding is $r_w^{-j} P(y)$ or $(r, \phi - 2\pi j/w)$ in polar coordinate with pointer at tag j.

For each quadruple, we let $v_i = r_w^{r_i+1}$ denote the weight in the absolute embedding for tag r_i, where $1 \leq i \leq 4$. We also let θ be the polar angle of v_4. As seen before, the absolute embeddings are $v_1 = -e^{-i\theta}, v_2 = -e^{i\theta}, v_3 = e^{-i\theta}, v_4 = e^{i\theta}$, and the corresponding polar angles are $\phi(v_1) = \pi - \theta, \phi(v_2) = \pi + \theta, \phi(v_3) = 2\pi - \theta, \phi(v_4) = \theta$.

We now compute the change of $P(y)$, the absolute embedding, and polar angle of $P(x)$ (we use the phrase "phase" for it) throughout a round as in the following table:

Since θ increases by $2\pi/w$ each round, the ending condition for one round matches the starting condition for next round. Before the first round (as we do one pure rotation before first quadruple in the sequence of rotations), the polar angle is at $\pi - 2\pi/w$, which matches the starting condition for round 1 with $\theta = 2\pi/w$.

As long as $\theta < \pi/3$ (or in our constructions $i < w/6$), during all rotations the polar angle stays between 0 and π, meaning it stays strictly above the real line and thus avoids $\mathcal{M}_{\sigma,w}$. □

2.6.3 Construction and Verification for Odd Window Length

We focus on a particular portion of the path from last section with the property that all w-mers in the path are well above the real axis. We also define the set of *critical embeddings* for a round as the set of embeddings right before or after an impure rotation (or a write in the tape model). For a quadruple, given θ, the absolute embedding (defined in the proof of Lemma 26) and the polar angle of all critical embeddings can be read from the table above.

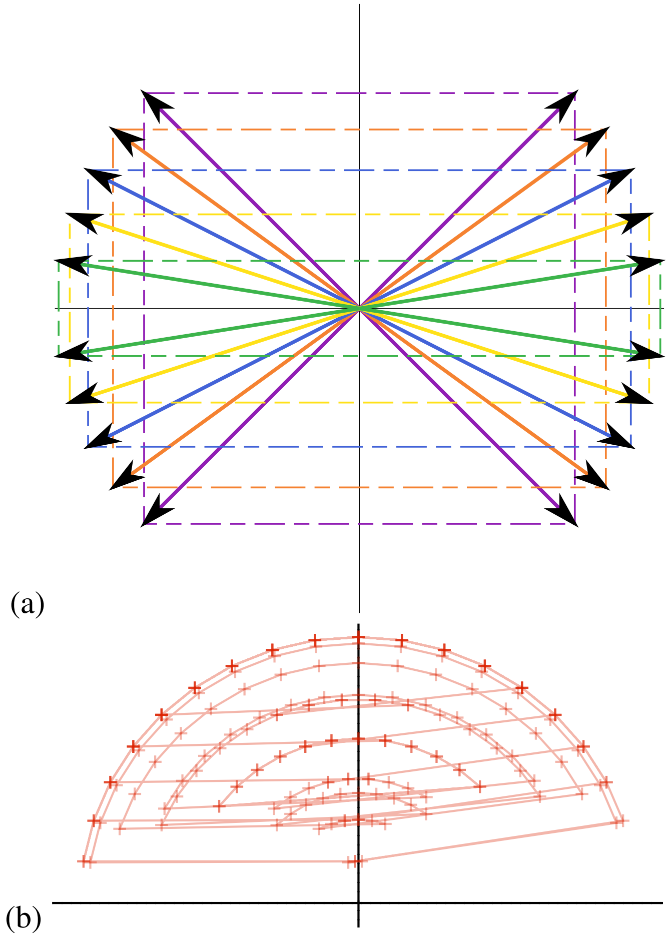

Figure 2.4: Overview for constructive proof of lower path length bounds of Mykkeltveit sets. (a) For $w = 40$, each set of 4 arrows of the same color represent a quadruple set of root of unity. There are a total of 5 sets. They were crafted so that the 4 vector in each set cancel out. (b) The path generated 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.

Lemma 27. *For sufficiently large $w = 2m$, the movement sequences defined above from $j = w/20$ to $j = w/10$ satisfies $\mathrm{Im}(P(x)) > 0.05$ at every step.*

Proof. Note that the choice of j means θ is between $2\pi/10$ and $2\pi/20$. As pure rotations are arcs over upper halfplane, $\mathrm{Im}(P(x))$ is the lowest at the endpoints of pure rotation, or as we defined above, the critical embeddings. However, at these points, the shortest embedding is $2\sin(\theta/2) > 0.3$, and the smallest polar angle is θ with $\sin(\theta) > 0.3$. We then have $\mathrm{Im}(P(x)) = r(P(x))\sin(\phi(P(x))) > 0.3 \times 0.3 > 0.05$. □

Now let $w = 2m+1$ and again assume binary alphabet $\sigma = 2$. Let $a_0 = m-1, a_1 = m, b = w-1 = 2m$. The corresponding roots of unity (weights for tags in the absolute embedding) are $r_w^{b+1} = 1$ for b, and $r_w^{a_0+1}$ and $r_w^{a_1+1}$ are the roots of unity directly above and below the vector -1. The starting w-mer is full 1 except a_0, a_1 and b set to zero. The resulting absolute embedding is on the real axis with value $2\cos(\pi/w) - 1 = -1 + 4\sin^2(\pi/2w) = -1 + O(w^{-2})$. We now construct a sequence of quadruples such that at the end of every quadruple the absolute embedding is still on the real axis with value close to -1.

We let j range from $w/20$ to $w/10$ as described before, but increment it by 2 every step.

Definition 11 (Imperfect Quadruples)**.** *For each j, we construct two candidate quadruples: $Q_j^+ = \{a_0 - j, a_1 + j, b - j, b + j\}, Q_j^- = \{a_0 - j + 1, a_1 + j - 1, b - j, b + j\}$. Both quadruples satisfy that sum of their corresponding roots of unity is on real axis, which we denote as q_j^+ and q_j^-. We have $|q_j| = 2(\cos(\theta + \pi/w) - \cos(\theta)) = O(1/w)$ where θ is either $2\pi j/w$ or $2\pi j/w + \pi/w$, and are of opposite sign: $q_j^+ > 0, q_j^- < 0$.*

We define l_j to be the embedding by setting all bases in quadruples $\{Q_k \mid k \leq j\}$ to zero from the initial w-mer. Our construction of the imperfect quadruples ensures l_j is a real number.

We decide the imperfect quadruple to use depending on the sign of $l_{j-2} + 1$. If l_{j-2} is smaller than -1, we pick Q_j^- and we have $l_j = l_{j-2} - q_j^-$. Otherwise, we pick Q_j^+ and we have $l_j = l_{j-2} - q_j^+$. In both cases, we assured $|l_j + 1| \leq \max(|q_j|, |l_{j-2} + 1|)$, which is $O(1/w)$ by induction on j.

The sequence of rotations is defined in exactly the same way as before. The analyses are similar, as there are between $w+1$ and $w+3$ moves every round and $O(w^2)$ total steps, no two quadruples share tags, and we finish the proof with the following lemma:

Lemma 28. *For every round of moves using imperfect quadruples, the embedding satisfies $\mathrm{Im}(P(x)) > 0$ at all times.*

Proof. Similar to our previous argument, we only need to show $\mathrm{Im}(P(x)) > 0$ at the critical embeddings. We start by constructing the following (perfect) quadruple for $2w$-mers: $Q'_j = \{w - 1 - 2j, w - 1 + 2j, 2w - 1 - 2j, 2w - 1 + 2j\}$. As seen in last lemma, the sequence generated by this quadruple satisfies $\mathrm{Im}(P(x)) > 0.05$ at all times, so it also holds at the critical embeddings. We can also map the tags in $Q_j^\pm$ onto $2w$-mers by keeping the corresponding roots of unity the same: $Q_j'^+ = \{w - 2 - 2j, w + 2j, 2w - 1 - 2j, 2w - 1 + 2j\}$ and $Q_j'^- = \{w - 2j, w - 2 + 2j, 2w - 1 - 2j, 2w - 1 + 2j\}$.

Now we fix one embedding in the critical set. For example, at the end of writing 0 to tag r_3, the absolute embedding is $-1 - v_1 - v_2 - v_3 = -1 + v_4$ and the polar angle is $(\pi + 3\theta)/2$ for the perfect quadruple. We will prove that for the imperfect quadruple, the embedding at this moment is similar.

The absolute embedding is a combination of v_0 (which is -1 for the perfect quadruple, and $l_{j-2} = -1 + O(1/w)$ for the imperfect one) and $\{v_i\}$s. v_3 and v_4 are the same for the two quadruples, while v_1 and v_2 are off by π/w degrees, translating to $O(1/w)$ distance on the complex plane. This means the absolute embedding differs by $O(1/w)$.

The polar angle relative to the absolute embedding is simply one of the $\phi(v_i)$, which is off by at most π/w. This corresponds to an extra π/w rotation in either direction, and since the length of the embedding is $O(1)$, it moves by $O(1/w)$ on top of the previous argument.

Combining both arguments, we show that if z is the embedding for the perfect quadruple at this moment, the embedding for the imperfect quadruple z' satisfies $|z' - z| = O(1/w)$. However, $\mathrm{Im}(z) > 0.05$, so $\mathrm{Im}(z') > 0$ holds for sufficiently large w. This proof works for all critical embeddings, and since $\mathrm{Im}(z')$ is the lowest at critical embeddings, $\mathrm{Im}(z') > 0$ also holds for the whole round. □

2.7 Lexicographical and Random Minimizers

2.7.1 Preliminary

We first recall the following definition, now limited to minimizers only:

Definition 12 (Contexts and Charged Contexts)**.** *A "context" of S is a substring of length $(w+k)$, or equivalently, two overlapping windows. The minimizer is applied to both the first and last windows, and a context is "charged" if different positions are picked.*

The idea for charged contexts is to attribute picked k-mers to the window that first picked it (it is charged for picking a new k-mer). To determine if a window is actually picking a new k-mer, it is sufficient to look back exactly one window due to the fact that minimizers pick k-mers in a forward manner, and the context is the union of the two windows necessary to determine whether this happens. This means counting picked k-mers in a string is equivalent to counting charged contexts (in other words, only charged contexts contribute to the density). Consequently, the (non-specific) density of a minimizer equals the probability that a context drawn from uniform distribution over Σ^{w+k} is a charged one.

We denote by $W = \Sigma^{w+k-1}$ the set of all possible windows and $C = \Sigma^{w+k}$ the set of all contexts. The following lemma gives a slightly stronger condition on the positions of selected k-mers in a charged context. A similar lemma was proved by Schleimer et al. [103] and a proof is given here for clarity.

Lemma 29. *For a minimizer, a context is charged if and only if the minimizer picks either the first k-mer of the first window or last k-mer in the last window.*

Proof. On one hand, if the minimizer picks either the first or the last k-mer in the context, it cannot be picked in both windows. On the other hand, if the minimizer does not pick either the first or the last k-mer, it will pick the same k-mer in both windows. Assuming otherwise, both picked k-mers are in both window, so this means one of them is not minimal, leading to a contradiction. □

We also define the concept of density factor. This concept is very useful for comparison.

Definition 13 (Density Factor)**.** *The density factor of a minimizer is its density multiplied by* $(w+1)$. *Intuitively, this is the expected number of selected locations in a random context.*

2.7.2 Lexicographical Minimizers are Asymptotically Optimal

Culminating with Theorem 4, we first prove that to have asymptotically optimal minimizers (i.e., minimizers with density of $O(1/w)$), the value k (for the k-mers) must be sufficiently large compared to the length w of the windows, and that this condition is necessary and sufficient. Again, to be more precise, we treat k as a function of w, and study the density as w grows to infinity. Next, we prove a density bound for random minimizers (which are by default asymptotically optimal), and its implications for an efficient algorithm for constant-factor optimal universal hitting set approximation.

Minimizers with Exceedingly Small k

If k is exceedingly small, in the sense that k does not even grow as fast as the logarithm of w —i.e., $\log_\sigma(w) - k \to \infty$ as w grows— no minimizer will obtain density $O(1/w)$. To see this, for any order $\mathcal{O}$, let y be the smallest of all k-mers. Any context starting with y is charged, and the proportion of such context is σ^{-k}. The density calculated from these contexts only is already $> \Theta(1/w)$, as $w\sigma^{-k} = \sigma^{\log_\sigma(w)-k} \to \infty$.

For this reason, in the following we are only interested in the case where k is large enough. That is, there exists a fixed constant c such that $k \geq \log_\sigma(w) - c$ for all sufficiently large w.

Lexicographic Minimizers with Specific k

We first prove the special case that the lexicographic minimizer achieves $O(1/w)$ density with parameter $k = \lfloor \log_\sigma(w/2) \rfloor - 2$. Recall $W = \Sigma^{k+w-1}$ is the set of all windows, and $C = \Sigma^{k+w}$ is the set of all contexts. Let $z \in W$ be a window, $f(z) : W \to \{0, 1, \dots, w-1\}$ be the minimizer function, and $\mathcal{C} \subset C$ be the set of charged contexts for this minimizer. Let $W^+ = \{z \in W \mid f(z) = 0\}$ be the set of windows where the minimizer picks the first k-mer, and similarly $W^- = \{z \in W \mid f(z) = w-1\}$. By Lemma 29, we know $|\mathcal{C}| \leq \sigma(|W^+| + |W^-|)$.

We now use the notion $\#$ to denote any nonzero character of Σ, and 0^d to denote d consecutive zeroes. Let A_*^+ be the set of windows whose first k-mer is 0^k, and for $1 \leq i \leq k$, let $A_i^+ = \{z \in W \mid z = 0^{i-1}\#\cdots, f(z) = 0\}$, that is, the set of windows that starts with exactly $i-1$ zeros and have the minimizer function pick the first k-mer. All A_i^+ and A_*^+ are mutually disjoint. Since the minimizer always pick 0^k at the start of the window, we have $W^+ = A_*^+ \cup \bigcup_{i=1}^k A_i^+$.

Lemma 30. *For* $1 \leq i \leq k$, $A_i^+ \subseteq B_i^+$, *where* $B_i^+ = \{z \in W \mid z = 0^{i-1}\#st, |s| = w-1, |t| = k-i, 0^i \notin s\}$, *that is, the set of windows that starts with* $0^{i-1}\#$ *and does not contain* 0^i *in the next* $w-1$ *bases.*

Proof. We need to show that if a window z starts with $0^{i-1}\#$ and is not in B_i^+, $f(z) \neq 0$. As $z \notin B_i^+$, there is a stretch of 0^i in z before the last $k-i$ characters. This means there is a k-mer of form $0^i\cdots$ in z, and since the first k-mer is of form $0^{i-1}\#\cdots$, the minimizer will never pick the first k-mer. □

Earlier in Section 2.3.2 of this dissertation, we proved the following:

Lemma 31. *The probability that a random string of length ℓ does not contain 0^d anywhere is at most $3(1-1/\sigma^{d+1})^{\ell}$.*

Setting $\ell = w-1$ and $d = i$ and noting there are σ^{k-i} choices for t in B_i, we know $|B_i^+| \le 3\sigma^{w+k-i}(1-1/\sigma^{i+1})^{w-1}$ for every i. This is sufficient to prove the following:

Lemma 32. $|W^+| = O(\sigma^w)$.

Proof. Let $b_i = 3\sigma^{w+k-i}(1-1/\sigma^{i+1})^{w-1}$, combined with the fact that $|A_*^+| = \sigma^{w-1}$, we know $|W^+| \le \sigma^{w-1} + \sum_{i=1}^{k} b_i$. It remains to bound the summation term.

We next prove $b_i > 2b_{i-1}$ for $2 \le i \le k$:

$$\begin{aligned} b_i/b_{i-1} &= \frac{3\sigma^{w+k-i}(1-1/\sigma^{i+1})^{w-1}}{3\sigma^{w+k-i+1}(1-1/\sigma^{i})^{w-1}} \\ &= \frac{1}{\sigma}\left(1+\frac{\sigma-1}{\sigma^{i+1}-\sigma}\right)^{w-1} \\ &> \frac{1}{\sigma}\left(1+\frac{w-1}{\sigma^{i+1}-\sigma}\right) > \frac{w}{\sigma^{i+2}} \end{aligned}$$

Note that we also use the fact $(1+x)^t > 1+xt$ and $\sigma - 1 \ge 1$ in the last line. The right-hand side is minimum when $i = k$. By our choice of k, $\sigma^{k+2} < w/2$, so the term is lower bounded by 2.

This implies $\sum_{i=1}^{k} b_i < 2b_k$, and since $b_k = O(\sigma^w)$, we have $|W^+| = O(\sigma^w)$. □

The bound for $|W^-|$ is computed similarly. It is different from W^+ as in case of ties for the minimal k-mer, the leftmost one is picked. Hence, we define $W^\ddagger$ as the set of windows such that the last k-mer is one of the minimal k-mers in the window. We have $W^- \subseteq W^\ddagger$, as the last k-mer needs to be the minimal, with no ties, for the minimizer to pick it.

Similarly, we define A_*^- as the set of windows that ends with 0^k. For $1 \le i \le k$, we define $A_i^- = \{z \in W^\ddagger \mid z = s0^{i-1}\#t, |s| = w-1, |t| = k-i\}$. This is the set of windows whose last k-mer starts with $0^{k-1}\#$ while satisfying the condition for $W^\ddagger$. Again, A_*^- and all A_i^- are mutually disjoint, and we have $W^\ddagger = A_*^- \cup \bigcup_{i=1}^{k} A_i^-$. There is an analogous lemma for bounding $|A_i^-|$:

Lemma 33. *For $1 \le i \le k$, $A_i^- \subseteq B_i^-$, where $B_i^- = \{z \in W \mid z = s0^{i-1}\#t, |s| = w-1, |t| = k-i, 0^i \notin s\}$.*

Proof. We need to show that if a window ends with a k-mer of form $0^{i-1}\#t$ and contains 0^i before last k-mer, it is not in $W^\ddagger$. In that case the window contains a k-mer of form $0^i\cdots$, which is strictly smaller than the last k-mer of the form $0^{i-1}\#\cdots$, violating the condition of $W^\ddagger$. □

Note that B_i^+ and B_i^- have highly similar expressions. In fact, we can simply bound the size of B_i^- by b_i (defined in the proof of Lemma 32) using the identical argument. This immediately means we have the exactly same bound for $|W^\ddagger|$ and $|W^+|$, as A_*^- also has the same size as A_*^+.

Theorem 6. *The lexicographic minimizer with $k_0 = \lfloor \log_\sigma(w/2) \rfloor - 2$ has density $O(1/w)$.*

Proof. We have $|\mathcal{C}| \le \sigma(|W^+| + |W^-|) \le \sigma(|W^+| + |W^\ddagger|) = O(\sigma^w)$, and the density is $|\mathcal{C}|/\sigma^{w+k_0} = O(\sigma^{-k_0}) = O(1/w)$. □

Lexicographic Minimizers with All Values of k

Next, we extend this result to show that this bound holds for all k as long as $k > \log_\sigma(w) - c$ for some constant c. As $k_0 < \log_\sigma w$, the following lemmas establishes our claim for small k:

Lemma 34. *The lexicographic minimizer with $k = k_0 - c$ for constant $c \geq 0$ has density $O(1/w)$.*

Proof. We define W'^+ and W'^- in the same way for this new minimizer with parameter k. We also call the new minimizer as k-minimizer, and the minimizer in Theorem 6 as k_0-minimizer.

We now claim $|W'^+| \leq |W^+|$. This is because given a window in W'^+, we can always append 0^c before the start of the window to get a window (for the k_0-minimizer) in W^+. Assume otherwise, we let $z_0 = 0^c \dots$ be the k_0-mer at the start of the extended window and z_1 be the k_0-mer picked by the k_0-minimizer. This requires z_1 to also have form of $0^c \cdots$, and that the k-mer after 0^c is strictly smaller than the last k-mer of z_0. However, if this is the case, the original window will not be in W'^+ as the first k-mer is not minimal.

Similarly, we claim $|W'^-| \leq |W^-|$ as given a window in W'^- we can append 0^c after the end of the window to get a window in W^-. Assume otherwise, we let $z_0 = \cdots 0^c$ be the last k_0-mer at the end of the extended window and z_1 be the k_0-mer picked by the k_0-minimizer. Since z_0 ends with 0^c and z_1 is smaller or equal to z_0, the k-prefix of z_1 must be smaller or equal to the k-prefix of z_0. This means the original window will not be in W'^- as there is a k-mer that is smaller or equal to the last k-mer.

Therefore $|W'^+| = O(\sigma^w)$ and $|W'^-| = O(\sigma^w)$, and the density is $\sigma(|W'^+|+|W'^-|)/\sigma^{w+k} = O(\sigma^{-k}) = O(1/w)$. □

The following lemma establishes our claim for large k:

Lemma 35. *The lexicographic minimizer with $k > k_0$ has density $O(1/w)$.*

Proof. We define W'^+ and W'^- in the same way for this new minimizer, and again call the new minimizer the k-minimizer.

We first claim $|W'^+| \leq \sigma^{k-k_0}|W^+|$. This is because given a window in W'^+, we can remove the last $k - k_0$ bases to get a window in W^+. To see this, we only need to show for the shorter window, the first k_0-mer is less than or equal to every other k_0-mer. Since the original window is in W'^+, the first k-mer in the original window is less than or equal to every other k-mer. This means the k_0-long prefix of the first k-mer is also less or equal to the k_0-long prefix of every other k-mer in the original window, which is exactly the condition for the short window in W^+. For every window x in W^+, there are σ^{k-k_0} windows in Σ^{w+k-1} that x is a prefix of, which means $|W'^+| \leq \sigma^{k-k_0}|W^+|$.

We now claim $|W'^-| \leq \sigma^{k-k_0}|W^\ddagger|$. Similarly, given a window in W'^-, we can again remove the last $k - k_0$ bases to get a window in $W^\ddagger$. As the original window is in W'^-, the last k-mer in the original window is less than every other k-mer. This means the k_0-long prefix of the last k-mer is less or equal to the k_0-long prefix of every other k-mer in the original window. (Note that if a k-mer is less than another, their prefix can be equal.) The k_0-long prefixes of k-mers in the original window are exactly the k_0-mers of the shorter window, so the short window is in $W^\ddagger$. With a similar argument, we have $|W^-| \leq \sigma^{k-k_0}|W^\ddagger|$.

These two facts combined means we can calculate the density of the new minimizer as follows:

$$\begin{aligned}\sigma(|W'^{+}|+|W'^{-}|)/\sigma^{w+k} &\leq \sigma^{k-k_0+1}(|W^{+}|+|W^{\ddagger}|)/\sigma^{w+k}\\ &= (|W^{+}|+|W^{\ddagger}|)/\sigma^{w+k_0-1}\\ &= O(\sigma^{-k_0}) = O(1/w). \qquad \square\end{aligned}$$

Combining everything we know, we have the following theorem.

Theorem 4. *For $k \geq \log_\sigma(w) - c$ with constant c, the lexicographic minimizer achieves density of $O(1/w)$. Otherwise, no minimizer can achieve density of $O(1/w)$.*

2.7.3 Density Bounds for Random Minimizers

In Schleimer et al. [103], it was estimated the expected density of random minimizers is $2/(w+1)$ with several assumptions on the string (which do not strictly hold in practice), and our main theorem (Theorem 5) achieves the same result up to $o(1/w)$ with a single explicit hypothesis between w and k. In this section, we prove the main theorem.

In the estimation of the expected density for random minimizers, there are two sources of randomness: (1) the order $\mathcal{O}$ on the k-mers is selected at random among all the permutations of Σ^k and (2) the input string is a very long random string with each character chosen IID. The key tool to this part is the following lemma to control the number of "bad cases" when a window contains two or more identical k-mers. Chikhi et al. [21] proved a similar statement to Lemma 36 with slightly different methods.

Lemma 36. *For any $\epsilon > 0$, if $k > (3+\epsilon)\log_\sigma w$, the probability that a random window of w k-mers contains two identical k-mers is $o(1/w)$.*

Proof. We start with deriving the probability that two k-mers in fixed locations i and j are identical in a random window. Without loss of generality, we assume $i < j$. If $j - i \geq k$, the two k-mers do not share bases, so given they are both random k-mers independent of each other, the probability is $\sigma^{-k} = 1/w^{3+\epsilon} = o(1/w^3)$.

Otherwise, the two k-mers intersect. We let $d = j - i$, and m_i to denote i^{th} k-mer of the window. We use x to denote the substring from the start of m_i to the end of m_j with length $k + d$ (or equivalently, the union of m_i and m_j). If $m_i = m_j$, the n^{th} character of m_i is equal to the n^{th} character of m_j, meaning $x_n = x_{n+d}$ for all $0 \leq n < k$. This further means x is a repeating sequence of period d, so x is uniquely determined by its first d characters and there are σ^d possible configurations of x. The probability a random x satisfies $m_i = m_j$ is then $\sigma^d/\sigma^{k+d} = \sigma^{-k} = o(1/w^3)$, which is also the probability of $m_i = m_j$ for a random window.

The event that the window contains two identical k-mers is the union of events of form $m_i = m_j$ for $i < j$, and each of these events happens with probability $o(1/w^3)$. Since there are $\Theta(w^2)$ events, by the union bound, the probability that any of them happens is upper bounded by $o(1/w)$. □

We are now ready to prove the following.

Theorem 5. *For $k > (3+\epsilon)\log_\sigma(w+1)$, the expected density of a random minimizer is $2/(w+1) + o(1/w)$.*

Proof. Given a context $c \in \Sigma^{w+k}$, we use $I(c)$ to denote the event that c has two identical k-mers. As c has $(w+1)$ k-mers, by Lemma 36, $P_c(I(c)) = o(1/w)$ assuming c is a random context.

Recall that a random minimizer means the order $\mathcal{O}$ is randomized, and $\mathcal{C}$ is the set of charged contexts. For any context c that does not have duplicate k-mers, we claim $P_{\mathcal{O}}(c \in \mathcal{C}) = 2/(w+1)$. This is because given all k-mers in c are distinct, under the randomness of $\mathcal{O}$, each k-mer has probability of exactly $1/(w+1)$ to be the minimal. By Lemma 29, $c \in \mathcal{C}$ if and only if the first or the last k-mer is the minimal, and as these two events are mutually exclusive, the probability of either happening is $2/(w+1)$. The expected density of the random minimizer then follows:

$$\begin{aligned} P_{c,\mathcal{O}}(c \in \mathcal{C}) &= P(I(c))P(c \in \mathcal{C} \mid I(c)) + P(\overline{I(c)})P(c \in \mathcal{C} \mid \overline{I(c)}) \\ &\leq P(I(c)) + P(c \in \mathcal{C} \mid \overline{I(c)}) \\ &= o(1/w) + 2/(w+1). \end{aligned}$$

□

Approximately Optimal Universal Hitting Sets

One interesting implication of Theorem 5 is on construction and approximation of compact UHS. In Section 2.2, we proved a connection between UHS and forward schemes. We restate it with minimizers, as follows.

Theorem 7. *For any minimizer $(w, k, \mathcal{O})$, the set of charged contexts $\mathcal{C}$ over a de Bruijn sequence of order $w+k$ is a UHS over $(w+k)$-mers with path length w, and with relative size identical to the density of the minimizer.*

This theorem, combined with the density bound derived last section, allows efficient construction of an approximately optimal UHS. We say an algorithm for constructing UHS is efficient if it runs in poly$(w, k)\sigma^{w+k}$ time, as the output length of such algorithms is already at least σ^{w+k}.

Lemma 1. *For sufficiently large k and for arbitrary w, there exists an efficient randomized algorithm to generate a $(2+o(1))$-approximation of a minimum-size UHS over k-mers with path length w.*

This is also the first known efficient algorithm to achieve constant approximation ratio, as previous algorithms [24, 28, 85] use path cover heuristics with approximation ratio dependent on w and k. We prove this by discussing the case with $k > w$ and $k < w$ separately. In the first case, the best UHS has its relative size lower bounded by the decycling set, and in the second case, a stronger bound of $1/w$ is available. The following two Lemmas give an efficient construction of a universal hitting set with bounded size in two cases: when w is large compared to k, and when it is small.

Lemma 37. *For sufficiently large k and $w > k - (3+\epsilon)\log_\sigma(k+1)$, there exists a UHS of relative size $2/k + o(1/k)$ with path length w. This is a $2 + o(1)$ approximation of the minimum size UHS.*

Proof. We pick $k' = (3+\epsilon)\log_\sigma(k+1)$ (ignoring rounding for simplicity), and let $w' = k - k'$. By our setup, $w \geq w'$, and $k' = o(k)$ (implying $w' = k - o(k)$). By Theorem 5, the charged context set of a random minimizer $(w', k', \mathcal{O}')$ has relative size $2/w' + o(1/w') = 2/k + o(1/k)$. By Theorem 7, the charged context set is a UHS over k-mers with path length w. This finishes the first part of the proof.

For the second part, by construction of the Mykkeltveit sets the minimum size UHS, regardless of path length, will have relative size at least $1/k - o(1/k)$. As our proposed UHS has relative size $2/k + o(1/k)$, it is a $2 + o(1)$ approximation of the minimum size UHS. □

Lemma 38. *For sufficiently large k and $w \leq k - (3 + \epsilon)\log_\sigma(k + 1)$, there exists a UHS of relative size $2/w + o(1/w)$ with path length w. This is a $2 + o(1)$ approximation of the best possible UHS.*

Proof. We pick $w' = w$ and $k' = k - w$. By our setup, $k' \geq (3 + \epsilon)\log_\sigma(w' + 1)$ as $w < k$. By the identical argument as seen in the last lemma, we obtain a UHS over k-mers with path length w and relative size $2/w + o(1/w)$.

For the second part, note that the minimal relative size of a UHS with path length w is $1/w$ as it need to hit every one of every w k-mer on the de Bruijn sequence of order k, where every k-mer appears exactly once. As our set has relative size $2/w + o(1/w)$, it is a $2 + o(1)$ approximation. □

These two lemmas combined give us the desired proof for Lemma 1.

Chapter 3

Low-Density, Sequence-Blind Minimizer Sketches

In this chapter, we develop a minimizer sketch (or rather, a distribution of minimizer sketches) with expected density strictly below $2/(w+1)+o(1/w)$, the random minimizer density bound. The construction works as long as $w < xk$ for some constant x. One other method exists to create minimizers with density below $2/(w+1)$ [74], but it requires $w \ll k$, a much more restrictive condition.

The name "Miniception" is shorthand for "Minimizer Inception", a reference to its construction that uses a smaller minimizer to construct a larger minimizer (the 2010 movie *Inception* set the stage at dreams within dreams, thus the reference). In the estimation of the expected density of Miniception minimizers, there are multiple sources of randomness: the choice of orders in the small and in the large minimizers, and the chosen context. The construction and the proof of Miniception uses these sources of randomness to ensure its good performance on average.

In this chapter, when the context is clear, we use minimizers as a shorthand to minimizer sketches. A version of the contents in this chapter (alongside with Section 2.7) was published in ISMB 2020 [128].

3.1 Notation

For the sake of clarity, we re-introduce the concepts we will use in this chapter, specializing to minimizers when appropriate.

In the following, $\Sigma = \{0, 1, \ldots, \sigma - 1\}$ is an alphabet (mapped to integers) of size σ and we assume that $\sigma \geq 2$ and is fixed. If $S \in \Sigma^*$ is a string, we use $|S|$ to denote the length of S.

Definition 14 (Minimizer and Windows)**.** *A "minimizer" is characterized by $(w, k, \mathcal{O})$ where w and k are integers and $\mathcal{O}$ is a complete order of Σ^k. A "window" is a string of length $(w+k-1)$, consisting of exactly w overlapping k-mers. Given a window as input, the minimizer outputs the location of the smallest k-mer according to $\mathcal{O}$, breaking ties by preferring leftmost k-mer.*

When $\mathcal{O}$ is the dictionary order, it is called a *lexicographic minimizer*. A minimizer created by randomly choosing a permutation of Σ^k uniformly over all possible permutations is called a *random minimizer*. See Figure 3.1 for another illustration of these and the following concepts.

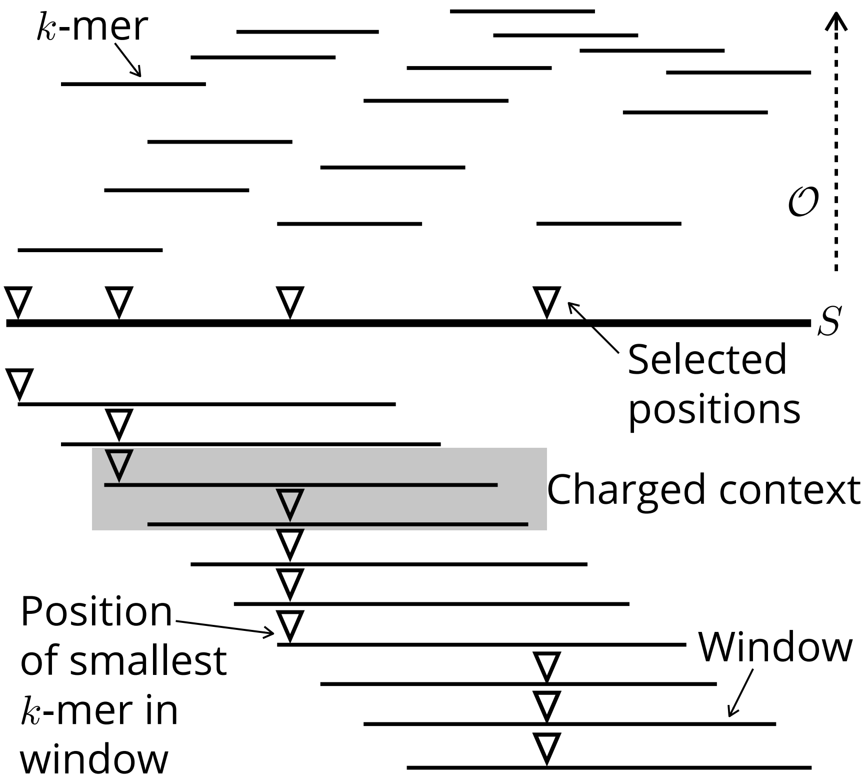

Figure 3.1: Another illustration of minimizer sketches, and charged windows. The string S is broken into k-mers. In each window (w consecutive k-mers) the position of the lowest (smallest according to $\mathcal{O}$) k-mer is selected. The gray context (2 consecutive windows) is an example of a charged window because the first and second window selected different positions. There are a total of 3 charged contexts in this example.

Definition 15 (Density). *Given a string $S \in \Sigma^*$ and a minimizer, a position in S is selected if the minimizer picks the k-mer at that position in any window of w consecutive k-mers. The specific density of a minimizer on S is the number of selected positions divided by the total number of k-mers in S. The density of a minimizer is the expected specific density on a sufficiently long random string.*

Note that the density is calculated by expectation over random strings, and is independent from S.

For the ease of comparison, we will also use the concept of contexts and density factors:

Definition 16 (Contexts and Charged Contexts). *A "context" of S is a substring of length $(w+k)$, or equivalently, two overlapping windows. The minimizer is applied to both the first and last windows, and a context is "charged" if different positions are picked.*

Definition 17 (Density Factor). *The density factor of a minimizer is its density multiplied by $(w+1)$. Intuitively, this is the expected number of selected locations in a random context.*

The idea for charged contexts is to attribute a chosen k-mer to the window that first picked it (in other words, the window is charged for picking a new k-mer). To determine if a window is actually picking a new k-mer, it is sufficient to look back exactly one window due to the fact that minimizers pick k-mers in a forward manner, and the context is the union of the two windows necessary to determine whether this happens. This means counting picked k-mers in a string is equivalent to counting charged contexts (in other words, only charged contexts contribute to the density). Consequently, the (non-specific) density of a minimizer equals the probability that a context drawn from uniform distribution over Σ^{w+k} is a charged one.

We denote by $W = \Sigma^{w+k-1}$ the set of all possible windows and $C = \Sigma^{w+k}$ the set of all contexts. The following lemma gives a slightly stronger condition on the positions of selected k-mers in a charged context. We have proved the following in the last chapter:

Lemma 29. *For a minimizer, a context is charged if and only if the minimizer picks either the first k-mer of the first window or last k-mer in the last window.*

As discussed last chapter, *universal hitting sets* (UHS) [85] are central to the analysis of minimizers [28, 73, 74], and we need to re-introduce them.

Definition 18 (Universal Hitting Sets)**.** *Assume $U \subseteq \Sigma^k$. If U intersects with every w consecutive k-mers (or equivalently, the set of k-mers in every $S \in W_{w,k}$), it is a UHS over k-mers with path length w and relative size $|U|/\sigma^k$.*

We have discussed Mykkeltveit sets as minimal decycling sets before. In this chapter, we are not using the Mykkeltveit sets, however we will need to know its size for comparison purposes.

Lemma 39 (Minimal Decycling Sets)**.** *([82]) Any UHS over k-mers has relative size at least $1/k - o(1/k)$.*

3.2 Constructing the Miniception

3.2.1 A Tale of Two UHS

UHS are connected to minimizers in two ways. The first connection, via the charged context set of a minimizer, is described in Section 2.2 and further explored in Section 2.7. The second and more known connection is via the idea of *compatible minimizers*. Detailed proof of the following properties are available in Marçais et al. [73, 74].

Definition 19 (Compatibility)**.** *A minimizer $(w, k, \mathcal{O})$ is said to be compatible with a UHS U, if the path length of U is at most w and for any $m \in U, m' \notin U$, $m < m'$ under $\mathcal{O}$.*

Construction of compatible minimizers from a universal hitting set can be done in many ways. A deterministic way is to start from an existing ordering of k-mers (such as the lexicographical one). From here, construct a compatible minimizer by first ordering k-mers within the UHS using the known ordering, then ordering the rest arbitrarily.

Lemma 40 (Properties of Compatible Minimizers)**.** *If a minimizer is compatible with a UHS, (1) any k-mer outside the UHS will never be picked by the minimizer, and (2) the relative size of the UHS is an upper bound to the density of the minimizer.*

The Miniception is a way of constructing minimizers that uses the UHS in both ways. Assume we have a minimizer $(w_0, k_0, \mathcal{O}_0)$. By Theorem 7, its charged context set $\mathcal{C}_0$ is a UHS over $(w_0 + k_0)$-mers with path length w_0. According to Definition 19, we construct a minimizer $(w, k, \mathcal{O})$ that is compatible with $\mathcal{C}_0$, where $k = w_0 + k_0$, $w \geq w_0$ and any k-mer in $\mathcal{C}_0$ is less than any k-mer outside $\mathcal{C}_0$ according to $\mathcal{O}$.

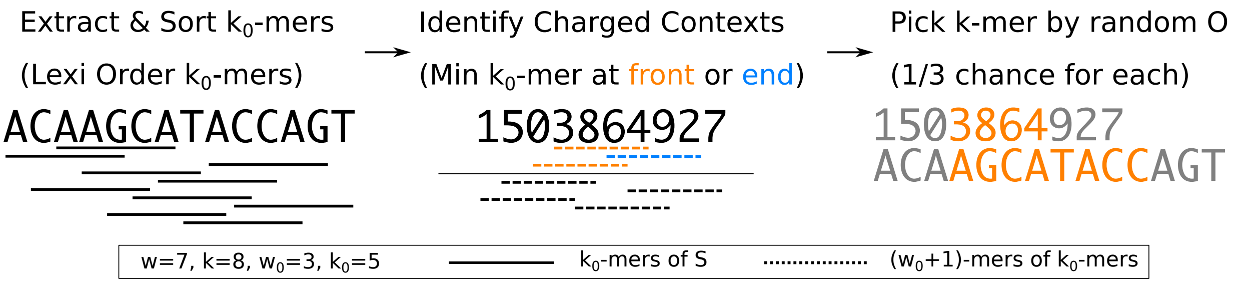

Figure 3.2: An example of running the Miniception in a window. The k_0-mers and (w_0+1)-mers are displayed by their order in $\mathcal{O}$ and $\mathcal{O}_0$, where the minimal elements appear at the top. We take $\mathcal{O}_0$ to be lexicographic order for simplicity, and $\mathcal{O}$ is a random order. The idea of sorting k_0-mers will be important in deriving the theoretical guarantees of the Miniception.

We assume that the smaller minimizer $(w_0, k_0, \mathcal{O}_0)$ is a random minimizer, and that the larger minimizer $(w, k, \mathcal{O})$ is a random compatible minimizer (meaning the order of k-mers within $\mathcal{C}_0$ is random in $\mathcal{O}$). The Miniception is formally defined as follows:

Definition 20 (The Miniception)**.** *Given parameters w, k and k_0, set $w_0 = k - k_0$. The Miniception is a minimizer with parameters w and k constructed as follows:*

- *A random minimizer $(w_0, k_0, \mathcal{O}_0)$ called the "seed" is generated.*
- *The set of charged contexts $\mathcal{C}_0 \subset \Sigma^k$ is calculated from the seed minimizer (note that $k = w_0 + k_0$, so the set of charged contexts are in fact set of k-mers).*
- *the order $\mathcal{O}$ of the resulting minimizer is constructed by generating a random order within $\mathcal{C}_0$ and having every other k-mer compare larger than any k-mers in $\mathcal{C}_0$.*

By Lemma 40, the order within k-mers outside $\mathcal{C}_0$ does not matter in constructing $\mathcal{O}$. In the following sections, we will prove the following theorem:

Theorem 8. *With $w = w_0 + 1$, $k = w_0 + k_0$ and $k_0 > (3 + \epsilon)\log_\sigma(2w_0 + 2)$, the density of the Miniception is upper bounded by $1.67/w + o(1/w)$.*

As $k = w_0 + k_0$, for large values of w_0, we can take for example $k_0 = 4\log_\sigma w_0$, meaning $w \approx k$ in these cases. This makes the Miniception the only known construction with guaranteed density $< 2/(w+1) + o(1/w)$ and with practical parameters.

Figure 3.2 provides an example of the Miniception. As a side note, the following theorem also constructs a universal hitting set from a minimizer, but is not relevant for our discussion.

Theorem 9. *Given a minimizer $(w, k, \mathcal{O})$, let S be the de Bruijn sequence of order $w + k$. The set of k-mers selected by the minimizer over S is a universal hitting set, whose relative size is lower bounded by the density of underlying minimizer.*

3.2.2 Implementing the Miniception

The Miniception can be implemented efficiently in practice, assuming efficient access to a random hash. Assuming that a random order is computed with a hash function in $O(k)$ time for a k-mer, determining the set of picked k-mers in a string S in takes $O(k|S|)$ time. This is as fast as a random minimizer. In particular, there is no need to precompute the set $\mathcal{C}_0$. In this section, we provide details of the implementation.

The implementation takes a sequence S as input (as well as other parameters specifying the minimizer), and returns the list of picked locations. We will derive a linear time implementation, meaning the algorithm runs in time $O(k|S|)$ as it takes $O(k)$ time just to process and compare k-mers. We will discuss our algorithms based on the assumption that a k-mer fits in a word (This means $k \leq 32$ for 64-bit systems), Value of k beyond this limit is rare in practice, and our algorithm also easily adapts to general values of k if needed.

While the Miniception is defined with multiple sources randomness, implementing it means sticking to one "instantiation", and the orders $\mathcal{O}_0$ and $\mathcal{O}$ are fixed. We will assume the alphabet is $\{0, 1, 2, 3\}$ so a k-mer can be written as an integer in $[4^k]$. We assume $O(k_0)$ time access to $\mathcal{O}_0$ as a function $f_0 : [4^{k_0}] \to \mathbb{R}$, such that $x_0 < x_1$ in $\mathcal{O}_0$ if and only if $f_0(x_0) < f_0(x_1)$, and similarly a function $f : [4^k] \to \mathbb{R}$ for some order function that agrees with $\mathcal{O}$ when restricted to $\mathcal{C}_0$, meaning both f_0 and f can simply be a random hash function.

From the construction of the Miniception, we can recover $\mathcal{C}_0$ from $\mathcal{O}_0$. In most cases, this means we only need to store a small minimizer to implement the Miniception on the fly. In comparison, existing methods with precomputed UHS requires storage of the whole set.

Implementation of minimizers usually use monotone queues for linear time complexity, under the same assumption that k-mers fit into words. The monotone queue is a special kind of deque that ensures the item in the queue are both in non-decreasing order and are inserted recent enough (within last T time slot where T is a fixed parameter, for example). Such data structures allow solving the problem of finding minimum element in every T-long window in input linear time, which is exactly the problem of identifying picked k-mers for a given sequence, as the chosen locations are simply locations that are minimal in a w-long window. We will provide pseudocode for implementing monotone queues later this section.

As the Miniception consists of two minimizers, we use two monotone queues, one implementing the seed minimizer and one implementing the actual minimizer that identifies picked locations. As we only care whether a context is charged for the seed minimizer, the first monotone queue has expiry time $w_0 + 1$ (meaning elements inserted into the monotone queue $w_0 + 1$ time slots ago will be removed). The second monotone queue is implemented as in a normal minimizer.

We provide a pseudocode for implementing the monotone queue in Algorithm 3.2.2. For simplicity, we will ignore the warmup part (before the first full window). As described before, monotone queues can be considered as deques with an extra parameter T (expiry time) and additional operations.

With the monotone queue, we can implement the Miniception in Algorithm 3.2.2. As mentioned before, we assume each k-mer can be held in a word.

Algorithm 1 The Monotone Queue with Expiry Time T

```
procedure INITIALIZE($l$)
    Allocate a deque with max capacity $l$
end procedure
procedure SETTIME($t$)                          ▷ this procedure removes expired items
    while queue is not empty do
        $x', t' \leftarrow$ front of queue
        if $t' \leq t - T$ then
            Pop $x', t'$ from queue
        else
            return
        end if
    end while
end procedure
procedure INSERT($x$, $t$)                      ▷ input is (item, time) pair
    Set current time to $t$
    while queue is not empty do
        $x', t' \leftarrow$ end of queue
        if $x < x'$ then                        ▷ minimizers prefer leftmost $k$-mers
            Pop $x', t'$ from queue
        else
            break
        end if
    end while
    Append $x, t$ to the end of queue
end procedure
```

Algorithm 2 The Miniception

```
procedure MINICEPTIONPICKS($S$)
    $Q_0 = \text{MONOQUEUE}(w_0 + 1)$                              ▷ $w_0 + 1$ is the window length
    $Q = \text{MONOQUEUE}(w)$
    for $i = (w + k - 1)$ to $|S|$ do                              ▷ Setups ignored for clarity
        $m_0 \leftarrow (m_0 \times 4 + S[i]) \mod 4^{k_0}$                    ▷ Latest $k_0$-mer
        $m \leftarrow (m \times 4 + S[i]) \mod 4^{k}$                          ▷ Latest $k$-mer
        $Q_0.\text{INSERT}(f_0(m_0), i)$
        $t \leftarrow$ time of $Q_0[0]$                    ▷ for current minimum $k_0$-mer in window
        if $(t = i)$ or $(t = (i - w_0))$ then
            $Q.\text{INSERT}(f(m), i)$                                   ▷ latest $k$-mer is in $\mathcal{C}_0$
        else
            $Q.\text{SETTIME}(i)$                                   ▷ remove out-of-window $k$-mer
        end if
        $p \leftarrow$ time of $Q[0]$                    ▷ this is the end of picked $k$-mer in window
        Pick $p - (k - 1)$ if not picked already
    end for
end procedure
```

3.3 Analyzing the Miniception

3.3.1 The Permutation Argument

We now focus on the setup outlined in Theorem 8. Our goal is to measure the density of the Miniception, which is equivalent to measuring the expected portion of charged contexts, that is, $P(c \in \mathcal{C})$. There are three sources of randomness here: (1) the randomness of the seed minimizer, (2) the randomness of the order within $\mathcal{C}_0$ (which we will refer to as the randomness of $\mathcal{O}$), and (3) the randomness of the context.

A context of the Miniception is a $(w+k) = (2w_0+k_0+1)$-mer, which contains $(2w_0+2)$ k_0-mers. By our choice of k_0 and Lemma 36, the probability that the context contains two identical k_0-mers is $o(1/w_0) = o(1/w)$ (as $w = w_0 + 1$). Similar to our reasoning in proving Theorem 5, let I_0 denote the event of duplicate k_0-mers in a Miniception context:

$$\begin{aligned} P(c \in \mathcal{C}) &= P(I_0(c))P(c \in \mathcal{C} \mid I_0(c)) + P(\overline{I_0(c)})P(c \in \mathcal{C} \mid \overline{I_0(c)}) \\ &\leq P(I_0(c)) + P(c \in \mathcal{C} \mid \overline{I_0(c)}) \\ &= o(1/w) + P(c \in \mathcal{C} \mid \overline{I_0(c)}). \end{aligned}$$

This means we only need to consider contexts without duplicate k_0-mers, which would incur only $o(1/w)$ error in density. Recall the way we determine whether a k-mer is in $\mathcal{C}_0$: we check if it is a charged context of the seed minimizer, which involves only comparisons between its constituent k_0-mers. This means given a context of the Miniception, we can determine if each of its k-mer is in the UHS $\mathcal{C}_0$ only using the order between all k_0-mers. We use $\mathcal{O}(c)$ to denote the order of k_0-mers within c according to $\mathcal{O}_0$. Conditioned on any c with $\overline{I_0}(c)$, over the randomness of $\mathcal{O}_0$, the order of the k_0-mers inside c now follows a random permutation of $(2w_0+2)=2w$, which we denote as $\mathcal{R}(2)$.

Next, we consider fixing both c and $\mathcal{O}_0$ (the only randomness is in $\mathcal{O}$), and calculate probability that the context is charged. Note that fixing c and $\mathcal{O}_0$ means fixed $\mathcal{O}(c)$, and fixed set of k-mers in $\mathcal{C}_0$. The order of the k-mers is still random due to randomness in $\mathcal{O}$. For simplicity, if a k-mer is in $\mathcal{C}_0$, we call it a *UHS k-mer*. A *boundary UHS k-mer* is a UHS k-mer that is either the first or the last k-mer in the context c.

Lemma 41. *Assume a fixed context c with no duplicate k_0-mers and a fixed $\mathcal{O}_0$. Denote $m_{boundary}$ as the number of boundary UHS k-mers ($m_{boundary}\in\{0,1,2\}$) and let m_{total} be the number of total UHS k-mers in the context. The probability that the context is charged, over the randomness of $\mathcal{O}$, is $m_{boundary}/m_{total}$.*

Proof. We first note that $m_{\text{total}}\geq 1$ for any context and any $\mathcal{O}_0$, due to $\mathcal{C}_0$ being a UHS over k-mers with path length $w_0\leq w$, so the expression is always valid. Furthermore, there are also no duplicate k-mers as $k>k_0$. As $\mathcal{O}$ is random, every UHS k-mer in the window has equal probability to be the minimal k-mer. The context is charged if one of the boundary UHS k-mers is chosen in this process, and the probability is $1/m_{\text{total}}$ for each boundary UHS k-mer, so the total probability is $m_{\text{boundary}}/m_{\text{total}}$. □

This proof holds for every c and $\mathcal{O}_0$ satisfying $\overline{I_0}$. In this case, both m_{boundary} and m_{total} are only dependent on $\mathcal{O}(c)$. This means we can write the probability of charged context conditioned on $\overline{I_0}$ with a single source of randomness, as follows:

$$\begin{aligned}P(c\in\mathcal{C}) &\leq P_{c,\mathcal{O}_0,\mathcal{O}}(c\in\mathcal{C}\mid\overline{I_0(c)})+o(1/w)\\ &=\mathbb{E}_{c,\mathcal{O}_0}(m_{\text{boundary}}/m_{\text{total}})+o(1/w)\\ &=\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(m_{\text{boundary}}/m_{\text{total}})+o(1/w)\end{aligned}$$

Next, we use E_0 to denote the event that the first k-mer in the context is a UHS k-mer, and E_1 to denote the event for the last k-mer. These two events are also only dependent on $\mathcal{O}(c)$. We then have $m_{\text{boundary}}=\mathbf{1}(E_0)+\mathbf{1}(E_1)$, where $\mathbf{1}(A)$ evaluates to 1 is event A happens and 0 otherwise. By linearity of expectation, we have the following:

$$\begin{aligned}P(c\in\mathcal{C}) &=\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(m_{\text{boundary}}/m_{\text{total}})+o(1/w)\\ &=\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}((\mathbf{1}(E_0)+\mathbf{1}(E_1)/m_{\text{total}})+o(1/w)\\ &=\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}}\mid E_0)P(E_0)+\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}}\mid E_1)P(E_1)+o(1/w)\end{aligned}$$

As the E_0 and E_1 are symmetric, it suffices to solve one term. We have $P(E_0)=2/w$, because E_0 is true if and only if the minimal k_0-mer in the first k-mer is either the first or the last one, and there are w k_0-mers in a k-mer. The only term left is $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}}\mid E_0)$.

In the next two sections, we will upper bound this last term, which in turn bounds $P(c\in\mathcal{C})$. It helps to understand why this argument achieves a bound better than a purely random minimizer, even though the Miniception looks very randomized. The context contains two UHS k-mers on average, because the relative size of $\mathcal{C}_0$ is $2/w+o(1/w)$, so it may appear the expectation

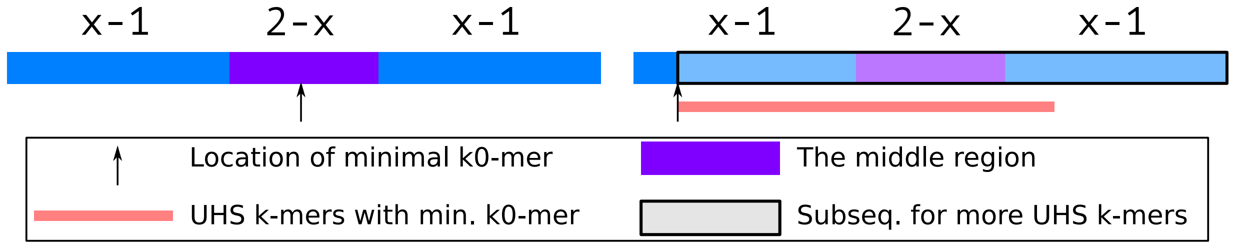
x-1
2-x
x-1
x-1
2-x
x-1
Location of minimal k0-mer
The middle region
UHS k-mers with min. k0-mer
Subseq. for more UHS k-mers

Figure 3.3: Setup for derivation of $P_n(x)$ with $n \geq 1$ and $1 \leq x \leq 2$. The text denotes the relative length of the corresponding substrings. If the minimal k_0-mer fall into the middle region (left panel), there are zero UHS k-mers in the sequence. Otherwise (right panel), there is at least one UHS k-mer with possibility for more from the substring.

term is close to 0.5, which leads to a density bound of $2/w + o(1/w)$, identical to a random minimizer. However, conditioned on E_0, the context provably contains at least one other UHS k-mer, and with strictly positive chance contains two or more other UHS k-mers, which brings the expectation down strictly below 0.5.

3.3.2 Deriving the Unconditional Distribution of UHS k-mer Counts

In this section, we bound the quantity $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}} \mid E_0)$ by deriving the distribution of m_{total}, where $\mathcal{O}(c)$ is sampled from $\mathcal{R}(2)$ conditioned on E_0. We emphasize that at this point the actual k_0-mers are not important and only their order matters. It is beneficial to view the sequence simply as the order $\mathcal{O}(c)$. To prepare for the derivation, we will first derive the distribution of m_{total} assuming $\mathcal{O}(c) \sim \mathcal{R}(2)$ without extra conditions.

We are interested in the asymptotic bound, meaning $w \to \infty$, so we use the following notation.

Definition 21. *Let $\mathcal{R}(x)$ denote the distribution of random order of xw elements. (This is consistent with previous definition of $\mathcal{R}(2)$, as a context contains $2w$ k_0-mers.) The relative length of a sequence is defined by its number of k_0-mers divided by w.*

Definition 22. *Given a sequence of relative length x, where the order of its constituent k_0-mers follows $\mathcal{R}(x)$, $P_n(x)$ denotes the probability that the sequence contains exactly n UHS k-mers.*

As a context is a sequence of relative length 2, we are interested in the value of $P_n(x)$ for $x \leq 2$ using recurrence.

To start, we fix x, the relative length of the sequence, and iterate over the location of the minimal k_0-mer and let its location be tw where $0 \leq t \leq x$. There are two kinds of UHS k-mers for this sequence. The first kind contains the minimal k_0-mer of the sequence, and there can be at most two of them: one starting with that k_0-mer and one ending with that k_0-mer. The second kind does not contain the minimal k_0-mer, so it is either to the left of the minimal k_0-mer or to the right of the minimal k_0-mer, in the sense that it does not contain the minimal k_0-mer in full. Precisely, it is from the substring that contains exactly the set of k_0-mers left to the minimal k_0-mer, or from the substring that contains exactly the set of k_0-mers right to the minimal k_0-mer: these two substrings have an overlap of $k_0 - 2$ bases but do not share any k_0-mer, and neither contain the minimal k_0-mer. We refer to these sequences as the *left substring* and *right substring* for conciseness.

This divides the problem of finding n UHS k-mers into two subproblems: finding UHS k-mers left of location tw, and finding UHS k-mers right of location tw. If we sample an order from $\mathcal{R}(x)$, conditioned on the minimal k_0-mer on location tw, the order of the k_0-mers left of the minimal k_0-mer follows $\mathcal{R}(t)$, and similarly $\mathcal{R}(x-t)$ for the k_0-mers right of the minimal k_0-mer. As we assume $w \to \infty$, we ignore that two substrings combined have one less k_0-mer. We will show later in Section 3.3.6 that such simplification introduces a negligible error. This means the subproblems have an identical structure to the original problem.

We are now ready to write down the recurrence relations for $P_n(x)$, starting with the boundary conditions for both x and n. For $x < 1$, corresponding to a sequence with relative length less than 1, it contains no k-mer so with probability 1 the sequence contains no UHS k-mer. This means $P_0(x) = 1$ and $P_n(x) = 0$ for $n \geq 1$. We next derive the value of $P_0(x)$, that is, the probability the sequence contains no UHS k-mer for $1 \leq x \leq 2$. Define the middle region as the set of k_0-mer locations that are at most $w-2$ k_0-mers away from both the first and the last k_0-mer. The whole sequence contains no UHS k-mer if and only if the minimal k_0-mer falls within the middle region, as only in this case every k-mer contains the minimal k_0-mer but none have it at the boundary. The relative length of the middle region is $2-x$, as we assume $w \to \infty$ (see Figure 3.3). As the order of k_0-mers follows $\mathcal{R}(x)$, every k_0-mer has equal probability to be the minimal and it is in the middle region with probability $(2-x)/x = 2/x - 1$, meaning $P_0(x) = 2/x - 1$.

Now, we handle the general case where $1 \leq x \leq 2$ and $n \geq 1$ (as seen in Figure 3.3). We define the middle region in an identical way as in previous paragraph, whose relative length is again $2-x$. If the minimal k_0-mer is in the middle region, the sequence has exactly zero UHS k-mers. Otherwise, by symmetry we assume it is to the left of the middle region (that is, at least $w-1$ k_0-mers away from the last k_0-mer in the sequence), with location tw where $0 \leq t < x-1$. The sequence now always has one UHS k-mer, that is the k-mer starting with the minimal k_0-mer, and all other $(n-1)$ UHS k-mers come from the substring right to the minimal k_0-mer. The substring has relative length $x-t$, and as argued above, the probability of observing exactly $(n-1)$ UHS k-mers from the substring is $P_{n-1}(x-t)$. Averaging over t, we have the following:

$$P_n(x) = \frac{2}{x}\int_0^{x-1} P_{n-1}(x-t)\mathrm{d}t, n \geq 1, 1 \leq x \leq 2$$

Given $P_0(x) = 2/x - 1$, we can solve for the next few P_n for $1 \leq x \leq 2$ (we will show the results for analytical integration later in this section). Recall our goal is to upper bound $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}} \mid E_0)$. For this purpose, $P_n(x)$ is not sufficient as the expectation is conditioned on E_0. We next derive an analogue to $P_n(x)$, but conditional on E_0.

3.3.3 Deriving the Conditional Distribution of UHS k-mer Counts

We now define the events E_0^+ and E_0^-. E_0^+ is the event that the first k-mer of the Miniception context is a UHS k-mer, because inside the first k-mer the minimal k_0-mer is at the front. Similarly, E_0^- is the event where the first k-mer is a UHS k-mer, because the last k_0-mer in the first

k-mer is minimal. These events are mutually exclusive and have equal probability of $1/w$, so $P(E_0^+ \mid E_0) = P(E_0^- \mid E_0) = 1/2$.

Definition 23 (Restricted Distribution)**.** *$\mathcal{R}^+(x)$ for $x \geq 1$ is the distribution of random permutations of xw elements, conditioned on the event that the first element is minimum among first w elements. Similarly, $\mathcal{R}^-(x)$ for $x \geq 1$ is the distribution of random permutations of xw elements, conditioned on the event that the last element is minimum among first w elements.*

We now have the following:

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}} \mid E_0)\\
&= \mathbb{E}(1/m_{\text{total}} \mid E_0^+)P(E_0^+ \mid E_0) + \mathbb{E}(1/m_{\text{total}} \mid E_0^-)P(E_0^- \mid E_0)\\
&= (\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^+(2)}(1/m_{\text{total}}) + \mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^-(2)}(1/m_{\text{total}}))/2
\end{aligned}
$$

Based on this, we define $Q_n^+(x)$ to be the probability that a sequence of relative length xw, where the order of k_0-mers inside the sequence follows $\mathcal{R}^+(x)$, contains exactly n UHS k-mers. Our goal now is to determine $Q_n^+(x)$ for $x \leq 2$, which also bounds $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^+(2)}(1/m_{\text{total}})$.

The general idea of divide-and-conquer stays the same in deriving a recurrence for $Q_n^+(x)$. It is however trickier to apply this idea with a conditional distribution. We solve this issue by defining the following:

Definition 24 (Restricted Sampling)**.** *With fixed x and w, the restricted sampling process samples a permutation of length xw, then swap the minimum element in the first w element with the first element.*

Lemma 42. *Denote the distribution generated by the restricted sampling process as $\mathcal{S}^+(x)$, then $\mathcal{S}^+(x) = \mathcal{R}^+(x)$.*

Proof. We prove the two distributions are identical in two parts. First, we show they have the same support (generate the same set of permutations). Any permutations sampled from $\mathcal{S}^+(x)$ satisfies E_0^+ by the swapping process, and any permutation satisfying E_0^+ can be sampled from $\mathcal{S}^+(x)$ as it can simply be the initial distribution and unchanged by the swapping process. Second, given $\mathcal{R}^+(x)$ is a uniform distribution over permutations satisfying E_0^+, we only need to prove for any two permutation satisfying E_0^+ they are generated by $\mathcal{S}^+(x)$ with the same probability. This is true because every permutation satisfying E_0^+ has exactly w preimages in the swapping process. □

As the distributions are the same, we redefine $Q_n^+(x)$ with $\mathcal{S}^+(x)$. The boundary condition for $Q^+(x)$ is $Q_0^+(x) = 0$ for all x, because the first k-mer is guaranteed to be a UHS k-mer (note that $Q_n^+(x)$ is defined only with $x \geq 1$).

For $n \geq 1$ and $x \leq 2$, from the process of restricted sampling, we know with probability $1/x$ the minimal k_0-mer in the sequence is the first k_0-mer overall. In this case, the first k-mer is the only UHS k-mer that contains the minimal k_0-mer, and all other UHS k-mers come from the substring without the first k_0-mer whose relative length is still x as we assume $w \to \infty$. We claim the following:

Lemma 43. *Given an order of k_0-mers sampled from $\mathcal{S}^+(x)$, conditioned on the first k_0-mer being overall minimal, the k_0-mer order excluding the first k_0-mer follows the unrestricted distribution $\mathcal{R}(x)$.*

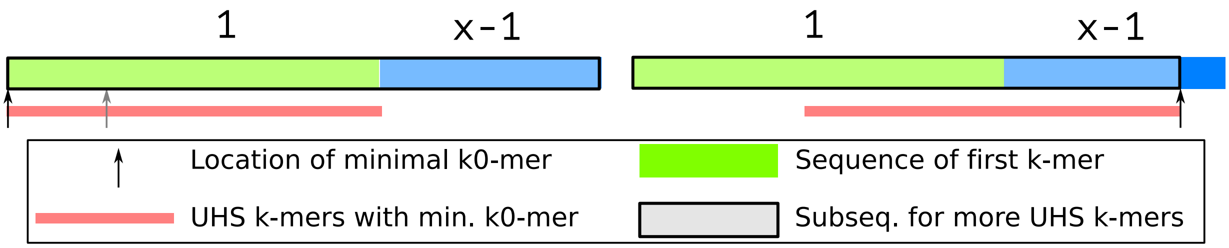
1
x-1
1
x-1
Location of minimal k0-mer
Sequence of first k-mer
UHS k-mers with min. k0-mer
Subseq. for more UHS k-mers

Figure 3.4: Setup for derivation of $Q_n^+(x)$ with $n \geq 1$ and $1 \leq x \leq 2$. The text denotes the relative length of the corresponding substrings. If the minimal k_0-mer is in the first k-mer, it will be the first k_0-mer overall due to the restricted sampling process. In this case, there is one guaranteed UHS k-mer and possibility more in the substring without first k_0-mer. Otherwise, the analysis is similar to the derivation of $P_n(x)$.

Proof. Our goal is to prove the two distributions are equal, and we will use the same two-step process. First, every permutation can be generated by both processes. Second, as proved before, $\mathcal{S}^+(x)$ generates each eligible permutation with identical probability, and this holds for the conditional distribution as each permutation has exactly one preimage. This means it is the same distribution as $\mathcal{R}(x)$ where each permutation is also generated with the same probability. □

This lemma means the probability of observing $(n-1)$ UHS k-mers outside the first k_0-mer is $P_{n-1}(x)$. Otherwise, we use the same argument as before by setting the location of the minimal k_0-mer to be tw, where $1 \leq t \leq x$. Only one UHS k-mer contains the minimal k_0-mer with probability 1 (if $x = 2$, $t = 2$ happens with probability 0), and all other UHS k-mers come from the substring to the left of the minimal k_0-mer. By a similar argument, the order of k_0-mers within the left substring follows $\mathcal{S}^+(t)$. These arguments are also shown in Figure 3.4. Averaging over t, we have the following recurrence for $Q_n^+(x)$, valid for $1 < x \leq 2$ and $n \geq 1$:

$$Q_n^+(x) = \frac{1}{x}\left(P_{n-1}(x) + \int_1^x Q_{n-1}^+(t)\mathrm{d}t\right).$$

Replacing $\mathcal{R}^+(x)$ with $\mathcal{R}^-(x)$, we can similarly define and derive the recurrence for $Q_n^-(x)$ given $1 \leq x \leq 2$. The process is highly symmetric to the previous case for $Q_n^+(x)$. To start with, we similarly define the restricted sampling process (where we swap the minimal element in the first w elements with the w^{th} element instead). We define $Q_n^-(x)$ to denote the same quantity as $Q_n^+(x)$, except the order is now sampled from the new restricted sampling process $\mathcal{S}^-(x)$. The derivation of $Q_n^-(x)$ is identical when the minimal k_0-mer in the sequence is not the w^{th} one. With probability $1/x$, the minimal k_0-mer will be the w^{th} one. In this case, if $x < 2$, again only one UHS k-mers will contain the minimal k_0-mer, but with $x = 2$ two k-mers will contain it (with $x = 2$, the k_0-mer is in the middle of the sequence, with at least $w - 1$ k_0-mers away from each end). All other UHS k-mers now come from the substring right to the minimal k_0-mer, whose relative length is now $x - 1$, and similar to our previous argument, the order within the substring follows $\mathcal{R}(x-1)$. This yields the following recurrence:

$$Q_n^-(x) = \frac{1}{x}(P_{n-1}(x-1) + \int_1^x Q_{n-1}^-(t)\mathrm{d}t), n \geq 1, x < 2$$
$$Q_n^-(x) = \frac{1}{x}(P_{n-2}(x-1) + \int_1^x Q_{n-1}^-(t)\mathrm{d}t), n \geq 1, x = 2$$

Similar to $P_n(x)$, we can derive the analytical solution to these integrals. Here, we perform symbolic integration and show the results for $n \leq 6$.

$$\begin{aligned}
P_0(x) &= 2/x - 1 \\
P_1(x) &= -2 + 4\ln(x)/x + 2/x \\
P_2(x) &= -4 + 4\ln^2(x)/x + 4\ln(x)/x + 4/x \\
P_3(x) &= -8 + 8\ln^3(x)/3x + 4\ln^2(x)/x + 8\ln(x)/x + 8/x \\
P_4(x) &= -16 + 4\ln(x)^4/3x + 8\ln(x)^3/3x + 8\ln(x)^2/x + 16\ln(x)/x + 16/x \\
P_5(x) &= -32 + 8\ln(x)^5/15x + 4\ln(x)^4/3x + 16\ln^3(x)/3x + 16\ln^2(x)/x + 32\ln(x)/x + 32/x \\
P_6(x) &= -64 + 8\ln(x)^6/45x + 8\ln^5(x)/15x + 8\ln^4(x)/3x + 32\ln^3(x)/3x + 32\ln^2(x)/x \\
&\quad + 64\ln(x)/x + 64/x \\
Q_0^+(x) &= 0 \\
Q_1^+(x) &= 2/x^2 - 1/x \\
Q_2^+(x) &= \ln(x)(4/x^2 - 1/x) \\
Q_3^+(x) &= \ln^2(x)(4/x^2 - 1/2x) \\
Q_4^+(x) &= \ln^3(x)(8/3x^2 - 1/6x) \\
Q_5^+(x) &= \ln^4(x)(4/3x^2 - 1/24x) \\
Q_6^+(x) &= \ln^5(x)(8/15x^2 - 1/120x) \\
Q_0^-(x) &= 0 \\
Q_1^-(x) &= \begin{cases} 1/x & x < 2 \\ 0 & x = 2 \end{cases} \\
Q_2^-(x) &= \begin{cases} \ln(x)/x & x < 2 \\ (1 + \ln(x))/x & x = 2 \end{cases} \\
Q_3^-(x) &= \ln^2(x)/2x \\
Q_4^-(x) &= \ln^3(x)/6x \\
Q_5^-(x) &= \ln^4(x)/24x \\
Q_6^-(x) &= \ln^5(x)/120x
\end{aligned}$$

Using the definition of $Q_n^+(x)$, we now bound $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^+(2)}(1/m_{\text{total}})$ by truncating the distribution's tail, as follows (omitting the condition for clarity):

$$\begin{aligned}\mathbb{E}(1/m_{\text{total}}) &= \sum_{i=1}^{\infty} Q_i^+(2)/i \\ &\leq \sum_{i=1}^{n} Q_i^+(2)/i + \left(1 - \sum_{i=1}^{n} Q_i^+(2)\right)/(n+1)\end{aligned}$$

We can derive a similar formula for the symmetric term $\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^-(2)}(1/m_{\text{total}})$. For both Q^+ and Q^-, at $n = 6$ the tail probability $1 - \sum_{i=1}^{n} Q_i(2) < 0.01$, so we bound both terms using $n = 6$, resulting in the following:

$$\begin{aligned}\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}(2)}(1/m_{\text{total}} \mid E_0) &= (\mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^+(2)}(1/m_{\text{total}}) + \mathbb{E}_{\mathcal{O}(c)\sim\mathcal{R}^-(2)}(1/m_{\text{total}}))/2 \\ &< 0.417\end{aligned}$$

Finally, we bound the density of the Miniception, now also using the symmetry conditions (omitting the condition $\mathcal{O}(c) \sim \mathcal{R}(2)$ for clarity):

$$\begin{aligned}P(c \in \mathcal{C}) &= \mathbb{E}(m_{\text{boundary}}/m_{\text{total}}) + o(1/w) \\ &\leq \mathbb{E}(1/m_{\text{total}} \mid E_0)P(E_0) + \mathbb{E}(1/m_{\text{total}} \mid E_1)P(E_1) + o(1/w) \\ &= 4\mathbb{E}(1/m_{\text{total}} \mid E_0)/w + o(1/w) \\ &< 1.67/w + o(1/w)\end{aligned}$$

3.3.4 Density Bounds for Other Configurations

We can derive the recurrence for $P_n(x)$, $Q_n^+(x)$ and $Q_n^-(x)$ for $x > 2$, corresponding to the scenario where $w \approx (x-1)k > k$. By similar techniques, with suitably chosen n, we can upper bound the density of the Miniception from the values of $Q_i^+(x)$ and $Q_i^-(x)$ with $i \leq n$. The resulting bound has form of $D(x)/w + o(1/w)$, where $D(x)$ is the density factor bound. We then have the following theorem:

Theorem 10. *With $x \geq 2$, $w = (x-1)(w_0+1)$, $k = w_0 + k_0$ and $k_0 > (3+\epsilon)\log_\sigma(x(w_0+1))$, the expected density of the Miniception is upper bounded by $D(x)/w + o(1/w)$. (In other words, in the last section we showed that $D(2) \approx 1.67$.)*

For the rest of this section, we derive $D(x)$. Note that as $w = w_0 + 1$ no longer holds, the relative length of a sequence is defined as the number of k_0-mers in the sequence divided by w_0+1, and $\mathcal{R}(x)$ is now a random permutation of $x(w_0+1)$ elements. Definition for $\mathcal{S}^{\pm}(x)$ and the desired quantities change accordingly.

We will follow the same general argument as before, by iterating over the location of minimal k_0-mer within the sequence as $t(w_0+1)$ for $0 \leq t \leq x$, and discuss the different scenarios. The extra consideration comes from the fact that now it is possible that both left and right substrings produce UHS k-mers, so we also need to iterate over the number of UHS k-mers from one substring. We will define the convolution operator to represent this iteration process:

$$[A(s) * B(t)]_n = \sum_{m=0}^{n} A_m(s)B_{n-m}(t)$$

We start with the derivation of $P_n(x)$ for $x > 2$. For $n = 0$ and $x > 2$, $P_n(x) = 0$ as wherever the minimal k_0-mer is, there will be one UHS k-mer containing it. For the rest, we define the middle region as the subregion that is at least w_0 k_0-mers away from both the first and the last k_0-mer in the sequence. Its relative length is $x - 2$. If the minimal k_0-mer falls in this region, two UHS k-mers (recall a k-mer consists of $(w_0 + 1)$ k_0-mers) will contain the k_0-mer and both substrings split by the minimal k_0-mer may contain more UHS k-mers. If the minimal k_0-mer falls outside this region, the analysis is identical to the previous case. This yields the following recurrence:

$$\begin{aligned}P_n(x) &= \frac{1}{x}(2\int_0^1 P_{n-1}(x-t)\mathrm{d}t + \int_1^{x-1}\sum_{m=0}^{n-2} P_m(t)P_{n-2-m}(x-t)\mathrm{d}t)\\ &= \frac{1}{x}(2\int_0^1 P_{n-1}(x-t)\mathrm{d}t + \int_1^{x-1}[P(t) * P(x-t)]_{n-2}\mathrm{d}t)\end{aligned}$$

For $Q_n^+(x)$, we similarly define the middle region. If the minimal k_0-mer before swapping process falls to the left of the middle region (with probability $1/x$), it falls within the first k-mer and its order is swapped with the first k_0-mer. The probability of observing $n-1$ UHS k-mers outside the first k_0-mer is the same $P_{n-1}(x)$. If the minimal k_0-mer is to the right of the middle region, one UHS k-mer contains this k_0-mer and all other UHS k-mers comes from the left substring, following $\mathcal{S}^+(x-1)$. If the minimal k_0-mer is in the middle region, two UHS k-mers contain the k_0-mer and both substrings may contain more UHS k-mers, with the order in the left substring conditioned on $\mathcal{S}^+(t)$ and the order in the right substring conditioned on $\mathcal{R}(x-t)$. This yields the following recurrence:

$$Q_n^+(x) = \frac{1}{x}(P_{n-1}(x) + \int_1^{x-1}[Q^+(t) * P(x-t)]_{n-2}\mathrm{d}t + \int_{x-1}^{x} Q_{n-1}^+(t)\mathrm{d}t)$$

The derivation for $Q_n^-(x)$ is extremely similar to that of $Q_n^+(x)$. Note that now $x \geq 2$, two UHS k-mers are guaranteed when the minimal k_0-mer is the w^{th} one.

$$Q_n^-(x) = \frac{1}{x}(P_{n-2}(x-1) + \int_1^{x-1}[Q^-(t) * P(x-t)]_{n-2}\mathrm{d}t + \int_{x-1}^{x} Q_{n-1}^-(t)\mathrm{d}t)$$

Following the methods of truncating distributions outlined in the main text, noting that now $P(E_0^+) = P(E_0^-) = 1/(w_0+1) = (x-1)/w$, we can calculate density factor bound as follows:

$$\begin{aligned}M_i &= (Q_i^+(x) + Q_i^-(x))/2\\ D(x) &= 4(x-1)(\sum_{i=1}^{n} M_i/i + (1 - \sum_{i=1}^{n} M_i)/(n+1))\end{aligned}$$

assuming the integrals are derived up to n UHS k-mers. The final density bound will be $D(x)/w + o(1/w)$, as promised at the beginning of this section.

3.3.5 Approximate Calculation of Density for Other Configurations

As $D(x)$ represents the density of the Miniception when $w \approx (x-1)k$, with $k \to \infty$, it is natural to approximate $D(x)$ by selecting a large value of k and calculate the density of the Miniception with $w = (x-1)k$. Here, we make the assumption that all k_0-mers in the string are unique and $k_0 \ll k$, so we simply take $k_0 = 1, w = (x-1)k$ and assume the order of the k_0-mers in the context follows $\mathcal{R}(x)$. As we will show later (Section 3.3.6), the final density bound will be off by $O(1/k)$ compared to the integrals, so by choosing k to be large enough we can ensure the derived bound is close to the integral bound, which again by Section 3.3.6 approximates the behavior of the Miniception for arbitrary large values of k.

Now, let $P[l, n]$ be the probability that given permutation of l k_0-mers, there are exactly n UHS k-mers. Recall a k-mer is a UHS k-mer if its first or last k_0-mer is the smallest in the substring, which is of length k now. We again enumerate over the location of the smallest k_0-mer in permutation, and denote this value i. Each specific location is the minimum with probability $1/l$. If $i \geq k-1$, the k-mer that ends at i (meaning its last k_0-mer is this one) is a UHS k-mer. By symmetry, if $i \leq l-k$, the k-mer starting at i is a UHS k-mer. The rest of UHS k-mers will come from the two substrings, obtained by removing the i^{th} k_0-mer from the sequence, and as before, we need to enumerate the number of UHS k-mers from either substrings. This results in the following recurrence, where we use $f_{l,n}(i)$ to denote the number of UHS k-mers that need to come from either substrings:

$$f_{l,n}(i) = n - \mathbf{1}(i \geq k-1) - \mathbf{1}(i \leq l-k)$$

$$P[l, n] = \frac{1}{l} \sum_{i=0}^{l-1} (P[i] * P[l-1-i])_{f_{l,n}(i)}$$

Here $\mathbf{1}(C)$ evaluates to 1 if C is true and 0 otherwise. We also use $*$ as the standard convolution operator: $(A * B)_n = \sum_{i=0}^{n} A[x, i]B[y, n-i]$. With the dynamic programming, we only need to provide that $P[i, 0] = 1, P[i, n] = 0$ for $i < k$ and $n \geq 1$. We can then calculate the values of $P[l, n]$ up to $l = xk$ and some preset value of n with the requirement $n > 2x$.

We can similarly generate the recurrence relationship for $Q^+[l, n]$ and $Q^-[l, n]$ (derivations are highly similar and omitted here):

$$Q^+[l, n] = \frac{1}{l}(kP[l-1, n-1] + \sum_{i=k}^{l-1} (Q^+[i] * P[l-1-i])_{f_{l,n}(i)})$$

$$Q^-[l, n] = \frac{1}{l}(kP[l-k, n-1-\mathbf{1}(l \geq 2k-1)] + \sum_{i=k}^{l-1} (Q^-[i] * P[l-1-i])_{f_{l,n}(i)})$$

The approximate density factor bound (we use the density factor bound for numerical stability and ease of comparison), which we denote as $D_k(x)$, can be calculated as follows using our previous argument of truncating the distribution.

$$M'_i = (Q^+[xk, i] + Q^-[xk, i])/2$$
$$D_k(x) = 4(x-1)(\sum_{i=1}^{n} M'_i/i + (1 - \sum_{i=1}^{n} M'_i)/(n+1))$$

The final density bound for the Miniception under this configuration will be $D_k(x)/w + o(1/w)$, where the $o(1/w)$ term comes from the event that some k_0-mers in a context can be identical.

3.3.6 Bounding the Errors for the Integral Methods

In the derivation for the Miniception, we assume $k \to \infty$ and use an integral instead of summation for derivation of $P_n(x)$ and other quantities. In this section, we show the error introduced by this method is small enough to not affect the final result asymptotically. Specifically, we show the following:

Lemma 44. *Let $D(x)$ be the density factor bound derived from the integrals, and $D_k(x)$ be the density factor bound calculated from the dynamic programming with window length $(x-1)k$ and k-mer length k (measured in number of k_0-mers), assuming all k_0-mers are distinct. We have $|D(x) - D_k(x)| = O(nx/k)$, where n is the largest number of UHS k-mers considered by both processes.*

This lemma serves a dual purpose. It means $D(x)$ is a good approximation to any $D_k(x)$ up to asymptotically negligible errors, and by calculating $D_k(x)$ for sufficiently large k, we can estimate $D(x)$ accurately which in turn approximates other $D_k(x)$. In other words, $D(x)$ serves as a bridge to connect the density factor bound $D_k(x)$ for different values of k, which then bound the density of the actual minimizer as long as k_0 satisfies the condition for Lemma 10. Note that in derivation of $D_k(x)$ the concept of k_0-mers are already abstracted away, and for simplicity we assume $k_0 = 1$ in that process, meaning every k-mer now consists of k k_0-mers.

Recall that we set up the integrals to derive the distribution of m_{total} conditioned on first k-mer being a UHS k-mer. We now assume $k_0 = 1$, meaning $k = w_0 + 1$ (as done in Section 3.4.1 when we try to use dynamic programming to approximate $D(x)$). The context has exactly xk k_0-mers, and each k-mer is of length k. All lengths in the section are measured in k_0-mers. As we have shown before, conditioned on E_0, the context will contain at least two UHS k-mers.

We have calculated a general formula for $P_n(x)$ where the final formula contains integrals. An alternative view of the integral recurrence is that we determine at most $n+1$ locations that contain the minimal k_0-mers in a certain substring and calculate the number of UHS k-mers based on the order and the distance of these locations. For example with $x = 2$, if the first location (the minimal k_0-mer in the whole context) is on the left half of the context, the second location is for the minimal k_0-mer in the substring right to the first location. If the second location is in the middle region (See Section 3.3.2 for definition) of that substring, we determined the number of UHS k-mers (1 in this case) using two locations. If instead, the second location is not inside the middle region, a third location is determined inside the sub-sub-sequence that could still generate a UHS k-mer (which for our case is the either the sequence between the first and second location, or the sequence between second location and the right end), and these three locations can collectively determine how many UHS k-mers are in the whole context. This process continues until no substrings can generate more UHS k-mers, or $n+1$ UHS k-mers have been generated. In this process, we claim there are at most $n+2$ determined locations. This is because of the fact that after the first step, each time we need to determine a new location inside a subsequence, a new UHS k-mer has been determined in the step that splits into this subsequence. Similarly, when we derive the values of $Q_n^+(x)$ and $Q_n^-(x)$, we can take an alternative view where determine the location of the minimal k_0-mer before swapping.

We denote the sequential decision process to determine the number of UHS k-mers, conditioned on E_0^+ (corresponding to $Q_n^+(x)$) and E_0^- (corresponding to $Q_n^-(x)$), as $m \sim \mathcal{P}_n^+(x)$ and $m \sim \mathcal{P}_n^-(x)$, respectively. Formally:

$$P_{m\sim\mathcal{P}^+(x)}(m=i) = Q_i^+(x) \qquad\qquad i \le n$$

$$P_{m\sim\mathcal{P}^+(x)}(m=n+1) = 1 - \sum_{j=1}^{n} Q_j^+(x) \qquad\qquad i = n+1$$

And this is symmetric for $\mathcal{P}^-(x)$. With this definition, we have:

$$D(x) = 2(x-1)\left(\mathbb{E}_{m\sim\mathcal{P}^+(x)}(1/m) + \mathbb{E}_{m\sim\mathcal{P}^-(x)}(1/m)\right)$$

Similarly, we can propose an alternative view of the dynamic programming, where we determine at most $n+1$ locations that contains the minimal k_0-mers in a certain sequence. The difference is that in the previous cases, the locations are picked on a real axis $[0,x]$, but now it is picked from a discrete set of locations in $[xk]$. We will now argue the two processes are not that different.

We first establish a mapping $M(t) : \mathbb{R} \to [xk]$ from real numbers to discrete locations, by multiplying the real number by k and rounding down. When a real number is sampled uniformly from $[0,x]$, its mapping will be a uniform distribution over $[xk]$ (that is, $1/xk$ chance to map into each location ignoring truncation errors as we assume $k \to \infty$). Now, we consider emulating the decision process from the dynamic programming (which we denote $\mathcal{P}_k^+(x)$, for simplicity). Each time a location is randomly sampled (the location of minimal k_0-mer in current substring of consideration) from $\mathcal{P}^+(x)$, we emulate one step of $\mathcal{P}_k^+(x)$ by selecting the mapped discrete location as the location of the minimal k_0-mer. The process continues until the end of decision process.

We now examine the emulation process in more detail. At each step of the decision process, we sample the location of the minimal k_0-mer in current substring of consideration, before the swapping process if the substring of consideration is still conditioned on E_0^+ or E_0^-. The boundary of the substring is determined by two previous samples (or simply the boundary), which we denote L and R. The discrete decision process we are emulating will be sampling the location of the minimal k_0-mer in current substring of consideration, which will be the substring from $M(L)$ to $M(R)$, including neither ends. It can be seen that we are sampling from a slightly longer range in the original process. As long as our sampled location T satisfies $M(L) < M(T) < M(R)$, the distribution of $M(T)$ is uniform over possible selections. If $M(T) = M(L)$ or $M(T) = M(R)$ at any step of the emulation, we call the whole process failed and do not continue. (in other words, if during the $n+2$ steps we sampled two locations into the same "bucket" indexed by M, the whole process fails.) The swapping process can be addressed similarly, as in this case $L = 0$ is guaranteed, and the emulated and original decision process will never disagree on whether the swap should be triggered.

Assume $M(T)$ passes this check, meaning it is uniformly sampled from possible locations between $M(L)$ and $M(R)$. We will now decide on the next step in the process. This includes two parts: Counting number of UHS k-mers including the minimal k_0-mer, and determining if a substring can still produce more UHS k-mers. Both parts involves two distance checks: If $T - L > 1$ in the original decision process, and if $M(T) - M(L) \geq w_0$ in the discrete decision process, for checking if the k-mer ending with the minimal k_0-mer in this substring constitutes a valid UHS k-mer. If the original decision process and the emulated decision process will give different answer to this check, we call the whole process failed and do not continue. For deciding if the left substring can produce more UHS k-mers, we replace w_0 with $w_0 + 1$ (e.g. checking if $M(T) - M(L) \geq w_0 + 1$, failing in case of disagreement between the continuous simulation and discrete sampling) and the analysis is the same. It is also symmetric for the right substring and checking the k-mer starting with minimal k_0-mer.

Now, if a decision process does not fail (by either of the previously defined criteria), we say it perfectly emulates a discrete decision process. By the way it is defined, the emulated process randomizes the location of minimal k_0-mers correctly, make the same decision as the original process at every step, and count the same number of UHS k-mers. We now characterize these "safe decision processes" with the following notation. Let $\mathcal{S}_k^+$ denote a distribution whose support is $\{\times, 1, 2, 3, \ldots, n+1\}$ derived from the aforementioned emulation process. × denotes a failed process, and an integer denotes the number of UHS k-mers determined from the process. We claim $P_{m\sim\mathcal{S}_k^+}(m = i) \leq P_{m\sim\mathcal{P}^+}(m = i)$ and $P_{m\sim\mathcal{S}_k^+}(m = i) \leq P_{m\sim\mathcal{P}_k^+}(m = i)$ simultaneously for all $i \in \{1, 2, 3, \ldots, n+1\}$. This essentially means that the "safe decision process" is a conservative estimate to both the continuous and the discrete selection process. If the "bad cases" are rare, the safe process is a close estimate to both and the two processes are similar.

The first part of the statement can be proved by the definition of the emulation process: $P_{m\sim\mathcal{S}_k^+}(m = i)$ is the probability that the emulation successes and returns i UHS k-mers, which is never greater than the probability of i UHS k-mers regardless of the emulation outcome ($P(A \cap B) \leq P(A)$). The second part is can be proved by viewing the emulation from the perspective of the discrete process (which we call a reverse emulation, however they are in fact the same process): At each step of the discrete decision process, we first randomly select the location of the minimal k_0-mer, and randomly choose a real number that maps to this location as its real coordinate. We then do a failure check by rolling a random number inside $[L, R]$ (from the real number locations selected in previous steps), and fail the process if the rolled location collide. Similarly, at each distance check we first get the outcome from discrete locations, then fail the process if the real number locations disagree with the outcome. In this perspective, we can prove the second part of the statement in the same way: $P_{m\sim\mathcal{S}_k^+}(m = i)$ is the probability that this reverse emulation successes and returns i UHS k-mers, and is never greater than the probability of getting i UHS k-mers regardless of the outcome of reverse emulation.

We now bound the probability of failure, that is $P_{m\sim\mathcal{S}_k^+}(m = \times)$. As the decision process consists of $n + 2$ steps at most, we will bound the probability at each step. At each step, the interval for random selection has at least relative length of 1 (otherwise no UHS k-mers will be from this section), and there are at least k k_0-mers for selection. The key observation here is there are only 6 ways to fail (colliding with $M(L)$, disagreement on whether left substring contains more UHS k-mers, disagreement on whether the k-mer ending at smallest k_0-mer is valid UHS k_0-mer, and their symmetric counterparts), and for each of them, there is a interval of length at most $1/k$ such that the fail event is triggered if and only if the minimal k_0-mer is inside the interval. In other words, the probability to fail at each decision step is at most $6/k = O(1/k)$, and the probability of failure at any step is $O(n/k)$.

We now have the tools to bound $|D(x) - D_k(x)|$, the error we get from the dynamic programming process. We first bound the term $\mathbb{E}_{m\sim\mathcal{P}^+(x)}(1/m)$. We denote S as the event that the emulation is successful, and $\mathcal{F}^+(x)$ as the distribution of UHS k-mer count when the emulation fails:

$$\begin{aligned}\mathbb{E}_{m\sim\mathcal{P}^+(x)}(1/m) &= \sum_{i=1}^{n+1} P_{m\sim\mathcal{P}^+(x)}(m=i)/i\\ &= P(S)\sum_{i=1}^{n+1} P_{m\sim\mathcal{S}_k^+(x)}(m=i)/i + P(\bar{S})\sum_{i=1}^{n+1} P_{m\sim\mathcal{F}^+(x)}(m=i)/i.\end{aligned}$$

And similarly, let $\mathcal{F}_k^+(x)$ as the distribution of UHS k-mer count when the reverse emulation fails:

$$\begin{aligned}\mathbb{E}_{m\sim\mathcal{P}_k^+(x)}(1/m) &= \sum_{i=1}^{n+1} P_{m\sim\mathcal{P}_k^+(x)}(m=i)/i\\ &= P(S)\sum_{i=1}^{n+1} P_{m\sim\mathcal{S}_k^+(x)}(m=i)/i + P(\bar{S})\sum_{i=1}^{n+1} P_{m\sim\mathcal{F}_k^+(x)}(m=i)/i.\end{aligned}$$

This leads to the following:

$$\begin{aligned}&|\mathbb{E}_{m\sim\mathcal{P}^+(x)}(1/m) - \mathbb{E}_{m\sim\mathcal{P}_k^+(x)}(1/m)|\\ &= P(S)|\sum_{i=1}^{n+1}(P_{m\sim\mathcal{F}^+(x)}(m=i) - P_{m\sim\mathcal{F}_k^+(x)}(m=i))/i|\\ &\leq P(S) = O(n/k).\end{aligned}$$

Substitute this back, we get $|D(x) - D_k(x)| = O(nx/k)$.

We should also point out that the bound is not tight. This is because we overestimate the difference term $\sum_{i=1}^{n+1} P_{m\sim\mathcal{S}_k^+(x)}(m = i)/i$ by simply upper bounding it by 1. However, for larger value of k and x, the context is almost guaranteed to have around $2x$ UHS k-mers, and the difference here is more on the order of $O(1/x^2)$. This means in practice the bound is more like $O(1/k)$ (considering $n = \Theta(x)$ is a reasonable choice) with the possibility of even smaller (as we do not consider that positive terms and negative terms cancel out each other), which also explains why we are comfortable to pick $k = 2500$ in our simulations as shown next section.

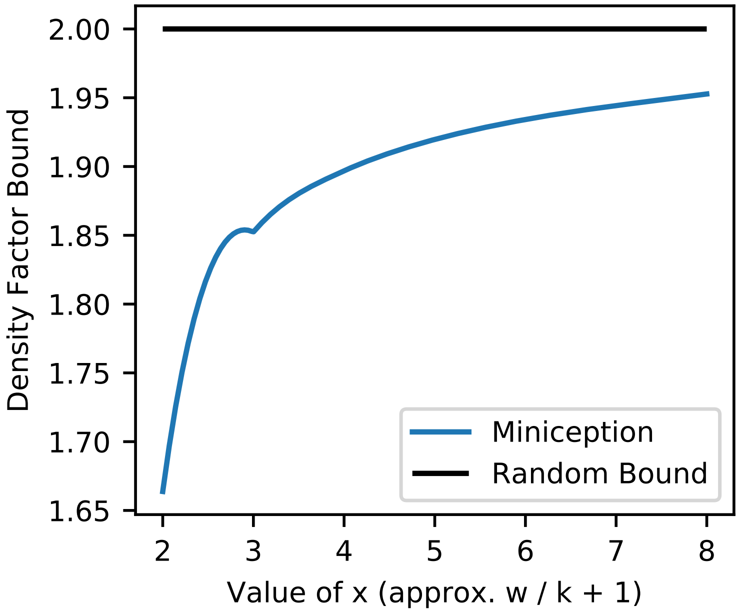

Figure 3.5: Density factor for the Miniception when $w \approx (x-1)k$ and $k = 2500$. The random minimizer achieves density factor of constant 2 and is plotted for comparison.

3.4 Empirical Evaluation of the Miniception

3.4.1 Asymptotic Performance of the Miniception

We use the dynamic programming formulation described in Section 3.3.5 to calculate the density factor of the Miniception given $w \approx (x-1)k$, with a large value of $k = 2500$. As analyzed in Section 3.3.6 this accurately approximates $D(x)$, which in turn approximates the density factor of the Miniception with other values of k up to an asymptotically negligible error. Figure 3.5 shows estimated $D(x)$ for $2 \leq x \leq 8$.

Consistent with Section 3.3, $D(2) \approx 1.67$, as $x = 2$ corresponds to the case $w \approx k$. There is no analytical form for $D(x)$ with $x > 2$, but this experiment suggests that as x grows, $D(x)$ increases while staying below 2, the density factor of a random minimizer. That is, as w gets increasingly larger than k, the Miniception performance regresses to that of a random minimizer. We conjecture that $D(x) = 2 - \Omega(1/x)$ as x grows. (that is, when the windows are x long measured in relative length, the savings measured in density factor is still more than $\Theta(1/x)$.)

3.4.2 Designing Minimizers with Long k-mers

Evaluation Using a Random Sequence

As seen in the implementation of Miniception (Section 3.2.2), the run time of the Miniception minimizer is the same as a random minimizer. Therefore, it can be used even for large values of k and w. This contrasts with PASHA ([28]), the most efficient minimizer design algorithm prior to our work, which only scales up to $k = 16$.

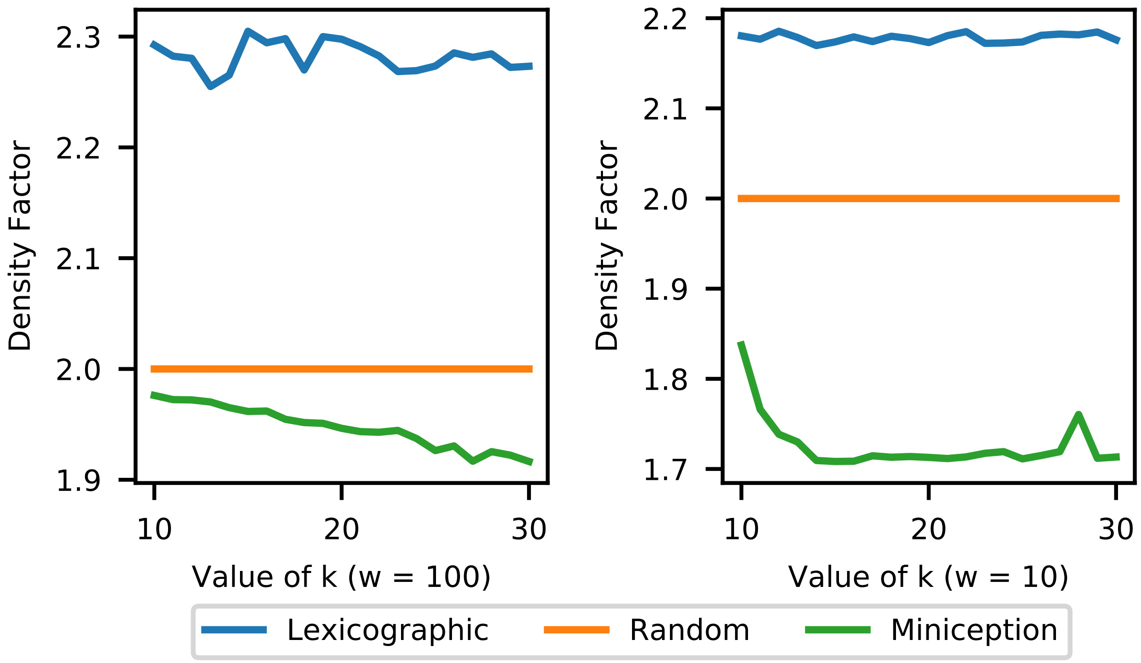

Figure 3.6: Comparing density of the Miniception against lexicographic and random minimizers. Left side: Fixed $w = 10$, right side: fixed $w = 100$.

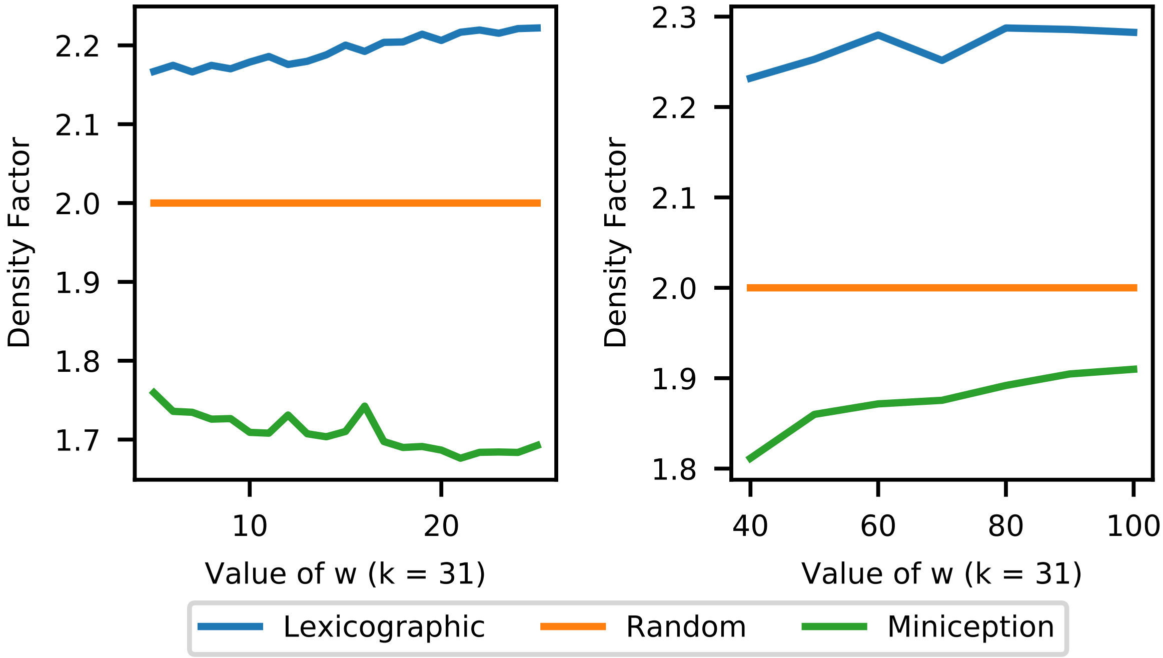

Figure 3.7: Comparing density of the Miniception against lexicographic and random minimizers. Left side for smaller value of w, right side for larger values of w, all with $k = 31$. Note that PASHA is computationally limited to $k \leq 15$ and thus cannot handle these cases.

We implemented the Miniception and calculated its density by sampling $1\,000\,000$ random contexts to estimate the probability of a charged context (which is equivalent to estimating the density as discussed before). The Miniception has a single tunable parameter k_0. It is important to pick an appropriate value of k_0: too small value of k_0 invalidates the assumption that most k_0-mers are unique in a window, and too large value of k_0 increases the value of x. Following this idea, our recommendation is to set k_0 close to $k - w$ if k is larger than w (which corresponds roughly to $x = 2$ in our analysis of the Miniception), and a constant multiple of $\log_\sigma w$ if w is larger than k (roughly corresponding to $x = w/k+1$ in our analysis). A parameter search around the recommended range is also recommended if performance is a critical concern.

We tune this parameter in our experiments by setting $k_0 = k - w$ if k is larger than $w + 3$. If this does not hold, we test the scheme with $3 \leq k_0 \leq 7$ and report the best-performing one. We show the results for Miniception against lexicographic and random minimizers in four setups: two with fixed w and two with fixed k (Figure 3.6 and Figure 3.7). These parameter ranges encompass values used by bioinformatics software packages.

The Miniception consistently performs better than the lexicographic and random minimizers in all tested scenarios. For the setups with fixed $w = 10$, k is larger than w and the Miniception achieves density factor of ≈ 1.72 for $k \geq 13$. Given that $1/w = 0.1$, our bound on the density factor of $1.67 + o(1/w)$ holds relatively well for these experiments. The same conclusion holds for the setup with fixed $k = 31$ and $5 \leq w \leq 20$.

For the experiments with $w = 100$, w is larger than k and we observe the same behavior as in Section 3.4.1: the performance degrades when k becomes smaller than w. Same conclusion holds for the setup with fixed $k = 31$ and $40 \leq w \leq 100$. Our theory also correctly predicts this behavior, as the decrease of $x \approx w/k + 1$ improves the density bound as seen in Section 3.3.

3.4.3 Comparison with PASHA (Prior SoTA)

Evaluation on Random Sequences (for Density)

We compare the Miniception with PASHA [28], an efficient and highly parallelized algorithm that constructs small universal hitting sets similar to DOCKS [85]. We downloaded the PASHA generated UHS from the project website, and implemented the compatible minimizers (Section 3.2.1) according to Ekim et al. [28]. Our test consists of two parts. For the first part, we fix $k = 13$ and vary w from 20 to 200. For the second part, we fix $w = 100$ and vary k from 7 to 15. This setup showcases some of the largest minimizers that can be computed by PASHA (we are unable to parse the UHS file for $k = 16$ as some of the lines are not exactly 16-mers). In Figure 3.8, we observe that the Miniception performs better than the random minimizers for these configurations on a random sequence, but PASHA, even though it is a heuristic without a density guarantee, overall holds the edge.

Evaluation on hg38

We perform experiments on the hg38 human reference genome sequence, where we observe similar results with a smaller performance edge for PASHA.

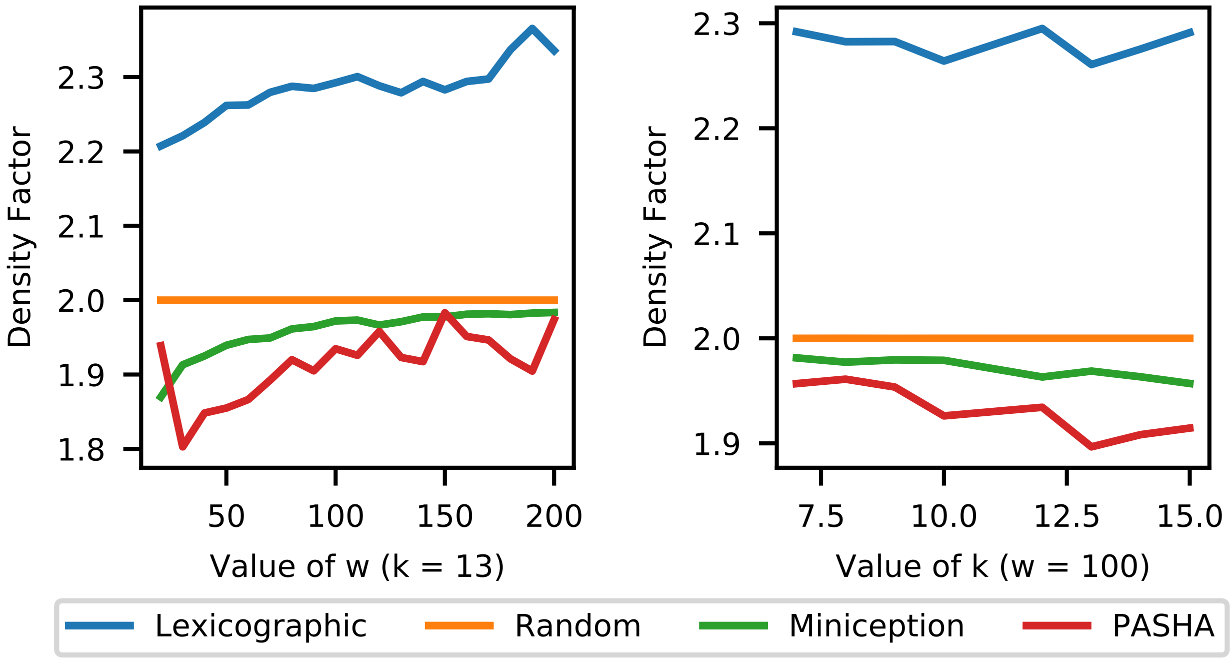

Figure 3.8: Comparing density of the Miniception against lexicographic, random and PASHA minimizers. Here, the configurations follow those as done in the PASHA paper, including its extreme cases.

As observed by Roberts et al. [95, 96], ordering the alphabet by reverse frequency may improve performance of the lexicographic minimizer. For this reason, we use the order $C < G < A < T$ for all minimizers where lexicographical order is needed. We also consider two strategies for constructing a compatible minimizer given the UHS from PASHA. The authors suggested a lexicographic order within the UHS, but we also implement a minimizer where the order within the UHS is randomized, like the Miniception.

Although PASHA has lower density than Miniception, it is limited to $k \leq 16$. Moreover, computing minimizers with PASHA requires storing in memory a relatively large set of k-mers (5 to 10 millions mers for $k = 13$). The Miniception, as we discussed before, does not need to store any set.

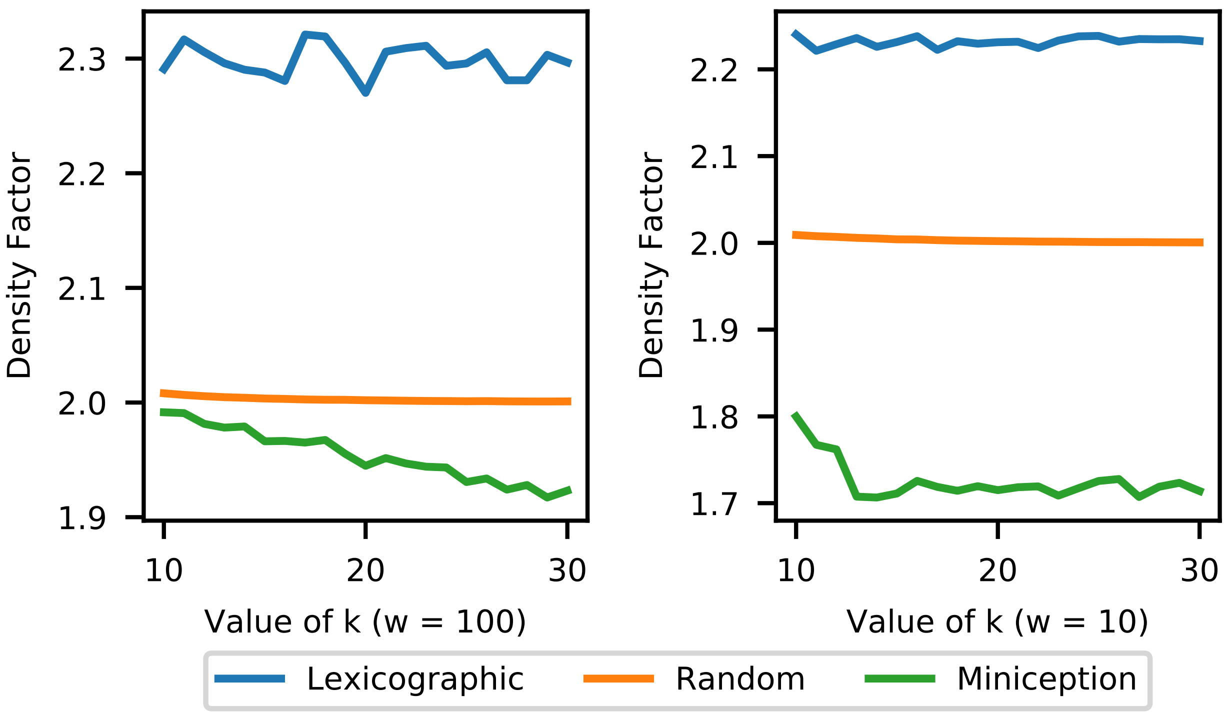

Figure 3.9: Comparing density of the Miniception against lexicographic and random minimizers on hg38. We experiment with $w = 10$ (left) and $w = 100$ (right).

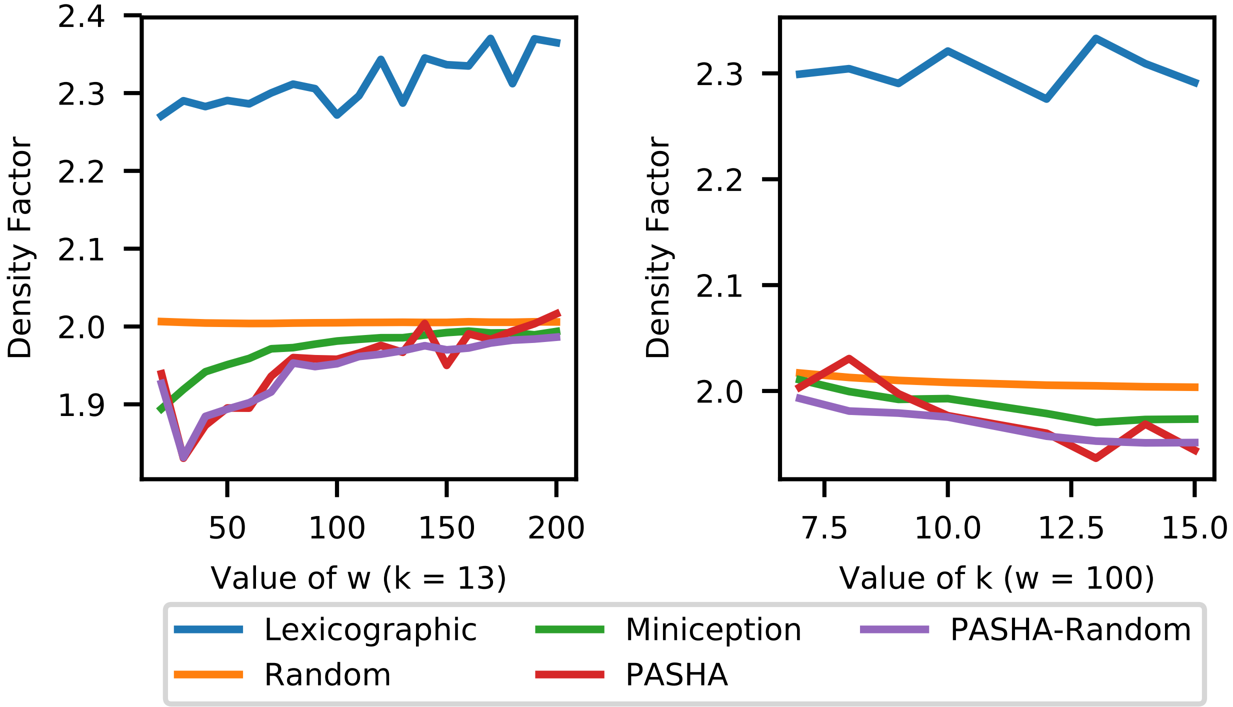

Figure 3.10: Density factor comparison on hg38 for Miniception with minimizers derived from PASHA outputs. Left: experiments with fixed $k = 13$ and varying w. Right: experiments with fixed $w = 100$ and varying k.

Chapter 4

Low-Density, Sequence-Specific Minimizer Sketches

In this chapter, we explore optimizing low-density minimizer sketches for sequence-specific scenarios (See Section 1.4.2 for more detailed discussion). The idea of constructing sequence sketches tailored to a specific sequence has been explored before [21, 24, 46], but it remains less understood than the average case. Random sequences have nice properties that allow for simplified probabilistic analysis, which are absent when the sequence is given (possibly in an adversarial way). Consequently, different analytic tools are needed to analyze sequence-specific minimizers. Previous minimizers designed to have low density in the average case often offer only modest improvements on sequences of interest such as reference genomes (for example, the Miniception as introduced in the previous chapter has this property).

The current theory for minimizers with low density in average is tightly linked to the theory of *universal hitting sets* (UHS) [52, 74, 85]. As the name suggests, a UHS is a set of k-mers that "hits" every w-long window of every possible sequence (hence the universality; it is an unavoidable set of k-mers). Universal hitting sets of small size generate minimizers with a provable upper-bound on their density. Universal hitting sets are less useful in the sequence-specific case as the requirement to hit every window of every sequence is too strong, and UHSs are too large to provide a meaningful upper-bound on the density in the sequence-specific case. New theoretical tools are needed to analyze the sequence-specific case.

We set the stage with notations in Section 4.1. In Section 4.2, we focus on the theory side of sequence-specific minimizer sketches. We will explore several shortcomings of the current universal hitting sets framework, propose the polar set, and show that the model has many desirable properties. In Section 4.3, we focus on the practical side of polar set derived minimizer sketches. We start by designing and analyzing an algorithm to optimize polar sets given a particular input string, then implement the algorithm and perform thorough comparison with existing algorithms on a variety of cases in Section 4.4. We finish with Section 4.5, a short discussion about the fact that polar sets can be extended in many ways, even including universal hitting sets as a special case while enabling more fine-grained analyses.

A version of this chapter was published in ISMB 2021 [129].

4.1 Prelimaries and Overview

We redefine some terms that we need to use within this chapter, for self-containment.

The sequence S is a string on the alphabet Σ of size $\sigma = |\Sigma|$. The parameters k and w define respectively the length of the k-mers and the window size. We assume that S is relatively long compared to these parameters: $|S| \gg w + k$.

Definition 25 (Minimizer and Windows)**.** *A minimizer is characterized by* $(w, k, \mathcal{O})$ *where* $\mathcal{O}$ *is a complete order of* Σ^k*. A window is a sequence of length* $(w + k - 1)$ *consisting of exactly* w k*-mers. Given a window as input, the minimizer outputs the location of the smallest* k*-mer according to* $\mathcal{O}$*, breaking ties by preferring the leftmost* k*-mer.*

The minimizer $(w, k, \mathcal{O})$ is applied to the sequence S by finding the position of the smallest k-mer in every window of S. Because two consecutive windows in S have a large overlap, the same k-mer is often selected in these two windows, and hence the minimizer returns a sampling of positions in the sequence S. The *specific density* of the minimizer on S is defined as the number of selected positions divided by the length $|S|$ (In previous chapter we set the divisor as number of k-mers in S, but for long sequences the difference is negligible).

The density is between $1/w$, because at least one k-mer in every window must be picked, and 1, because it is a sampling of the positions of S. Therefore the goal is to find orders $\mathcal{O}$ that have a density as close to $1/w$ as possible. A minimizer with density $1/w$ is a *perfect minimizer*. For simplicity, when stating the density of a minimizer we ignore any additive term that is $o(1/w)$ (i.e., asymptotically negligible compared to $1/w$).

A *random minimizer* is defined by choosing at random one of the permutations of all k-mers. The expected density of a random minimizer over a random string is $2/(w+1)$ in most cases (see Section 2.7 for a more rigorous argument). Equivalently, the expected distance between adjacent selected k-mers is $(w + 1)/2$. The random minimizers will serve as a baseline against which to compare.

Defining orders. For practical reasons, we define orders by defining a set U (not necessarily a universal hitting set) and considering orders that are *compatible* with U: an order $\mathcal{O}$ is compatible with U if every element of U compares less than any element not in U under $\mathcal{O}$. That is, only the smallest elements for $\mathcal{O}$ are specified (the elements of U) and a minimizer using an order compatible with U will preferentially select the elements of U. There exist many orders that are compatible with U as the relative order between the elements within U is not specified.

Universal Hitting Sets. A set U is a universal hitting set if for every one of σ^{w+k-1} possible windows (recall σ is the size of the alphabet), it contains a k-mer from U. In the average case, minimizers compatible with U have densities upper bounded by $|U|/\sigma^k$, because only k-mers from the universal hitting set can be selected. The next section provides a more detailed discussion of why this bound provided by universal hitting sets does not always apply for sequence-specific minimizer analysis, and why universal hitting sets do not specialize well.

Short sequences. On a short random sequence (in a sense made precise by Lemma 49) most k-mers are unique (i.e., they occur only once in the sequence S). Therefore, it is likely that there is a set U of unique k-mers of S that are exactly w bases apart in S, and a minimizer compatible with U is perfect. Unfortunately most sequences of interest (e.g., reference genomes) are too long, too repetitive and in general do not satisfy the hypothesis of Lemma 49. For most sequences it is not possible to find a set of "perfect seeds" of k-mers spaced exactly w apart.

Polar sets. A *polar set* is a relaxed version of a perfect set: any pair of k-mers m_1 and m_2 from a polar set A is always more than $w/2$ bases apart in S (see the more general Definition 27). The intuition behind this definition is that for a minimizer compatible with A, any k-mer from A selected by the minimizer is at distance $\geq (w+1)/2$ from the previous and the next selected k-mer. Hence, k-mers selected from A are at least as sparse, and usually more sparse than k-mers selected using a random minimizer in expectation.

We will also prove a technical lemma, which we actually proved the first time in Lemma 36 back in Section 2.7.3. We do it again for self-containment.

Lemma 45. *Given a random sequence and a pair of locations $i < j$, the probability that the k-mer starting at i equals the k-mer starting at j is exactly σ^{-k}.*

Proof. If $j - i \geq k$, the two k-mers do not share bases, so given they are both random k-mers independent of each other, the probability is σ^{-k}. Otherwise, the two k-mers intersect. We let $d = j - i$, and use m_i to denote the k-mer starting at location i. We use s to denote the substring from the start of m_i to the end of m_j with length $k + d$ (or equivalently, the union of m_i and m_j). If $m_i = m_j$, the p^{th} character of m_i is equal to the p^{th} character of m_j, meaning $s_p = s_{p+d}$ for all $0 \leq p < k$. This further means s is a repeating sequence of period d, so s is uniquely determined by its first d characters and there are σ^d possible configurations of s. The probability a random s satisfies $m_i = m_j$ is then $\sigma^d/\sigma^{k+d} = \sigma^{-k}$. □

4.2 Theory of Sequence-Specific Minimizers and Polar Sets

4.2.1 A Walkthrough of Universal Hitting Set-Based Methods

Universal hitting sets have been an important component in constructing practical minimizers. In this section, we provide a more formal and technical discussion on universal hitting sets. First, we formally define UHS and discuss why existing heuristics to construct UHS are not adequate for sequence-specific minimizer. Then, we discuss the two existing methods to analyze compatible minimizers of UHSes, and show that these approaches both have issues that make them unfit for our goal.

Definitions and Inelasticity of UHS

Definition 26 (Universal Hitting Sets). *Let U be a set of k-mers. If U intersects with every w consecutive k-mers, it is a UHS over k-mers with path length w and relative size $|U|/\sigma^k$.*

A decycling set is a set of k-mers that intersect with any sufficiently long strings. Any universal hitting set must be a decycling set, so lower bound on the size of decycling sets applies to all universal hitting sets.

Lemma 46 (Minimal Decycling Sets)**.** *Any UHS over k-mers with finite path length has relative size $\Omega(1/k)$.*

With a universal hitting set, it is guaranteed that any compatible minimizer will only select k-mers within the UHS on any sequence. Currently, the most popular approach for constructing efficient minimizers is via construction of a compact universal hitting set, followed by constructing a compatible minimizer. These universal hitting sets are usually constructed by expanding from a minimal decycling set (MDS). As we have shown in Theorem 3 back in Section 2.5, the Mykkeltveit MDS [82], the MDS that is predominantly used as the starting point, already covers all windows of length $O(k^3)$. Empirically, with larger value of w only a few k-mers needs to be added to satisfy the universal hitting condition. As a result, UHSes constructed for different references look like each other, and the compatible minimizers do not specialize well.

A related concern about using UHSes on specific sequences is on handling of repetitive k-mers. As we have discussed, repetitive k-mers are prevalent in human reference genome. Any universal hitting set always contains homomers like $AAA\cdots A$ as it is required to cover a sequence of all As. This argument also extends to other repetitive k-mers. Such homomers, or repetitive k-mers, would then be preferred when using compatible minimizers for sequence sketching. This problem of prioritizing repetitive k-mers is also present in fixed interval sampling. Meanwhile, existing literature [46, 63] suggests it is in fact beneficial to not select these k-mers for read mapping, while proposing different remedies to this issue. Our proposed methods also have the effect of avoiding repetitive k-mers, as it is likely some of repetitive k-mer appears close to another copy of itself, automatically disqualifying for inclusion. (However, if the priority set is not universal hitting as we will do this section, exclusion from the priority set does not guarantee that a k-mer will never be selected by a compatible minimizer.)

Analysis via Density Upper Bound

There are two existing ways to analyze the density of compatible minimizers. The first is via the following lemma, as we have mentioned in the main text:

Lemma 47. *If U is a UHS over k-mers, any compatible minimizer has density at most $|U|/\sigma^k$.*

This lemma is universally applicable, and it does not depend on the ordering within U. It also applies to sequence-specific cases, by replacing $|U|/\sigma^k$ by the number of UHS k-mer occurrences divided by number of k-mers in the input sequence. However, this is an upper bound which becomes non-informative with $w > 2k$ and sufficiently large k. Because any universal hitting set is at least as large as a minimal decycling set (Lemma 46), and a random $(w, k)-$minimizer achieves density of approximately $2/(w+1)$, Lemma 26 at best tells us the compatible minimizer is no worse than a random one when $w > 2k$. We can reach a similar conclusion for the sequence-specific case, where k-mers in UHS occur so often that their occurrence does not imply selection in the majority of cases.

Analysis via Probability of Single UHS Contexts

There is a second approach to analysis of compatible minimizers from universal hitting sets [73]. The key lemma reads as follows (slightly rephrased):

Lemma 48. *If U is a UHS over k-mers, let $SP(U)$ be the probability that a context contains only one element in U. Under certain assumptions, the expected density of a random minimizer compatible with U is $2(1 - SP(U))/(w + 1)$.*

We now show this lemma depends on assumptions that highly depends on the structure of U.

We start with some notations, slightly different from the original paper. Fix a context, let m_i denote the i^{th} k-mer in the context. We also let $z_i = \mathbf{1}(m_i \in U)$, let H denote the event that the context is charged, and let $Z = \sum_{i=0}^{w} z_i$. Let $C(n, k)$ be the binomial coefficients. The proof involves the following equation (we only list the first term - there are four analogous terms):

$$P(H \mid Z = j) = \frac{C(w-1, j-1)}{C(w+1, j)} P(H \mid Z = j, z_0 = 1, z_w = 0) + \cdots$$

which involves a counting argument: Given $Z = \sum_{i=0}^{w} z_i = j$, there are $C(w + 1, j)$ different configurations of z, and $C(w - 1, j - 1)$ of them satisfies $z_0 = 1$ and $z_w = 0$. However, by invoking this counting argument, it is implicitly assumed that every configuration satisfying $\sum z_i = j$ happens with the same probability, as stated (again, we only keep the terms with $z_0 = 1$ and $z_w = 0$ and hide the rest of terms):

Assumption 1. *Let $P(z)$ be the probability of generating a random context and observing $z_i = \mathbf{1}(m_i \in U)$. If $\sum z_i = \sum z_i'$, $P(z) = P(z')$.*

If this is true, we also have $P(z \mid Z = j) = 1/C(w + 1, j)$. We now recover the statement as follows:

$$\begin{aligned} P(H \mid Z = j, z_0 = 1, z_w = 0) &= \sum_{Z=j, z_0=1, z_w=0} P(H \mid z) P(z \mid Z = j, z_0 = 1, z_w = 0) \\ &= \sum_{Z=j, z_0=1, z_w=0} P(H \mid z)/C(w, j-1) \end{aligned}$$

and

$$\begin{aligned} P(H \mid Z = j) &= \sum_{\sum Z = j} P(H \mid z) P(z \mid Z = j) \\ &= \sum_{Z=j, z_0=1, z_w=0} P(H \mid z) P(z \mid Z = j) + \cdots \\ &= \sum_{Z=j, z_0=1, z_w=0} P(H \mid z)/C(w+1, j) + \cdots \\ &= \frac{C(w-1, j-1)}{C(w+1, j)} P(H \mid Z = j, z_0 = 1, z_w = 0) + \cdots \end{aligned}$$

The assumption is true in expectation if the UHS itself is a random subset of Σ^k, which is not the case as that set also has to satisfy the UHS condition. For a general set U, the probability that a k-mer is in U is highly dependent on whether the preceding intersecting k-mers are in U, and the above assumption is likely not valid in most scenarios.

Universal hitting sets may also be constructed in a specific way to enable better analysis of compatible minimizers, like the Miniception we discuss in last chapter. However, as we have seen before, analysis for the Miniception does not apply to other universal hitting sets.

4.2.2 Perfect Sketches: A Case for Sequence-Specific Constructions

A perfect minimizer is a minimizer that achieves density of exactly $1/w$ (probably up to asymptotically negligible terms and rounding error). While the only known examples of perfect minimizers are in the asymptotic case where $w \ll k$ [74], perfect sequence-specific minimizers exist with high probability for short sequences. The following lemma is an example of the difference between optimizing sequence-specific minimizers and non-sequence-specific minimizers.

Lemma 49. *If $|S| < \sqrt{\epsilon w}\sigma^{k/2}$, with at least $1-\epsilon$ probability, a sequence of length $|S|$ where each character is uniformly randomly selected has a perfect minimizer.*

Proof. The optimal minimizer is constructed with fixed interval sampling. More specifically, we take every w k-mer in S and denote the resulting k-mer set U, then construct a minimizer compatible with U. The resulting minimizer is perfect if and only if the k-mers in U only appear in the selected locations. There are $|S|/w$ selected locations and $(1-1/w)|S|$ locations not selected, and for each pair of selected and not selected locations, the k-mers at these two locations are identical with probability σ^{-k} from Lemma 45. By union bound, the probability that any k-mer in U appears outside selected locations is at most $|S|^2\sigma^{-k}/w < \epsilon$, and the sequence has a perfect minimizer with probability at least $1-\epsilon$. □

4.2.3 Core Concepts for Polar Sets

We now define polar sets, the key component for our proposed methods.

Definition 27 (Polar set)**.** *Given sequence S and parameters (w, k, s) with $0 \le s < 1/2$, a* polar set *A of slackness s is a set of k-mers satisfying the following: If two k-mers in S are both in A, they must be at least $(1-s)w$ bases apart in S. (In other words, if a k-mer in S falls in A, there will be no other k-mers in A that are $(1-s)w$ bases close.)*

This can be viewed as a complementary idea to the universal hitting sets or a relaxed form of perfect sets. A universal hitting set requires the set to hit every w consecutive k-mers at least once, while a polar set with $s = 0$ requires the set to hit every w consecutive k-mers at most once. A set of perfect seeds, if it exists, is both a polar set with zero slackness and a universal hitting set. See Figure 4.1 for a more concrete example.

The condition $s < 1/2$ is critical for our analysis. Specifically, this condition is required to obtain a lower bound on the specific density of compatible minimizers, not just an upper bound.

Definition 28 (Link energy)**.** *Given sequence S, parameters (w, k) and a polar set A, if two k-mers on S are $l \le w$ bases apart and are both in A, the link energy of the pair is defined as $2l/(w+1) - 1 \ge 0$. The total link energy of A is the sum of link energy across all eligible pairs.*

Any two k-mers from A in S must be more than $w/2$ bases apart (remember $s < 1/2$), so two k-mers cannot form a link if there is a third k-mer from A between them. With $s = 0$, the link energy is fixed to be $2w/(w+1) - 1 = 1 - 2/(w+1) \approx 1$ for each eligible pair, and the total link energy is approximately the number of pairs that form a link, which in turn is the number of k-mer pairs in the polar set that are exactly w bases away on S.

In the following sections, we introduce and discuss the backbone of the polar set framework, which revolves around closer inspection of how a random minimizer works on a specific sequence, and drawing contrast between sequence-specific minimizers and non-sequence-specific

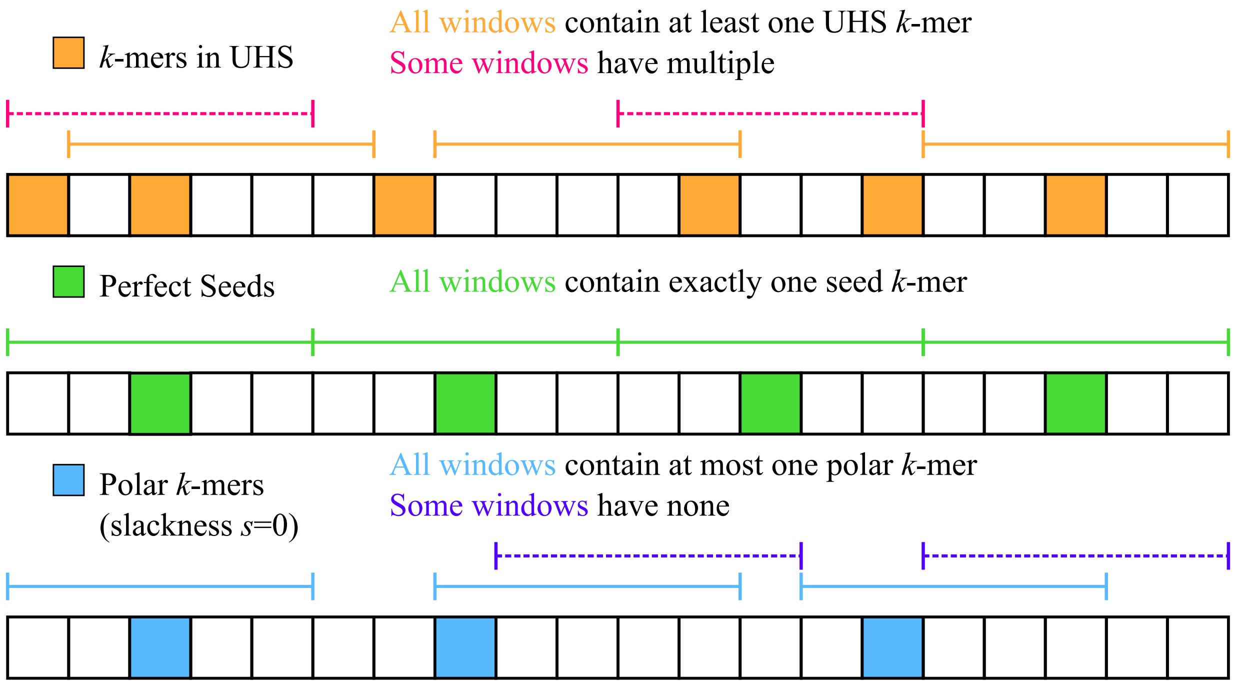
k-mers in UHS
All windows contain at least one UHS k-mer
Some windows have multiple
Perfect Seeds
All windows contain exactly one seed k-mer
Polar k-mers
(slackness s=0)
All windows contain at most one polar k-mer
Some windows have none

Figure 4.1: Comparing universal hitting sets, perfect seeds (compatible minimizers become perfect minimizers) and polar sets. Each block indicates a k-mer, and each segment indicates a window of length 5 ($w = 5$). To provide a better contrast with universal hitting sets, we show polar sets with slackness $s = 0$ (see Definition 27).

minimizers. We use the term "non-sequence-specific minimizers" (sequence-blind as in the introduction in a slightly different context) to refer to constructions of minimizer that do not specifically target a certain sequence, but rather aim to minimize density, the expected specific density on a random string.

Context Energy and Energy Savers

Contexts provide an alternative way to measure the density of a minimizer (See Section 2.7). These play a central role in the analysis of polar sets.

Definition 29 (Charged Contexts)**.** *A* context *of S is a substring of length $(w+k)$, or equivalently $(w+1)$ consecutive k-mers, or equivalently 2 consecutive windows.*

A context is charged *if the minimizer selects a different k-mer in the first window than in the second window.*

See top left of Figure 4.2 for examples of charged contexts. Intuitively, a charged context corresponds to the event that a new k-mer is picked, and counting picked k-mers is equivalent to counting charged contexts.

Lemma 50 (Specific Density by Charged Contexts)**.** *For a given sequence S and a minimizer, the number of selected locations by the minimizer equals the number of charged contexts plus 1.*

Given a context c, define $E(c)$ as the probability that c will be charged with a random minimizer (one with a random ordering of k-mers), which we call the *energy* of c.

Lemma 51. *The expected number of picked k-mers in S under a random minimizer is $1+E_0(S)$, where $E_0(S) = \sum_c E(c)$ is called the initial energy of S and the summation is over every context of S.*

This is proved by combining the linearity of expectation and Lemma 50. This implies that the total energy of a sequence is directly related to the specific density of random minimizers, which is number of picked locations in S divided by number of k-mers in S. $E(c)$ admits a simple formula:

Lemma 52. $E(c) = 2/u(c)$ *if the last k-mer in the context is unique,* $1/u(c)$ *otherwise, where $u(c)$ denotes the number of unique k-mers in c.*

Proof. Consider an imaginary minimizer with $w' = w + 1$ and identical k. The context of a $(w, k)-$minimizer is a window of the imaginary minimizer, and it is charged if and only if the imaginary minimizer picks either the first or the last k-mer. If the imaginary minimizer does not pick either end, the two constituent windows of the context share the same minimal k-mer, and the context is not charged.

With a random minimizer, the probability that the first k-mer is picked in the imaginary window is $1/u(c)$. The probability that the last k-mer is picked is $1/u(c)$ if the last k-mer is unique, 0 otherwise, because the minimizer break ties by preferring the k-mer to the left. The two events are mutually exclusive, so $E(c)$ is the sum of these two terms. □

If all k-mers in a context are unique, $E(c) = 2/(w + 1)$ is guaranteed, which we call the *baseline*. If this holds for all windows, a random minimizer will have specific density of $2/(w + 1)$, similar to applying a random minimizer to a random sequence. As lower $u(c)$ only increases

$E(c)$, $E(c) < 2/(w+1)$ only if the last k-mer in c is not unique and there are over $(w+1)/2$ unique k-mers in the context.

Definition 30. *A context c is called an* energy saver *if $E(c) < 2/(w+1)$, and its energy deficit is defined as $2/(w+1) - E(c)$. The energy deficit of S, denoted $D(S)$, is the total energy deficit across all energy savers: $D(S) = \sum_c \max(0, 2/(w+1) - E(c))$.*

In general, the value of $D(S)$ is very small due to the fact that energy saver contexts satisfying $E(c) < 2/(w+1)$ are rare.

Lemma 53. *For a random context, the probability that it is an energy saver is at most $w\sigma^{-k}$.*

Proof. We bound the probability that the last k-mer in a context is not unique. The probability that the last k-mer equals a specific k-mer in another location is σ^{-k} (Lemma 45). Applying union bound over w other k-mers (as each context has $(w+1)$ k-mers) we get the desired result. □

There are examples of sequences where energy saver contexts are abundant. An extreme scenario is when the sequence S is has a period of w, and has w distinct k-mers. In this case, all contexts become energy saver contexts. These scenarios are rare in practice.

Similarly, we can define energy spenders and energy surplus as follows:

Definition 31. *A context c is called an* energy spender *if $E(c) > 2/(w+1)$, and its surplus is defined as $E(c) - 2/(w+1)$. The energy surplus of S, denoted $X(S)$, is the total energy surplus across all energy spenders: $X(S) = \sum_c \max(0, E(c) - 2/(w+1))$.*

Contexts with energy surpluses are more common than energy savers, but still fairly rare in a random sequence with suitable choice of w and k:

Lemma 54. *For a random context, the probability that it is an energy spender is at most $w(w+1)\sigma^{-k}/2$.*

Proof. A context becomes an energy spender if the last k-mer is unique, and some k-mers appears twice. We bound the probability that some k-mers in the context appear twice. Any two k-mers in a given context are identical to each other with probability σ^{-k} (Lemma 45), and we apply a union bound of size $w(w+1)/2$ (enumerating over pairs of k-mers) to obtain the desired result. □

4.2.4 Main Theorem for Polar Set

With the proper tools, we now state the main theorem of the Polar Sets.

Theorem 11. *Given a sequence S and a polar set A on S, let $E_0(S)$ be the initial energy of S, $D(S)$ be the total energy deficit, $X(S)$ be the total energy surplus, and $L(S,A)$ be the total link energy from the polar set. The number of selected k-mers over S for a random minimizer compatible with A is at most $1 + E_0(S) + D(S) - L(S,A)$, and at least $1 + E_0(S) - X(S) - L(S,A)$.*

Proof. We first prove the upper bound part. We start by elevating the energy of every energy saver context to the baseline $2/(w+1)$. By definition, this increases the total energy of S by $D(S)$, so number of selected k-mers is now upper bounded by $1 + E_0(S) + D(S)$. Formally, $\sum E(x) \le 1 + E_0(S) + D(S)$.

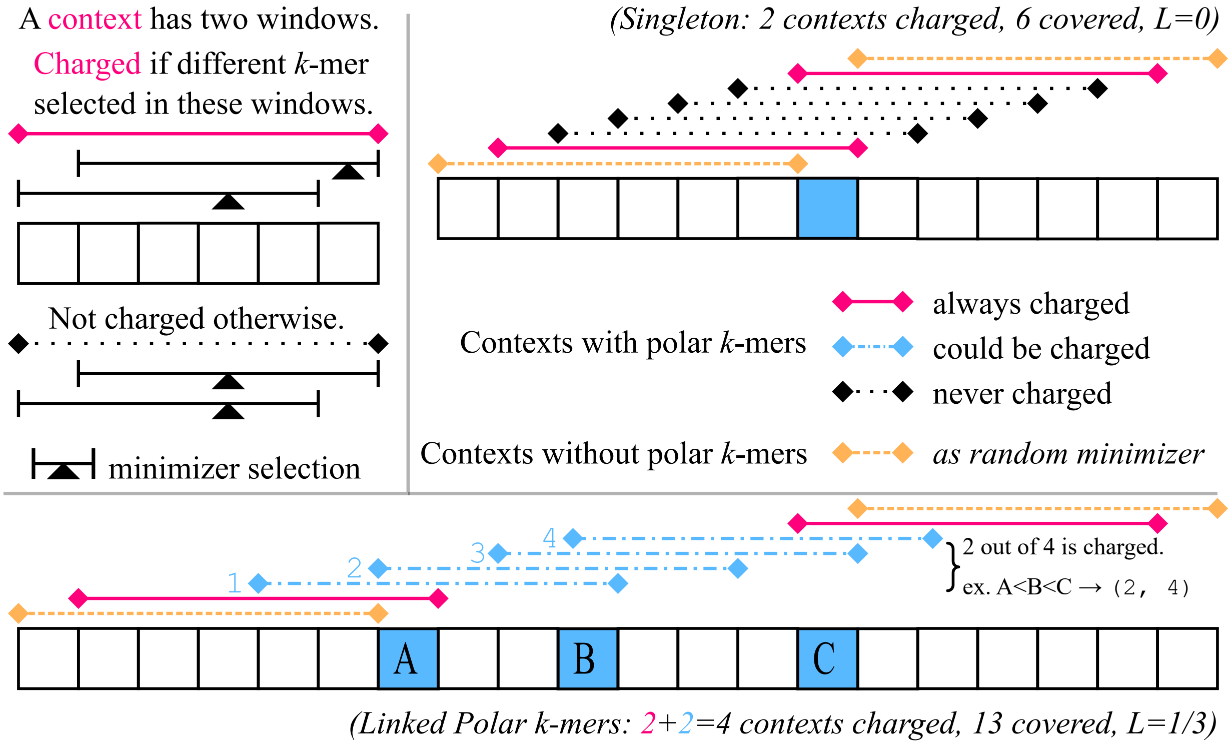
A context has two windows.
Charged if different k-mer selected in these windows.
Not charged otherwise.
minimizer selection
(Singleton: 2 contexts charged, 6 covered, L=0)
Contexts with polar k-mers
always charged
could be charged
never charged
Contexts without polar k-mers
as random minimizer
1
2
3
4
2 out of 4 is charged.
ex. A<B<C → (2, 4)
A
B
C
(Linked Polar k-mers: 2+2=4 contexts charged, 13 covered, L=1/3)

Figure 4.2: Examples for our argument for the polar set density bound with $w = 5$. Left top: Legend for a context, and when it is charged. Right top: Case for a singleton polar k-mer without links. In this case, $L(S, A) = 0$. Bottom: Case for three linked polar k-mers. Whatever the ordering between the three polar k-mers, two of the four contexts marked in blue will be charged. The link energy $L(S, A) = 1/3$: A and B are $l = 3$ bases away with no energy, B and C are $l = 4$ bases away with energy $2l/(w + 1) - 1 = 1/3$.

Consider the minimizer compatible with A, with arbitrary ordering within A and random ordering outside A. We can still calculate the expected number of selected k-mers by summing up the probability of every context being charged, which we denote $E_A(x)$. Our goal for the rest of this proof is to show $\sum(E(x) - E_A(x)) \geq L(S, A)$.

If a context does not contain a k-mer from A, $E(x) = E_A(x)$. Thus, we only need to consider the contexts that contain at least a k-mer from A, which are split into continious segments by the set of contexts not containing k-mers from A.

If a segment only contains one k-mer from A, there are exactly $(w + 1)$ contexts in this segment (see Figure 4.2, upper right for an example). As each context now has energy at least $2/(w + 1)$, the total energy from $(w + 1)$ contexts is at least 2. However, $\sum E_A(x)$ across these contexts is exactly 2, as exactly two contexts will be always charged (the first and last in the segment), and every other context will never be charged. This means such segments can be ignored in the upper bound analysis, as it can only decrease the total energy: $\sum(E(x) - E_A(x)) \geq 0$.

We now focus on the segments with more than one k-mer from A (see Figure 4.2, bottom for an example). Let n be the number of contexts from this segment, we have $\sum E(x) \geq 2n/(w+1)$ because we have assumed $E(x) \geq 2/(w + 1)$ for every context. We next count the number of charged contexts, which is $\sum E_A(x)$. Because every two k-mers in A are more than $w/2$ bases apart, for every k-mer m in A, there is a window such that m is the only polar k-mer, and will be selected. Recall a context is charged if and only if a new k-mer is selected in its latter window. Given that each of the d k-mers in A is selected, this corresponds to d charged contexts in the segment. However, there is an extra charged context: The last context c in the segment, of which the last k-mer in A is the first k-mer of c. In the second window of c, a new k-mer outside A will be selected because being at the end of the segment, there are no more k-mers in A to choose from. Let d be the number of k-mers in A in this segment, we conclude that $\sum E_A(x) = d + 1$ regardless of the ordering, and $\sum(E(x) - E_A(x)) \geq 2n/(w + 1) - d - 1$.

Next, we calculate the total link energy. Recall the link energy is defined as $2l/(w + 1) - 1$ for two polar k-mers $l \leq w$ bases away. The total link energy is thus $2(\sum l)/(w + 1) - (\sum 1)$, summed across all k-mer links. The latter term is simply counting number of links, which resolves to $d - 1$. The earlier term is summing up distance between adjacent k-mers in A. As this is a segment where every two adjacent k-mers have a link, $\sum l$ is the distance from the first k-mer in A to the last k-mer in A. There are n contexts in the segment. The first k-mer in A is the last k-mer in the first context, and the last k-mer in A is the first k-mer in the last context. The first context and the last context are $n - 1$ bases away. The first k-mer and the last k-mer in a context are w bases away. Thus, we have $\sum l = n - 1 - w$.

Finally, we have the following, using S' to denote the sequence that contains the contexts in a segment:

$$\begin{aligned}
L(S', A) &= \sum(2l/(w + 1) - 1) \\
&= 2(n - w - 1)/(w + 1) - (d - 1) \\
&= 2n/(w + 1) - d - 1 \\
&\leq \sum(E(x) - E_A(x)).
\end{aligned}$$

The summation is over each context in the segment. This inequality holds for every segment, and thus we have:

$$\begin{aligned}\sum E_A(x) &= \sum E(x) + \sum (E_A(x) - E(x)) \\ &\leq 1 + E_0(S) + D(S) - L(S, A).\end{aligned}$$

The lower bound uses a symmetric argument as we first upper bound the energy of each context by the baseline $2/(w+1)$. This decreases total energy by $X(S)$ and total expected number of selected k-mers is lower bounded by $1+E_0(S)-X(S)$. The identical argument (with signs flipped) will lead to the final lower bound $\sum E_A(x) \geq 1 + E_0(S) - X(S) - L(S, A)$. □

As $1+E_0(S)$ is the expected number of selected k-mers with a completely random minimizer, we can provably outperform random minimizers if $L(S, A) > D(S)$. For a ballpark estimate, we assume S is a random sequence, and assume the slackness parameter $s = 0$ in construction of the polar set. In this setup, each link has exactly $1 - 2/(w + 1) \approx 1$ energy. As seen in Lemma 53, a context is an energy saver with probability $w\sigma^{-k}$, and its deficit is at most $2/(w + 1) - 1/w \approx 1/w$, meaning $D(S) \approx \sigma^{-k}|S|$. This further means we need the number of links to be at least $\sigma^{-k}|S|$ to provably beat a random minimizer. On the other hand, ignoring the effect of $D(S)$, in order to beat the specific density of a random minimizer by $\epsilon/(w + 1)$, total link energy of $\epsilon|S|/(w + 1)$ is needed. Assuming no slackness, this means the number of links need to be at least $\epsilon|S|/(w - 1)$. Intuitively, ϵ portion of the sequence needs to be covered by links between close enough k-mers in polar set.

A proper polar set requires $s < 1/2$ for the main theorem to hold. When $s \geq 1/2$, only the upper bound part of the theorem holds with an alternative definition of link energy, which we will discuss later in this chapter.

4.2.5 Abstraction and Hardness of Optimization

The link energy formulation of polar sets allows us to cast the problem in graph theoretical framework. Consider an undirected, weighted graph where every unique k-mer is a vertex. An edge connects two k-mers with the following conditions: If these two k-mers ever appear within fewer than $(1 - s)w$ bases of each other in S, the weight is $-\infty$. Otherwise, the weight of this edge is the total link energy by selecting only these two k-mers, which might establish several links given each k-mer may appear in S multiple times. Additionally, the graph contains self-loops. The edge weight of the loop is $-\infty$ if the k-mer appears within $(1 - s)w$ bases of its own copy. Otherwise, the weight of the self-loop is the total link energy by selecting only that k-mer. The problem of finding optimal polar sets becomes the problem of finding an induced subgraph with maximum weight.

The general maximum induced subgraph problem is well known to be NP-hard via reduction from max-clique. We now show optimization of polar sets are also NP-hard just like max-clique. Here, we show a reduction from the problem of maximal independent set to the problem of optimal polar set, with an alphabet of 3. Let $G = (V, E)$ be the instance for maximal independent set, and without loss of generality, let $|V| = 2^d$. We use $\Sigma = \{X, 0, 1\}$ as the alphabet, and for the polar set instance, we let $w = 2d + 1$, $k = d$ and $s = 0$. This means, we want to find subset of $d-$mers that form many links exactly $2d+1$ bases away, but no two $d-$mers in the polar set can be fewer than $2d+1$ bases from each other. With $s=0$, link energy is equivalent to number of links up to a scaling factor, so we are optimizing number of links that can be formed. We now construct the query string for polar set, which we divide into three sections.

Disqualification Gadget. Given an arbitrary $d-$mer $z \in \Sigma^d$, we let the disqualification gadget be the following string:

$$DQ(z) = X^{2d+1}zX^dzX^{2d+1}$$

With presence of $DQ(z)$, z cannot appear in the polar set, because it appears twice exactly $2d$ bases away in the disqualification gadget. The X^{2d+1} section on both ends of the gadget is to prevent d-mers within the gadget to form links with adjacent gadgets or sections, as X^d is not in the polar set.

Disqualification Section. We append a disqualification gadget to the query string for every d-mer (there are at most $3^k = n^{1.5}$ of them), **except** all d-mers containing only 0 and 1.

Vertex Section. For each vertex v in G, let a be its binary representation. We add $X^{2d+1}aX^{d+1}aX^{2d+1}$ to the query string.

Edge Section. For each edge (u,v) in G, let a,b be the binary representation of the two ends. We add $X^{2d+1}aX^dbX^{2d+1}$ to the query string.

The final query string is formed by the concatenation of three sections.

Theorem 12. *The maximal independent set can be solved by solving the optimal polar set of aforementioned query string.*

Proof. We claim any polar set of the query string corresponds to an independent set V' of G, with $|V'|$ links. All d-mers in the polar set are those representing vertexes in G, as other d-mers (those containing X) cannot appear due to the disqualification section. For each d-mer in the polar set, we get one link from the vertex section of the query string. If $(u,v) \in E$, the two d-mers representing u and v cannot be selected into the polar set at the same time, because in the edge section these two d-mers are separated by exactly $2d$ bases, violating the polar set condition. On the other hand, all independent sets of G can be represented by a polar set, with total links $|V'|$ using the same argument.

We conclude that the optimal polar set of the query string is the representation of a maximal independent set of G, which proves the statement. □

This reduction also suggests hardness of approximately solving optimal polar sets, because maximal independent sets are known to be hard to approximate (For example, see Bazgan et al. [5]).

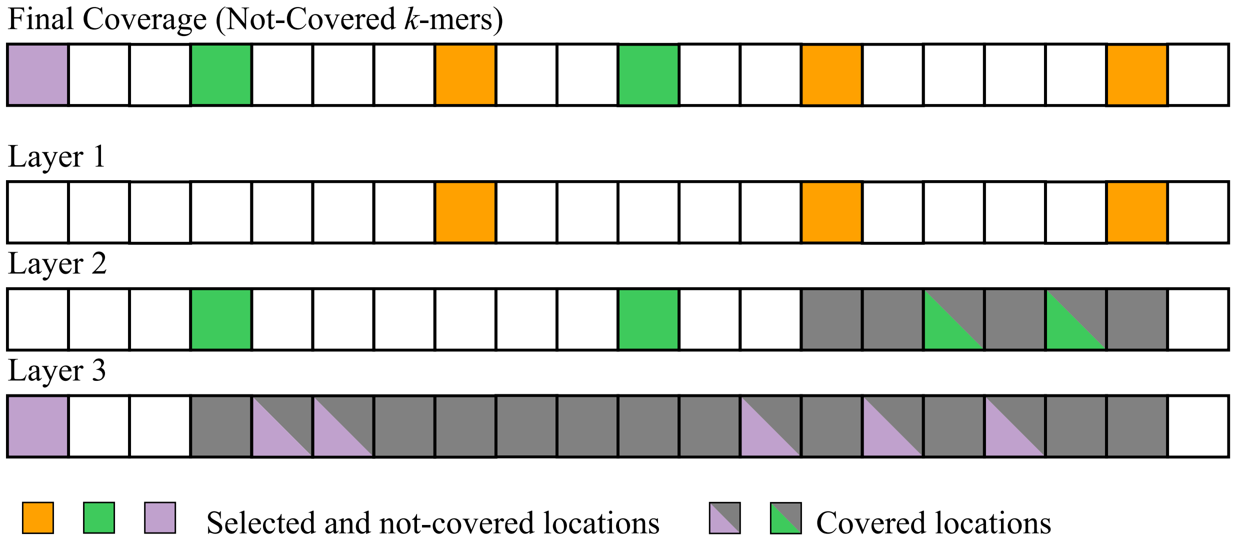
Final Coverage (Not-Covered k-mers)
Layer 1
Layer 2
Layer 3
Selected and not-covered locations
Covered locations

Figure 4.3: Examples of layered polar sets, with three layers. Without layered polar sets, the k-mers from layer 2 and 3 could not be selected as in the polar set because of self-collision. The whole sequence is covered in this case (every window contains a polar k-mer from one layer). Layer 1 is the one with highest priority and our layered heuristics construct it first.

4.3 Optimizing a Polar Set

In this section, we propose a practical extension to polar sets, and formally introduce our heuristics to construct a polar set.

4.3.1 Layered Polar Sets

Assume we have already constructed a polar set A that covers some segments of the reference sequence. Here, covered means that every window contains a k-mer from the polar set, or equivalently, A acts as a universal hitting set on these segments.

Now, to cover the rest of the reference, we shall extend A so more k-mers become polar k-mers. It is natural to consider generating a polar set over the uncovered portion of the reference sequence, then merge this set with A. This however leads to problems. Let A' be a polar set over the uncovered portion of the reference sequence. $A \cup A'$ might not always be a valid polar set, because a k-mer $m' \in A'$ may appear in the already-covered part of the reference sequence, and appear close to another k-mer $m \in A$, thus violating the polar set condition for $A \cup A'$.

On the other hand, the reason we set up the constraint for polar sets is to ensure that k-mers in the polar set will always be selected by any compatible minimizer. In other words, we want to ensure we know exactly the set of k-mers that will be selected. The fact that $A \cup A'$ is not a polar set means that $m' \in A'$ might not always be selected by a compatible minimizer. However, from the perspective of constructing efficient minimizers, we do not need m' to be selected everywhere, as in some places the reference sequence is already covered with k-mers in A (and we know it). By forcing $m < m'$ for any $m \in A$, we ensure that m' will only be selected outside the segments covered by A.

Applying this argument to all k-mers in A', we can essentially ignore the sequence segments already covered by A when constructing A', as long as $m < m'$ for any $m \in A$ and $m' \in A'$ is satisfied. This gives a way to progressively construct the layers of polar sets: at each layer we only need to consider regions of the reference sequence that are not yet covered by previous layers. Formally:

Definition 32. *A layered polar set is a list of sets of k-mers $\{A_i\}$, for $1 \leq i \leq m$. With slackness s, the layered polar set condition is satisfied if for any k-mer in A_j, for each of its appearance at location t in the reference sequence, either of the following holds:*

- *It is at least $(1 - s)w$ bases apart from any k-mer in $\{A_1, A_2, \dots, A_j\}$.*
- *It is covered: There are two k-mers in $\{A_1, A_2, \dots, A_{j-1}\}$ (importantly this does not include A_j), appearing at location l and h, satisfying $l < t < h$ and $h - l \leq w$.*

Similarly, a compatible order for $\{A_i\}$ is an order that places all k-mers from A_1 first in arbitrary order, then those in A_2, ..., then those in A_m and finally those not in any of $\{A_i\}$ in a random order. The link energy $L(\{A_i\}, S)$ is similarly defined over the pairs of close k-mer appearances that are not covered. More formally:

Definition 33. *For a layered polar set, if two k-mers in the layered polar sets, not necessarily from the same layer, appear $l \leq w$ bases apart in S, and neither are covered (as defined in Definition 32), the link energy between them is $2l/(w + 1) - 1 > 0$. $L(\{A_i\}, S)$ is the total link energy across all pairs.*

These definitions of layered polar sets and link energy have two important properties. First, the link energy is non-decreasing as more layers are added to the set. Second, an almost identical argument proves the same bounds for layered polar sets as for polar sets in Theorem 11. See Figure 4.3 for a concrete example of layered polar sets.

4.3.2 Polar Set Heuristic

While more k-mers in the polar set does not always mean higher link energy in theory, larger polar sets usually provide for higher link energy in practice. Thus, we consider a simple heuristic to generate a polar set that somewhat attempt to include a lot of k-mers. The core idea is to select as many k-mers as possible from the set of k-mers that appear exactly w bases away from each other (that is, the set of k-mers that could form a set of perfect seeds). We cannot select all of them as it may violate the polar set condition due to some k-mers appearing multiple times. Because reference sequences are long strings (in the range of billions of bases for mammalian genomes), we consider algorithms that scale well with the length of the reference sequence, preferably close to linear.

Fix an offset $o \in [0, w - 1]$, we start by listing all locations t such that $t = o \mod w$ in the reference sequence S. We then randomly shuffle the locations, and for each location t in this random order, add the k-mer at location t to the polar set. When we add a k-mer m to the polar set, we also locate and remove all k-mers in the polar set that appear fewer than $(1 - s)w$ bases away from m. Additionally, if a k-mer appears at multiple locations in the list, it is only added once, when we first encounter that k-mer in the shuffled list. This is to prioritize k-mers that appear less often; Frequent k-mers are expected to be processed early, because they appear many times in the shuffled list, and k-mers that are added early are more likely to be absent in the final polar set, because they are more likely to be removed due to conflicts with a later-processed k-mer. Jain et al. [46] has explored a similar idea in building tiered random minimizers using a biased hash function.

Our algorithm also has a variant, which we call "monotonic". In the monotonic variant of the heuristics, for every k-mer m we add to the polar set, we calculate the link energy before removing conflicting k-mers and adding m, then calculate it again after doing these operations. (We cover how the link energy is actually calculated in Section 4.3.4.) If the link energy decreases after these operation, we revert them by removing m and adding back k-mers that conflict with m. This variant is slower but results in more efficient polar sets.

We filter k-mers before they are considered for addition to polar sets. If a k-mer appears fewer than $(1 - s)w$ bases away from its own copy (collides with itself) in the reference sequence, it cannot be in the polar set, and the k-mer is discarded from consideration. We also filter out k-mers by their frequency in the reference sequence (see Section 4.3.3 for the threshold value), for practical speedups, as these k-mers will be considered very early in the procedure and is highly unlikely to present in the final set.

Algorithm 3 shows the pseudocode for the heuristic. As seen in the pseudocode, our implementation uses a number of data structures not mentioned before. We describe the data structures in Section 4.3.4, and analyze the time complexity in Section 4.3.5.

Algorithm 3 Pseudocode for Polar Set Heuristics

```
function PolarSet($S, w, k$)
    Start with an empty set $A \leftarrow \{\}$ and a random offset $o$
    Shuffle list of locations $t = o \mod w$ for $0 \leq t < |S|$
    for each $t$ in the list and the $k$-mer $m_t$ at location $t$ do
        Skip if $m_t$ is filtered, or has been processed previously
        Obtain list $l$ of occurrences of $m_t$ via suffix array
        Obtain list of conflicting $k$-mers via linked blocks
        Remove all conflicting $k$-mers and add $m_t$ to the polar set $A$
        For the monotonic variant: revert on the previous line if link energy decreases
    end for
    return $A$
end function
```

4.3.3 Layered Heuristics and Hyperparameters

We construct layered polar sets with a similar algorithm. The properties of layered polar sets guarantee that new layers cannot decrease the final link energy of the polar set.

We rerun the polar set heuristic multiple times, each time with a new random offset o. Each round is run with the current layers of polar sets, and the resulting polar set is added as a new layer. The algorithm for each layer is mostly identical to the single-layer version, with a few changes.

- When processing a k-mer, we skip all of its occurrences that are covered by existing layers of the polar set.
- We skip k-mers at non-covered locations t that are fewer than $(1-s)w$ bases away from a k-mer in a previous layer. These k-mers cannot be in the layer without violating the layered polar set condition.
- At the end of each round, we remove all k-mers selected in the current layer that do not form a link with any k-mers.

We also gradually increase the threshold of k-mer frequency at each round to prioritize low frequency k-mers. In our experiments, we use a total of 7 rounds, with last two rounds being monotonic. The frequency threshold is set at the value to include 85% of locations of the reference in the first round, gradually increasing to 95% in the last round.

The slackness s is also a tunable parameter, which determines when a pair of k-mers is considered in collision. A lower value of s ensures the distance between adjacent polar k-mers are large and have higher link energy for every pair of linked k-mers, but results in smaller number of k-mers selected, implying fewer links. A higher value of s means larger polar sets covering more of the reference sequence and more links formed, but adjacent polar k-mers may be closer to each other resulting in lower energy per link.

In our experiments, we use a fixed slackness $s = 0.4$ after parameter search. This results in approximately 20% less efficient links (average link energy compared to theoretical maximum), but higher total link energy due to inclusion of more links. A more thorough parameter tuning might suggest a gradually increasing value of s between rounds. We provide pseudocode for the layered heuristics at Algorithm 4.

Algorithm 4 Pseudocode for Layered Polar Set Heuristics

```
for each round in the layered heuristics do
    Shuffle list of locations $t = o \mod w$ for a new offset $o$
    Remove locations covered by previous layers of $k$-mers
    Run the For-Loop in Algorithm 3
    Remove all singleton $k$-mers from current layer
    Add all present $k$-mers as a new layer
end for
```

4.3.4 Supporting Data Structures

Our heuristics require some data structures to operate efficiently both in theory and in practice.

Suffix Array

In order to quickly index k-mers and obtain the list of occurrences of a k-mer, we precompute the suffix array [72], the inverse suffix array and the heights table (also known as the LCP array) of the reference sequence. The inverse suffix array ISA satisfies $ISA[SA[i]] = i$ for all i. The heights table $h[i]$ is the longest common prefix for suffixes of S starting at $SA[i]$ and $SA[i+1]$.

All can be computed in linear time [49]. This allows us to find the list of T locations that share the same k-mer as location t, in $O(T)$ time.

Linked Blocks

The layered polar set property ensures that in any stretch of $w/2$ bases, at most one k-mer at one location is selected into any layer of the layered polar sets, excluding covered locations. We use a data structure called linked blocks to represent the set of these selected locations of k-mers. Let $h = \lfloor w/2 \rfloor$, we divide the locations in the reference sequence into h-long blocks, and use an array of length $|S|/h$ to represent these blocks. Each value in the array $C[b]$ is either -1, meaning there are no selected location within this block spanning location $[bh, (b+1)h)$, or a nonnegative integer j, indicating that the k-mer at location $bh + j$ is selected. With linked blocks we can do the following operation quickly:

Definition 34. *`PeekL(x)` returns the closest selected location to the left of x, up to w bases. If there are no selected location to the left of x within w bases, it returns nothing.*

`PeekL` is fast because we only need to query up to three blocks. Adding a location and removing a location also only involves a single block. Similarly we can define `PeekR(x)` returning the closest selected location to the right of x up to w bases. With this data structure, we can implement many critical operations in the aforementioned heuristics. The step of filtering k-mers, more specifically determining whether a k-mer collides with itself, is done using this data structure, in similar fashion to bucket sorting. By maintaining two linked blocks, one for the current layer and one for all previous layers, we can determine whether a location is covered by the previous layers, and list collisions on the current layer.

Calculating Link Energy

In the monotonic variant of our heuristics, we need to calculate the total link energy before and after adding a k-mer. In our implementation, we update the link energy of the polar set as we add and remove locations to the linked blocks, using the following alternative formula for link energy:

$$L(\{A_i\}, S) = 2A_{\text{cov}}/(w+1) - A_{\text{ele}} - A_{\text{seg}}.$$

Here, A_{cov} is the number of contexts that contain a k-mer from the polar set, A_{ele} is the number of non-covered location of selected k-mers, and A_{seg} is the number of continuous segments of windows that contain a k-mer from the polar set. When adding and removing a location to the linked blocks, the changes to these three values are calculated using linked block primitives in constant time, so we can update the link energy in constant overhead. As a sanity check, we see that when adding an isolated k-mer, A_{cov} increases by $(w + 1)$ and the other two values increase by 1, resulting in a net link energy gain of zero, consistent with the original definition. We can also compute the link energy of the polar k-mers in bottom part of Figure 4.3 using this formula, where $A_{\text{cov}} = 13$, $A_{\text{ele}} = 3$ and $A_{\text{seg}} = 1$, resulting in the total link energy of $1/3$.

4.3.5 Time Complexity Analysis

We now analyze the time complexity of the layered polar sets heuristic, assuming no monotonic rounds for now. Let n be the length of the reference sequence, and assume a constant-sized alphabet. We assume a word of constant size can hold an integer in $[0, n]$, and that accessing an element in an array of length n takes constant time. These conditions hold for genomes and 64-bit machines. This means the primitive operations on linked blocks take constant time, and operations involving the suffix array also take constant time.

Consider a worst case scenario: By iterating k-mers that appear exactly w bases away from each other, we iterated over all k-mers in the reference sequence. Assume a k-mer m occurs T times in the reference sequence. In filtering phase, we first fetch the list of T locations in $O(T)$ time using the suffix array, and we want to determine if there are two elements whose difference is less than $(1 - s)w$. This can be done using the linked blocks in $O(T)$ time. In the case of layered polar sets, we also want to determine if each of the locations is covered by previous layers, and if it is fewer than $(1 - s)w$ bases away from a location in a previous layer. As we use one linked block for all previous layers, this can be done in $O(T)$ time. The filtering phase thus finishes in $O(T)$ time.

The main algorithm is split into three parts: detecting k-mers that are close to m in the reference sequence, removing those k-mers from the polar set, and adding m to the polar set. Detecting and listing k-mers that are close to m takes $O(T)$ time, as each location reports only four collisions at most, two to the left and two to the right. Removing a k-mer that occurs T' times takes $O(T')$ time, but since each k-mer is only added and removed once in one round, this amortizes to $O(T)$ time. Adding m to the polar set also takes $O(T)$ time. The singleton detection step (removing k-mers forming no links) also takes $O(T)$ time for checking if m is a singleton.

As each k-mer is only visited once in the main algorithm, and in the worst case scenario every k-mer in S is visited, we conclude that the layered polar set heuristics runs in $\sum O(T) = O(n)$ time for each layer, and as a special case the (non-layered) polar set heuristics runs in $O(n)$.

In practice, the running time changes widely between different configurations on the same sequence, due to many factors. The worst case scenario (every k-mer is visited when we visit the set of k-mers at location $t = o \mod w$) is not satisfied in many configurations, especially those with large w. In extreme cases when a perfect minimizer exists, a round runs in $O(n/w)$ time instead. Many practical considerations also affect the runtime of the heuristics, such as the fact that we resort to a simple sort-and-scan algorithm for detecting self-collisions in the filtering step due to the time and memory overhead for using an additional linked block. The monotonic variant of the heuristic can in theory run in $O(n^2)$ time per round. The extra complexity comes from the fact that each k-mer can be added and removed up to $O(n)$ times in a round, whenever another k-mer is being considered (in the original version, a k-mer is only added and removed once). However, in our experiments, the monotonic variant is not significantly slower.

4.4 Experimental Results

All the experiments are run using the human reference genome hg38. To facilitate the performance comparison across a range of parameter values of w and k, we report the *density factor*

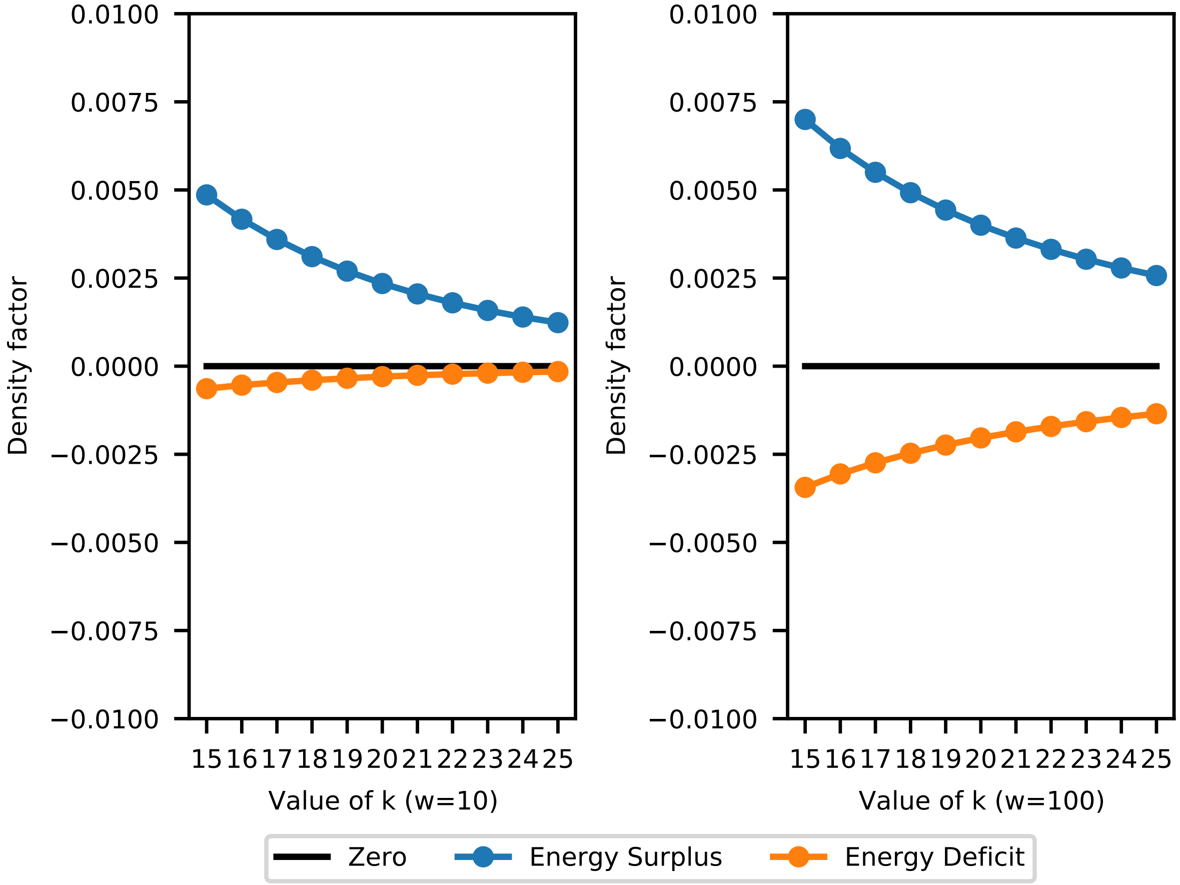

Figure 4.4: Energy surplus and deficit for short ($w = 10$) and for long ($w = 100$) windows, computed on the human reference sequence hg38. The difference between the two lines is the difference between the upper and lower bound of Theorem 11. It is very small and the bounds are very good estimates in practice.

(See Section 2.7 and Marçais et al. [73]) instead of the density. The density factor is the density multiplied by $(w + 1)$. Regardless of the value of w, the random minimizer has an expected density factor of 2 and a perfect minimizer has a density factor of ≈ 1 ($1 + 1/w$ to be exact, which is close to 1 for large w.)

4.4.1 Energy Deficit and Energy Surplus

First, we calculate the average energy deficit $X(S)/|S|$ and average energy surplus $D(S)/|S|$. The results are in Figure 4.4.

The reference genome is more repetitive than a purely random sequence. However, empirically the energy surplus and deficit are still small, well below 0.01 measured in density factor, implying a relative error of at most 1% when estimating specific density with link energy. Thus, when constructing efficient minimizers by (layered) polar sets, using link energy to estimate specific density is efficient and accurate. For reference, on a random sequence the average energy

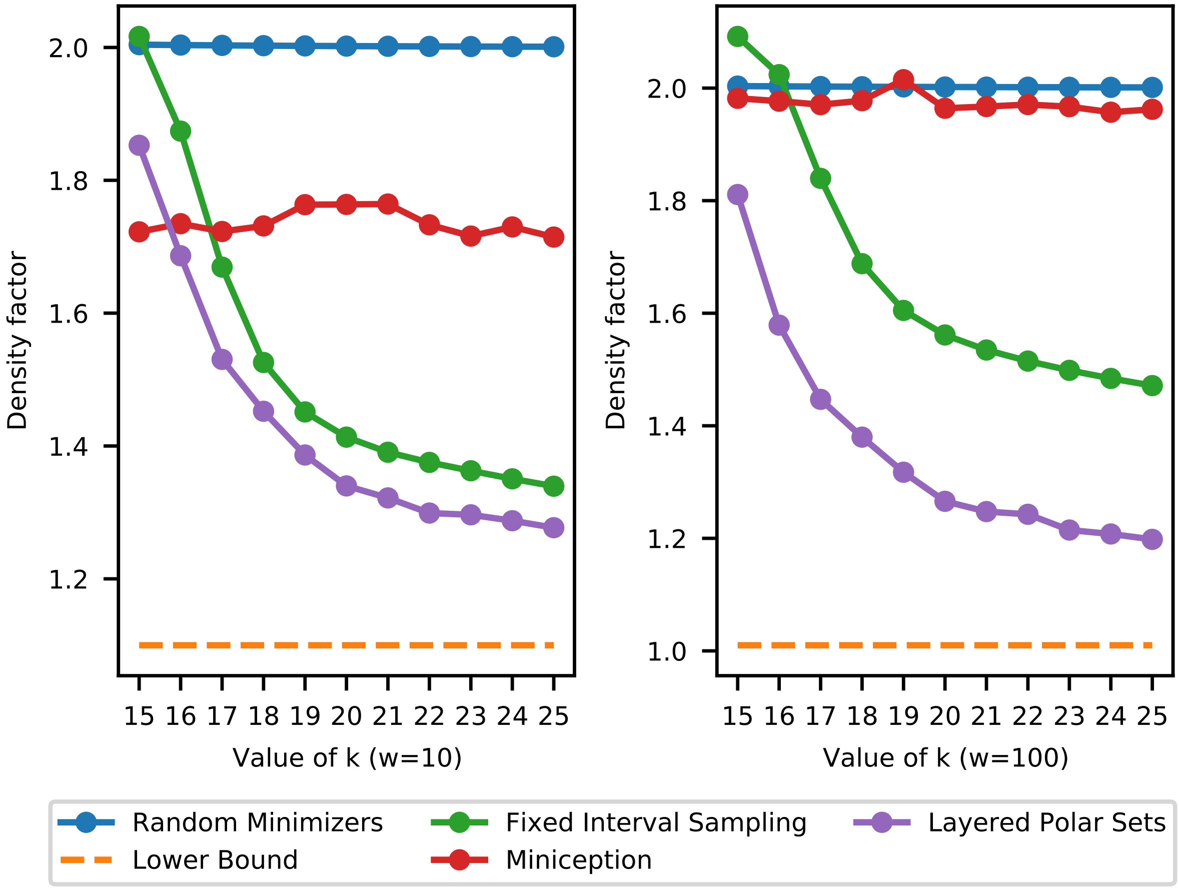

Figure 4.5: Density factor for the proposed methods, for short (left) and long (right) windows, computed on hg38. The bottom orange dashed line is the theoretical minimum density (perfect minimizers).

surplus and deficit are below 10^{-7} in absolute value, for the parameter range in which we are interested.

4.4.2 Polar Set Main Evaluation: Density Comparison

We next evaluate our proposed algorithms for layered polar sets. We implemented the algorithm with Python3. Experiments are run in parallel and the longest ones finish within a day on a Intel(R) Xeon(R) Platinum 8275CL CPU. The peak memory usage stands at 100 GB, which happens at the start when loading the precomputed suffix array using Python `pickle`.

We compare our results against some other candidates:

- Random Minimizers. Achieves density factor of 2 in theory (Section 2.7) and in practice (Section 3.4.2 for example).
- Lower Bound. This corresponds to the density factor for perfect minimizers. While our

theory predicts existence of perfect minimizers matching the lower bound with large value of k, this rarely happens with practical parameter values.

- Fixed Interval Sampling. Fix a random offset $o \in [0, w)$, this method constructs a priority set by selecting all k-mers at location $iw + o$ for integer i. The final minimizer is a random compatible minimizer with the priority set.
- The Miniception (see previous chapter), a practical algorithm that provably achieves lower density than random minimizers in many scenarios. Its hyperparameter k_0 is set to $\max(5, k-w)$ for our experiments (See Section 3.4.2 for brief discussion on hyperparameters).

We do not include existing algorithms for constructing compact universal hitting sets because these methods do not scale to values of $k > 14$. Our heuristics work the best when k-mers do not appear too frequently, or roughly speaking, when $\sigma^k > |S|$ where S is the reference sequence. This choice of parameter is common in bioinformatics analysis. With the sequence at the size of human reference genome, our heuristics work well starting at $k = 15$. Additionally, the Miniception achieves comparable performance with leading UHS-based heuristics, so its performance also serves as a reasonable proxy.

We consider two scenarios, first with short windows ($w = 10$) and second with long windows ($w = 100$). The results are shown in Figure 4.5. Our experiments indicate that our simple heuristics yield efficient minimizers, greatly outperforming random minimizers and the Miniception, while maintaining a consistent edge over fixed interval sampling methods, in both short windows and long windows settings. The improvement is more pronounced when the windows are long, which arguably is more common in practice.

4.4.3 Detailed Profiling of Polar Set Performance

We perform a number of controlled experiments to better analyze how polar set based minimizer performs.

Density Factor of Layered Polar Sets by Round

To show that our proposed layered anchor set heuristics is useful, in Figure 4.6 we plot the density factor after each round of optimization on the human reference genome hg38. All algorithms are run for a total of 7 rounds, with last two being monotonic rounds. We select 7 to ensure the resulting sets are not too complicated and can be computed in reasonable amount of time. With more rounds, many of the results can be further improved.

Experiments on Human Chromosome 1

To show the effect of reference sequence length on the performance of sequence-specific minimizers, in Figure 4.7 we show the performance plot when we build sequence-specific minimizers for chr1 only. The human chromosome 1 sequence is around 10% of the whole hg38 sequence, and consistent with our theory, the time and memory spent to run these experiments on chr1 are around 10% of that for hg38 ones.

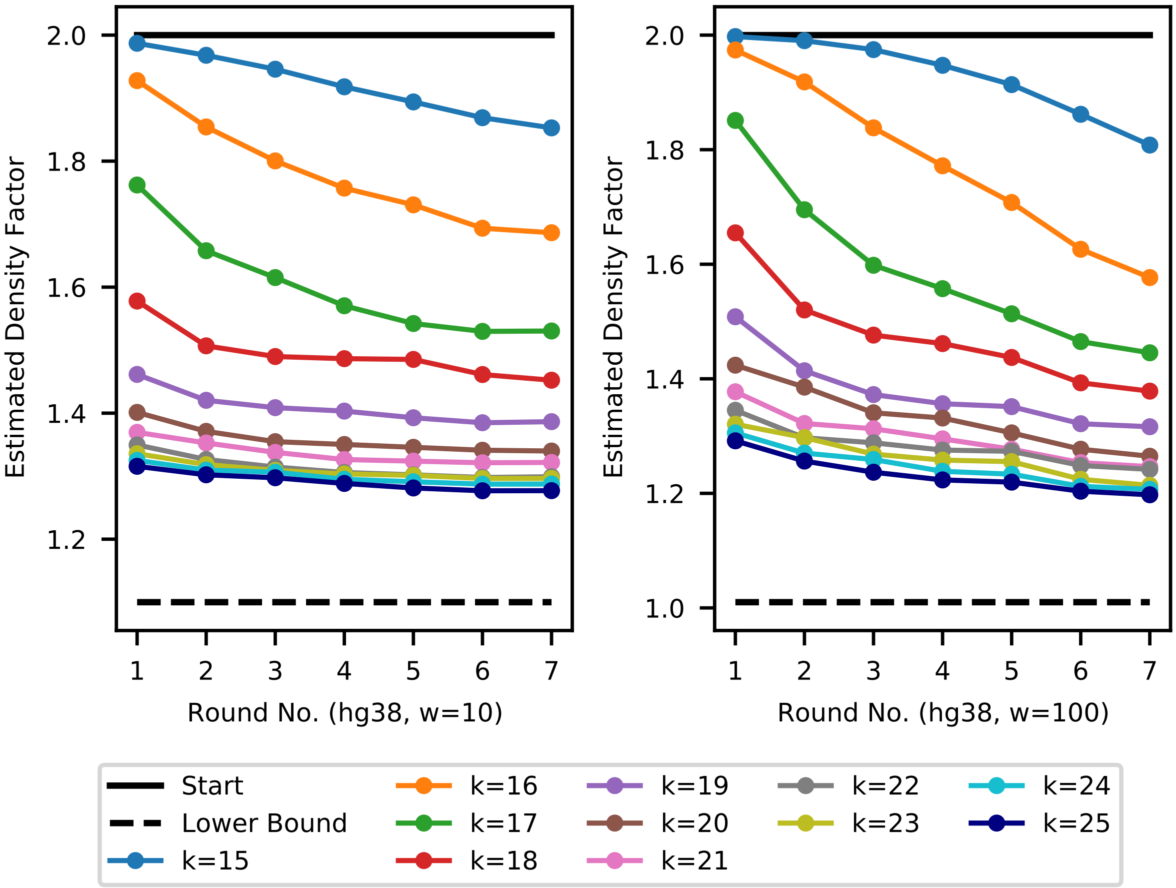

Figure 4.6: Density factor of layered anchor sets after each round of the optimization, corresponding to the experiments shown in Figure 4.5.

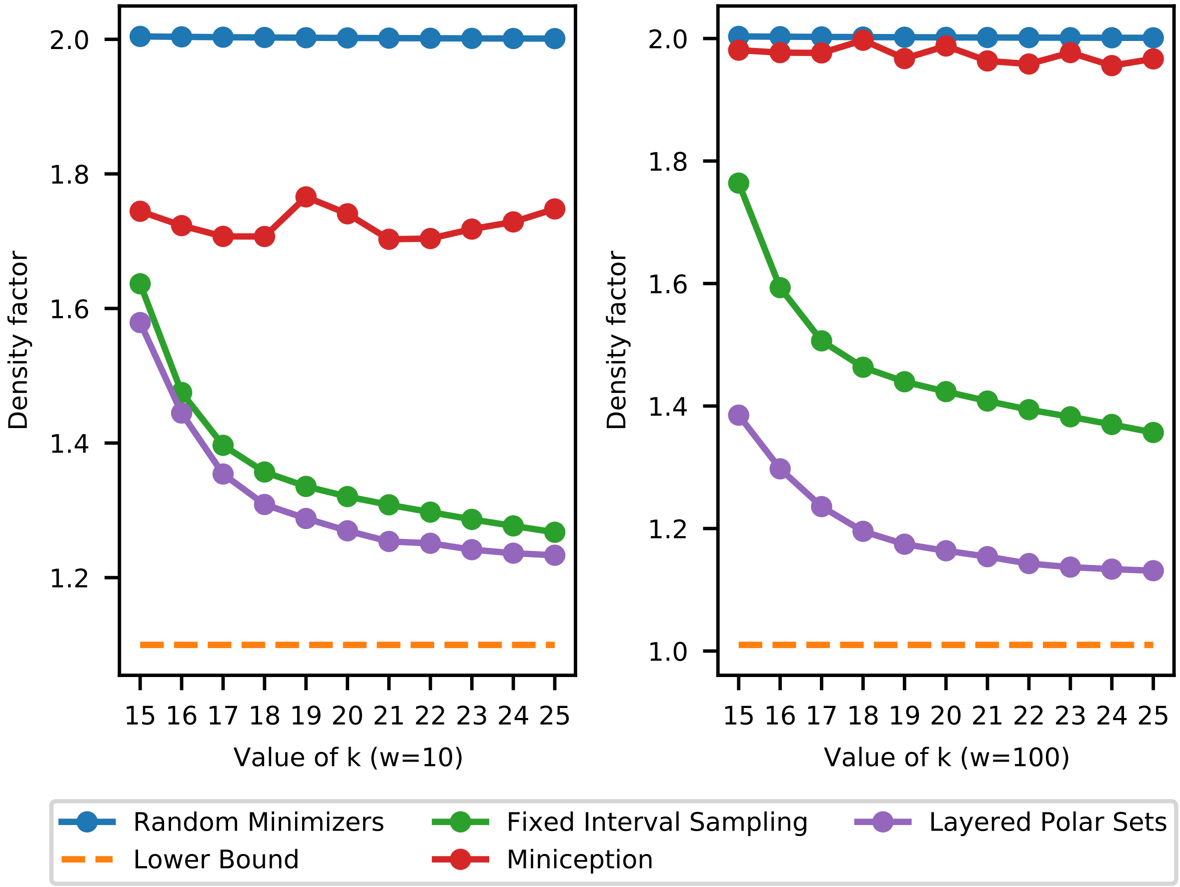

Figure 4.7: Performance of sequence-specific minimizers, optimized and tested on human chromosome 1 instead of whole human genome. The parameters are $w = 10$ (left) and $w = 100$ (right).

Viability of Sequence-Specific Minimizers on Non-target Sequences

To validate that optimization of sequence-specific density does not come at the cost of higher (non-sequence-specific) density, we generate the sequence-specific minimizers for hg38 reference genome, then apply these minimizers on a random sequence. Figure 4.8 shows the results. We expect these to perform close to random minimizers when $\sigma^k \gg N$ where N is the length of the reference sequence. In these cases, most k-mers in a random sequence is not seen in the reference sequence, and optimized sequence-specific minimizers behave just like random minimizers in most cases. More formally:

Lemma 55. *If a polar set contains N k-mers satisfying $N \leq \epsilon\sigma^k/(w+1)$, minimizers derived from the polar set satisfies that its density differs from that of a random minimizer by at most ϵ.*

Proof. The density of a minimizer equals the probability a random context is charged (see Section 2.7 for intuition and proof). If a context does not contain any k-mer from the polar set, the minimizers derived from the polar set behaves exactly the same as a random minimizer. The probability that a random k-mer is in the polar set is N/σ^k. Using a union bound over all k-mers in a context, the probability that a random context contains a k-mer from the polar set is at most $(w+1)N/\sigma^k \leq \epsilon$. In other words, the polar-set-derived minimizer behaves differently from a random minimizer with probability ϵ, so their density differs at most by the same amount. □

The performance for the Miniception is almost identical to that in hg38, and is not shown in Figure 4.8. The layered polar sets is also arguably more robust at lower values of k, as its density stays close to that of a random minimizer.

Building Sequence-Specific Minimizers on Random Sequences

To further show that human reference genome is highly repetitive and construction of efficient sequence-specific minimizers is hard in such setup, we run the algorithms to generate sequence-specific minimizers on a random sequence of length $230\,000\,000$, similar to that of chromosome 1. Figure 4.9 shows the performance of layered polar sets and fixed interval sampling method. Compared with Figure 4.7, we observe it is much easier to build efficient minimizers on a random sequence, and to match the theoretical lower bound, even given the reference sequences has similar length.

Experiments on Human Centromere

The human reference genome contains highly repetitive regions, the most notorious one being the centromere. Even assembling the sequence of centromere requires significant collaborative effort [80]. In this section, we test performance of different minimizers on the complete chrX sequence, including the newly assembled centromere. These sequence-specific minimizers are optimized and tested on the chrX sequence. Instead of showing density across different configurations of w and k, we fix $w = 100$ and $k = 15$ as a representative configuration, and plot the density of the minimizers along the genomic coordinate. While the density for both layered polar sets and fixed interval sampling methods fluctuates over different regions in the chromosome,

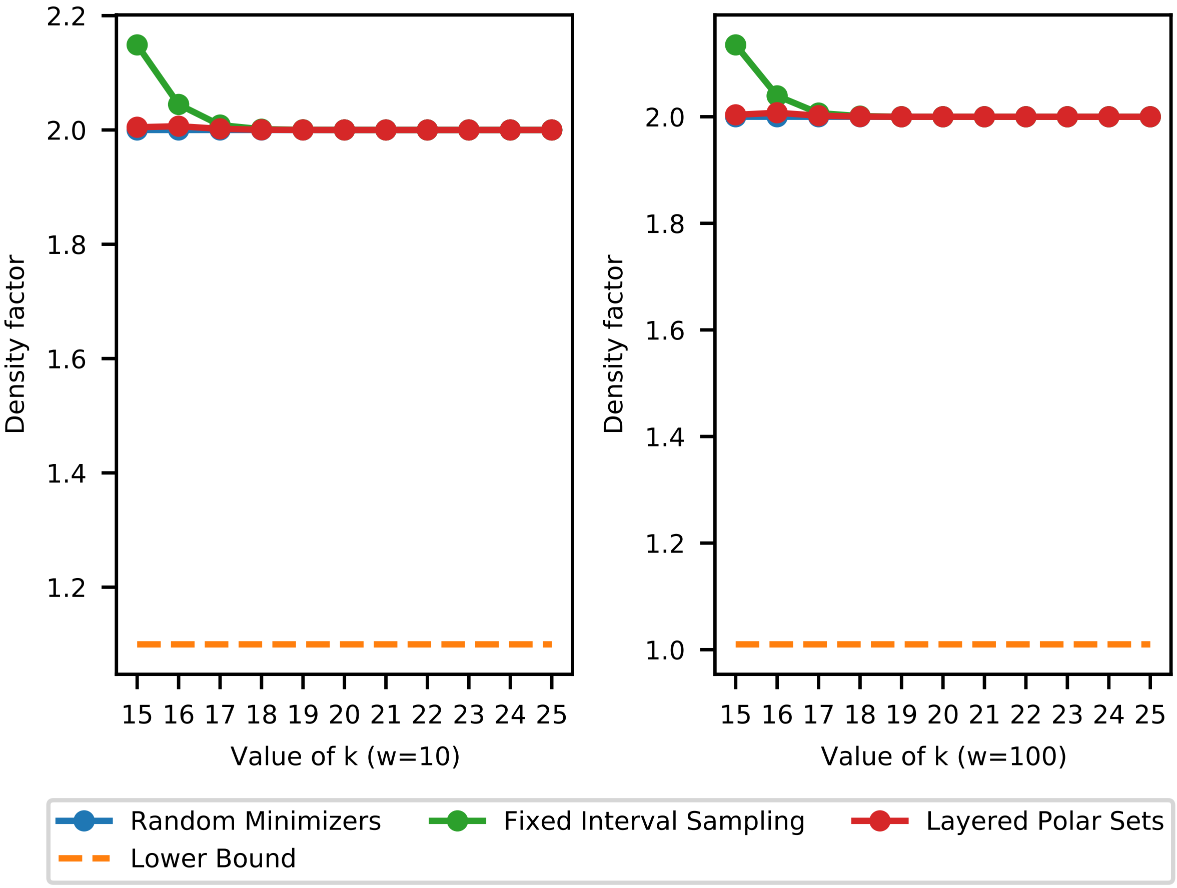

Figure 4.8: Performance of sequence-specific minimizers on random sequences (optimized on hg38). The parameters are $w = 10$ (left) and $w = 100$ (right). This is different from Figure 4.9, here the specific density is measured on a unrelated random sequence.

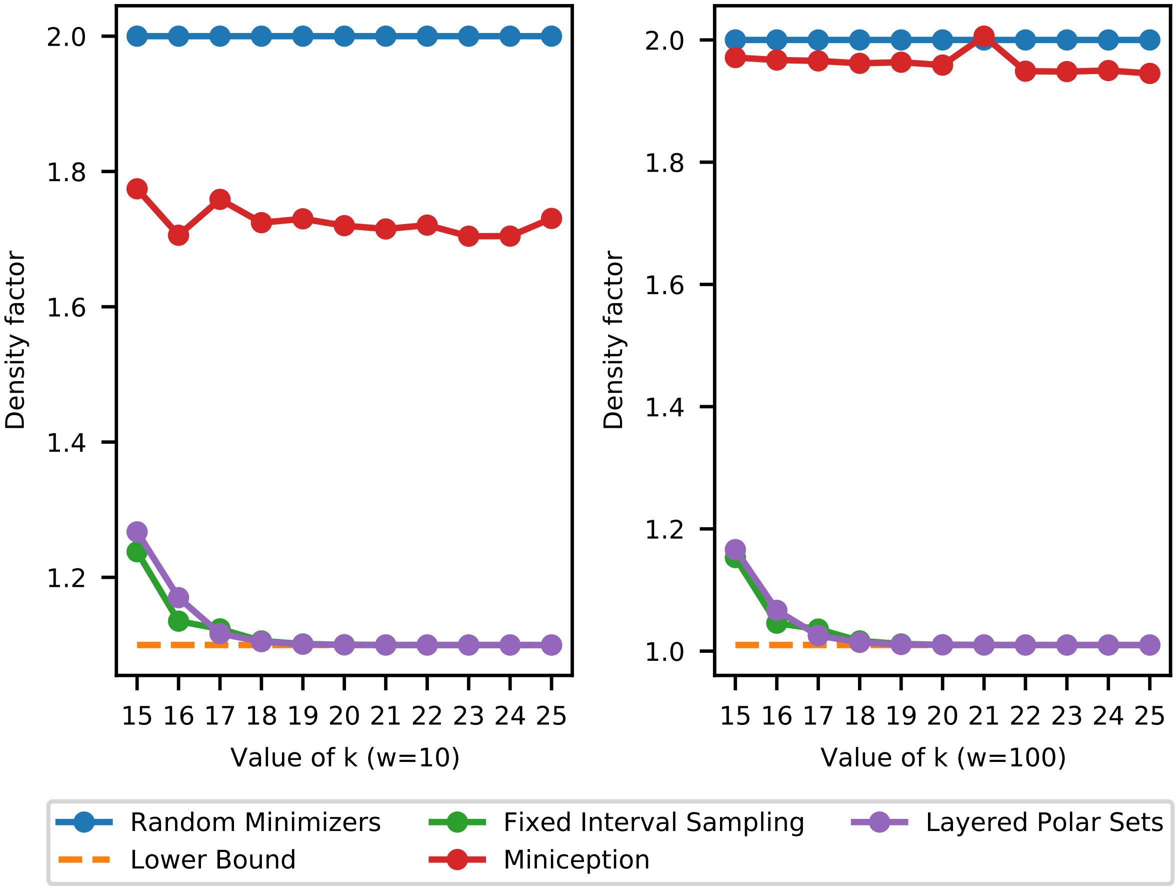

Figure 4.9: Performance of sequence-specific minimizers, optimized and tested on a $230\,000\,000$ −long random sequence. The parameters are $w = 10$ (left) and $w = 100$ (right). This is different from Figure 4.8, here the specific density is measured on the same sequence on which the minimizers were optimized.

the most significant spike coincides with the centromere region (shaded in Figure 4.10). Other configurations of w and k also lead to a similar conclusion.

Intuitively, because k-mers are so repetitive within the centromere, the polar set heuristics refuse to select k-mers within the region, and the resulting minimizers operate like a random minimizer over the centromere. We also note that the sketching method used by Winnowmap [46] in fact selects more k-mers from the centromere region compared to a random minimizer (called "standard" in the Winnowmap paper), implying higher density. However, the resulting read mapper performs well over the centromere measured by mapping error rate.

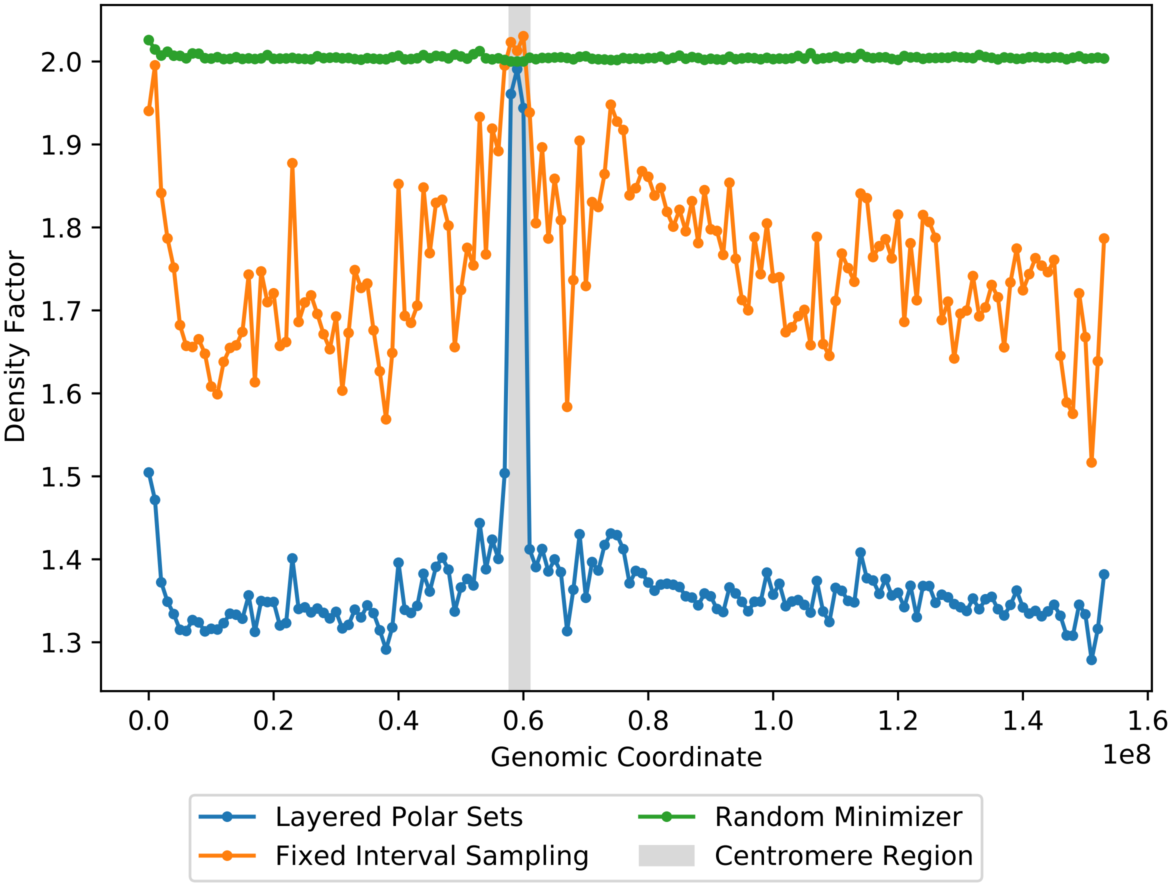

Figure 4.10: Performance of sequence-specific minimizers on the chrX centromere region. The parameters are $w = 100, k = 15$, optimized and tested on the completed chrX sequence including the centromere region. The X axis is the genomic coordinate, and the shaded region indicate the centromere of chrX.

4.5 Generalization of Polar Sets

4.5.1 Non-Sequence-Specific Polar Sets

For the sake of simplicity, in this section we only discuss polar sets with $s = 0$. The discussion about $s > 0$ is highly similar. As we have discussed, the density of a minimizer is the expected specific density over a random sequence. Equivalently, it equals the specific density on the de Bruijn sequence of order at least $w + k$. Therefore, one may construct polar sets on the de Bruijn sequence of sufficient order, to build non-sequence-specific minimizers. However, this is impossible with long windows:

Lemma 56. *No non-trivial polar set exists when $w > k$ and S is the de Bruijn sequence of order $w + k$.*

Proof. We simply show no k-mers can be in the set. For every k-mer m, the sequence mm exists within S, because S is the de Bruijn sequence of order at least $2k$. Picking m violates the condition for polar set because it appears twice with $k < w$ bases apart in S. □

Polar sets exist on de Bruijn sequences of order $w + k$, when $w \leq k$. With $w = k$, these polar sets become *non-overlapping k-mers* [59], that is, the set of k-mers where no proper prefix of a k-mer equals a proper suffix of another k-mer. The problem of finding large set of non-overlapping k-mers is hard in general, although constructive algorithms exist [9] for constant factor approximation. With $w < k$ we obtain *minimally-overlapping k-mers*, a concept that has also been studied in other contexts [35]. We believe the concept of non-sequence-specific polar sets is of both practical and theoretical interest.

4.5.2 Improper Polar Sets and UHSes

The alternative formula for link energy, as described in Section 4.3.4, allows us to define the link energy of any subset of k-mers, not just those satisfying the polar set condition. The main theorem for polar set still holds, but only the upper bound part. Interestingly, if we plug in a universal hitting set, we get $A_{\text{cov}} = n, A_{\text{ele}} = |U|, A_{\text{seg}} = 1$ and the link energy of $2n/(w + 1) - |U| - 1$, where n is the number of k-mers in the reference sequence and $|U|$ is the total number of times a k-mer in UHS appear in the reference sequence. Plugging this into the main polar set theorem, we recover the specific density upper bound $|U|/n$ for universal hitting sets, up to an error of $D(S)/n$. In this sense, universal hitting sets can be seen as a specific and extreme case of an improper polar set.

In general, any set of k-mers can be improper polar sets, which serves as a bridge between the two extreme definitions: universal hitting sets where the whole sequence needs to be covered but coverage can be arbitrarily dense, and polar sets where the coverage by k-mers needs to be sparse everywhere but no sequence is required to be covered at all. Optimizing a "priority set" of k-mers to construct compatible minimizers in general, using the link energy of an improper polar set as the guide (similar to what we have done in monotonic variant of the polar set heuristics) may be an interesting approach worthy of more exploration.

Chapter 5

Conclusion

5.1 Contributions

In this dissertation, we explore the theory and practice of low-density minimizer sketches, and made a number of important contributions to the field.

On the theory side, our contributions are:

- We settled the question of asymptotically optimal minimizers, and showed that lexicographical minimizers are asymptotically optimal;
- We calculated the densities of lexicographical and random minimizer sketches under common conditions;
- Casting Mykkeltveit sets as special cases of universal hitting sets, we upper and lower bounded their path lengths using novel geometric techniques;
- We provided the first set of structural theorems for local and forward schemes, which are previously proposed generalizations of minimizers.

These results settle various previously open questions regarding low-density minimizers, and open many venue for future endeavors.

On the practice side, our contributions are:

- For sequence-blind optimization of low-density minimizer sketches, we propose the Miniception, which is
    - the first minimizer sketch that achieves provably lower-than-random density for practical values of w and k;
    - very efficient and requires minimal overhead to implement;
    - empirically validated to achieve low density as predicted by theory.
- For sequence-specific optimization of low-density minimizer sketches, we propose the Polar Sets and their associated heuristics, which are
    - a complementary framework to existing constructions with Universal Hitting Sets;
    - accompanied by a clear optimization objective that is provably linked to specific density (which is our optimization objective), with bounded errors on both sides;

- a framework that can be optimized using simple yet efficient heuristics, with experimental validation
- with performance that has been thoroughly profiled in different scenarios for better understanding of inner workings.

Both methods are still current state-of-the-art for density optimization at least in some configurations of w and k. Further, the ideas presented in the construction and analysis provide routes to potentially push for future improvements. As minimizer skecthes are essential to many pipelines in computational genomics, these improvements will provide great benefits to the research area.

5.2 Open Theoretical Questions

While this dissertation has advanced the theory behind universal hitting sets and minimizer sketches, there are many more open questions on the theoretical front.

5.2.1 Settling Path Length for Minimal Decycling Sets

There is more than one decycling set of minimum size (MDS) for given w. The Mykkeltveit [82] set is one possible construction, and a construction based on very different ideas is given in Champarnaud *et al.* [17]. There are many more possible MDS than these two. Empirically, for small values of w, we can exhaustively search all the MDS on the binary alphabet: for $2 \leq w \leq 7$ the number of MDS is respectively 2, 4, 30, 28, 68 288 and 18 432.

While experiments suggest the longest remaining path in a Mykkeltveit depathing set defined in Mykkeltveit [82] and Section 2.4 (slightly different from the original Mykkeltveit set) 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.

5.2.2 Tighter Connection between UHS and Selection Schemes

In Section 2.2, we showed that universal hitting sets and forward selection schemes are somewhat convertible to each other:

- Given a (w, k)-UHS with relative size d, the compatible (w, k)-minimizer (which is also a forward scheme) has density upper bounded by d.
- Given a (w, k)-forward scheme with density d, its charged contexts give a $(w, w+k)$-UHS with relative size exactly d.

This conversion, however, is not perfect, because in converting from schemes to UHSes we need to pay the price of inflating k. If forward schemes are replaced by local schemes, the charged contexts give a $(w, 2w + k)$-UHS instead, an even larger price. In general, given a (w, k)-minimizer with density d (let's not consider forward schemes for now), there are no known algorithms that can generate a (w, k)-UHS with relative size *upper* bounded by d. It should be noted that using Theorem 9, we would indeed end up with a (w, k)-UHS, but its relative size would be *lower*-bounded by d.

We believe the problem in general is impossible, so one open question is the price of approximation: Is there a constant factor $c > 1$ such that given a (w, k)-minimizer with density d, we can always construct a (w, k)-UHS with relative size cd? As a start, we would have $c \leq 2$ trivially, because in Section 2.7.3 we showed that there exists universal constructions that are $(2 + o(1))$-approximations to optimal density UHSes.

5.2.3 Low-Density Forward and Local Schemes

The $1/w$ bound is not the only density lower bound for minimizers. Specifically, Marçais et al. [74] proved the following lower bound:

$$\frac{1.5 + \frac{1}{2w} + \max(0, \left\lfloor \frac{k-w}{w} \right\rfloor)}{w + k}. \tag{5.1}$$

As w grows compared to k, this implies that the density factor of the minimizers are lower bounded by a constant up to 1.5. We also observed that performance of minimizers, both the Miniception and the PASHA compatible ones, regress to that of a random minimizer when w increases. Unfortunately, this is inherent to minimizers. With a fixed k, as the window size w grows, the k-mers become increasingly decoupled from each other and the ordering $\mathcal{O}$ plays less of a role in determining the density.

As discussed before, the minimizers are not the only class of methods to sample k-mers from strings, and local schemes are generalizations of minimizers that are defined by a function $f : \Sigma^{w+k-1} \to \{0, 1, \cdots, w - 1\}$ with no additional constraints. Local schemes are not limited by Equation 5.1, and because they are generalizations of minimizers, they will be at least as good as minimizers for densities. Unpublished experiments via brute force search on small w and k show that the minimal density for forward and local schemes are lower than that of minimizers. However, we do not have a non-trivial construction of low-density forward or local schemes (non-trivial in the sense that it is not also a minimizer). Construction of such objects and finding use cases for them are interesting directions for future work.

5.2.4 Perfect UHSes and Perfect Selection Schemes

Given w, we can construct the following three universal hitting sets over w-mers:

- The Mykkeltveit set is a UHS with relative size $(1 + o(1))/w$ and path length $\geq w^2/100$ (Section 2.6).
- We can use the charged contexts of the Miniception to obtain a UHS with relative size $(1.68 + o(1))/w$ and path length $2w$ (Section 3.3).
- We can use a random minimizer with suitable w and k, to obtain a UHS with relative size $(2 + o(1))/w$ and path length w (Section 2.7.3).

This presents a clear tradeoff between relative size and path length, which intuitively makes sense: As we have more w-mers, the path length we must leave uncovered decreases. It is natural to question if these relatively simple constructions are the best we can do. More specifically, is it

Method	**Path Length**	**Relative Size**
Mykkeltveit Set	$\geq w^2/100$	$(1 + o(1))/w$
Charged Contexts of Miniception	$2w$	$(1.68 + o(1))/w$
Random Minimizer Construction	w	$(2 + o(1))/w$
Optimal UHS	w	$(1 + o(1))/w$

Table 5.1: Comparing Our UHS Construction and the Hypothetical Optimal Construction.

possible to obtain a perfect UHS, that is, (w, w)-UHS with relative size $(1+o(1))/w$? Table 5.2.4 provides a succinct overview of this discussion.

Also, as we have discussed in Section 5.2.2, it is in fact strictly no harder to construct a (w, w)-minimizer with density $(1 + o(1))/w$ than to construct a perfect UHS, because existence of a perfect UHS would imply this construction. We can even further relax and only require a local scheme, however, even this is an open question as discussed in Section 5.2.3 because there are no non-trivial constructions of local schemes yet.

5.3 Open Practical Questions

While we believe both the Miniception and the Polar Sets are promising methods that can be used for pipelines in computational genomics, we acknowledge there are a lot of practical issues to solve. Here we list a number of interesting future avenues for research into the practical use of minimizers.

5.3.1 Improving Polar Sets

Limitations of Polar Sets

While the concept of polar sets is interesting and leads to improvements in state-of-the-art sequence-specific minimizer design, we should acknowledge its limitations. First, it cannot be used in designing sequence-blind minimizers when $w > k$ as we discussed in Section 4.5.1. (However, one can also argue this means the method is more tailored for sequence-specific minimizers.)

Relatedly, the performance of polar sets deteriorates quickly for small k, for longer sequences, and for repetitive sequences. This can be seen from our experiments: Performance improves quickly for larger k and for shorter sequences (Section 4.4.2 and Section 4.4.3). Further improvements are observed when we replace chromosome 1 with a pure random sequence of similar length, where the theoretical minimum is easily reached (Section 4.4.3). This unfortunately means the polar sets algorithm is not yet ready for a certain set of parameters, like those with $k \leq 10$.

Additionally, the current optimization algorithm for polar sets cannot be parallelized, and thus it runs very slowly for longer sequences, especially when we run the monotonic variant. We believe improvements to the polar sets framework and algorithm are possible to resolve these issues, and our proposed improper polar sets (Section 4.5) can be a good start.

Optimal Polar Sets

With better computing power and more efficient algorithms, it is desirable to compute an optimal polar set. Thanks to our link energy formulation, the problem of optimal polar set can be formed with integer linear programming (ILP), each k-mer being a binary variable. For moderately-sized reference sequences, an optimal polar set can be found. However, no such convenient formulation exists for layered polar sets, and it is an interesting question whether there is a tractable optimization problem for minimizers in general.

5.3.2 Lifting the Window Guarantee

As discussed back in Section 1.3.2, the window guarantee is an important reason why minimizers are so widely used. However, as discussed in the Syncmer paper [26], the window guarantee is sometimes too strong and limits design space for efficient sequence sketches. In fact, many applications of minimizers already skip k-mers that are too frequent, such as done in minimap2 [63]. This breaks the window guarantee because repetitive windows have no k-mer selected at all. Another interesting attempt is done in Winnowmap [46], where the idea of *Robust Winnowing*, first proposed in 2004 [103] is revived for read mapping. One side effect of using such robust winnowing scheme is breaking the window guarantee, however as Jain et al. [46] shows, this change alongside others bring considerable improvements to read mapping performance. We believe developing systematic approaches to lift the window guarantee and make the minimizer method in general more versatile is important.

Approximate UHS

A related potential line of research links back to universal hitting sets: relax the constraints on universal hitting sets and study approximate UHSes. This is a concept not yet defined in published work, but for example, a $(w, k, p)-$UHS can be defined as a set of k-mers such that a random window of w k-mers has probability at least $1 - p$ to contain a k-mer from the set. Bell [6] introduces a similar notion in combintorics context. We can similarly define approximate minimizers that are only required to select a k-mer with probability at least $1-p$. Our preliminary experiments has shown that both Mykkeltveit sets and open syncmers [26] are both better-than-random approximate UHSes, in the sense that they can hit more windows (lower p) compared to a random set of k-mers of the same size. (Although, any UHS has $p = 0$ and are always better than random.) We believe this concept is interesting from both theoretical and practical perspective.

5.3.3 Better Implementation of Sketches-by-Optimization

The polar sets can be used wherever universal hitting sets are used, in most cases. Given that our heuristics for layered polar sets only produce a small number of layers, implementation of a compatible minimizer with layered polar sets is not fundamentally different from that with a universal hitting set. The fixed interval sampling method is very similar to previously proposed methods [3, 35, 53], where the sketch of a reference sequence is simply the set of k-mers appearing at locations divisible by w. Polar sets might not be able to directly replace fixed interval sampling, however it can be readily expanded into a set of seeds that covers the whole reference sequence.

These approaches are currently relatively underused, compared to more traditional approach of minimizers like lexicographical, random or slight variants of these. A significant reason for the unpopularity of UHS-family methods is the fact that using these methods requires looking up a table of k-mers, be it a set of polar k-mers or universal hitting k-mers, for every k-mer in the query sequence. In contrast, for a random minimizer implemented using a hash function, no lookup is required during the sequence sketch generation process. Since these lookup tables are usually the result of sequence-specific optimization, we say these methods fall into the category of "sketches-by-optimization". This contrast leads to interesting tradeoffs in efficiency. For example, using a polar-set-compatible minimizer generates a more compact sequence sketch, but might take more computation at query time compared to using a random minimizer, due to the time spent in loading and querying the set of polar k-mers.

We believe better implementation of k-mer lookup tables and better optimization of sequence sketches, possibly in a joint manner, will popularize sketches-by-optimization, and this is an interesting problem to work on in the future. Existing methods already take step towards this goal. Jain et al. [46] uses a compact lookup table to index frequent k-mers, and Liu et al. [67] uses a Bloom filter to perform approximate query over fixed interval samples. Techniques like lock-free k-mer hash [75], k-mer Bloom filters [89] and k-mer bit trees [54] might also further help the performance.

5.3.4 Tuning window length and k-mer length

Throughout this dissertation, we have not touched the optimization of w and k. Tuning these parameters is highly application dependent, and in general is beyond the scope of this dissertation. In general, larger w means lower sensitivity, but more efficient for indexing and succinct for storage. The meaning of k is much more complicated. Larger k could mean lower collision rate, higher sensitivity, more buckets, larger lookup tables, among other things. But at least from the density optimization perspective, larger k usually means easier optimization of density. For extension of our existing work, tuning w and k alongside the ordering could lead to much better trade-off and performance gain.

5.3.5 Alternative Measurements of Efficiency

Throughout this dissertation our goal has been the optimization of density in either sequence-blind or sequence-specific scenarios. Lower density leads to smaller sequence sketches (in expectation), and for many applications this is desirable. However, depending on the way one uses the sequence sketch, alternative measurements of efficiency may be more desirable (also see discussion in Edgar [26]). For example, in k-mer counting, minimizers are used to place k-mers into buckets. In this case, the specific density is less relevant, and we are more concerned about the number of buckets, and the load balance between different buckets [73, 83]. For read mapping, smaller sequence sketches have their advantages, while some may prefer reducing the number of matches (which are especially troublesome for highly repetitive regions), or reducing the number of false positive seed matches in general. We believe many of these objectives are correlated with each other, and we are interested in both further exploring benefits of a small sequence sketch, and optimization techniques for alternative measurements of efficiency. Some of the alternative measurements may also come with interesting theoretical questions that can be answered.

5.4 Concluding Remark

Sequence sketching algorithms are indispensable parts of modern computational biology, and minimizer sketches are one of the most popular sketches for a variety of applications. We believe the popularity of minimizer sketches will only grow due to its simplicity and versatility, and our work will serve as important foundations for future method development in the field.

Bibliography

[1] Mohamed Ibrahim Abouelhoda, Stefan Kurtz, and Enno Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *J. Discrete Algorithms*, 2:53–86, 2004. 1.2

[2] Ludmil B. Alexandrov, Serena Nik-Zainal, David C. Wedge, Samuel Aparicio, Sam Behjati, Andrew V. Biankin, Graham Bignell, Niccolo' Bolli, Åke Borg, Anne-Lise Børresen-Dale, S Boyault, Birgit Burkhardt, Adam P. Butler, Carlos Caldas, H. Davies, Christine Desmedt, Roland Eils, Jórunn Erla Eyfjörd, John A. Foekens, Mel Greaves, F. Hosoda, Barbara Hutter, Tomislav Ilicic, Sandrine Imbeaud, Marcin Imielinsk, Natalie Jäger, David T. W. Jones, David Jones, Stian Knappskog, Marcel Kool, Sunil R. Lakhani, Carlos López-Otín, Sancha Martin, Nikhil C. Munshi, Hiromi Nakamura, Paul A. Northcott, Marina Pajic, Elli Papaemmanuil, Angelo Virgilio Paradiso, John V. Pearson, Xosé S. Puente, Keiran M Raine, Manasa Ramakrishna, Andrea L. Richardson, Julia Richter, Philip Rosenstiel, Matthias Schlesner, Ton N. M. Schumacher, Paul N. Span, Jon W. Teague, Yasushi Totoki, Andrew N.J. Tutt, Rafael Valdés-Mas, Marit M. van Buuren, Laura J van 't Veer, Anne Vincent-Salomon, Nicola Waddell, Lucy R Yates, Jessica Zucman-Rossi, P. Andrew Futreal, Ultan McDermott, Peter Lichter, Matthew L Meyerson, Sean M. Grimmond, Reiner Siebert, Elías Campo, Tatsuhiro Shibata, Stefan M. Pfister, Peter J. Campbell, and Michael R. Stratton. Signatures of mutational processes in human cancer. *Nature*, 500:415 – 421, 2013. 1.1

[3] Meznah Almutairy and Eric Torng. Comparing fixed sampling with minimizer sampling when using k-mer indexes to find maximal exact matches. *PLOS ONE*, 13(2):e0189960, 2018. 5.3.3

[4] Haihua Bai, Xiaosen Guo, Narisu Narisu, Tianming Lan, Qizhu Wu, Yanping Xing, Yong Zhang, Stephen R Bond, Zhili Pei, Yanru Zhang, et al. Whole-genome sequencing of 175 mongolians uncovers population-specific genetic architecture and gene flow throughout north and east asia. *Nature Genetics*, 50(12):1696–1704, 2018. 1.1

[5] Cristina Bazgan, Bruno Escoffier, and Vangelis Th Paschos. Completeness in standard and differential approximation classes: Poly-(D) APX-and (D) PTAS-completeness. *Theoretical Computer Science*, 339(2-3):272–292, 2005. 4.2.5

[6] Jason P Bell. Unavoidable and almost unavoidable sets of words. *International Journal of Algebra and Computation*, 15(04):717–724, 2005. 5.3.2

[7] Michael A. Bender, Martin Farach-Colton, Rob Johnson, Bradley C. Kuszmaul, Dzejla Medjedovic, Pablo Montes, Pradeepa Anand Shetty, Richard P. Spillane, and Erez Zadok. Don't thrash: How to cache your hash on flash. *Proc. VLDB Endow.*, 5:1627–1637, 2011.

1.2

[8] Konstantin Berlin, Sergey Koren, C. Chin, James P Drake, Jane M. Landolin, and Adam M. Phillippy. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature Biotechnology*, 33:623–630, 2015. 1.2

[9] S. R. Blackburn. Non-overlapping codes. *IEEE Transactions on Information Theory*, 61 (9):4890–4894, 2015. doi: 10.1109/TIT.2015.2456634. 4.5.1

[10] Francine Blanchet-Sadri, Naomi C Brownstein, Andy Kalcic, Justin Palumbo, and Tracy Weyand. Unavoidable sets of partial words. *Theory of Computing Systems*, 45(2):381–406, 2009. 2.1

[11] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13:422–426, 1970. 1.2

[12] Dorret I Boomsma, Cisca Wijmenga, Eline P Slagboom, Morris A Swertz, Lennart C Karssen, Abdel Abdellaoui, Kai Ye, Victor Guryev, Martijn Vermaat, Freerk Van Dijk, et al. The genome of the Netherlands: design, and project goals. *European Journal of Human Genetics*, 22(2):221–227, 2014. 1.1

[13] Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997. 1.2

[14] C. Titus Brown and Luiz C. Irber. sourmash: a library for MinHash sketching of DNA. *J. Open Source Softw.*, 1:27, 2016. 1.2

[15] Michael B. Burns, Nuri A. Temiz, and Reuben S. Harris. Evidence for APOBEC3B mutagenesis in multiple human cancers. *Nature Genetics*, 45:977–983, 2013. 1.1

[16] Alexander Burstein and Sergey Kitaev. On unavoidable sets of word patterns. *SIAM Journal on Discrete Mathematics*, 19(2):371–381, 2005. 2.1

[17] Jean-Marc Champarnaud, Georges Hansel, and Dominique Perrin. Unavoidable sets of constant length. *International Journal of Algebra and Computation*, 14(2):241–251, April 2004. ISSN 0218-1967. doi: 10.1142/S0218196704001700. 1.4.3, 2.1, 5.2.1

[18] Kok Hao Chen, Alistair N Boettiger, Jeffrey R Moffitt, Siyuan Wang, and Xiaowei Zhuang. Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233), 2015. 1.1

[19] Rayan Chikhi and Guillaume Rizk. Space-efficient and exact de Bruijn graph representation based on a bloom filter. *Algorithms for Molecular Biology : AMB*, 8:22 – 22, 2012. 1.2

[20] Rayan Chikhi, Antoine Limasset, Shaun Jackman, Jared T. Simpson, and Paul Medvedev. On the representation of de Bruijn graphs. *Journal of Computational Biology*, 22(5):336–352, January 2015. ISSN 1066-5277. doi: 10.1089/cmb.2014.0160. 1.3.3

[21] Rayan Chikhi, Antoine Limasset, and Paul Medvedev. Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, 32(12):i201–i208, 2015. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw279. 1.3.3, 1.4.4, 2.7.3, 4

[22] Ananyo Choudhury, Michèle Ramsay, Scott Hazelhurst, Shaun Aron, Soraya Bardien, Gerrit Botha, Emile R Chimusa, Alan Christoffels, Junaid Gamieldien, Mahjoubeh J Sefid-Dashti, et al. Whole-genome sequencing for an enhanced understanding of genetic variation among South Africans. *Nature Communications*, 8(1):1–12, 2017. 1.1

[23] Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M. Rueda, Mark J. Dunning, Doug Speed, Andy G. Lynch, Shamith A. Samarajiwa, Yinyin Yuan, Stefan Gräf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, Steven McKinney, Anita Langerød, Andrew R. Green, Elena Provenzano, Gordon C. Wishart, Sarah E. Pinder, Peter H. Watson, Florian Markowetz, Leigh Murphy, Ian O. Ellis, Arnie Purushotham, Anne-Lise Børresen-Dale, James D. Brenton, Simon Tavaré, Carlos Caldas, and Samuel Aparicio. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486:346 – 352, 2012. 1.1

[24] Dan DeBlasio, Fiyinfoluwa Gbosibo, Carl Kingsford, and Guillaume Marçais. Practical universal k-mer sets for minimizer schemes. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '19, pages 167–176, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6666-3. doi: 10.1145/3307339.3342144. 1.4.4, 1.5.2, 2.7.3, 4

[25] Sebastian Deorowicz, Marek Kokot, Szymon Grabowski, and Agnieszka Debudaj-Grabysz. KMC 2: Fast and resource-frugal k-mer counting. *Bioinformatics*, 31(10): 1569–1576, 2015. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btv022. 1.3.3

[26] Robert C Edgar. Syncmers are more sensitive than minimizers for selecting conserved k-mers in biological sequences. *bioRxiv*, 2020. 1.4.4, 5.3.2, 5.3.2, 5.3.5

[27] Jesper Eisfeldt, Gustaf Mårtensson, Adam Ameur, Daniel Nilsson, and Anna Lindstrand. Discovery of novel sequences in 1,000 Swedish genomes. *Molecular Biology and Evolution*, 37(1):18–30, 2020. 1.1

[28] Barış Ekim, Bonnie Berger, and Yaron Orenstein. A randomized parallel algorithm for efficiently finding near-optimal universal hitting sets. January 2020. 1.4.4, 1.5.2, 2.7.3, 3.1, 3.4.2, 3.4.3

[29] Chee-Huat Linus Eng, Michael Lawson, Qian Zhu, Ruben Dries, Noushin Koulena, Yodai Takei, Jina Yun, Christopher Cronin, Christoph Karp, Guo-Cheng Yuan, et al. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+. *Nature*, 568 (7751):235–239, 2019. 1.1

[30] Marius Erbert, Steffen Rechner, and Matthias Müller-Hannemann. Gerbil: a fast and memory-efficient k-mer counter with GPU-support. *Algorithms for Molecular Biology*, 12(1):1–12, 2017. 1.3.3

[31] Li Fan, Pei Cao, J. Almeida, and A.Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM Transactions on Networking*, 8(3):281–293, 2000. doi: 10.1109/90.851975. 1.2

[32] Paolo Ferragina and Giovanni Manzini. Indexing compressed text. *J. ACM*, 52:552–581, 2005. 1.2

[33] National Center for Biotechnology Information. SRA growth, 2021. URL `https://www.ncbi.nlm.nih.gov/sra/docs/sragrowth/`. 1.1

[34] Laurent C Francioli, Andronild Menelaou, Sara L Pulit, Freerk Van Dijk, Pier Francesco Palamara, Clara C Elbers, Pieter BT Neerincx, Kai Ye, Victor Guryev, Wigard P Kloosterman, et al. Whole-genome sequence variation, population structure and demographic history of the Dutch population. *Nature Genetics*, 46(8):818–825, 2014. 1.1

[35] Martin C Frith, Laurent Noé, and Gregory Kucherov. Minimally-overlapping words for sequence similarity search. *bioRxiv*, 2020. 4.5.1, 5.3.3

[36] Erik Garrison, Jouni Sirén, Adam M Novak, Glenn Hickey, Jordan M Eizenga, Eric T Dawson, William Jones, Shilpa Garg, Charles Markello, Michael F Lin, et al. Variation graph toolkit improves read mapping by representing genetic variation in the reference. *Nature Biotechnology*, 36(9):875–879, 2018. 1.1

[37] Robert Giegerich and Stefan Kurtz. From Ukkonen to McCreight and Weiner: A unifying view of linear-time suffix tree construction. *Algorithmica*, 19:331–353, 1997. 1.2

[38] Deepti Gurdasani, Tommy Carstensen, Fasil Tekola-Ayele, Luca Pagani, Ioanna Tachmazidou, Konstantinos Hatzikotoulas, Savita Karthikeyan, Louise Iles, Martin O Pollard, Ananyo Choudhury, et al. The African genome variation project shapes medical genetics in Africa. *Nature*, 517(7534):327–332, 2015. 1.1

[39] Peter S. Hammerman, Douglas Voet, Michael S. Lawrence, Douglas Voet, Rui Jing, Kristian Cibulskis, Andrey Y. Sivachenko, Petar Stojanov, Aaron McKenna, Eric S. Lander, S. Gabriel, Gad Getz, Marcin Imieliński, Elena Helman, Bryan Hernandez, Nam Pho, Matthew L Meyerson, Andy Chu, Jung E. Hye-Chun, Andrew J. Mungall, Erin D. Pleasance, A. Gordon Robertson, Payal Sipahimalani, Dominik Stoll, Miruna Balasundaram, Inanç Birol, Yaron S. N. Butterfield, Eric Chuah, Robin Jn Coope, Richard Corbett, Noreen Dhalla, Ranabir Guin, Ann He, Carrie Hirst, Martin Hirst, Robert A. Holt, Darlene Lee, Haiyan I. Li, Michael Mayo, Richard A. Moore, Karen L. Mungall, Ka Ming Nip, Adam B. Olshen, Jacqueline E. Schein, Jared R. Slobodan, Angela Tam, Nina Thiessen, Richard Varhol, Thomas Zeng, Yongjun Zhao, Steven J. M. Jones, Marco A. Marra, Gordon Saksena, Andrew D. Cherniack, Steven E Schumacher, Barbara Tabak, Scott L. Carter, Huy Nguyen, Roberto Onofrio, Andrew Crenshaw, Kristin G. Ardlie, Rameen Beroukhim, Wendy Winckler, Alexei I. Protopopov, Jianhua Zhang, Angela Hadjipanayis, Semin Lee, Ruibin Xi, Lixing Yang, Xiaojia Ren, Hailei Zhang, Sachet A. Shukla, Peng-Chieh Chen, Psalm S Haseley, Eunjung Alice Lee, Lynda Chin, Peter J. Park, Raju Kucherlapati, Nicholas D. Socci, Yupu Liang, Nikolaus Schultz, Laetitia Borsu, Alex E. Lash, Agnes J. Viale, Chris Sander, Marc Ladanyi, James T. Auman, Katherine A. Hoadley, Matthew D. Wilkerson, Yan Shi, Christina L. Liquori, Shao-Wu Meng, Ling Li, Yidi J. Turman, Michael D. Topal, Donghui Tan, Scot Michael Waring, Elizabeth Buda, Jesse L. Walsh, Corbin D. Jones, Piotr A. Mieczkowski, Darshan Singh, Junyuan Wu, Anisha Gulabani, Peter Dolina, Tom Bodenheimer, Alan Hoyle, Janae V. Simons, Matthew G. Soloway, Joshua M. Stuart, Lisle E. Mose, Stuart R. Jefferys, Saianand Balu, Brian D. O'Connor, Derek Y. Chiang, Jinze Liu, David Neil Hayes, Charles M. Perou, Leslie M. Cope, Daniel J. Weisenberger, Ludmila V. Danilova, D. Weisenberger,

Dennis T. Maglinte, Fei Pan, David J. Van Den Berg, Timothy J. Triche, James Gordon Herman, Stephen B. Baylin, Peter W. Laird, Michael S. Noble, Nils Gehlenborg, Danielle Dicara, Jinhua Zhang, Chang-Jiun Wu, Spring Yingchun Liu, Lihua Zou, Pei Lin, Juok Cho, Marc-Danie Nazaire, James Robinson, Helga Thorvaldsdóttir, Jill P. Mesirov, Rileen Sinha, Giovanni Ciriello, Ethan G. Cerami, Benjamin E. Gross, Anders S. Jacobsen, Jianjiong Gao, Bülent Arman Aksoy, Nils Weinhold, Ricardo Ramirez, Barry S. Taylor, Yevgeniy Antipin, Boris Reva, Qianxing Mo, Venkatraman E. Seshan, Paul Paik, Rehan Akbani, Nianxiang Zhang, Bradley M. Broom, Tod D. Casasent, Anna Unruh, Chris Wakefield, Ronald C Cason, Keith A. Baggerly, John N. Weinstein, David Haussler, Christopher C. Benz, Jingchun Zhu, Christopher Szeto, Gary K. Scott, Christina Yau, Sam Ng, Theodore Goldstein, Peter Waltman, Artem Sokolov, Eric A. Collisson, Kyle Ellrott, Daniel R. Zerbino, Singer Ma, Brian Craft, Ying Du, Christopher R. Cabanski, Vonn Walter, J. S. Marron, Yufeng Liu, Kai Wang, Jan Prins, Chad J. Creighton, Yiqun Zhang, William D. Travis, Natasha Rekhtman, Joanne Eunhee Yi, Marie Christine Aubry, Richard T. Cheney, Sanja Dacic, Douglas B. Flieder, William K. Funkhouser, Peter B. Illei, Jerome B. Myers, Ming-Sound Tsao, Robert J. Penny, David W Mallery, Troy Shelton, Martha Hatfield, Scott Morris, Peggy Yena, Candace Shelton, Mark E. Sherman, Joseph D. Paulauskis, Ramaswamy Govindan, Ijeoma A. Azodo, David G. Beer, Ron Bose, Lauren Averett Byers, David P. Carbone, Li-Wei Chang, Elizabeth Chun, Li Ding, John V. Heymach, Cristiane M. Ida, Bruce Evan Johnson, Igor Jurisica, Jacob M. Kaufman, Farhad Kosari, David J. Kwiatkowski, Christopher A. Maher, Andrew J. Mungall, William Pao, Martin Peifer, Gordon Robertson, Valerie Rusch, Jill M. Siegfried, Carrie Sougnez, Roman K. Thomas, Sandra C. Tomaszek, Charles J. Vaske, David A. Wheeler, Dennis A. Wigle, Christopher Wilks, Jianjua John Zhang, Mark A. Jensen, Robert Sfeir, Ari B. Kahn, Anna L. Chu, Prachi Kothiyal, Zhining Wang, E. E. Snyder, Joan U. Pontius, Todd Pihl, Brenda Ayala, Mark Backus, Jessica L Walton, Julien Baboud, Dominique L. Berton, Matthew C. Nicholls, Deepak Srinivasan, Rohini Raman, Stanley Girshik, Peter A. Kigonya, Shelley Alonso, Rashmi N. Sanbhadti, Sean P. Barletta, J. M. Greene, David Pot, Bizhan Bandarchi-Chamkhaleh, Jeff Boyd, JoEllen Weaver, Ijeoma A. Azodo, Christiane M. Ida, Ping Yang, Malcolm V. Brock, Kristen Rogers, Marian Rutledge, Travis Brown, Beverly Lee, James J. Shin, Dante C Trusty, Rajiv Dhir, Olga Potapova, Konstantin V. Fedosenko, Elena Nemirovich-Danchenko, Maureen F. Zakowski, Mary V. Iacocca, Jennifer Brown, Brenda Rabeno, Christine Czerwinski, Nicholas J. Petrelli, Zhen Fan, Nicole Todaro, John Eckman, W. Rathmell, Leigh B. Thorne, Meijuan Huang, L. Peter Boice, Ashley D. Hill, Erin E. Curley, Carl D. Morrison, Carmelo Gaudioso, John M. S. Bartlett, Sugy Kodeeswaran, Brent W. Zanke, Harmanjatinder S. Sekhon, Kerstin A. David, Hartmut Juhl, Xuan Van Le, Bernard Kohl, Richard A. Thorp, Nguyen Viet Tien, Nguyen Van Bang, Howard H. Sussman, Bui Duc Phu, Richard Hajek, Nguyen Phi Hung, Khurram Zaman Khan, Thomas Muley, Kenna R. Mills Shaw, Margi Sheth, Liming Yang, Kenneth H. Buetow, Tanja Davidsen, John A. Demchok, Greg Eley, Martin L Ferguson, Laura A. L. Dillon, Carl F. Schaefer, Mark Guyer, Bradley A Ozenberger, Jacqueline D. Palchik, Jane Peterson, Heidi J. Sofia, Elizabeth J. Thomson, and Ronglai Shen. Comprehensive genomic characterization of squamous cell lung cancers. *Nature*, 489:519 – 525, 2012. 1.1

[40] Peter M Higgins and Christopher J Saker. Unavoidable sets. *Theoretical computer science*, 359(1-3):231–238, 2006. 2.1

[41] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC '98*, 1998. 1.2

[42] Guy S Jacobs, Georgi Hudjashov, Lauri Saag, Pradiptajati Kusuma, Chelzie C Darusallam, Daniel J Lawson, Mayukh Mondal, Luca Pagani, François-Xavier Ricaut, Mark Stoneking, et al. Multiple deeply divergent denisovan ancestries in papuans. *Cell*, 177(4):1010–1021, 2019. 1.1

[43] Chirag Jain, Alexander T. Dilthey, Sergey Koren, Srinivas Aluru, and Adam M. Phillippy. A fast approximate algorithm for mapping long reads to large reference databases. *Journal of Computational Biology*, 25 7:766–779, 2018. 1.2

[44] Chirag Jain, Sanchit Misra, Haowen Zhang, Alexander Dilthey, and Srinivas Aluru. Accelerating sequence alignment to graphs. In *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 451–461, 2019. doi: 10.1109/IPDPS.2019.00055. 1.1

[45] Chirag Jain, Arang Rhie, Nancy Hansen, Sergey Koren, and Adam M. Phillippy. A long read mapping method for highly repetitive reference sequences. *bioRxiv*, page 2020.11.01.363887, November 2020. doi: 10.1101/2020.11.01.363887. 1.3.3

[46] Chirag Jain, Arang Rhie, Haowen Zhang, Claudia Chu, Brian P Walenz, Sergey Koren, and Adam M Phillippy. Weighted minimizer sampling improves long read mapping. *Bioinformatics*, 36(Supplement_1):i111–i118, July 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa435. 1.3.3, 1.4.4, 4, 4.2.1, 4.3.2, 4.4.3, 5.3.2, 5.3.3

[47] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology*, 36(4):338–345, 2018. 1.1

[48] Cyriac Kandoth, Michael D. McLellan, Fabio Vandin, Kai Ye, Beifang Niu, Charles Lu, Mingchao Xie, Qunyuan Zhang, Joshua F. McMichael, Matthew A. Wyczalkowski, Mark D. M. Leiserson, Christopher A. Miller, John S. Welch, Matthew J. Walter, Michael C. Wendl, Timothy J. Ley, Richard K. Wilson, Benjamin J. Raphael, and Li Ding. Mutational landscape and significance across 12 major cancer types. *Nature*, 502:333 – 339, 2013. 1.1

[49] Juha Kärkkäinen and Peter Sanders. Simple linear work suffix array construction. In *International colloquium on automata, languages, and programming*, pages 943–955. Springer, 2003. 4.3.4

[50] Rongqin Ke, Marco Mignardi, Alexandra Pacureanu, Jessica Svedlund, Johan Botling, Carolina Wählby, and Mats Nilsson. In situ sequencing for RNA analysis in preserved tissue and cells. *Nature Methods*, 10(9):857–860, 2013. 1.1

[51] Birte Kehr, Anna Helgadottir, Pall Melsted, Hakon Jonsson, Hannes Helgason, Adalbjörg Jonasdottir, Aslaug Jonasdottir, Asgeir Sigurdsson, Arnaldur Gylfason, Gisli H Halldors-

son, et al. Diversity in non-repetitive human sequences not found in the reference genome. *Nature Genetics*, 49(4):588–593, 2017. 1.1

[52] Dominik Kempa and Tomasz Kociumaka. String synchronizing sets: sublinear-time BWT construction and optimal LCE data structure. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 756–767, 2019. 4

[53] Nilesh Khiste and Lucian Ilie. E-MEM: efficient computation of maximal exact matches for very large genomes. *Bioinformatics*, 31(4):509–514, 2015. 5.3.3

[54] Carl Kingsford and Rob Patro. Reference-based compression of short-read sequences using path encoding. *Bioinformatics*, 31(12):1920–1928, 2015. 5.3.3

[55] Gregory Kucherov and Kamil Salikhov. Using cascading bloom filters to improve the memory usage for de brujin graphs. *Algorithms for Molecular Biology : AMB*, 9:2 – 2, 2013. 1.2

[56] Stefan Kurtz, Adam M. Phillippy, Arthur L. Delcher, Michael E. Smoot, Martin Shumway, Corina Antonescu, and Steven L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biology*, 5:R12 – R12, 2003. 1.2

[57] Ben Langmead and Steven L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature Methods*, 9:357–359, 2012. 1.2

[58] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10: R25 – R25, 2008. 1.2

[59] Vladimir Iosifovich Levenshtein. Maximum number of words in codes without overlaps. *Problemy Peredachi Informatsii*, 6(4):88–90, 1970. 4.5.1

[60] Douglas A. Levine, Gad Stacey B. Kristian Eric Andrey Carrie Mike Cyriac Getz Gabriel Cibulskis Lander Sivachenko Sougnez L, Gad Getz, S. Gabriel, Kristian Cibulskis, Eric S. Lander, Andrey Y. Sivachenko, Carrie Sougnez, Mike S Lawrence, Cyriac Kandoth, David J. Dooling, Robert S. Fulton, Lucinda Fulton, Joelle M. Kalicki-Veizer, Michael D. McLellan, Michelle D. O'Laughlin, Heather K. Schmidt, Richard K. Wilson, Kai Ye, Li Ding, Elaine R. Mardis, Adrian Ally, Miruna Balasundaram, Inanç Birol, Yaron S. N. Butterfield, Rebecca Carlsen, Candace D. Carter, Andy Chu, Eric Chuah, Hye-Jung E. Chun, Noreen Dhalla, Ranabir Guin, Carrie Hirst, Robert A. Holt, Steven J. M. Jones, Darlene Lee, Haiyan I. Li, Marco A. Marra, Michael Mayo, Richard A. Moore, Andrew J. Mungall, Patrick Plettner, Jacqueline E. Schein, Payal Sipahimalani, Angela Tam, Richard Varhol, A. Gordon Robertson, Andrew D. Cherniack, Itai M. Pashtan, Gordon Saksena, Roberto Onofrio, Steven E Schumacher, Barbara Tabak, Scott L. Carter, Bryan Hernandez, Jeff Gentry, Helga Birgitte Salvesen, Kristin G. Ardlie, Wendy Winckler, Rameen Beroukhim, Matthew L Meyerson, Angela Hadjipanayis, Semin Lee, Harshad S. Mahadeshwar, Peter J. Park, Alexei I. Protopopov, Xiaojia Ren, Sahil Seth, Xingzhi Song, Jiabin Tang, Ruibin Xi, Lixing Yang, Dong Zeng, Raju Kucherlapati, Lynda Chin, Jianhua Zhang, J. Todd Auman, Saianand Balu, Tom Bodenheimer, Elizabeth Buda, David Neil Hayes, Alan Hoyle, Stuart R. Jefferys, Corbin D. Jones, Shao-Wu Meng, Piotr A. Mieczkowski, Lisle E. Mose, Joel S. Parker, Charles M. Perou, Jeff Roach, Yan Shi,

Janae V. Simons, Mathew G. Soloway, Donghui Tan, Michael D. Topal, Scot Michael Waring, Junyuan Wu, Katherine A. Hoadley, Stephen B. Baylin, Moiz S. Bootwalla, Phillip H. Lai, Timothy J. Triche Jr, David J. Van Den Berg, Daniel J. Weisenberger, Peter W. Laird, Hui Shen, Juok Cho, Danielle Dicara, Scott R. Frazer, David I. Heiman, Rui Jing, Pei Lin, William Mallard, Petar Stojanov, Douglas Voet, Hailei Zhang, Lihua Zou, Michael S. Noble, Sheila M. Reynolds, Ilya Shmulevich, Bülent Arman Aksoy, Yevgeniy Antipin, Giovanni Ciriello, Gideon Dresdner, Jianjiong Gao, Benjamin E. Gross, Anders S. Jacobsen, Marc Ladanyi, Boris Reva, Chris Sander, Rileen Sinha, Selcuk Onur Sumer, Barry S. Taylor, Ethan G. Cerami, Nils Weinhold, Nikolaus Schultz, Ronglai Shen, Stephen C. Benz, Theodore Goldstein, David Haussler, Sam Ng, Christopher Szeto, Joshua M. Stuart, Christopher C. Benz, Christina Yau, Wei Zhang, Matti Annala, Bradley M. Broom, Tod D. Casasent, Zhenlin Ju, Han Liang, Guoyan Liu, Yiling Lu, Anna Unruh, Chris Wakefield, John N. Weinstein, Nianxiang Zhang, Yuexin Liu, Russell R. Broaddus, Rehan Akbani, Gordon B. Mills, Christopher Adams, Thomas Barr, Aaron D. Black, Jay Bowen, John Deardurff, J Frick, Julie M. Gastier-Foster, Tom Grossman, Hollie A. Harper, Melissa Hart-Kothari, Carmen Helsel, Aaron Hobensack, Harkness Kuck, Kelley Kneile, Kristen M. Leraas, Tara M. Lichtenberg, Cynthia Mcallister, Robert Pyatt, Nilsa C. Ramirez, Teresa R. Tabler, Nathan Vanhoose, Peter White, Lisa Wise, E J Zmuda, Nandita Barnabas, Charlenia Berry-Green, Victoria M Blanc, L. Peter Boice, Michael Button, Ádám Farkas, Alex E. Green, Jean Rgn Mackenzie, Dana Nicholson, Steve E Kalloger, C. Blake Gilks, Beth Y. Karlan, Jenny Lester, Sandra Orsulic, Mark E. Borowsky, Mark G. Cadungog, Christine Czerwinski, Lori Huelsenbeck-Dill, Mary V. Iacocca, Nicholas J. Petrelli, Brenda Rabeno, Gary Witkin, Elena Nemirovich-Danchenko, Olga Potapova, Daniil Rotin, Andrew Berchuck, Michael J. Birrer, P J Disaia, Laura Monovich, Erin E. Curley, Johanna Gardner, David W Mallery, Robert J. Penny, Sean C. Dowdy, Boris J. Winterhoff, Linda N. Dao, Bobbie S. Gostout, Alexandra Meuter, Attila Teoman, Fanny Dao, Narciso Olvera, Faina Bogomolniy, Karuna Garg, Robert A. Soslow, Douglas A. Levine, Mikhail Abramov, John M. S. Bartlett, Sugy Kodeeswaran, Jeremy R. Parfitt, F. V. Moiseenko, Blaise Alexander Clarke, Marc T. Goodman, Michael E. Carney, Rayna K. Matsuno, Jennifer C. Fisher, Meijuan Huang, W. Kimryn Rathmell, Leigh B. Thorne, Linda Van Le, Rajiv Dhir, Robert A. Edwards, Esther Elishaev, Kristin K Zorn, Paul J. Goodfellow, David G. Mutch, Ari B. Kahn, Daphne W Bell, Pamela. M. Pollock, Chen Wang, David A.Wheeler, Eve Shinbrot, Brenda Ayala, Anna L. Chu, Mark A. Jensen, Prachi Kothiyal, Todd Pihl, Joan U. Pontius, David Pot, E. E. Snyder, Deepak Srinivasan, Kenna R. Mills Shaw, Margi Sheth, Tanja Davidsen, Greg Eley Martin L. Ferguson, John A. Demchok, Liming Yang, Mark Guyer, Bradley A Ozenberger, and Heidi J. Sofia. Integrated genomic characterization of endometrial carcinoma. *Nature*, 497:67 – 73, 2013. 1.1

[61] Timothy J. Ley, Christopher B. Miller, Li xia Ding, Benjamin J. Raphael, Andrew J. Mungall, A. Gordon Robertson, Katherine A. Hoadley, Timothy J. Triche, Peter W. Laird, Jack D. Baty, Lucinda L. Fulton, Robert S. Fulton, Sharon E. Heath, Joelle M. Kalicki-Veizer, Cyriac Kandoth, Jeffery M. Klco, Daniel C. Koboldt, Krishna L. Kanchi, Shashikant Kulkarni, Tamara L. Lamprecht, David E. Larson, Ling Lin, Charles Lu, Michael D. McLellan, Joshua F. McMichael, Jacqueline E. Payton, Heather K. Schmidt,

David H. Spencer, Michael H. Tomasson, John W. Wallis, Lukas D Wartman, Mark A. Watson, John S. Welch, Michael C. Wendl, Adrian Ally, Miruna Balasundaram, Inanç Birol, Yaron S. N. Butterfield, Readman Chiu, Andy Chu, Eric Chuah, Hye-Jung E. Chun, Richard Corbett, Noreen Dhalla, Ranabir Guin, Ann He, Carrie Hirst, Martin Hirst, Robert A. Holt, Steven Jones, Aly Karsan, Darlene Lee, Haiyan I. Li, Marco A. Marra, Michael Mayo, Richard A. Moore, Karen L. Mungall, Jeremy D K Parker, Erin D. Pleasance, Patrick Plettner, Jacquie E. Schein, Dominik Stoll, Lucas Swanson, Angela Tam, Nina Thiessen, Richard Varhol, Natasja H. Wye, Yongjun Zhao, S. Gabriel, Gad Getz, Carrie Sougnez, Lihua Zou, Mark D. M. Leiserson, Fabio Vandin, Hsin-Ta Wu, Frederick Applebaum, Stephen B. Baylin, Rehan Akbani, Bradley M. Broom, Ken Chen, Thomas C. Motter, Khanh V.T. Nguyen, John N. Weinstein, Nianziang Zhang, Martin L Ferguson, Christopher Adams, Aaron D. Black, Jay Bowen, Julie M. Gastier-Foster, Tom Grossman, Tara M. Lichtenberg, Lisa Wise, Tanja Davidsen, John A. Demchok, Kenna R. Mills Shaw, Margi Sheth, Heidi J. Sofia, Liming Yang, James R. Downing, and Greg Eley. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *The New England Journal of Medicine*, 368 22:2059–74, 2013. 1.1

[62] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32 14:2103–10, 2016. 1.3.3, 1.3.3

[63] Heng Li and Inanc Birol. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty191. 1.3.3, 1.4.2, 4.2.1, 5.3.2

[64] Ruiqiang Li, Chang Yu, Yingrui Li, Tak Wah Lam, Siu-Ming Yiu, Karsten Kristiansen, and Jun Wang. Soap2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25 15:1966–7, 2009. 1.2

[65] Ruiqiang Li, Yingrui Li, Hancheng Zheng, Ruibang Luo, Hongmei Zhu, Qibin Li, Wubin Qian, Yuanyuan Ren, Geng Tian, Jinxiang Li, et al. Building the sequence map of the human pan-genome. *Nature Biotechnology*, 28(1):57–63, 2010. 1.1

[66] Yang Li et al. MSPKmerCounter: a fast and memory efficient approach for k-mer counting. *arXiv preprint arXiv:1505.06550*, 2015. 1.3.3

[67] Yuansheng Liu, Leo Yu Zhang, and Jinyan Li. Fast detection of maximal exact matches via fixed sampling of query k-mers and bloom filtering of index k-mers. *Bioinformatics*, 35(22):4560–4567, 2019. 5.3.3

[68] Ditte Lovatt, Brittani K Ruble, Jaehee Lee, Hannah Dueck, Tae Kyung Kim, Stephen Fisher, Chantal Francis, Jennifer M Spaethling, John A Wolf, M Sean Grady, et al. Transcriptome in vivo analysis (TIVA) of spatially defined single cells in live tissue. *Nature Methods*, 11(2):190–196, 2014. 1.1

[69] Eric Lubeck, Ahmet F Coskun, Timur Zhiyentayev, Mubhij Ahmad, and Long Cai. Single-cell in situ rna profiling by sequential hybridization. *Nature Methods*, 11(4):360–361, 2014. 1.1

[70] Ana-Teresa Maia, Stephen-John Sammut, Ana Jacinta-Fernandes, and Suet-Feung Chin. Big data in cancer genomics. *Current Opinion in Systems Biology*, 4:78–84, 2017. ISSN

2452-3100. doi: https://doi.org/10.1016/j.coisb.2017.07.007. 1.1

[71] Swapan Mallick, Heng Li, Mark Lipson, Iain Mathieson, Melissa Gymrek, Fernando Racimo, Mengyao Zhao, Niru Chennagiri, Susanne Nordenfelt, Arti Tandon, et al. The simons genome diversity project: 300 genomes from 142 diverse populations. *Nature*, 538 (7624):201–206, 2016. 1.1

[72] Udi Manber and Gene Myers. Suffix arrays: a new method for on-line string searches. *siam Journal on Computing*, 22(5):935–948, 1993. 4.3.4

[73] Guillaume Marçais, David Pellow, Daniel Bork, Yaron Orenstein, Ron Shamir, and Carl Kingsford. Improving the performance of minimizers and winnowing schemes. *Bioinformatics*, 33(14):i110–i117, July 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx235. 1.4.3, 1.5.2, 2.1, 2.1.1, 3.1, 3.2.1, 4.2.1, 4.4, 5.3.5

[74] Guillaume Marçais, Dan DeBlasio, and Carl Kingsford. Asymptotically optimal minimizers schemes. *Bioinformatics*, 34(13):i13–i22, July 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty258. 1.4.3, 1.4.4, 1.5.2, 2.1, 2.1.1, 2.1.1, 3, 3.1, 3.2.1, 4, 4.2.2, 5.2.3

[75] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27 6:764–70, 2011. 5.3.3

[76] Guillaume Marçais, Arthur L. Delcher, Adam M. Phillippy, Rachel Coston, Steven L. Salzberg, and Aleksey V. Zimin. Mummer4: A fast and versatile genome alignment system. *PLoS Computational Biology*, 14, 2018. 1.2

[77] Guillaume Marçais, Dan DeBlasio, Prashant Pandey, and Carl Kingsford. Locality sensitive hashing for the edit distance. *bioRxiv*, page 534446, 2019. 1.2

[78] Chiara Medaglia, Amir Giladi, Liat Stoler-Barak, Marco De Giovanni, Tomer Meir Salame, Adi Biram, Eyal David, Hanjie Li, Matteo Iannacone, Ziv Shulman, et al. Spatial reconstruction of immune niches by combining photoactivatable reporters and scRNA-seq. *Science*, 358(6370):1622–1626, 2017. 1.1

[79] Páll Melsted and Jonathan K. Pritchard. Efficient counting of k-mers in DNA sequences using a bloom filter. *BMC Bioinformatics*, 12:333 – 333, 2011. 1.2

[80] Karen H Miga, Sergey Koren, Arang Rhie, Mitchell R Vollger, Ariel Gershman, Andrey Bzikadze, Shelise Brooks, Edmund Howe, David Porubsky, Glennis A Logsdon, et al. Telomere-to-telomere assembly of a complete human X chromosome. *Nature*, 585(7823): 79–84, 2020. 4.4.3

[81] Donna M. Muzny, Matthew N. Bainbridge, Kyle Chang, Huyen Dinh, J Drummond, Gerald R. Fowler, Christie L. Kovar, Lora L Lewis, Margaret B. Morgan, Irene Newsham, Jeffrey S. Reid, Jireh Santibanez, Eve Shinbrot, Lisa R. Treviño, Yuan qing Wu, Min Wang, Preethi H. Gunaratne, Lawrence A. Donehower, Chad J. Creighton, David A. Wheeler, Richard A. Gibbs, Michael S. Lawrence, Douglas Voet, Rui Jing, Kristian Cibulskis, Andrey Y. Sivachenko, Petar Stojanov, Aaron McKenna, Eric S. Lander, S. Gabriel, Li Ding, Robert S. Fulton, Daniel C. Koboldt, Todd N. Wylie, Jason R. Walker, David J. Dooling, Lucinda Fulton, Kim D. Delehaunty, Catrina C. Fronick, Ryan T. Demeter,

Elaine R. Mardis, Richard K. Wilson, Andy Chu, Hye-Jung E. Chun, Andrew J. Mungall, Erin D. Pleasance, A. Gordon Robertson, Dominik Stoll, Miruna Balasundaram, Inanç Birol, Yaron S. N. Butterfield, Eric Chuah, Robin Jn Coope, Noreen Dhalla, Ranabir Guin, Carrie Hirst, Martin Hirst, Robert A. Holt, Darlene Lee, Haiyan I. Li, Michael Mayo, Richard A. Moore, Jacqueline E. Schein, Jared R. Slobodan, Angela Tam, Nina Thiessen, Richard Varhol, Thomas Zeng, Yongjun Zhao, Steven J. M. Jones, Marco A. Marra, Adam J. Bass, Alex H. Ramos, Gordon Saksena, Andrew D. Cherniack, Steven E Schumacher, Barbara Tabak, Scott L. Carter, Nam Pho, Huy Nguyen, Roberto Onofrio, Andrew Crenshaw, Kristin G. Ardlie, Rameen Beroukhim, Wendy Winckler, Matthew L Meyerson, Alexei I. Protopopov, Angela Hadjipanayis, Eunjung Alice Lee, Ruibin Xi, Lixing Yang, Xiaojia Ren, Narayanan Sathiamoorthy, Peng-Chieh Chen, Psalm S Haseley, Yonghong Xiao, Semin Lee, Jonathan G. Seidman, Lynda Chin, Peter J. Park, Raju Kucherlapati, James T. Auman, Katherine A. Hoadley, Ying Du, Matthew D. Wilkerson, Yan Shi, Christina L. Liquori, Shao-Wu Meng, Ling Li, Yidi J. Turman, Michael D. Topal, Donghui Tan, Scot Michael Waring, Elizabeth Buda, Jesse L. Walsh, Corbin D. Jones, Piotr A. Mieczkowski, Darshan Singh, Junyuan Wu, Anisha Gulabani, Peter Dolina, Tom Bodenheimer, Alan Hoyle, Janae V. Simons, Matthew G. Soloway, Lisle E. Mose, Stuart R. Jefferys, Saianand Balu, Brian D. O'Connor, Jan Prins, Derek Y. Chiang, David Neil Hayes, Charles M. Perou, Toshinori Hinoue, Daniel J. Weisenberger, Dennis T. Maglinte, Fei Pan, Benjamin Paul Berman, David J. Van Den Berg, Hui Shen, Timothy J. Triche, Stephen B. Baylin, Peter W. Laird, Gad Getz, Michael S. Noble, Doug Voat, Nils Gehlenborg, Danielle Dicara, Juinhua Zhang, Hailei Zhang, Chang-Jiun Wu, Spring Yingchun Liu, Sachet A. Shukla, Lihua Zhou, Pei Lin, Richard W. Park, Marc-Danie Nazaire, James Robinson, Helga Thorvaldsdóttir, Jill P. Mesirov, Vésteinn Thorsson, Sheila M. Reynolds, Brady Bernard, Richard Kreisberg, Jake Lin, Lisa E. Iype, Ryan Bressler, Timo Erkkilä, Madhumati Gundapuneni, Yuexin Liu, Adam Norberg, Thomas Robinson, Da Yang, Wei Zhang, Ilya Shmulevich, Jorma J. de Ronde, Nikolaus Schultz, Ethan G. Cerami, Giovanni Ciriello, Arthur P. Goldberg, Benjamin E. Gross, Anders S. Jacobsen, Jianjiong Gao, Bogumil Kaczkowski, Rileen Sinha, Bülent Arman Aksoy, Yevgeniy Antipin, Boris Reva, Ronglai Shen, Barry S. Taylor, Marc Ladanyi, Chris Sander, Rehan Akbani, Nianxiang Zhang, Bradley M. Broom, Tod D. Casasent, Anna Unruh, Chris Wakefield, Stanley R. Hamilton, Ronald C Cason, Keith A. Baggerly, John N. Weinstein, David Haussler, Christopher C. Benz, Joshua M. Stuart, Stephen C. Benz, J. Zachary Sanborn, Charles J. Vaske, Jingchun Zhu, Christopher Szeto, Gary K. Scott, Christina Yau, Sam Ng, Theodore Goldstein, Kyle Ellrott, Eric A. Collisson, Aaron E. Cozen, Daniel R. Zerbino, Christopher Wilks, Brian Craft, Paul T. Spellman, Robert J. Penny, Troy Shelton, Martha Hatfield, Scott Morris, Peggy Yena, Candace Shelton, Mark E. Sherman, Joseph D. Paulauskis, Julie M. Gastier-Foster, Jay Bowen, Nilsa C. Ramirez, Aaron D. Black, Robert Pyatt, Lisa Wise, Peter White, Monica M. Bertagnolli, Jennifer Brown, Timothy A. Chan, Gerald C. Chu, Christine Czerwinski, Frederick Denstman, Rajiv Dhir, Arnulf Dörner, Charles S. Fuchs, José G. Guillem, Mary V. Iacocca, Hartmut Juhl, Andrew Kaufman, Bernard Kohl, Xuan Van Le, M. C. Mariano, Elizabeth N. Medina, Michael Meyers, Garrett Nash, Phillip B. Paty, Nicholas J. Petrelli, Brenda Rabeno, William G Richards, David B. Solit, Patricia Swanson, Larissa Temple, Joel E. Tepper, Richard A.

Thorp, Efsevia Vakiani, Martin R. Weiser, Joseph E. Willis, Gary Witkin, Zhao shi Zeng, Michael J. Zinner, Carsten Zornig, Mark A. Jensen, Robert Sfeir, Ari B. Kahn, Anna L. Chu, Prachi Kothiyal, Zhining Wang, E. E. Snyder, Joan U. Pontius, Todd Pihl, Brenda Ayala, Mark Backus, Jessica L Walton, John H. Whitmore, Julien Baboud, Dominique L. Berton, Matthew C. Nicholls, Deepak Srinivasan, Rohini Raman, Stanley Girshik, Peter A. Kigonya, Shelley Alonso, Rashmi N. Sanbhadti, Sean P. Barletta, J. M. Greene, David Pot, Kenna R. Mills Shaw, Laura A. L. Dillon, Kenneth H. Buetow, Tanja Davidsen, John A. Demchok, Greg Eley, Martin L Ferguson, Peter Fielding, Carl F. Schaefer, Margi Sheth, Liming Yang, Mark Guyer, Bradley A Ozenberger, Jacqueline D. Palchik, Jane Peterson, Heidi J. Sofia, and Elizabeth J. Thomson. Comprehensive molecular characterization of human colon and rectal cancer. *Nature*, 487:330 – 337, 2012. 1.1

[82] Johannes Mykkeltveit. A proof of Golomb's conjecture for the de Bruijn graph. *Journal of Combinatorial Theory, Series B*, 13(1):40–45, August 1972. ISSN 0095-8956. doi: 10.1016/0095-8956(72)90006-8. 1.4.4, 2.1.1, 2.4.1, 2.4.1, 39, 4.2.1, 5.2.1

[83] Johan T Nyström-Persson, Gabriel Keeble-Gagnère, and Niamat Zawad. Compact and evenly distributed k-mer binning for genomic sequences. *bioRxiv*, 2020. 5.3.5

[84] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. Mash: fast genome and metagenome distance estimation using minhash. *Genome Biology*, 17, 2016. 1.2

[85] Yaron Orenstein, David Pellow, Guillaume Marçais, Ron Shamir, and Carl Kingsford. Compact universal k-mer hitting sets. In *Algorithms in Bioinformatics*, Lecture Notes in Computer Science, pages 257–268. Springer, Cham, 2016. ISBN 978-3-319-43680-7 978-3-319-43681-4. doi: 10.1007/978-3-319-43681-4_21. 1.4.3, 1.4.4, 1.5.2, 2.7.3, 3.1, 3.4.3, 4

[86] Prashant Pandey, Michael A Bender, Rob Johnson, and Rob Patro. debgr: an efficient and near-exact representation of the weighted de bruijn graph. *Bioinformatics*, 33(14): i133–i141, 2017. 1.2

[87] Prashant Pandey, Michael A. Bender, Rob Johnson, and Robert Patro. A general-purpose counting filter: Making every bit count. *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017. 1.2

[88] Prashant Pandey, Fatemeh Almodaresi, Michael A Bender, Michael Ferdman, Rob Johnson, and Rob Patro. Mantis: A fast, small, and exact large-scale sequence-search index. *Cell Systems*, 7(2):201–207, 2018. 1.2

[89] David Pellow, Darya Filippova, and Carl Kingsford. Improving bloom filter performance on sequence data using k-mer bloom filters. *Journal of Computational Biology*, 24(6): 547–557, 2017. 5.3.3

[90] Nelson Pérez, Miguel Gutierrez, and Nelson Vera. Computational performance assessment of k-mer counting algorithms. *Journal of Computational Biology*, 23(4):248–255, 2016. 1.3.3

[91] Charles M. Perou, Therese Sørlie, Michael B. Eisen, Matt van de Rijn, Stefanie S. Jeffrey, Christian A. Rees, Jonathan R. Pollack, Douglas T. Ross, Hilde Johnsen, Lars A.

Akslen, Øystein Fluge, Alexander Pergamenschikov, Cheryl F. Williams, Shirley X. Zhu, Per Eystein Lønning, Anne-Lise Børresen-Dale, Patrick O. Brown, and David Botstein. Molecular portraits of human breast tumours. *Nature*, 406:747–752, 2000. 1.1

[92] Sophie Petropoulos, Daniel Edsgärd, Björn Reinius, Qiaolin Deng, Sarita Pauliina Panula, Simone Codeluppi, Alvaro Plaza Reyes, Sten Linnarsson, Rickard Sandberg, and Fredrik Lanner. Single-cell RNA-seq reveals lineage and X chromosome dynamics in human preimplantation embryos. *Cell*, 165(4):1012–1026, 2016. 1.1

[93] Goran Rakocevic, Vladimir Semenyuk, Wan-Ping Lee, James Spencer, John Browning, Ivan J. Johnson, Vladan Arsenijevic, Jelena Nadj, Kaushik Ghose, Maria C. Suciu, Sun-Gou Ji, Gülfem Demir, Lizao Li, Berke Ç. Toptaş, Alexey Dolgoborodov, Björn Pollex, Iosif Spulber, Irina Glotova, Péter Kómár, Andrew L. Stachyra, Yilong Li, Milos Popovic, Morten Källberg, Amit Jain, and Deniz Kural. Fast and accurate genomic analyses using genome graphs. *Nature Genetics*, 51(2):354–362, February 2019. ISSN 1546-1718. doi: 10.1038/s41588-018-0316-4. 1.1

[94] Mikko Rautiainen, Veli Mäkinen, and Tobias Marschall. Bit-parallel sequence-to-graph alignment. *Bioinformatics*, 35(19):3599–3607, 03 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz162. 1.1

[95] Michael Roberts, Wayne Hayes, Brian R. Hunt, Stephen M. Mount, and James A. Yorke. Reducing storage requirements for biological sequence comparison. *Bioinformatics*, 20 (18):3363–3369, 2004. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/bth408. 1.3.1, 1.3.3, 2.1.1, 3.4.3

[96] Michael Roberts, Brian R. Hunt, James A. Yorke, Randall A. Bolanos, and Arthur L. Delcher. A preprocessor for shotgun assembly of large genomes. *Journal of Computational Biology*, 11(4):734–752, 2004. ISSN 1066-5277. doi: 10.1089/cmb.2004.11.734. 1.3.1, 1.3.3, 2.1.1, 2.1.1, 3.4.3

[97] Steven A Roberts, Michael S. Lawrence, Leszek J. Klimczak, Sara A. Grimm, David C. Fargo, Petar Stojanov, Adam Kiezun, Gregory V. Kryukov, Scott L. Carter, Gordon Saksena, Shawn F. Harris, Ruchir R. Shah, Michael A. Resnick, Gad Getz, and Dmitry A. Gordenin. An APOBEC cytidine deaminase mutagenesis pattern is widespread in human cancers. *Nature Genetics*, 45:970–976, 2013. 1.1

[98] Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray, Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019. 1.1

[99] Roye Rozov, Ron Shamir, and Eran Halperin. Fast lossless compression via cascading bloom filters. *BMC Bioinformatics*, 15:S7 – S7, 2014. 1.2

[100] S. W. Golomb. Nonlinear shift register sequences. In *Shift Register Sequences*, pages 110–168. World Scientific, September 2014. ISBN 978-981-4632-00-3. doi: 10.1142/9789814632010_0006. 2.1.1, 2.1.1

[101] Christopher J Saker and Peter M. Higgins. Unavoidable sets of words of uniform length. *Information and Computation*, 173(2):222–226, 2002. 2.1

[102] Eric E Schadt, Steve Turner, and Andrew Kasarskis. A window into third-generation sequencing. *Human Molecular Genetics*, 19(R2):R227–R240, 2010. 1.1

[103] Saul Schleimer, Daniel S. Wilkerson, and Alex Aiken. Winnowing: Local Algorithms for Document Fingerprinting. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 76–85. ACM, 2003. ISBN 978-1-58113-634-0. doi: 10.1145/872757.872770. 1.4.4, 1.5.1, 2, 2.1, 2.7.1, 2.7.3, 5.3.2

[104] Sydney M Shaffer, Margaret C Dunagin, Stefan R Torborg, Eduardo A Torre, Benjamin Emert, Clemens Krepler, Marilda Beqiri, Katrin Sproesser, Patricia A Brafford, Min Xiao, et al. Rare cell variability and drug-induced reprogramming as a mode of cancer drug resistance. *Nature*, 546(7658):431–435, 2017. 1.1

[105] Sheel Shah, Eric Lubeck, Maayan Schwarzkopf, Ting-Fang He, Alon Greenbaum, Chang Ho Sohn, Antti Lignell, Harry MT Choi, Viviana Gradinaru, Niles A Pierce, et al. Single-molecule RNA detection at depth by hybridization chain reaction and tissue hydrogel embedding and clearing. *Development*, 143(15):2862–2867, 2016. 1.1

[106] Sheel Shah, Eric Lubeck, Wen Zhou, and Long Cai. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron*, 92 (2):342–357, 2016. 1.1

[107] Alex K Shalek, Rahul Satija, Xian Adiconis, Rona S Gertner, Jellert T Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, et al. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498(7453):236–240, 2013. 1.1

[108] John Shalf. The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A*, 378(2166):20190061, 2020. 1.1

[109] Rachel M Sherman and Steven L Salzberg. Pan-genomics in the human genome era. *Nature Reviews Genetics*, 21(4):243–254, 2020. 1.1

[110] Brad Solomon and Carl Kingsford. Fast search of thousands of short-read sequencing experiments. *Nature Biotechnology*, 34(3):300, 2016. 1.2

[111] Brad Solomon and Carl Kingsford. Improved search of large transcriptomic sequencing databases using split sequence bloom trees. *Journal of Computational Biology*, 25(7): 755–765, 2018. 1.2

[112] Jun Sone, Satomi Mitsuhashi, Atsushi Fujita, Takeshi Mizuguchi, Kohei Hamanaka, Keiko Mori, Haruki Koike, Akihiro Hashiguchi, Hiroshi Takashima, Hiroshi Sugiyama, Yutaka Kohno, Yoshihisa Takiyama, Kengo Maeda, Hiroshi Doi, Shigeru Koyano, Hideyuki Takeuchi, Michi Kawamoto, Nobuo Kohara, Tetsuo Ando, Toshiaki Ieda, Yasushi Kita, Norito Kokubun, Yoshio Tsuboi, Kazutaka Katoh, Yoshihiro Kino, Masahisa Katsuno, Yasushi Iwasaki, Mari Yoshida, Fumiaki Tanaka, Ikuo K. Suzuki, Martin C. Frith, Naomichi Matsumoto, and Gen Sobue. Long-read sequencing identifies GGC repeat expansions in NOTCH2NLC associated with neuronal intranuclear inclusion disease. *Nature Genetics*, 51(8):1215–1221, August 2019. ISSN 1546-1718. doi: 10.1038/s41588-019-0459-y. 1.1

[113] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016. 1.1

[114] Michael JT Stubbington, Tapio Lönnberg, Valentina Proserpio, Simon Clare, Anneliese O Speak, Gordon Dougan, and Sarah A Teichmann. T cell fate and clonality inference from single-cell transcriptomes. *Nature Methods*, 13(4):329–332, 2016. 1.1

[115] Peter H Sudmant, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015. 1.1

[116] Chen Sun, Robert S Harris, Rayan Chikhi, and Paul Medvedev. Allsome sequence bloom trees. In *International Conference on Research in Computational Molecular Biology*, pages 272–286. Springer, 2017. 1.2

[117] David Tamborero, Abel Gonzalez-Perez, Christian Perez-Llamas, Jordi Deu-Pons, Cyriac Kandoth, Jüri Reimand, Michael S. Lawrence, Gad Getz, Gary D Bader, Li Ding, and Núria López-Bigas. Comprehensive identification of mutational cancer driver genes across 12 tumor types. *Scientific Reports*, 3, 2013. 1.1

[118] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, et al. mRNA-seq whole-transcriptome analysis of a single cell. *Nature Methods*, 6(5):377–382, 2009. 1.1

[119] Shara Tibken. CES 2019: Moore's Law is dead, says Nvidia's CEO, 2019. URL `https://www.cnet.com/tech/computing/moores-law-is-dead-nvidias-ceo-jensen-huang-says-at-ces-2019/`. 1.1

[120] Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature Biotechnology*, 32(4):381, 2014. 1.1

[121] Erwin L van Dijk, Yan Jaszczyszyn, Delphine Naquin, and Claude Thermes. The third revolution in sequencing technology. *Trends in Genetics*, 34(9):666–681, 2018. 1.1

[122] Sanja Vickovic, Gökcen Eraslan, Fredrik Salmén, Johanna Klughammer, Linnea Stenbeck, Denis Schapiro, Tarmo Äijö, Richard Bonneau, Ludvig Bergenstråhle, José Fernandéz Navarro, et al. High-definition spatial transcriptomics for in situ tissue profiling. *Nature Methods*, 16(10):987–990, 2019. 1.1

[123] Roger Volden, Theron Palmer, Ashley Byrne, Charles Cole, Robert J Schmitz, Richard E Green, and Christopher Vollmers. Improving nanopore read accuracy with the R2C2 method enables the sequencing of highly multiplexed full-length single-cell cDNA. *Proceedings of the National Academy of Sciences*, 115(39):9726–9731, 2018. 1.1

[124] Peter Weiner. Linear pattern matching algorithms. In *SWAT*, 1973. 1.2

[125] Ye Yu, Jinpeng Liu, Xinan Liu, Yi Zhang, Eamonn Magner, Erik Lehnert, Chen Qian, and Jinze Liu. Seqothello: querying rna-seq experiments at scale. *Genome Biology*, 19(1): 1–13, 2018. 1.2

[126] Daniel R Zerbino and Ewan Birney. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Research*, 18(5):821–829, 2008. 1.3.3

[127] Haowen Zhang, Li Song, Xiaotao Wang, Haoyu Cheng, Chenfei Wang, Clifford A Meyer, Tao Liu, Ming Tang, Srinivas Aluru, Feng Yue, et al. Fast alignment and preprocessing of chromatin profiles with Chromap. *Nature communications*, 12(1):1–6, 2021. 1.3.3

[128] Hongyu Zheng, Carl Kingsford, and Guillaume Marçais. Improved design and analysis of practical minimizers. *Bioinformatics*, 36:i119–i127, 07 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa472. 2, 3

[129] Hongyu Zheng, Carl Kingsford, and Guillaume Marçais. Sequence-specific minimizers via polar sets. *Bioinformatics*, 37:i187–i195, 07 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab313. 4

[130] Hongyu Zheng, Carl Kingsford, and Guillaume Marçais. Lower density selection schemes via small universal hitting sets with short remaining path length. *Journal of Computational Biology*, 28(4):395–409, 2021. doi: 10.1089/cmb.2020.0432. 2