


A k-mer-Based Estimator of the Substitution Rate Between Repetitive Sequences

Haonan Wu 

Department of Computer Science and Engineering, The Pennsylvania State University,
University Park, PA, USA

Antonio Blanca[†] 

Department of Computer Science and Engineering, The Pennsylvania State University,
University Park, PA, USA

Paul Medvedev[†] 

Department of Computer Science and Engineering, The Pennsylvania State University,
University Park, PA, USA

Department of Biochemistry and Molecular Biology, The Pennsylvania State University,
University Park, PA, USA

Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA, USA

Abstract

K-mer-based analysis of genomic data is ubiquitous, but the presence of repetitive k-mers continues to pose problems for the accuracy of many methods. For example, the Mash tool (Ondov et al. 2016) can accurately estimate the substitution rate between two low-repetitive sequences from their k-mer sketches; however, it is inaccurate on repetitive sequences such as the centromere of a human chromosome. Follow-up work by Blanca et al. (2021) has attempted to model how mutations affect k-mer sets based on strong assumptions that the sequence is non-repetitive and that mutations do not create spurious k-mer matches. However, the theoretical foundations for extending an estimator like Mash to work in the presence of repeat sequences have been lacking.

In this work, we relax the non-repetitive assumption and propose a novel estimator for the mutation rate. We derive theoretical bounds on our estimator's bias. Our experiments show that it remains accurate for repetitive genomic sequences, such as the alpha satellite higher order repeats in centromeres. We demonstrate our estimator's robustness across diverse datasets and various ranges of the substitution rate and k-mer size. Finally, we show how sketching can be used to avoid dealing with large k-mer sets while retaining accuracy. Our software is available at https://github.com/medvedevgroup/Repeat-Aware_Substitution_Rate_Estimator.

2012 ACM Subject Classification Applied computing → Bioinformatics; Applied computing → Computational biology

Keywords and phrases k-mers, sketching, mutation rates

Digital Object Identifier 10.4230/LIPIcs.WABI.2025.20

Supplementary Material *Software (Source Code):*

https://github.com/medvedevgroup/Repeat-Aware_Substitution_Rate_Estimator [32]
archived at [swh:1:dir:258c949c42d162c56f1e09a0ece39722a5076601](https://www.swh.io/dir/258c949c42d162c56f1e09a0ece39722a5076601)

Funding This material is based upon work supported by the National Science Foundation under Grants No. DBI2138585 and OAC1931531. Research reported in this publication was supported by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM146462. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

[†] The last two authors contributed equally.



Acknowledgements We thank Amatur Rahman for initial work on the project and Qunhua Li and David Koslicki for helpful discussions. We thank Mahmudur Rahman for the helpful discussion about hash functions. We thank Bob Harris for the idea of using dynamic programming to compute the probability of the destruction of all k -spans.

1 Introduction

K -mer-based analysis of genomic data is ubiquitous. e.g. in genome assembly [1], error correction [2], read mapping [13], variant calling [29], genotyping [30, 7], database search [14, 9], metagenomic sequence comparison [26], and alignment-free sequence comparison [28, 20, 24]. One of the major challenges is the presence of repetitive k -mers, which adversely affects the practical performance as well as the theoretical analysis of downstream algorithms. One example is that heuristic read aligners like minimap2 [15] and even more rigorous ones like Eskemap [25] filter out highly repetitive k -mers in order to avoid explosive run times. Another example is the recent paper [27] that proved that sequence alignment can on average be done in almost $\mathcal{O}(n \log n)$ time but could not account for sequences with a high number of repeats.

One of the major advantage of k -mer-based methods is that they lend themselves more easily to sketching [16, 22], which is important for scaling to large-scale data. The groundbreaking Mash paper [20] was able to estimate the mutation rate between two genomes fast enough to be able to construct a phylogeny of 17 primate species in a tiny fraction of the time it would take an alignment-based method. Their approach uses an estimator based on the Jaccard similarity between the k -mer sketches of two sequences. However, the derivation behind their estimator assumes that the genomes have no repeats, making it inaccurate in highly repetitive regions. Other methods for estimating mutation rates are not designed for and/or not tested on highly repetitive sequences [34, 10, 18, 23].

In this paper we tackle the challenge of accounting for repeats when estimating the mutation rate. We assume that a string t is generated from a string s through a simple substitution process [5], where every nucleotide of s mutates with a fixed probability r . Given the number of shared k -mers between s and t and the k -mer abundance histogram of s , we define our estimator \hat{r} as the solution to a polynomial equation, which can be solved using Newton's method. We give a theorem to bound its bias, in terms of properties of s (Theorem 3). Our estimator is designed to capture the most salient properties of the repeat structure of the genome, with the rest of the information being captured in the bias bounds. As a result, a user can decide *a priori* whether to trust our estimator, based on the quality of the bias bounds and on another heuristic we provide (Theorem 4).

We evaluate our estimator \hat{r} empirically on various sequences, including the alpha satellite centromeric region of human chr21 and the highly repetitive human RBMY gene. For such repetitive sequences, our estimator remains highly accurate, while the repeat-oblivious estimator of the kind used by Mash is unreliable. We make a comprehensive evaluation of \hat{r} across the spectrum of k and r values, which can guide a user towards choosing a k value for their analysis. We also show that our estimator can be used on top of a FracMinHash sketch, without systematically effecting the bias. Our software is available on GitHub [32].

2 Preliminaries

Let s be a string and let $k > 0$ be a parameter indicating the k -mer size. We will index string positions from 1. We further assume in this paper that s is at least k nucleotides long. We use L to denote the number of nucleotides in the string minus $(k - 1)$, or, equivalently,

the number of k -mers in s . For $1 \leq i \leq L$, let s_i be the k -mer starting at position i of s . Let $sp^k(s)$ be the set of all distinct k -mers in s , also called the k -spectrum of s . We let L_0 be the size of $sp^k(s)$, i.e. the number of distinct k -mers in s . Given a k -mer τ , we will use the shorthand $\tau \in s$ to mean $\tau \in sp^k(s)$. Given two strings s and t , we define $I(s, t) \triangleq |sp^k(s) \cap sp^k(t)|$ as the number of k -mers shared between them. We will usually use the shorthand of I for $I(s, t)$. Given two k -mers τ and ν , we use $\text{HD}(\tau, \nu)$ to denote their Hamming distance.

Let K be a set of k -mers and let s be a string. We let $\text{occ}(K)$ denote the number of positions i in s such that $s_i \in K$. When K consists of a single element τ , we simply write $\text{occ}(\tau)$. A set of positions \mathcal{J} is said to be a *set of occurrences* of K if for all $i \in \mathcal{J}$, we have $s_i \in K$. A set of occurrences is said to be *non-overlapping* if, for all distinct $i, j \in \mathcal{J}$, $|j - i| \geq k$. We let $\text{sep}(K)$ be the maximum size of a set of non-overlapping occurrences of K , also referred to as the *separated occurrence count*. Observe that $0 \leq \text{sep}(K) \leq \text{occ}(K)$. The *abundance histogram* of a string s is the sequence (a_1, \dots, a_L) where a_i is the number of k -mers in $sp^k(s)$ that occur i times in s . Note that $L_0 = \sum_{i=1}^L a_i$.

We will consider the following random *substitution process*, parameterized by a rate $0 \leq r \leq 1$. Given a string s , it generates an equal-length string where, independently, the character at each position is unchanged from s with probability $1 - r$ and changed to one of the three other nucleotides with probability $r/3$.

3 Problem overview and proposed solution

In this paper, we address the following problem. Let $0 \leq r \leq 1$ be a substitution rate. Let s be a string and let t be generated from s using the substitution process parametrized by r . Let $I_{\text{obs}} = I(s, t)$ be the observed spectrum intersection size. Given I_{obs} and the abundance histogram of s , the problem is to estimate the mutation rate r .

The *bias* of an estimator \hat{r} for r is defined as $\mathbb{E}[\hat{r}] - r$. A good estimator should have a small absolute bias, one that is within the error tolerance of downstream applications. For our problem, directly finding an estimator for r with provably small bias turned out to be technically challenging. Instead, we provide an estimator for $q \triangleq 1 - (1 - r)^k$, which corresponds to the probability that a k -mer occurrence contains at least one substitution. There is a natural one-to-one correspondence between an estimator \hat{q} of q and an estimator \hat{r} of r via the equation $\hat{q} = 1 - (1 - \hat{r})^k$. Thus, an alternative to bounding the bias of \hat{r} is to bound that of \hat{q} ; i.e., bound $\mathbb{E}[\hat{q}] - q = \mathbb{E}[1 - (1 - \hat{r})^k] - (1 - (1 - r)^k)$. While the difference between the two approaches may intuitively seem minor, bounding the bias of \hat{q} turned out to be more technically feasible.

The only previously known estimator for this problem is what we refer to as the *repeat-oblivious* estimator¹ and denote by r_{obl} . The derivation of this estimator assumes that 1) s has no repeats and 2) if a k -mer mutates, it never mutates to anything that is already in s . The estimator is then derived using a simple two step technique, called the method of moments [6]. The first step is to derive $\mathbb{E}[I]$ in terms of r . Under the assumptions (1) and (2), $\mathbb{E}[I] = L(1 - r)^k$. The second step is to take the observed value I_{obs} , plug it in place of $\mathbb{E}[I]$, and solve for r . In this case, one solves the equation $I_{\text{obs}} = L(1 - r)^k$ and gets the estimator $r_{\text{obl}} = 1 - \left(\frac{I_{\text{obs}}}{L}\right)^{1/k}$ and the corresponding estimator $q_{\text{obl}} = 1 - \frac{I_{\text{obs}}}{L}$. The

¹ What we describe is based on the estimators used in [8, 20, 12], but with two important differences. The first is that we use the modification adopted in the follow up work of [24] and described in Appendix A.6 of [3]. The second is that the original estimator was calculated from the Jaccard similarity between two sequences; however, under our substitution process model, we can state more simply in terms of the spectrum intersection size.

estimator q_{obl} is an unbiased estimator of q when (1) and (2) hold, but the bias for general sequences is not known. We are also not aware of any results about the bias of r_{obl} , even under assumptions (1) and (2). As we will show in Section 6, the repeat-oblivious estimator has a large empirical bias when the assumptions are substantially violated.

On the one hand, the repeat-oblivious estimator does not at all account for the repeat structure of s . On the other hand, an estimator that would fully account for it seems to be challenging to derive, analyze, and compute. Moreover, such an estimator would likely be superfluous for real data. Instead, our approach is intended to achieve a middle ground between accuracy and complexity by accounting for the most essential part of the repeat structure in the estimator and expressing the non-captured structure in the bias formula. We will show that under assumption (2) and the assumption that all k -mer occurrences are non-overlapping in s ,

$$\mathbb{E}[I] \approx L_0 - \sum_{i=1}^L a_i (1 - (1 - r)^k)^i. \quad (1)$$

Following the method of moments approach, we define our estimator \hat{r} as the unique solution (the uniqueness is shown in Lemma A.2) to this equation when plugging in I_{obs} in place of $\mathbb{E}[I]$. Though we are not able to analytically solve for \hat{r} , we can find the solution numerically using Newton's Method.

Note that the assumptions we make are not necessary to compute \hat{r} and only represent the ideal condition for our estimator. Our theoretical and experimental results will quantify more precisely how the deviation from our assumptions is reflected in the bias.

4 Estimator bias

Recall that we define $\hat{q} = 1 - (1 - \hat{r})^k$ and, as mentioned earlier, we will prove the theoretical results on the bias of \hat{q} , rather than \hat{r} . First, we need to derive the expectation and variance of the intersection size. A closed-form expression for even the expectation is elusive, so we will instead use an approximation and derive bounds on the error. The idea behind our bounds is that the error becomes small on the types of sequences that occur in biological data.

We want to underscore that when we make probabilistic statements, it is with regard to the substitution process and not with regard to s . We do not make any assumptions about s , and, in particular, we are not considering the situation where s itself is generated randomly.

First, it is useful to express $I \triangleq I(s, t)$ as a sum of indicator random variables. Let us define E_i^τ as event that $t_i = \tau$ and $E^\tau = \cup_{i=1}^L E_i^\tau$ as the event that at least one position in t contains τ . By linearity of expectation, we have

$$\mathbb{E}[I] = \sum_{\tau \in s} \Pr[E^\tau] = \sum_{\tau \in s} \Pr \left[\cup_{i=1}^L E_i^\tau \right].$$

Let $\mathcal{F}(q) \triangleq L_0 - \sum_{i=1}^L a_i q^i$. Relying on the approximation $\mathbb{E}[I] \approx \mathcal{F}(q)$ (i.e. Equation (1)), we define \hat{q} as the solution to $I_{\text{obs}} = \mathcal{F}(q)$, or, equivalently, $\hat{q} = \mathcal{F}^{-1}(I_{\text{obs}})$. We show that this approximation holds when we assume that 1) $\Pr[E_i^\tau] = 0$ when $s_i \neq \tau$ (see footnote²) and 2) all occurrences of τ are non-overlapping in s :

² We note that this assumption is not theoretically precise, because forbidding a k -mer at position i from mutating to $\tau \in s$ usually implies that there is at least one $\nu \notin s$ that the k -mer at position $i + 1$ can no longer mutate to. Because of these dependencies, there are downstream effects on the probability space that are complex to track. A theoretically robust alternative was given in [5] via the k -span formulation of the problem. It could be used to formalize the assumption here, however, in this paper, we only use the assumption to give intuition for the estimator and do not use it in any formal theorem statements or proofs.

$$\begin{aligned}
\mathbb{E}[I] &= \sum_{\tau \in s} \Pr[\cup_{i=1}^L E_i^\tau] \\
&\approx \sum_{\tau \in s} \Pr[\cup_{i:s_i=\tau} E_i^\tau] && \text{(because of (1))} \\
&= \sum_{\tau \in s} (1 - \Pr[\cap_{i:s_i=\tau} \neg E_i^\tau]) \\
&\approx \sum_{\tau \in s} (1 - \prod_{i:s_i=\tau} \Pr[\neg E_i^\tau]) && \text{(because of (2))} \\
&= \sum_{\tau \in s} (1 - q^{occ(\tau)}) \\
&= L_0 - \sum_{i=1}^L a_i q^i \\
&= \mathcal{F}(q)
\end{aligned}$$

The underlying philosophy for our estimator is that while these assumptions are not perfectly satisfied on real data, in most cases the contribution due to violations of these assumptions is small. To make this mathematically precise, we will bound the difference between $\mathbb{E}[I]$ and $\mathcal{F}(q)$ in terms of an expression that can be calculated for any s .

► **Theorem 1.** *We have that $L_E \leq \mathbb{E}[I] \leq U_E$, where*

$$\begin{aligned}
L_E &\triangleq \sum_{\tau \in s} 1 - q^{sep(\tau)}, \\
U_E &\triangleq \sum_{\tau \in s} 1 - q^{occ(\tau)} + \beta_\tau, \text{ where} \\
\beta_\tau &\triangleq \min \left\{ \sum_{\substack{i=1 \\ s_i \neq \tau}}^L (1-r)^{k-HD(s_i, \tau)} (r/3)^{HD(s_i, \tau)}, q^{sep(\tau)} \right\}.
\end{aligned}$$

The difference between $\mathcal{F}(q)$ and L_E (i.e. $\sum_{\tau \in s} q^{sep(\tau)} - q^{occ(\tau)}$) is close to 0 when the number of k -mers with overlapping occurrences is close to 0. On the other hand, the difference between $\mathcal{F}(q)$ and U_E (i.e. $\sum_{\tau \in s} \beta_\tau$) is never zero (except in corner cases). However, the largest terms contributing to this difference are due to pairs of non-identical k -mers that have a small Hamming distance to each other. Thus, the difference becomes small when the number of “near-repeats” is small.

Next, we upper bound the variance.

► **Theorem 2.** *We have that*

$$\begin{aligned}
\text{Var}[I] &\leq L_0 - \mathbb{E}[I] - (L_0 - \mathbb{E}[I])^2 + \sum_{\tau \in s} \sum_{\substack{v \in s \\ v \neq \tau}} q^{sep(\tau, v)} \\
&\leq U_{Var},
\end{aligned}$$

where

$$U_{Var} = \sum_{\tau \in s} \sum_{v \neq \tau} q^{sep(\{\tau, v\})} + \begin{cases} L_0 - U_E - (L_0 - U_E)^2 & \text{if } L_0 - U_E \geq 1/2; \\ 1/4. & \text{otherwise.} \end{cases}$$

Theorem 2 gives an upper bound on $\text{Var}[I]$ in two forms. The first one is more precise because it is a function of $\mathbb{E}[I]$. However, since we are not able to compute $\mathbb{E}[I]$ exactly, the second form allows us to plug in the upper bound on $\mathbb{E}[I]$ from Theorem 1.

Given the bounds on $\mathbb{E}[I]$ and $\text{Var}[I]$, we are now able to bound the bias of \hat{q} .

► **Theorem 3.** *Let s be a sequence with at least one k -mer that occurs exactly once. The bias of \hat{q} is $\mathbb{E}[\hat{q}] - q$ where $\mathbb{E}[\hat{q}]$ is bounded as*

$$\begin{aligned} \mathbb{E}[\hat{q}] &\geq f(U_E) - \text{Var}[I] \left(\frac{\mathcal{F}''(f(U_E))}{2(\mathcal{F}'(f(L_E)))^3} + \alpha \right) \geq f(U_E) - U_{\text{Var}} \left(\frac{\mathcal{F}''(f(U_E))}{2(\mathcal{F}'(f(L_E)))^3} + \alpha \right) \\ \mathbb{E}[\hat{q}] &\leq f(L_E) \end{aligned}$$

and where $f \triangleq \mathcal{F}^{-1}$ and $\alpha = \max\{L_0 - L_E, U_E\} \cdot \max_{x \in (0,1)} \left| \frac{1}{6} \frac{\mathcal{F}'''(x)\mathcal{F}'(x) - 3(\mathcal{F}''(x))^2}{-(\mathcal{F}'(x))^5} \right|$.

The derivatives of $\mathcal{F}(q)$ have straightforward closed-form expressions, since \mathcal{F} is a polynomial in q . We do not have a closed-form solution for f , but it can be evaluated numerically using Newton's method. Thus, for any given sequence s , we can precompute the bounds of our \hat{q} estimator bias for any value of q . Due to space limitations, we do not further elaborate on the algorithm to compute the bounds in Theorem 3 or on its runtime analysis.

When the observed intersection is empty, there is a loss of signal and it becomes challenging for any intersection-based estimator to differentiate the true substitution rate from 100%. The following theorem gives an upper bound on the probability that the intersection is empty, as a function of L, k , and r . In Section 6, we will show how it can be used to make a conservative decision that the computed estimate is unreliable.

► **Theorem 4.** *Let s be a string of length at least k . The probability that every interval of length k in $s[1..i + k - 1]$ has at least one substitution can be computed in $\Theta(ik)$ time with a dynamic programming algorithm that takes as input only L, r, k (not s).*

5 Proofs

This section contains the proofs of our theoretical results. In particular, we will prove Theorems 1–3 from the previous section. The proof of Theorem 4 is left for the Appendix. We start by proving a couple of preliminary facts that will be used in the proofs of these theorems. First, we consider the probability of the event E_i^τ , which is straightforward to derive.

► **Lemma 5.** *For all τ , $\Pr[E_i^\tau] = (1 - r)^{k - \text{HD}(s_i, \tau)} (r/3)^{\text{HD}(s_i, \tau)}$.*

Proof. In order for s_i to be equal to τ after the mutation process, exactly $k - \text{HD}(s_i, \tau)$ positions must remain unmutated (which happens with probability $(1 - r)^{k - d}$) and exactly $\text{HD}(s_i, \tau)$ positions must mutate to the needed nucleotide (which happens with probability $(r/3)^{\text{HD}(s_i, \tau)}$). ◀

Next, we will bound the probability that all the occurrences of a k -mer become mutated; i.e. a k -mer does not survive the mutation process.

► **Lemma 6.** *Let τ be a k -mer with occurrence locations denoted by $p_1 < \dots < p_{\text{occ}(\tau)}$. For all $2 \leq \ell \leq \text{occ}(\tau)$,*

1. $\Pr[\neg E_{p_\ell}^\tau \mid \cap_{i=1}^{\ell-1} \neg E_{p_i}^\tau] \geq q$, and
2. $\Pr[\cap_{i=1}^\ell \neg E_{p_i}^\tau] \geq q^\ell$.

Proof. We drop τ from the notation since it remains constant throughout the proof. We first prove the first statement of the lemma. Let us consider the intervals associated with E_{p_ℓ} and $E_{p_{\ell-1}}$, denoted by $[p_\ell, p_\ell + k - 1]$ and $[p_{\ell-1}, p_{\ell-1} + k - 1]$, respectively. If these intervals are disjoint, then we are done. Otherwise, the union of these intervals can be partitioned into three regions: 1) the part of the interval of $E_{p_{\ell-1}}$ that does not intersect with the interval of E_{p_ℓ} , 2) the intersection of the two intervals, and 3) the part of the interval of E_{p_ℓ} that does not intersect with the interval of $E_{p_{\ell-1}}$. We denote the lengths of these intervals as a , b , and c , respectively, and we denote the event that no mutation occurs in the intervals as A , B , and C , respectively. Let $X = \cap_{i=1}^{\ell-1} \neg E_{p_i}$, i.e. we need to calculate $\Pr[\neg E_{p_\ell} | X]$. First, we reduce the calculation to $\Pr[B | X]$ as follows:

$$\Pr[E_{p_\ell} | X] = \Pr[B, C | X] = \Pr[B | C, X] \Pr[C | X] = \Pr[B | X] \Pr[C] = \Pr[B | X] (1 - r)^c. \quad (2)$$

Next, to calculate $\Pr[B | X]$, we proceed by conditioning on A :

$$\Pr[B | X] = \Pr[B | A, X] \Pr[A | X] + \Pr[B | \neg A, X] \Pr[\neg A | X] \leq \Pr[B | A, X] + \Pr[B | \neg A, X].$$

First note that

$$A \cap X \implies A \cap \neg E_{p_{\ell-1}} \iff A \cap (\neg A \cup \neg B) \iff A \cap \neg B \implies \neg B,$$

and so $\Pr[B | A, X] = 0$. To bound $\Pr[B | \neg A, X]$, consider all the intervals E_{p_i} , for $i < \ell$, that intersect with B 's interval. Formally, let $\mathcal{J} = \{i < \ell \mid E_{p_i} \text{ intersects } B\text{'s interval}\}$. Note that all intervals indexed by \mathcal{J} necessarily contain A 's interval. Therefore, the event $\neg A$ implies $\cap_{i \in \mathcal{J}} \neg E_{p_i}$. We can now write

$$\Pr[B | \neg A, X] = \Pr[B | \neg A, \cap_{i \in \mathcal{J}} \neg E_{p_i}] = \Pr[B | \neg A] = \Pr[B] = (1 - r)^b.$$

Therefore, $\Pr[B | X] \leq (1 - r)^b$ and plugging this bound into Equation (2), we get the first statement of the lemma.

To prove the second statement of the lemma, we apply the chain rule together with the first statement:

$$\Pr[\cap_{i=1}^{\ell} \neg E_{p_i}] = \Pr[\neg E_{p_1}] \prod_{i=2}^{\ell} \Pr[\neg E_{p_i} | \neg E_{p_1}, \dots, \neg E_{p_{i-1}}] \geq q^\ell \quad \blacktriangleleft$$

We can now prove Theorem 1:

Proof of Theorem 1. It suffices to prove that for every k-mer $\tau \in s$, it holds that $1 - q^{sep(\tau)} \leq \Pr[\cup_{i=1}^L E_i^\tau] \leq 1 - q^{occ(\tau)} + \beta_\tau$. For the lower bound, let \mathcal{J} be a non-overlapping set of occurrences of τ of size $sep(\tau)$. Then we have

$$\begin{aligned} \Pr[\cup_{i=1}^L E_i^\tau] &\geq \Pr[\cup_{i:s_i=\tau} E_i^\tau] = 1 - \Pr[\cap_{i:s_i=\tau} \neg E_i^\tau] \geq 1 - \Pr[\cap_{i \in \mathcal{J}} \neg E_i^\tau] \\ &= 1 - \prod_{i \in \mathcal{J}} \Pr[\neg E_i^\tau] = 1 - \prod_{i \in \mathcal{J}} q = 1 - q^{sep(\tau)}, \end{aligned} \quad (3)$$

where we use the independence of the events $\{\neg E_i^\tau\}$ when they are non-overlapping. For the upper bound, let $A = \cup_{i:s_i=\tau} E_i^\tau$ and let $B = \cup_{i:s_i \neq \tau} E_i^\tau$. Then, by Lemma 6,

$$\Pr[\cup_{i=1}^L E_i^\tau] = \Pr[A \cup B] = \Pr[A] + \Pr[B \cap \neg A] \leq 1 - q^{occ(\tau)} + \Pr[B \cap \neg A]$$

To bound $\Pr[B \cap \neg A]$ observe that $\Pr[B \cap \neg A] \leq \min(\Pr[B], \Pr[\neg A])$, and by Lemma 5:

$$\Pr[B] \leq \sum_{i:s_i \neq \tau} \Pr[E_i^\tau] = \sum_{i:s_i \neq \tau} (1 - r)^k \left(\frac{r}{3(1 - r)} \right)^{HD(s_i, \tau)}.$$

Moreover, by Equation (3), $\Pr[\neg A] = 1 - \Pr[A] \leq q^{sep(\tau)}$ and the result follows. \blacktriangleleft

The proof of the variance bound is more straightforward:

Proof of Theorem 2. Since I is a sum of indicator random variables (i.e. $I = \sum_{\tau \in s} E^\tau$), we can write the variance as

$$\text{Var}[I] = (L_0 - \mathbb{E}[I]) - (L_0 - \mathbb{E}[I])^2 + \sum_{\tau \in s} \sum_{\substack{v \in s \\ v \neq \tau}} \Pr[\neg E^v, \neg E^\tau];$$

for completeness we include a proof of this fact in the appendix (Lemma A.1).

Consider some $\tau \neq v$ and let \mathcal{J} be a non-overlapping set of occurrences of $\{\tau, v\}$. Let $\mathcal{J}^\tau \subseteq \mathcal{J}$ be the positions where τ occurs and let $\mathcal{J}^v \subseteq \mathcal{J}$ be the positions where v occurs. Then,

$$\Pr[\neg E^\tau, \neg E^v] \leq \Pr[\cap_{i \in \mathcal{J}^\tau} \neg E_i^\tau, \cap_{i \in \mathcal{J}^v} \neg E_i^v] = \prod_{i \in \mathcal{J}^\tau} \Pr[\neg E_i^\tau] \cdot \prod_{i \in \mathcal{J}^v} \Pr[\neg E_i^v] = q^{\text{sep}(\{\tau, v\})}.$$

This gives the first form of the upper bound on the variance. The U_{Var} upper bound is derived from the fact that $f(x) = x - x^2$ is monotonically increasing on $[0, 1/2)$ and decreasing on $[1/2, \infty)$. Therefore, the maximum of $1/4$ is achieved at $x = 1/2$. ◀

Proof of Theorem 3. In Lemma A.2 in the Appendix, we show that f is well-defined. We will only consider f on the interval $[\mathcal{F}(1), \mathcal{F}(0)]$. Throughout the proof, we will rely on the facts that 1) on the interval $q \in [0, 1]$, $\mathcal{F}'(q) < 0$, $\mathcal{F}''(q) \leq 0$, $\mathcal{F}'''(q) \leq 0$; 2) for $y \in [\mathcal{F}(1), \mathcal{F}(0)]$, $f'(y) < 0$ and $f''(y) \leq 0$; 3) the first three derivatives of f can be expressed in terms of f and the derivatives of \mathcal{F} . These properties follow by basic calculus and are stated formally in Lemma A.2. Recall that $\mathbb{E}[\hat{q}] = \mathbb{E}[f(I)]$. To get the upper bound, we use the fact that f is decreasing and concave. We apply Jensen's inequality followed by Theorem 1 to get that $\mathbb{E}[f(I)] \leq f(\mathbb{E}[I]) \leq f(L_E)$.

For the lower bound, since we cannot analytically derive $f(I)$, we derive a reverse Jensen inequality using the Taylor expansion of f around $\mathbb{E}[I]$. Specifically, using the Lagrange Remainder, we know that there exists some ξ_I between I and $\mathbb{E}[I]$ such that

$$f(I) = f(\mathbb{E}[I]) + f'(\mathbb{E}[I])(I - \mathbb{E}[I]) + \frac{1}{2}f''(\mathbb{E}[I])(I - \mathbb{E}[I])^2 + \frac{1}{6}f'''(\xi_I)(I - \mathbb{E}[I])^3.$$

Since we are interested in the expected value of $f(I)$, we take expectations on both sides:

$$\mathbb{E}[f(I)] = f(\mathbb{E}[I]) + \frac{1}{2}f''(\mathbb{E}[I])\text{Var}[I] + \mathbb{E}\left[\frac{1}{6}f'''(\xi_I)(I - \mathbb{E}[I])^3\right].$$

We will bound the terms separately by writing $\mathbb{E}[f(I)] \geq T_1 + T_2 - T_3 \cdot \max_{y \in [F(1), F(0)]} T_4$ with $T_1 = f(\mathbb{E}[I])$, $T_2 = \frac{1}{2}f''(\mathbb{E}[I])\text{Var}[I]$, $T_3 = \mathbb{E}[|I - \mathbb{E}[I]|^3]$, and $T_4 = \frac{1}{6}|f'''(y)|$. For the first term, we use the fact that f is decreasing and apply Theorem 1 to get that $f(\mathbb{E}[I]) \geq f(U_E)$. For the second term T_2 , we first plug in the second derivative of f and then apply monotonicity properties together with Theorem 1 to get

$$T_2 = \frac{-\mathcal{F}''(f(\mathbb{E}[I]))}{2(\mathcal{F}'(f(\mathbb{E}[I])))^3} \text{Var}[I] \geq \frac{-\mathcal{F}''(f(U_E))}{2(\mathcal{F}'(f(L_E)))^3} \text{Var}[I].$$

For T_3 , we use the fact that $I \leq L_0$, which implies that $|I - \mathbb{E}[I]| \leq \max(L_0 - \mathbb{E}[I], \mathbb{E}[I])$, and thus

$$T_3 = \mathbb{E}[|I - \mathbb{E}[I]|(I - \mathbb{E}[I])^2] \leq \max(L_0 - \mathbb{E}[I], \mathbb{E}[I])\text{Var}[I] \leq \max(L_0 - L_E, U_E)\text{Var}[I].$$

For T_4 ,

$$\begin{aligned} \max_{y \in [\mathcal{F}(1), \mathcal{F}(0)]} T_4 &\leq \max_{y \in [\mathcal{F}(1), \mathcal{F}(0)]} \left| \frac{1}{6} \frac{\mathcal{F}'''(f(y))\mathcal{F}'(f(y)) - 3(\mathcal{F}''(f(y)))^2}{-(\mathcal{F}'(f(y)))^5} \right| \\ &\leq \max_{x \in [0,1]} \left| \frac{1}{6} \frac{\mathcal{F}'''(x)\mathcal{F}'(x) - 3(\mathcal{F}''(x))^2}{-(\mathcal{F}'(x))^5} \right| \end{aligned}$$

6 Experimental results

In this section, we evaluate the empirical accuracy and robustness of our estimator, used by itself or in combination with sketching.

6.1 Datasets

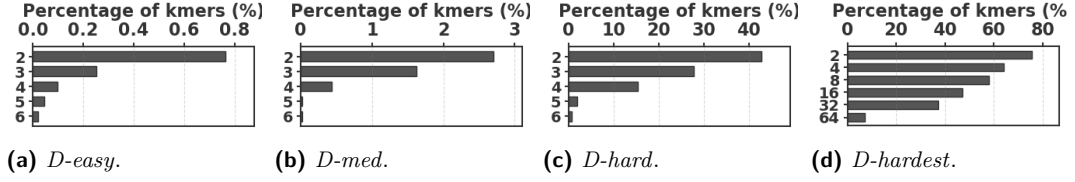
To evaluate our estimator, we use four sequences to capture various degrees of repetitiveness. The sequences are extracted from the human T2T-CHM13v2.0 reference [19]. The sequences and their coordinates are available at our reproducibility GitHub page [33]. Table 1 shows properties of these sequences and Figure 1 shows their cumulative abundance histograms.

1. *D-easy*: This is an arbitrarily chosen substring from chr6, which had no unusual repeat annotations. We set $k = 20$ for this sequence, which is similar to what was used in the Mash paper [20]. Less than 1% of the k -mers are non-singletons.
2. *D-med*: This is the sequence of RBMY1A1, a chrY gene that is composed of ALUs, SINEs, LINEs, simple repeats, and other repeat elements [Fig 2C in [21]]. We also use $k = 20$ for this sequence. Approximately 3% of k -mers are non-singletons.
3. *D-hard*: This is a subsequence of RBMY1A that is annotated as a simple repeat by Tandem Repeats Finder [4], containing 4.2 similar copies of a repeat unit of length 545nt. We use $k = 10$, which is large enough to avoid spurious repeats in this short sequence but small enough to capture its repetitive structure. More than 40% of the k -mers are non-singletons.
4. *D-hardest*: This is a subsequence (100k long) of a region that is annotated as ‘Active α Sat HOR’ in the chr21 centromere. The location of the subsequence within the region is arbitrary. Alpha satellite (α Sat) DNA consists of 171-bp monomers arranged into higher-order repeats, and is notoriously difficult to assemble or map to [17]. We use $k = 30$ for this sequence, as a user dealing with such a sequence is likely to choose a higher k value. Over 70% of the k -mers are non-singletons.

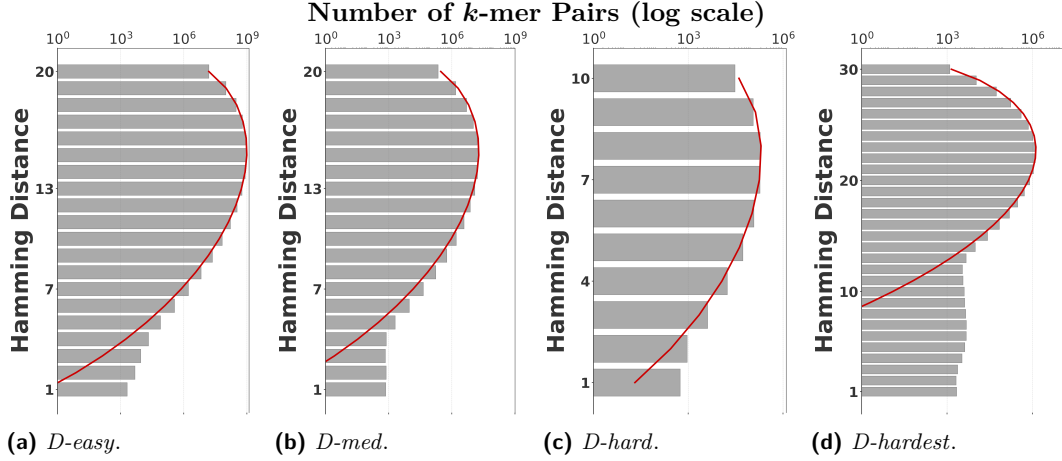
Table 1 Sequence properties of our four experimental datasets. A k -mer τ is *overlapping* if it overlaps itself at least once in the sequence, i.e. $sep(\tau) < occ(\tau)$.

Name	Default k	N. k -mers (L)	N. distinct k -mers (L_0)	N. of overlapping k -mers	Biological significance
<i>D-easy</i>	20	100,000	98,786	15	arbitrary region
<i>D-med</i>	20	14,400	13,727	2	RBMYA1 gene
<i>D-hard</i>	10	2,264	1,199	0	simple repeat
<i>D-hardest</i>	30	100,000	3,987	0	centromere

Before proceeding with experiments, we assess the validity of the two approximations made in the derivation of our estimator. The first approximation is ignoring the dependency between overlapping occurrences of a k -mer. The k -mers where this happens, i.e. k -mers τ where $sep(\tau) < occ(\tau)$, contribute to inaccuracy. As shown in Table 1, this is exceedingly rare. The second approximation is ignoring the possibility that a k -mer mutates to another



■ **Figure 1** Cumulative abundance histograms of our datasets. Each row labeled y shows the percentage of k -mers which occur at least y times.



■ **Figure 2** The distribution of all-vs-all k -mer Hamming distances. The theoretical Hamming distance distributions between random k -mers are shown in the red curves.

k -mer in the spectrum. K -mer pairs in s that have a low Hamming distance will contribute to the bias. Figure 2 shows the distribution of all-vs-all pairwise k -mer Hamming distances. The D -hard and D -hardest datasets indeed have a large amount of “near-repeat” k -mers, which should make these datasets challenging for our estimator.

6.2 Comparison of our estimator to the repeat-oblivious estimator

Figure 3 shows the performance on a range of substitution rates, $r \in (0.1\%, 33\%)$. For D -hard and D -hardest, our estimator has a high accuracy (within a few percent of the true value), in the range of around $r \in (0.1\%, 24\%)$. The r_{obl} estimator, on the other hand, has a much smaller reliability range, e.g. $r \in (10\%, 24\%)$ in D -hardest. For example, when the substitution rate is $r = 1.1\%$, the average of r_{obl} is 10.4%, while the average of \hat{r} is 1.1%. For $r > 24\%$, the observed intersection size was frequently 0; both estimators estimate $r = 100\%$ at this point, making them unstable. For D -easy and D -med, the performance of r_{obl} is nearly as good as our estimator, except at very low values of r (e.g. r_{obl} has a 230% relative error at $r = 0.1\%$ on D -med).

Figure 4 evaluates the estimators on D -hardest while fixing $r = 1\%$ and varying k . For $k \geq 690$, both estimators become unstable (not shown in figure); similar to the case of high substitution rates, the observed intersection size was frequently 0. For smaller k , our estimator performed much better than r_{obl} , e.g. for $k = 32$, the average r_{obl} was 10%.

The relative performance of the two estimators can be explained algebraically. The r_{obl} estimator is derived using the approximation that the probability that a k -mer τ from s remains after substitutions as $\text{occ}(\tau)(1 - q)$. Our estimator uses the approximation that τ remains as $1 - q^{\text{occ}(\tau)}$. For singleton k -mers, these probabilities are equal, but for repetitive

sequences, the effect of $occ(\tau) > 1$ cannot be neglected; therefore, r_{obl} gets progressively worse as the datasets become more repetitive. Furthermore, $occ(\tau, s)(1 - q) > 1 - q^{occ(\tau, s)}$ on $q \in (0, 1)$. Consequently, r_{obl} tends to be higher than \hat{r} . The difference between $1 - q^{occ(\tau, s)}$ and $occ(\tau, s)(1 - q)$ increases as q decreases. Hence, the gap between r_{obl} and \hat{r} is larger for smaller r and smaller k , as Figures 3 and 4 show. Finally, as q approaches 1, the probability of an empty intersection becomes greater, leading all estimators to output 1. This explains the pattern for large r in Figure 3d.

6.3 Combination with sketching

Sketching is a powerful technique that can make it possible to quickly compute all-pairs estimates on large datasets [20]. Our estimator lends itself to being applied on the sketched (rather than full) intersection, as follows. Given a threshold $0 < \theta < 1$, one can use a hash function to uniformly map each k -mer to a real number in $(0, 1)$. A *FracMinHash sketch* of a sequence s is defined as the subset of the k -spectrum of s that hashes below θ [11]. In this way, one can compute I^θ , the size of the intersection between the sketches of s and t .

Recall that our estimator is defined by finding the unique value r to solve $I_{obs} = L_0 - \sum_{i=1}^L a_i(1 - (1 - r)^k)^i$, where I_{obs} is the size of the observed (non-sketched) intersection. It is easy to show that the expected value of I^θ over the sketching process is θI . A natural extension is then to find the unique value of r to solve $\frac{I_{obs}^\theta}{\theta} = L_0 - \sum_{i=1}^L a_i(1 - (1 - r)^k)^i$. The only caveat is that in some rare cases for very low mutation rates, $\frac{I_{obs}^\theta}{\theta}$ may exceed L_0 and result in a lack of unique solution; in such cases, we hard code the estimator to return 0.

Figure 5a shows the accuracy of the resulting estimator on *D-hardest*, averaged over the combined replicates of the substitution and sketching process. The sketching does not introduce any systematic bias, but, as expected, increases the variance of our estimator. The variance is larger for smaller θ values. These results indicate that our estimator can indeed be applied to FracMinHashed sequences, with the threshold parameter θ controlling the trade-off between sketch size and the estimator's variance.

Figure 5b evaluates the isolated impact of the sketching process for a fixed string t , which better reflects the typical user scenario. For each substitution rate r , we generate a single mutated string t and compute the \hat{r} estimate based on the non-sketched intersection. We then replicate the sketching procedure for s and t and compare the distribution of the sketched estimator to the value of the non-sketched estimate (shown as red bar). The results demonstrate that sketching can accelerate the estimation process, at the cost of introducing controlled variance in the estimates.

6.4 Accuracy as a combined function of k and r

The accuracy of our estimator \hat{r} ultimately depends on an intricate interplay between k and r . A smaller k increases the number of repeats, making estimation more challenging. On the other hand, as r or k increases, the probability $q = 1 - (1 - r)^k$ increases, leading to a higher chance of an empty intersection size and an unreliable estimator. To more thoroughly explore the space of all values, Figure 6 evaluates the average relative absolute error, defined as $\frac{1}{n} \sum_{i=1}^n \frac{|\hat{r}_i - r|}{r}$, over a wide range of r and k . This combines our estimator's empirical bias and variance, indicating the parameter ranges at which our estimator is reliable.

We note that a user is usually able to choose k but not r . For r , they typically have only a rough range on what it might be. For instance, substitution rates of more than 25% are unlikely for biologically functional sequences. Therefore, choosing a k boils down to choosing a column from the heatmap that is good for the desired r range. Figure 6 shows that choosing a k in the range of 10 to 20 would work well for all of our datasets.

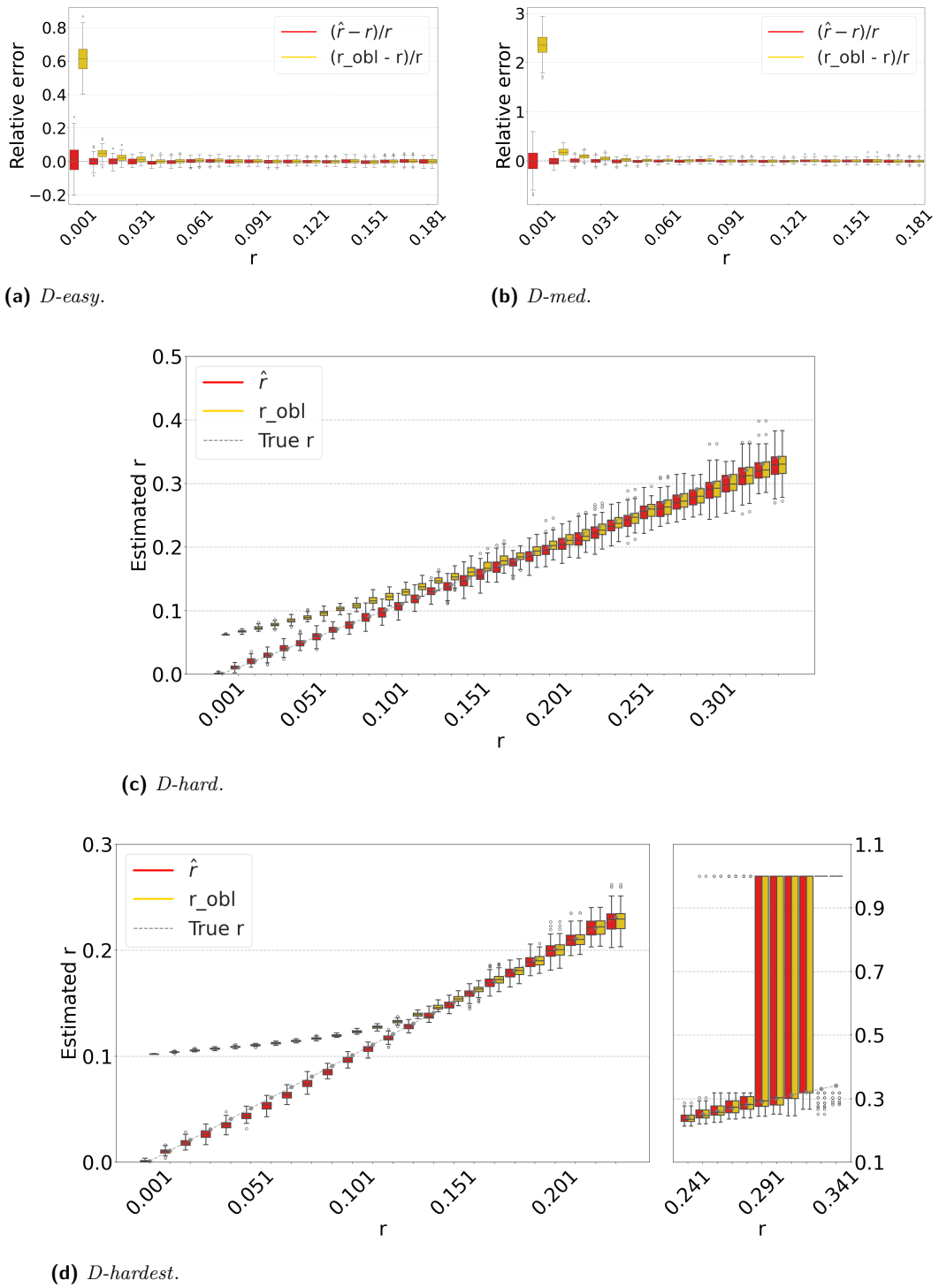


Figure 3 Comparison of our estimator \hat{r} with r_{obl} . For each r value, we simulate the random substitution process 100 times and show the box plot of the resulting estimates. For D -easy and D -med, the y-axis shows the relative error. For D -hard and D -hardest, the y-axis shows the actual estimator value instead, in order to reflect the bigger scale of the differences. For D -easy and D -med, the plots follow the same pattern if they were to be extended rightwards up to $r = 33\%$.

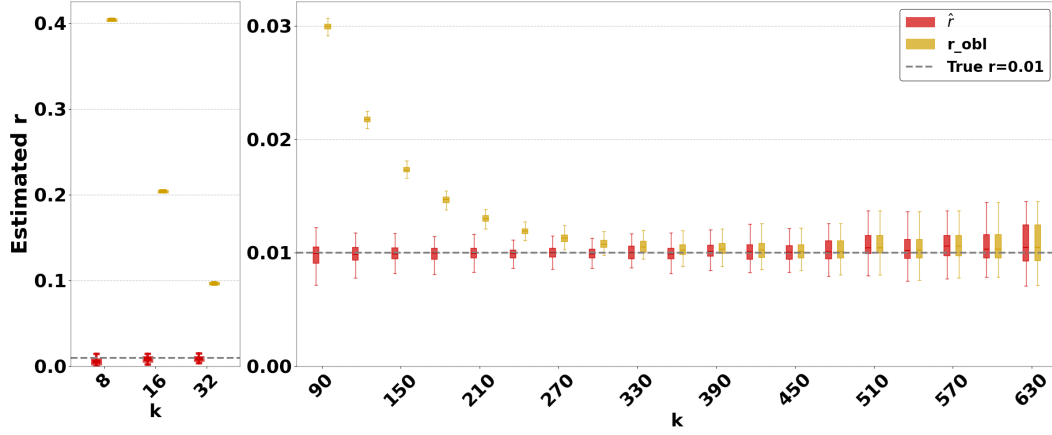


Figure 4 Comparison of our estimator with r_{obl} on *D-hardest*. We fixed $r = 1\%$ and varied k . For each k value, we simulate the random substitution process 100 times and show the box plot of the resulting estimators.

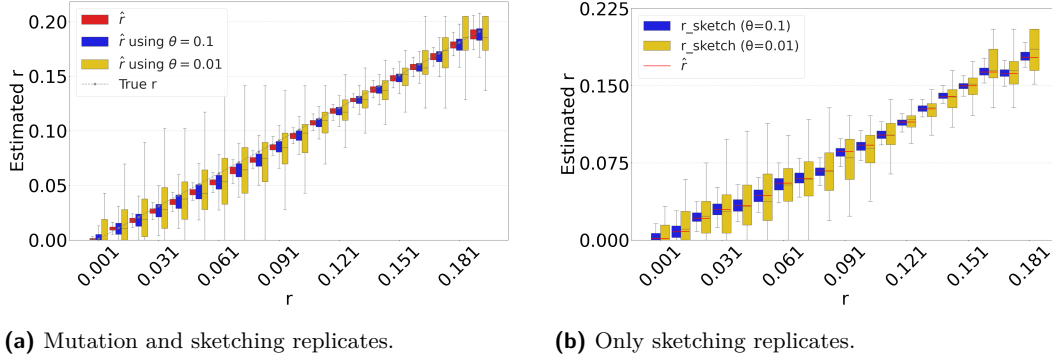


Figure 5 Sketching-based estimation results on *D-hardest*. In panel (a), for each r , we replicate the substitution process 100 times and, for each replicate, we replicate the sketching process 100 times. In panel (b), for each r , we generate one mutated string and replicate the sketching process 100 times.

6.5 Theoretical bounds on the bias

Theorem 3 gives theoretical bounds on the bias of \hat{q} . To validate these bounds empirically, we run simulations, using the same setup as in Figure 3. Figure 7 shows that the empirical mean usually lies within the bias bounds, as the theory predicts. In cases where it does not, the empirical variance is high, indicating that the empirical mean has not yet converged to within the bounds. Furthermore, we see that the upper bound is nearly tight. This is consistent with the fact that overlapping k -mers are rare (Table 1), implying that that $\mathcal{F}(q)$ is approximately equal to our lower bound on the expected intersection size (i.e. L_E).

The lower bound tracks the true value closely, except in the range of $r \in (0, 10\%)$ of *D-hardest*. We believe this is primarily due to the looseness of the variance upper bound U_{Var} in Theorem 2. When we plugged the observed empirical variance of I in place of U_{Var} in Theorem 3, the lower bound curve no longer behaved abnormally in *D-hardest* (plot not shown). Furthermore, when we additionally replaced both U_E and L_E with the observed empirical mean of I , the bounds closely captured the empirical mean of \hat{q} . These empirical

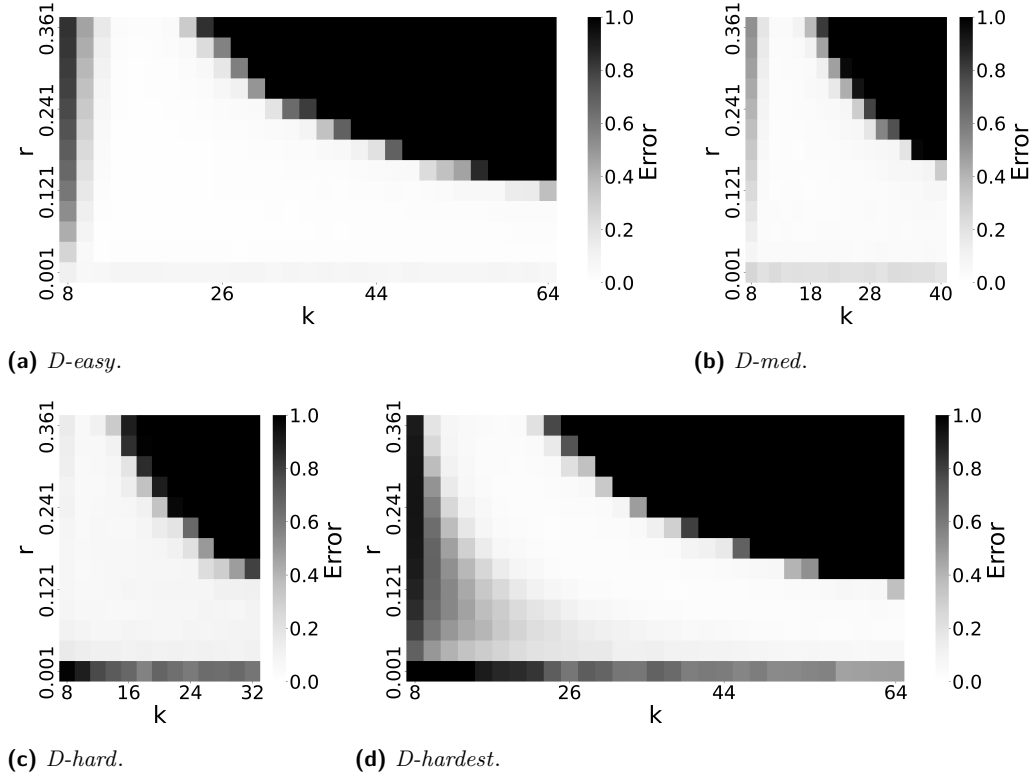


Figure 6 The accuracy of our estimator \hat{r} as a function of both k and r . Each cell shows the average relative absolute error of 100 replicates, e.g. an error of 0.5 means that the estimate is off by 50%. The errors are capped at 1.0, i.e. all errors greater than 1.0 are shown as 1.0.

results suggest that the estimator satisfies the approximation $\mathbb{E}[\hat{q}] = \mathbb{E}[\mathcal{F}^{-1}(I)] \approx \mathcal{F}^{-1}(\mathbb{E}[I])$. In other words, when we have looseness in the bias bounds, it is due to the looseness of Theorems 1 and 2 rather than Theorem 3.

6.6 Identifying unstable parameters using Theorem 4

Figure 6 indicates that when k and r are large enough to lead to a high q , our estimator becomes unstable. Our observations indicate that this happens because the intersection becomes empty, resulting in $\hat{r} = 100\%$ regardless of the true mutation rate. This limitation is anticipated and reflects a fundamental constraint shared by any intersection-based estimator. Figure 7 does not reflect this limitation, because in such cases, the relative error is small simply by virtue of q being close to $\hat{q} = 1$ (even though the estimate of r is not accurate). We therefore looked for an alternative method to *a priori* determine, given a high value of k , which values of high r make our estimator unstable.

We hypothesized that computing the probability of an empty intersection size *a priori* can identify such unstable regions of the parameter space, without needing to do simulations as for Figure 6. Though computing this probability is challenging in the general case, Theorem 4 gives an upper bound P_{empty} based on only L , k , and r . The upper bound is approximately tight when not considering the effect of repeats. We therefore hypothesized that when P_{empty} is high, our estimator becomes unstable.

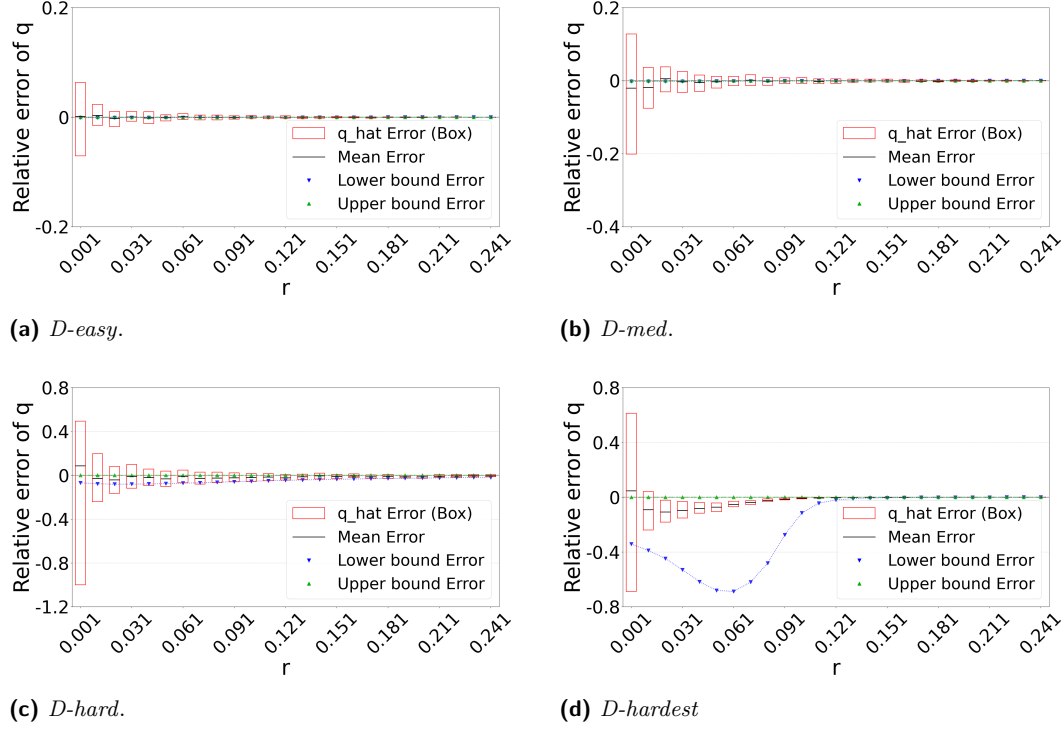


Figure 7 Theoretical bounds on the bias of \hat{q} . For each r , the box plot shows 100 replicates of the substitution process. For the box plots, the y-axis shows the distribution of \hat{q}_i/q . For the lower and upper bound curves, the y-axis corresponds to the ratio of the bound to the true q . The black bars in the center of each box represent the mean, rather than the median.

Figure 8a plots P_{empty} against the accuracy of our estimator. As hypothesized, the substitution rate at which our estimator starts to become unstable (around 24 – 28%) coincides with a sharp increase in P_{empty} . To test this more thoroughly, we computed P_{empty} for all values of k and r for which we evaluated D -hard in Figure 6c. Figure 8c shows that there is a close correspondence between k and r values where our estimator’s relative error is high and P_{empty} is high. These observations suggest that P_{empty} is a useful diagnostic criterion for determining values of k , and r when \hat{r} may fail.

7 Conclusion

In this paper, we propose an estimator for the substitution rate between two sequences that is robust in highly repetitive regions such as centromeres. Our experiments validated its performance across a broad range of k and r values. We provide theoretical bounds on our estimator’s bias (specifically on the bias of \hat{q}), and show that it accurately captures the estimator’s empirical mean in most scenarios.

For large values of k and r , i.e., when q is large, the intersection of the k -spectra tends to be empty with high probability, which is a foreseeable limitation for all intersection-based estimators. To address this, we introduce a heuristic criterion, P_{empty} , which depends only on the number of k -mers L , the k -mer size k , and the substitution rate r . This criterion allows us to heuristically identify parameter settings under which the estimator becomes unstable.

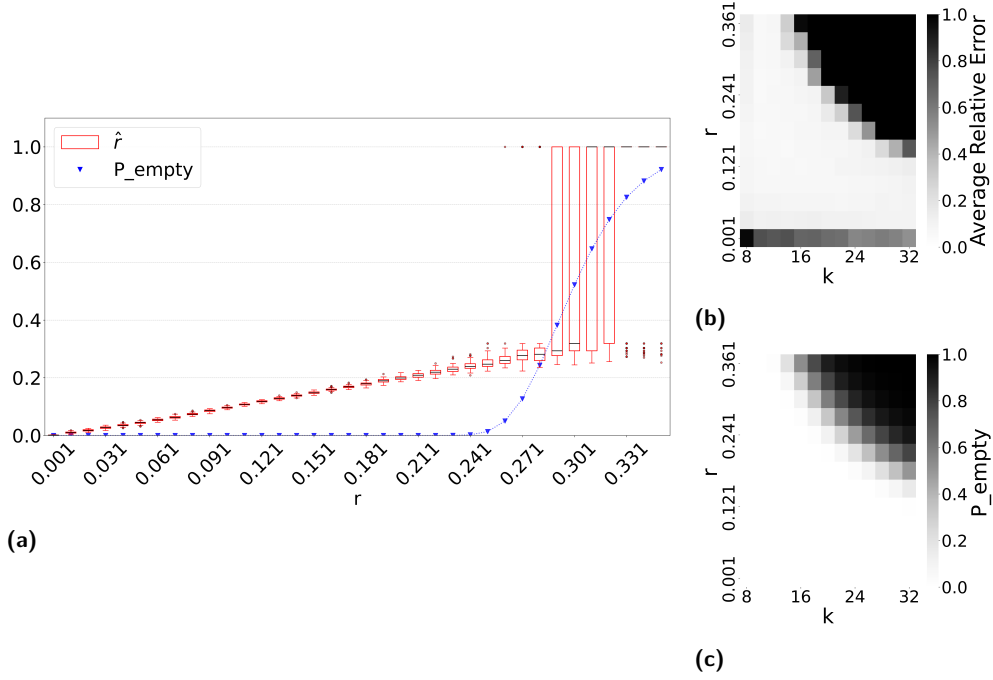


Figure 8 The usefulness of P_{empty} as a diagnostic criterion for when our estimator becomes unstable. Panel (a) overlays the estimator values on D -hardest with P_{empty} values. Panel (b) recapitulates the heatmap of Figure 6c, i.e. the estimator error on D -hard. Panel (c) shows the value of P_{empty} for the length of D -hard and the same parameter values in (b).

We also showed how our estimator can be easily combined with FracMinHashing. Empirical results show that sketching does not introduce systematic bias, albeit at the cost of increased variance.

We do not perform a runtime analysis of our estimator because it completes in less than a second on our data. The runtime of our estimator is the time it takes to solve an equation numerically using Newton’s method. Since $\mathcal{F}(q)$ is a polynomial and the solution is constrained to the interval $[0, 1]$, Newton’s method converges in $\mathcal{O}(\log \log(1/\epsilon))$ iterations, where ϵ is the target precision. Each iteration involves evaluating \mathcal{F} and its derivative, which takes time proportional to the number of non-zero a_i terms. Except for esoteric corner cases, the number of such terms is small in practice.

The immediate open problem is to tighten the theoretical bounds on the bias. Future work could thus focus on deriving a tighter variance bound to strengthen the theoretical characterization of $\mathbb{E}[\hat{q}]$. A bigger open question is how to derive confidence intervals. This is a more challenging problem than bounding the bias because it requires a deeper understanding of the estimator’s distribution.

Our estimator could potentially be extended to work on unassembled sequencing reads, as opposed to assembled genomes. Our method does not rely on the k -mer multiplicities in the intersection size, making it amenable to such a scenario. Still, one of the limitations of our estimator is the need to know the abundance histogram of the source string. A tool like GenomeScope [31] can estimate the abundance histogram from sequence data k -mer counts. Alternatively, the user may choose to use an abundance histogram from a related genome, as related genomes are likely to have similar abundance histograms. Fully adapting this estimator to work with sequencing data remains an important future work.

References

- 1 Anton Bankevich, Andrey V. Bzikadze, Mikhail Kolmogorov, Dmitry Antipov, and Pavel A. Pevzner. Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads. *Nature Biotechnology*, 40(7):1075–1081, 2022.
- 2 Anton Bankevich, Sergey Nurk, Dmitry Antipov, Alexey A Gurevich, Mikhail Dvorkin, Alexander S Kulikov, Valery M Lesin, Sergey I Nikolenko, Son Pham, Andrey D Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology*, 19(5):455–477, 2012. doi:10.1089/cmb.2012.0021.
- 3 Mahdi Belbasi, Antonio Blanca, Robert S Harris, David Koslicki, and Paul Medvedev. The minimizer jaccard estimator is biased and inconsistent. *Bioinformatics*, 38(Supplement_1):i169–i176, June 2022. doi:10.1093/bioinformatics/btac244.
- 4 G. Benson. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Research*, 27(2):573–580, 1999.
- 5 Antonio Blanca, Robert S Harris, David Koslicki, and Paul Medvedev. The statistics of k-mers from a sequence undergoing a simple mutation process without spurious matches. *Journal of Computational Biology*, 29(2):155–168, 2022. doi:10.1089/cmb.2021.0431.
- 6 George Casella and Roger L. Berger. *Statistical inference*. Duxbury, Pacific Grove, Calif., 2. ed. edition, 2002.
- 7 Luca Denti, Marco Previtali, Giulia Bernardini, Alexander Schönhuth, and Paola Bonizzoni. MALVA: genotyping by Mapping-free ALlele detection of known VARIants. *iScience*, 18:20–27, 2019.
- 8 Huan Fan, Anthony R Ives, Yann Surget-Groba, and Charles H Cannon. An assembly and alignment-free method of phylogeny reconstruction from next-generation sequencing data. *BMC genomics*, 16(1):522, 2015.
- 9 Robert S. Harris and Paul Medvedev. Improved representation of Sequence Bloom Trees. *Bioinformatics*, 36(3):721–727, 2020. doi:10.1093/bioinformatics/btz662.
- 10 Bernhard Haubold, Fabian Klötzl, and Peter Pfaffelhuber. andi: Fast and accurate estimation of evolutionary distances between closely related genomes. *Bioinformatics*, 31(8):1169–1175, 2014. doi:10.1093/bioinformatics/btu815.
- 11 Mahmudur Rahman Hera, N Tessa Pierce-Ward, and David Koslicki. Deriving confidence intervals for mutation rates across a wide range of evolutionary distances using fracminhash. *Genome research*, 33(7):1061–1068, 2023.
- 12 Chirag Jain, Luis M Rodriguez-R, Adam M Phillippy, Konstantinos T Konstantinidis, and Srinivas Aluru. High throughput ani analysis of 90k prokaryotic genomes reveals clear species boundaries. *Nature communications*, 9(1):1–8, 2018.
- 13 Bryce Kille, Erik Garrison, Todd J Treangen, and Adam M Phillippy. Minmers are a generalization of minimizers that enable unbiased local Jaccard estimation. *Bioinformatics*, 39(9), 2023. doi:10.1093/bioinformatics/btad512.
- 14 Téó Lemane, Nolan Lezsoche, Julien Lecubin, Eric Pelletier, Magali Lescot, Rayan Chikhi, and Pierre Peterlongo. Indexing and real-time user-friendly queries in terabyte-sized complex genomic datasets with kindex and ORA. *Nature Computational Science*, 4(2):104–109, 2024. doi:10.1038/s43588-024-00596-6.
- 15 Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018. doi:10.1093/bioinformatics/bty191.
- 16 Guillaume Marçais, Brad Solomon, Rob Patro, and Carl Kingsford. Sketching and sublinear data structures in genomics. *Annual Review of Biomedical Data Science*, 2(1):93–118, 2019.
- 17 Shannon M McNulty and Beth A Sullivan. Alpha satellite DNA biology: finding function in the recesses of the genome. *Chromosome research*, 26:115–138, 2018.
- 18 Burkhard Morgenstern, Bingyao Zhu, Sebastian Horwege, and Chris André Leimeister. Estimating evolutionary distances between genomic sequences from spaced-word matches. *Algorithms for Molecular Biology*, 10(1), 2015.

- 19 Sergey Nurk, Sergey Koren, Arang Rhie, Mikko Rautiainen, Andrey V Bzikadze, Alla Mikheenko, Mitchell R Vollger, Nicolas Altemose, Lev Uralsky, Ariel Gershman, et al. The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022.
- 20 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(1):132, 2016.
- 21 Arang Rhie, Sergey Nurk, Monika Cechova, Savannah J. Hoyt, Dylan J. Taylor, Nicolas Altemose, Paul W. Hook, Sergey Koren, Mikko Rautiainen, Ivan A. Alexandrov, Jamie Allen, Mobin Asri, Andrey V. Bzikadze, Nae-Chyun Chen, Chen-Shan Chin, Mark Diekhans, Paul Flicek, Giulio Formenti, Arkarachai Fungtammasan, Carlos Garcia Giron, Erik Garrison, Ariel Gershman, Jennifer L. Gerton, Patrick G. S. Grady, Andrea Guarracino, Leanne Haggerty, Reza Halabian, Nancy F. Hansen, Robert Harris, Gabrielle A. Hartley, William T. Harvey, Marina Haukness, Jakob Heinz, Thibaut Hourlier, Robert M. Hubley, Sarah E. Hunt, Stephen Hwang, Miten Jain, Rupesh K. Kesharwani, Alexandra P. Lewis, Heng Li, Glennis A. Logsdon, Julian K. Lucas, Wojciech Makalowski, Christopher Markovic, Fergal J. Martin, Ann M. Mc Cartney, Rajiv C. McCoy, Jennifer McDaniel, Brandy M. McNulty, Paul Medvedev, Alla Mikheenko, Katherine M. Munson, Terence D. Murphy, Hugh E. Olsen, Nathan D. Olson, Luis F. Paulin, David Porubsky, Tamara Potapova, Fedor Ryabov, Steven L. Salzberg, Michael E. G. Sauria, Fritz J. Sedlazeck, Kishwar Shafin, Valery A. Shepelev, Alaina Shumate, Jessica M. Storer, Likhitha Surapaneni, Angela M. Taravella Oill, Françoise Thibaud-Nissen, Winston Timp, Marta Tomaszekiewicz, Mitchell R. Vollger, Brian P. Walenz, Allison C. Watwood, Matthias H. Weissensteiner, Aaron M. Wenger, Melissa A. Wilson, Samantha Zarate, Yiming Zhu, Justin M. Zook, Evan E. Eichler, Rachel J. O'Neill, Michael C. Schatz, Karen H. Miga, Kateryna D. Makova, and Adam M. Phillippy. The complete sequence of a human y chromosome. *Nature*, 621(7978):344–354, 2023.
- 22 Will P. M. Rowe. When the levee breaks: a practical guide to sketching algorithms for processing the flood of genomic data. *Genome Biology*, 20(1):199, 2019.
- 23 Sophie Röhling, Alexander Linne, Jendrik Schellhorn, Morteza Hosseini, Thomas Dencker, and Burkhard Morgenstern. The number of k-mer matches between two dna sequences as a function of k and applications to estimate phylogenetic distances. *Plos one*, 15(2):e0228070, 2020.
- 24 Shahab Sarmashghi, Kristine Bohmann, M Thomas P Gilbert, Vineet Bafna, and Siavash Mirarab. Skmer: assembly-free and alignment-free sample identification using genome skims. *Genome Biology*, 20(1):1–20, 2019.
- 25 Tizian Schulz and Paul Medvedev. ESKEMAP: exact sketch-based read mapping. *Algorithms for Molecular Biology*, 19(1):19, 2024. doi:10.1186/s13015-024-00261-7.
- 26 Jim Shaw and Yun William Yu. Fast and robust metagenomic sequence comparison through sparse chaining with skani. *Nature Methods*, 20(11):1661–1665, 2023.
- 27 Jim Shaw and Yun William Yu. Proving sequence aligners can guarantee accuracy in almost $O(m \log n)$ time through an average-case analysis of the seed-chain-extend heuristic. *Genome Research*, 33(7):1175–1187, 2023.
- 28 Kai Song, Jie Ren, Gesine Reinert, Minghua Deng, Michael S Waterman, and Fengzhu Sun. New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Briefings in bioinformatics*, 15(3):343–353, 2014. doi:10.1093/bib/bbt067.
- 29 Daniel S Standage, C Titus Brown, and Fereydoun Hormozdiari. Kevlar: a mapping-free framework for accurate discovery of de novo variants. *iScience*, 18:28–36, 2019.
- 30 Chen Sun and Paul Medvedev. Toward fast and accurate snp genotyping from whole genome sequencing data for bedside diagnostics. *Bioinformatics*, 35(3):415–420, 2018. doi:10.1093/bioinformatics/bty641.
- 31 Gregory W Vulture, Fritz J Sedlazeck, Maria Nattestad, Charles J Underwood, Han Fang, James Gurtowski, and Michael C Schatz. Genomescope: fast reference-free genome profiling from short reads. *Bioinformatics*, 33(14):2202–2204, 2017. doi:10.1093/bioinformatics/btx153.

- 32 Haonan Wu, Antonio Blanca, and Paul Medvedev. Repeat-Aware_Substitution_Rate_Estimator. Software, swbId: swb:1:dir:258c949c42d162c56f1e09a0ece39722a5076601 (visited on 2025-08-04). URL: https://github.com/medvedevgroup/Repeat-Aware_Substitution_Rate_Estimator, doi:10.4230/artifacts.24318.
- 33 Haonan Wu, Antonio Blanca, and Paul Medvedev. Reproducibility repository. https://github.com/medvedevgroup/kmer-stats-repeat_Reproduce.
- 34 Huiguang Yi and Li Jin. Co-phylog: an assembly-free phylogenomic approach for closely related organisms. *Nucleic Acids Research*, 41(7):e75–e75, 2013.

A

 Appendix: Missing Proofs

► **Theorem 4.** *Let s be a string of length at least k . The probability that every interval of length k in $s[1..i+k-1]$ has at least one substitution can be computed in $\Theta(ik)$ time with a dynamic programming algorithm that takes as input only L, r, k (not s).*

Proof. Let M_i be the event that every interval of length k in $s[1..i+k-1]$ has at least one substitution. We claim that the following recurrence holds, which automatically leads to the dynamic programming algorithm of the desired time.

$$\Pr[M_i] = \begin{cases} q & \text{if } i = 1, \\ (1 - (1-r)^{i-1})q + (1-r)^{i-1}(1 - (1-r)^{k-i+1}) & \text{if } 1 < i \leq k, \\ \sum_{j=0}^{k-1} \Pr[M_{i-1-j}]r(1-r)^j & \text{if } i > k. \end{cases}$$

We will use k -span to denote an interval of length k in s . For the case that $i = 1$, M_i is the probability that the first k -mer mutates, which is q . For $1 < i \leq k$, we do the following. We denote A as the event that there is at least one substitution in $s[1, i-1]$, B as the event that there is at least one substitution in $s[i, k]$, and C as the event that there is at least one substitution in $s[k+1, i+k-1]$. Then we have

$$\Pr[M_i] = \Pr[A] \cdot \Pr[M_i|A] + \Pr[\neg A] \cdot \Pr[M_i|\neg A] \quad (4)$$

$$= \Pr[A] \Pr[B, C] + \Pr[\neg A] \Pr[B] \quad (5)$$

$$= (1 - (1-r)^{i-1})(1 - (1-r)^k) + (1-r)^{i-1} \cdot (1 - (1-r)^{k-i+1}) \quad (6)$$

For the last case ($i > k$), let L_j be the event that j is the position of the rightmost substitution in $s[1, i+k-1]$. Observe that for $j \neq \ell$, L_j and L_ℓ are mutually exclusive. Furthermore, observe that the rightmost mutation position must be at least i , otherwise the k -span starting at i is not mutated. Therefore, by the law of total probability, we have

$$\Pr[M_i] = \sum_{j=1}^{i+k-1} \Pr[M_i, L_j] = \sum_{j=i}^{i+k-1} \Pr[M_i, L_j] \quad (7)$$

Observe that if there is a substitution at position j , then all the k -spans beginning at positions $j-k+1, \dots, i$ are mutated. Therefore, $\Pr[M_i, L_j] = \Pr[M_{j-k}, L_j] = \Pr[M_{j-k}] \cdot \Pr[L_j]$. Hence,

$$\Pr[M_i] = \sum_{j=i}^{i+k-1} \Pr[M_{j-k}] \cdot \Pr[L_j] = \sum_{j=0}^{k-1} \Pr[M_{i-1-j}] \Pr[L_{i+k-1-j}] \quad (8)$$

$$= \sum_{j=0}^{k-1} \Pr[M_{i-1-j}] \cdot r(1-r)^j \quad (9)$$

◀

► **Lemma A.1.** Let X be a sum of random variables X_1, \dots, X_n , and let $\mu = \mathbb{E}[X]$. Then

$$\text{Var}[X] = n - \mu - (n - \mu)^2 + \sum_{i=1}^n \sum_{j \neq i}^n \Pr[X_i = 0, X_j = 0].$$

Proof.

$$\begin{aligned} \text{Var}[X] &= \mathbb{E}[X^2] - \mu^2 \\ &= \sum_i \sum_j \Pr[X_i = 1, X_j = 1] - \mu^2 \\ &= \sum_i \sum_j \Pr[X_i = 1, X_j = 1] + \Pr[X_i = 1, X_j = 0] - \Pr[X_i = 1, X_j = 0] \\ &\quad - \Pr[X_i = 0, X_j = 0] + \Pr[X_i = 0, X_j = 0] - \mu^2 \\ &= \sum_i \sum_j \Pr[X_i = 1] - \Pr[X_j = 0] + \Pr[X_i = 0, X_j = 0] - \mu^2 \\ &= n\mu - n(n - \mu) + \sum_i \sum_j \Pr[X_i = 0, X_j = 0] - \mu^2 \\ &= -(n - \mu)^2 + \sum_i \sum_j \Pr[X_i = 0, X_j = 0] \\ &= -(n - \mu)^2 + n - \mu + \sum_i \sum_{j \neq i} \Pr[X_i = 0, X_j = 0] \quad \blacktriangleleft \end{aligned}$$

► **Lemma A.2.** $\mathcal{F}(q)$ is invertible on $[0, 1]$. Moreover, if there exists at least one k -mer τ with $\text{occ}(\tau) = 1$, then, on the intervals $q \in [0, 1]$ and $y \in [\mathcal{F}(1), \mathcal{F}(0)]$, and letting f denote the inverse of \mathcal{F} , we have that

$$\mathcal{F}'(q) < 0, \mathcal{F}''(q) \leq 0, \mathcal{F}'''(q) \leq 0, \quad (10)$$

$$f'(y) = \frac{1}{\mathcal{F}'(f(y))} < 0 \quad (11)$$

$$f''(y) = -\frac{\mathcal{F}''(f(y))}{(\mathcal{F}'(f(y)))^3} \leq 0. \quad (12)$$

$$f'''(y) = \frac{\mathcal{F}'''(f(y))\mathcal{F}'(f(y)) - 3(\mathcal{F}''(f(y)))^2}{-(\mathcal{F}'(f(y)))^5} \quad (13)$$

Proof. Recall that a_i is the number of k -mers that have i copies, and that $\mathcal{F}(q) \triangleq L_0 - \sum_{i=1}^L a_i q^i$. Since all a_i values are non-negative, all derivatives of \mathcal{F} are non-positive on $q \in [0, 1]$. Moreover, since we assume that a_1 is strictly positive, the first derivative of \mathcal{F} is strictly negative on $q \in [0, 1]$. The derivatives of f can be expressed in terms of the derivatives of \mathcal{F} and f by applying the inverse function rule from basic calculus. ◀