K-Mer Index

Mathis Fituwi

May 7, 2025

1 Introduction

The k-mer index is a crucial tool in computational genomics for analyzing genome sequences. Similar to n-grams in linguistic analysis, k-mers help identify genome alignments for dozens of applications like disease research. In this paper, we will conduct a basic analysis of k-mer behavior, looking at how the number of seeds, that is occurrences of k-mers, changes as k increases. We also explore the impact of increasing the error rate, which represents the probability of a read containing a mutated base, on alignment performance. Our findings show that as k increases, the median number of seeds decreases exponentially, while a slight increase in the error rate leads to longer alignment runtimes.

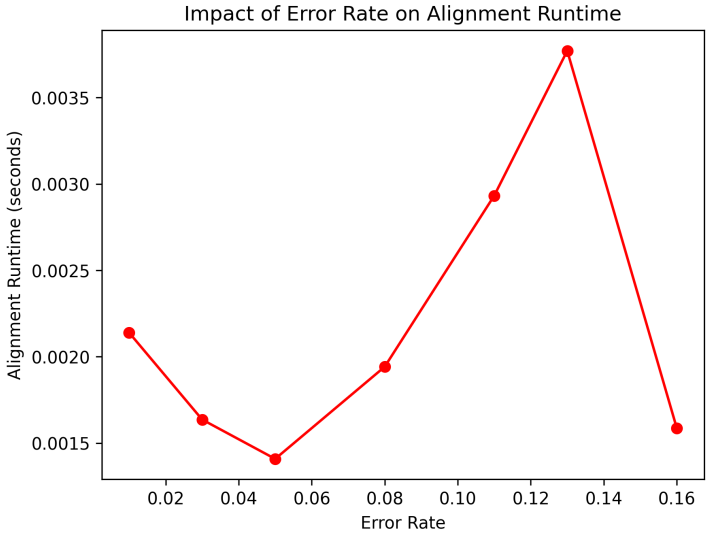

Figure 1: Error Rate vs Alignment Runtime (seconds)

As the error rate increases, alignment runtime also increases. This occurs because a higher error rate introduces more mutations, leading to a greater number of mismatches. More mismatches requires us to examine a larger number of potential alignments, increasing the time spent on forward and backward verification.

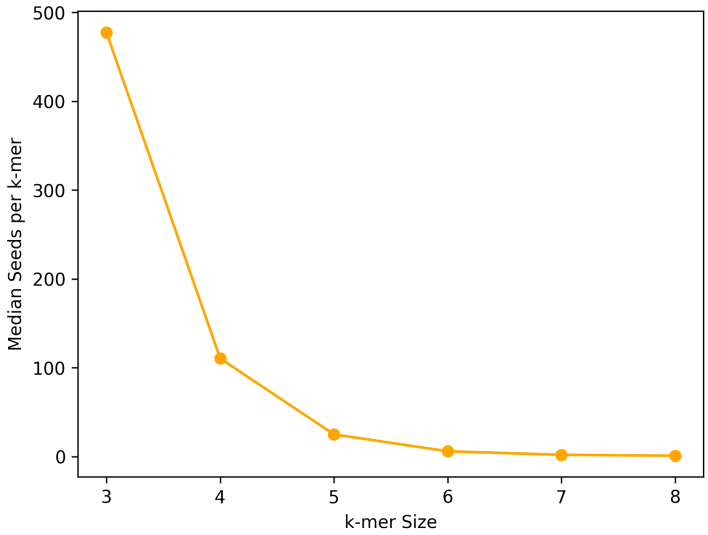

Figure 2: Size of k-mer vs Median Seeds per k-mer

As k-mer size increases, the median number of seeds per k-mer drops significantly. This happens because larger k-mers create more unique nucleotide substrings, which naturally appear less frequently. In contrast, smaller k-mers generate shorter, more general substrings that occur more often.

1.1 Method

Our apporach leverages k-mer indexing to efficiently align simulated sequencing reads to a reference genome. First, we preprocess the reference genome by constructing a k-mer index hash table that maps each k-mer to its index position within the sequence. Reads are then simulated with varying error rates using Poissons' distribution to introduce mutations for the purpose of making the alignment process more realistic.

With each read, we extract its k-mers and use the index to find potential alignment positions in the reference. These alignments checked to see if they fall below the mismatch threshold. Then, the best scoring alignment is determined using the Smith-Waterman algorithm, which evaluates matches, mismatches, and gaps..

To analyze performance, we experiment with different k-mer sizes and error rates. This required for hard coding the desired array of k-mer and error rate values we wanted to work with. We measure the median number of seeds per k-mer to assess sensitivity and track alignment runtime across varying error rates to understand it's computation speed. The results are visualized through line plots to highlight the impact of k-mer specificity and sequencing errors on alignment accuracy and runtime.

1.2 Reproducibility

To replicate these experiments, follow these steps:

```
$ git clone git@github.com:cu-compg-spring-2025/assignment-5-k-mer-index-MatFit.git
$ cd assignment-5-k-mer-index-MatFit/src
$ python kmer_idx.py \
    --kmer 5 \
    --reference ../data/wuhana-hu.fa.gz \
    --num_reads 50 \
```