LARGE SCALE DATA STRUCTURES & ORGANIZATION
Exhaustive, "fuzzy" Alignment with a Prefix Trie

Tsosie Schneider
Isaac Shaffer
Assiri Sareh
Daniel Kpormegbey

NORTHERN ARIZONA UNIVERSITY

Introduction

- The ability to detect the presence of rare variants in bacterial population can provide the signal necessary to fingerprint a sample.
- Subsequently, this provide vital information for the investigation and prosecution of bioterrorism attacks or attempts.
- Most popular heuristic algorithms used to align reads to genomes are not exhaustive

Project Objective

- We implement an exhaustive alignment algorithm using a Prefix Trie (Guaranteed to find an alignment when one exists).
- Algorithm outputting Query number, Genome Location, and SNP count.

Dataset

- Bacillus Anthracis bacteria as reference genome with size 5,227,294 bases (NC_003997.3 Bacillus anthracis str. Ames).
- A 15,901,538 100-mers read set.

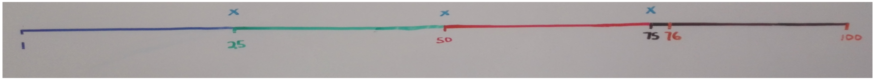
Exhaustive Algorithm

- We consider $5' \rightarrow 3'$ and $3' \rightarrow 5'$ (reverse complement)
- All possible query alignment with 3 SNP's.
- Upper bound on alignment:

$$[2(G.Size - ReadLength) \times (\#NumReads)] \approx 1.662408162 \times 10^{14}$$

Exhaustive Algorithm

- A worst case scenario
- Evenly distribution of the 3 SNP's



Overview of Algorithm (Modified Blast)

- **Prefix Trie** on Genome (+ Reverse Complement)
- **Bit Array**

  - Cluster

0	...	0	0	1	1	1	1	1	1	1	1	1	0	0	...	0

Overview of Algorithm (Modified Blast)

- **Smith-Waterman**

  - Misses edge SNPs

- **Needleman-Wunsch**

  - Entire alignment checked

- **Levenshtein**

Overview of Algorithm (Modified Blast)

Edit (Levenshtein) Distance

		A	C	G	A	A	C
	0	1	2	3	4	5	6
A	1	0	1	2	3	4	5
T	2	1	1	2	2	3	4
G	3	2	2	1	2	3	4
C	4	3	3	2	2	3	4
T	5	4	4	3	3	3	4
T	6	5	5	4	3	4	4

Results

TABLE: Time to completion for a search

Queries	Time (Seconds)
10	40
50	123
100	245
500	1015
1000	1620
5000	9994
10000	19913

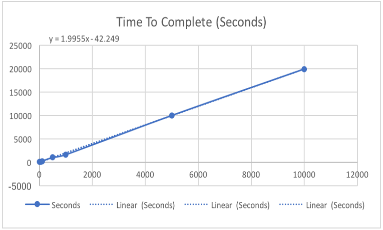
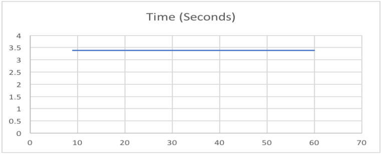


FIGURE: Graph of Time to completion for a search using different number of queries

$y = 1.9955(15901538) - 42.2489 = 31731476.83$. We have approximately 368 days for completion for our reads set.

Results

- Varying the number of seed cluster size from 9, 13, 28, 40, and 60, the time to completion remained 3:38 seconds.



- This may be due to the fact that reads only have a few positions that reach the cluster minimum. Therefore the sections either align well resulting in a bunch of seeds or very few.

Results

TABLE: Time to completion for a search

Seed Size	Time (Seconds)
19	37
20	38
21	39
22	40
23	41
24	41
25	42

+

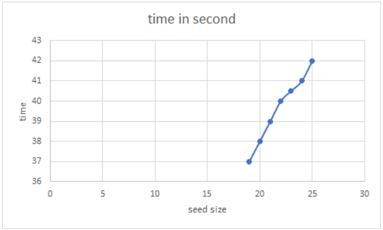


FIGURE: Graph of Time to completion for a search using different seed sizes

Conclusion

- This implementation is impractical on a single core.
- Seed size impacts alignment times.
- Further refinement is a good idea.

Conclusion