BJYZ

Bowen Yao, Jingwu Wang,
Yuqi Chen, Ken Chen

Team Member And Role

	Phase 1: Sept. 17th → Oct. 8th	**Phase 2: Oct 8th → Oct. 29th**	**Phase 3: Oct. 29th → Nov. 19th**	**Phase 4: Nov. 19th → Dec. 1st**
Project Manager	Bowen Yao	Zikang Chen	Yuqi Chen	Jingwu Wang
Documentation lead	Jingwu Wang	Bowen Yao	Zikang Chen	Yuqi Chen
Testing lead	Yuqi Chen	Jingwu Wang	Bowen Yao	Zikang Chen
I/O & Orchestration lead	Zikang Chen	Yuqi Chen	Jingwu Wang	Bowen Yao

Communication

Content

- Algorithm
- Testing
- Result and Metrics

System Requirements

- Python Version
  - Python 3.6 or later version
- Imported Library
  - biopython: Fasta/ Fastq file reader
  - psutil: for retrieving information on system utilization (CPU, memory, etc)

Project Structure

```
BJYZ/
├── src/
│   ├── fasta_reader.py     # Reader class for reference genome
│   ├── fastq_reader.py     # Reader class for sequence reads
│   ├── sam_writer.py       # Writer class for SAM Files
│   ├── substring_index.py  # Indexing + SW algorithm
│   ├── read_mapper.py      # Project main entry point
├── tests/
│   ├── test_fasta_reader.py    # Unit tests for fasta_reader
│   ├── test_fastq_reader.py    # Unit tests for fastq_reader
│   ├── test_sam_writer.py      # Unit tests for sam_writer
│   ├── test_substring_index.py # Unit tests for substring_index algorithm
│   ├── test_index_parallel.py  # Unit tests for parallel version
│   └── file/               # Short example input files
│       ├── example.fasta
│       └── example.fastq
├── data/                   # Folder containing input file
│   ├── fasta files
│   └── fastq files
├── README.md               # Description of project
└── requirements.txt        # Python dependencies
```

Algorithm

Some Algorithms We Tried

Reference Genome Length - n

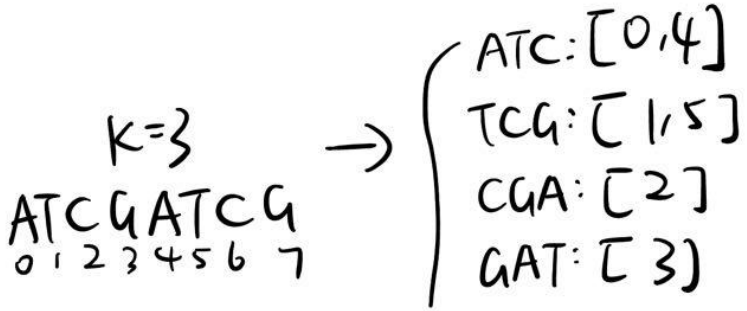
Read Genome Length - m

Seed Length - k (kmer)

- Suffix Array
  - Time complexity is O(m log n)
  - Takes several minutes for 1000 reads
- Smith-Waterman Algorithm
  - Gives the optimal approximate matching
  - Low Efficiency - Time Complexity O(nm)
  - Takes > 1 hour for 1000 reads
- Hash table + Seeds-and-Extends + Banded Smith-Waterman

Reference Genome

1. Decide a k (we choose 20)
2. Find all k-mers in the reference genome and store them along with their starting indices to a hash table



Split Reads - retrieve potential candidate

1. **Set seed length:** Let $k = 20$ represent the length of each seed. Define l as the number of seeds, where $l = \frac{|R|}{2}$, and $|R|$ is the length of the read R.
2. **Select seeds:** Divide the read R into l non-overlapping or overlapping seeds of length k. Denote these seeds as $S_1, S_2, \ldots, S_l$, where $S_i = R[i : i + k]$.
3. **Map seeds to reference genome:** For each seed S_i, use the precomputed hash table H to find the corresponding start indices in the reference genome G, denoted as

$$H(S_i) = \{I_1^i, I_2^i, \ldots\}$$

seed length=3, select seeds for every 2 index

ATCGATCG → [(ATC, 0), (CGA, 2), (ATC, 4)]

0 1 2 3 4 5 6 7

ATC: [0, 4]
TCG: [1, 5]
CGA: [2]
GAT: [3]

→ [([0, 4], 0), ([2], 2), ([0, 4], 4)]

→ [[0, 4], [0], [0]]

Split Reads - apply BandedSW algorithm

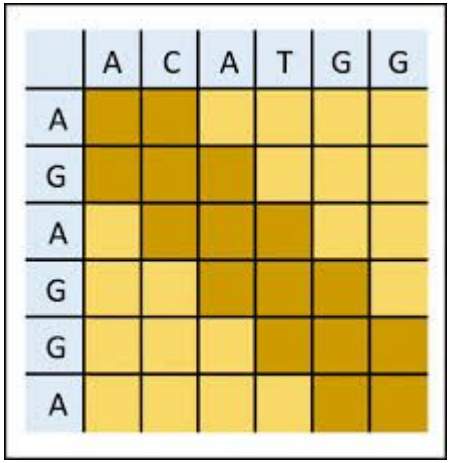
4. **Extract candidate regions:** For each retrieved start index I_j^i from the hash table and let p_i represent the position of seed S_i in R. Extract the corresponding region from the reference genome:

$$G[I_j^i - p_i : I_j^i - p_i + |R|]$$

This region will serve as the candidate for alignment.

5. **Apply banded Smith-Waterman:** Use a **banded dynamic programming** approach to align R to the extracted region. Define the bandwidth w as a parameter controlling the number of diagonal offsets to compute (e.g., $w = 5$). The alignment computation is restricted to cells (i, j) where $|i - j| \leq w$. Compute the alignment score $A(I_j^i)$ for each candidate region.

6. **Compute alignment score:** After performing the banded Smith-Waterman algorithm, let $A(I_j^i)$ denote the alignment score for the start index I_j^i. After obtaining the maximum score, we traced the path back to construct CIGAR string.



Split Reads - apply BandedSW algorithm

7. **Early exit:** Set the early exit score threshold as

$$\tau = 0.1 \times |R|$$

where $|R|$ is the length of the read. If $A(I_j^i) > \tau$, then exit the loop and claim I_j^i as the alignment for the read R. Otherwise, proceed to the next start index.

8. **Reverse complement case:** If no start index satisfies $A(I_j^i) > \tau$, compute the reverse complement of R, and repeat the process from step 2.

Parameters

- k: Seed Length [significantly impact the speed]
  - The larger k, the faster
  - The larger k, the lower possibility to find a match
- Seed_num: The number of seeds for each read
  - The larger, the greater possibility of finding a match
  - The larger, the more comparison that slows speed
- DP threshold: Matching similarity
- Bandwidth: BandedSW bandwidth
  - The lower bandwidth, the faster
  - The lower bandwidth, the accuracy may not be ensured

Testing

Testing on I/O

- ReadFasta and ReadFastq
  - Test on empty file
  - Test incorrect input path
  - Test on malformed file
  - Test on a simple correct file
- SAMWriter
  - Validate the SAM file output
  - Check correct header
  - Check correct mapping

Testing on Parallel Mapping

- Comparison of Expected Start and End Position
  - For each read in the dataset, compare the match index returned by the mapping algorithm with the corresponding expected start and end position from the ground truth data
- Performance Metrics Calculation:
  - Precision
  - Recall
  - Runtime

Result

Midterm Result

Midterm Metrics For Test Dataset 1:

Wall Clock Time (s)	CPU Time (s)	Reads Per Minute	Memory Usage (MB)
4.30	4.27	14,019	41.91

Break-down:

True Positive	False Positive	True Negative	False Negative
811	0	188	1

Precision	Recall	F1 Score
1.00	1.00	1

Midterm Metrics For Test Dataset 2:

Wall Clock Time (s)	CPU Time (s)	Reads Per Minute	Memory Usage (MB)
4.27	4.24	14,151	40.58

Break-down:

True Positive	False Positive	True Negative	False Negative
799	0	188	13

Precision	Recall	F1 Score
1.00	0.98	0.99

Final Result

Final Metrics For Challenging Test Dataset 1:

Wall Clock Time (s)	CPU Time (s)	Reads Per Minute	Memory Usage (MB)
37.60	33.34	531,822	614.60

Break-down:

True Positive	False Positive	True Negative	False Negative
259,851	7,322	66,132	0

Correctness	Precision	Recall	F1 Score
97.8%	0.97	1.00	0.99

Final Metrics For Challenging Test Dataset 2:

Wall Clock Time (s)	CPU Time (s)	Reads Per Minute	Memory Usage (MB)
39.37	35.48	507,888	632.62

Break-down:

True Positive	False Positive	True Negative	False Negative
258,145	8,856	66,199	105

Correctness	Precision	Recall	F1 Score
97.31%	0.97	1.00	0.98

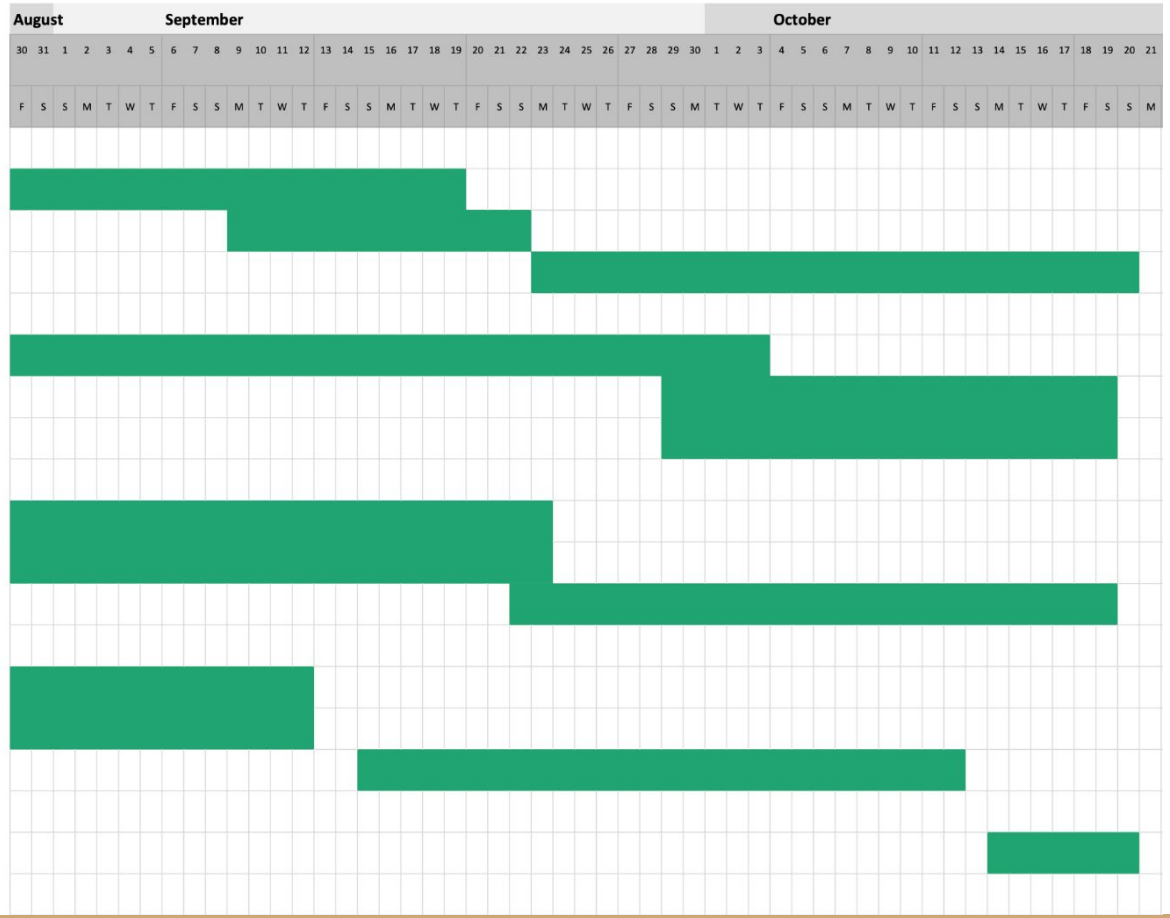
Midterm

Legend: On track | Low risk | Med risk | High risk | Unassigned

Project start date: 8/25/2024

Scrolling increment: 5

Milestone description	Category	Assigned to	Progress	Start	Days
Algorithm					
Hashing	On Track	Bowen Yao	100%	8/30/2024	21
Suffix array	On Track	Bowen Yao	100%	9/9/2024	14
SWT	On Track	Zikang Chen	100%	9/23/2024	28
Documentation					
Code comments	On Track	Jingwu Wang	100%	8/30/2024	35
Design Doc	On Track	Bowen Yao	100%	9/29/2024	21
Readme	On Track	Bowen Yao	100%	9/29/2024	21
Testing					
Testing fasta, fastq readers	On Track	Yuqi Chen	100%	8/30/2024	25
Testing output sam files	On Track	Yuqi Chen	100%	8/30/2024	25
Testing performance	On Track	Jingwu Wang	100%	9/22/2024	28
I/O					
Writing fasta readers	On Track	Zikang Chen	100%	8/30/2024	14
Writing fastq readers	On Track	Zikang Chen	100%	8/30/2024	14
Writing sam output	On Track	Yuqi Chen	100%	9/15/2024	28
Presentation Preparation					
Preparing slides	On Track	All	100%	10/14/2024	7



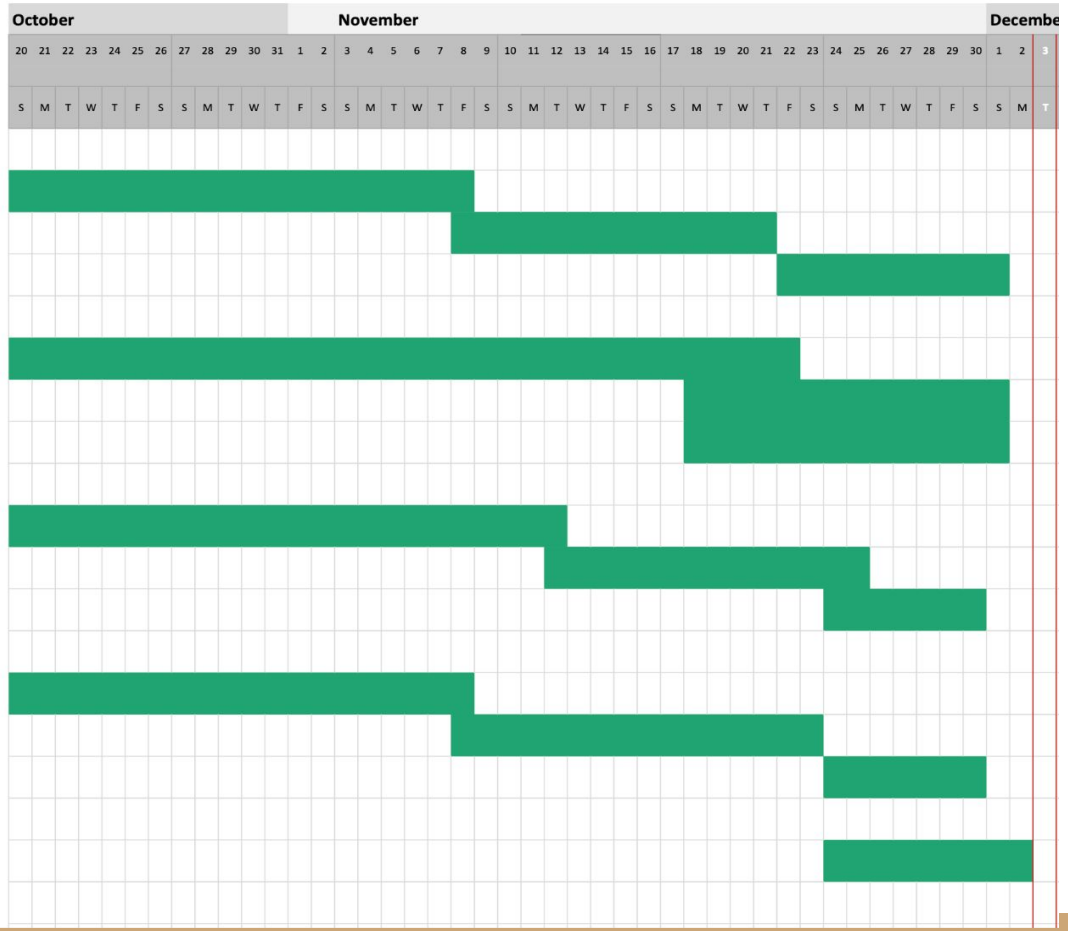
Final Project

Legend: On track | Low risk | Med risk | High risk | Unassigned

Project start date: 10/19/2024

Scrolling increment: 1

Milestone description	Category	Assigned to	Progress	Start	Days
Algorithm Optimization					
Implementing the parallel design	On Track	Zikang Chen	100%	10/19/2024	21
banned SWT	On Track	Yuqi Chen	100%	11/8/2024	14
fine-tune parameters	On Track	Jingwu Wang	100%	11/22/2024	10
Documentation					
Code comments	On Track	Bowen Yao	100%	10/19/2024	35
Design Doc	On Track	Zikang Chen	100%	11/18/2024	14
Readme	On Track	Yuqi Chen	100%	11/18/2024	14
Testing					
Testing performance	On Track	Jingwu Wang	100%	10/19/2024	25
Testing format of sam files	On Track	Bowen Yao	100%	11/12/2024	14
Testing on NOTs	On Track	Zikang Chen	100%	11/24/2024	7
I/O					
Writing CIGAR string	On Track	Yuqi Chen	100%	10/19/2024	21
Writing other fields	On Track	Jingwu Wang	100%	11/8/2024	16
SAM file formats	On Track	Bowen Yao	100%	11/24/2024	7
Presentation Preparation					
Preparing slides	On Track	All	100%	11/24/2024	9



Further Development

- Compile C code using Cython
- GPU acceleration

Conclusion

Hash table + Seeds-and-Extends + Smith-Waterman

~14000 reads/min (Midterm Goal)

Added “**Banded**” Smith-Waterman

~90000 reads/min

Added **Parallelism** on read level

~500000 reads/min

Citation

- Alser, M., Rotman, J., Deshpande, D., Taraszka, K., Shi, H., Baykal, P. I., Yang, H. T., Xue, V., Knyazev, S., Singer, B. D., Balliu, B., Koslicki, D., Skums, P., Zelikovsky, A., Alkan, C., Mutlu, O., & Mangul, S. (2021). Technology dictates algorithms: Recent developments in read alignment. *Genome Biology*, *22*(1). https://doi.org/10.1186/s13059-021-02443-7
- Y. -L. Liao, Y. -C. Li, N. -C. Chen and Y. -C. Lu, "Adaptively Banded Smith-Waterman Algorithm for Long Reads and Its Hardware Accelerator," 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Milan, Italy, 2018, pp. 1-9, doi: 10.1109/ASAP.2018.8445105.
- Treangen, T. J., & Salzberg, S. L. (2011). Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nature reviews. Genetics*, *13*(1), 36–46. https://doi.org/10.1038/nrg3117