Timothy Liu
005994409

# CS CM121 Project Option 2: Pseudoalignment Implementation Report

## Overview

My pseudoalignment implementation uses the naive approach with a hash table for indexing k-mers, generating k-mers from both forward and reverse complement DNA sequences and excluding k-mers containing ambiguous bases ('N'). It matches these k-mers against an index built from transcriptome data, grouping the matches into equivalence classes. New equivalence classes arise from the intersection of matched transcripts from both strands, accounting for novel mappings. This design efficiently handles the complexities of DNA sequences and ensures accurate pseudoalignment results. Code was written in Python with the immense help of Biopython.

# Building Index

**Build Index**
The build_index function constructs an index from the given transcriptome data. It reads the transcriptome file, extracts all possible k-mers of the specified length (k), and maps each k-mer to the corresponding transcript ID. This index has rapid lookups during pseudoalignment since it uses a hash table in the form of a python defaultdict object.

# Pseudoalignment

## Generating k-mers

The generate_kmers function creates a set of unique k-mers of length 'k' from a given sequence, excluding any k-mers that contain the ambiguous base 'N'. It iterates through the sequence, extracting and adding valid k-mers to the set to make sure that only unambiguous k-mers are included.

## Reverse Complement Function

The reverse_complement function calculates the reverse complement of a DNA sequence. This is needed for handling reads from both strands of DNA. It uses a dictionary to map each nucleotide to its complement and rebuilds the reverse complementary sequence one character at a time. Sequences with 'N' included were kept the same and are dealt with when extracting k-mers from the sequence.

## Pseudoalignment

The pseudoalign function performs the main pseudoalignment task. It processes paired-end reads, generating k-mers and their reverse complements for each read. These k-mers are then

matched against the index to identify transcripts containing those k-mers. The transcripts are grouped into equivalence classes, and the counts of reads mapping to each equivalence class are recorded.

More specifically, it creates a set of unique k-mers from a given sequence, excluding any k-mers with 'N' to avoid ambiguity. It iterates through the sequence, extracting each k-mer of length 'k' and adds it to the set if it does not contain 'N'. This ensures only unambiguous k-mers are considered. For each k-mer, it checks for matches in the index. If matches are found, it intersects these matches to determine the equivalence class of transcripts that align with the read. If no matches are found, an 'NA' equivalence class is assigned. The function then counts occurrences of each equivalence class, returning a dictionary of these counts.

In terms of dealing with 'N', I chose to disregard the 'N' bases when generating k-mers for several reasons. Primarily, 'N' represents any nucleotide base (A, T, C, or G) and so it introduces ambiguity that can significantly affect the accuracy and reliability of the k-mer matching process. Including k-mers containing 'N' could result in false-positive matches since 'N' could correspond to multiple possible k-mers, leading to a substantial increase in the potential k-mer space and the computational complexity of the alignment. By excluding k-mers with 'N', I make sure that only well-defined, unambiguous sequences are considered, which enhances the specificity and precision of the pseudoalignment. This simplifies the computational process and improves the quality of the matches by focusing on definitive nucleotide sequences, which can reduce noise and improve the overall robustness and accuracy of the alignment results.

# Miscellaneous

### Main Function

The run_pseudoalignment function does the entire pseudo alignment process. It builds the index, performs pseudoalignment, and outputs the results. It also calculates and prints statistics on the sizes of equivalence classes. The parameters given are: the reads FASTA file, the transcriptome FASTA, and the desired k-mer length.

### Equivalence Class Dataframe

The equivalence_classes_to_dataframe function converts equivalence classes and their counts into a pandas DataFrame. It just makes it easier to visualize and parse the data.

### Compare Results

The compare_results function compares pseudoalignment results with results from a traditional aligner like Bowtie.

Timothy Liu
005994409

### Reading in FASTA Files

Reading in FASTA files was done using Biopython's SeqIO module, which provides a python package for parsing sequence data. This allowed me to extract and manipulate the nucleotide sequences directly from the FASTA files for subsequent manipulation.

# Results and Statistics

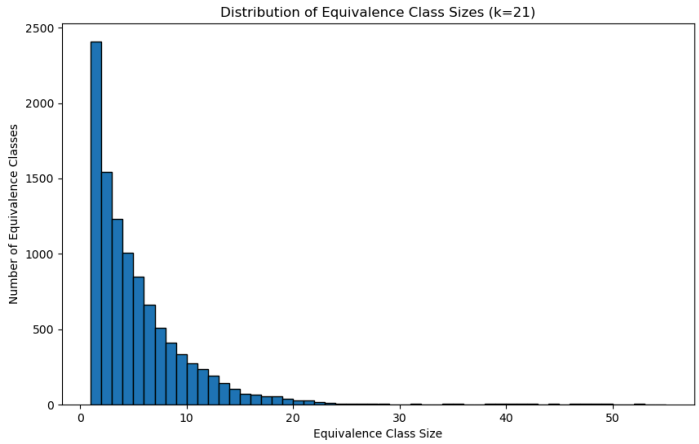
### k=21 Results

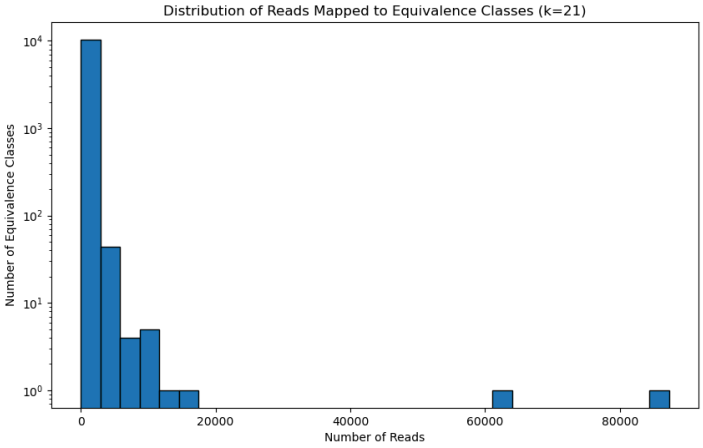
| | counts | number of items in equivalence class | isoforms in equivalence class |
|---|---|---|---|
| 32 | 87300 | 1 | NA |
| 4248 | 61258 | 2 | ENST00000329251,ENST00000496634 |
| 1718 | 15353 | 1 | ENST00000536684 |
| 2137 | 12309 | 7 | ENST00000227157,ENST00000379412,ENST0000039622... |
| 8796 | 11070 | 1 | ENST00000393067 |
| 3827 | 10449 | 5 | ENST00000345732,ENST00000389939,ENST0000053207... |
| 542 | 10012 | 3 | ENST00000321153,ENST00000530398,ENST00000530797 |
| 1968 | 9331 | 2 | ENST00000228140,ENST00000525634 |
| 9474 | 8799 | 2 | ENST00000527673,ENST00000532567 |
| 9767 | 7448 | 8 | ENST00000227378,ENST00000524552,ENST0000052611... |
| 9744 | 7294 | 1 | ENST00000260197 |
| 9473 | 6333 | 3 | ENST00000527673,ENST00000527791,ENST00000532567 |
| 9478 | 5989 | 1 | ENST00000527673 |
| 1451 | 5504 | 7 | ENST00000314138,ENST00000524496,ENST0000052656... |
| 1734 | 5329 | 4 | ENST00000339995,ENST00000396525,ENST0000052568... |

**Output of Top 15 counts for k=21**

Timothy Liu
005994409

## Plots




## Duration

Pseudoalignment (and index building) with k=21 took 2 minutes and 59 seconds on an M1 Air Macbook.

## Statistics

| | Min | Max | Median | Mean | STD |
|---|---|---|---|---|---|
| Counts | 1 | 87300 | 10 | 123.70 | 1149 |
| Number of Items in Equivalence Class | 1 | 54 | 4 | 5.071 | 5.30 |

Timothy Liu
005994409

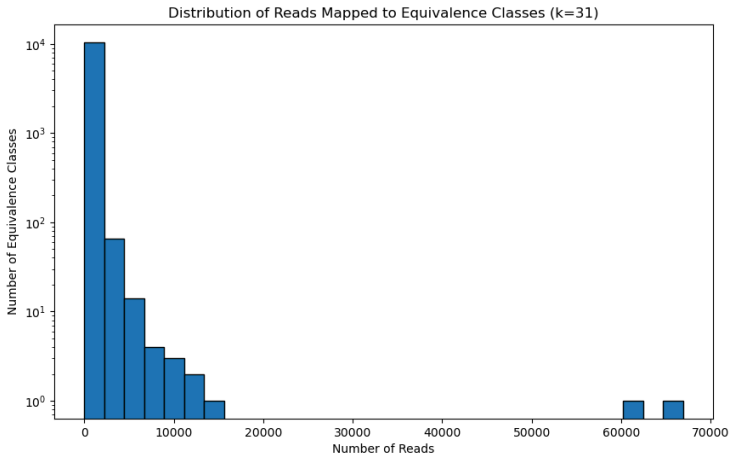
## k=31 Results

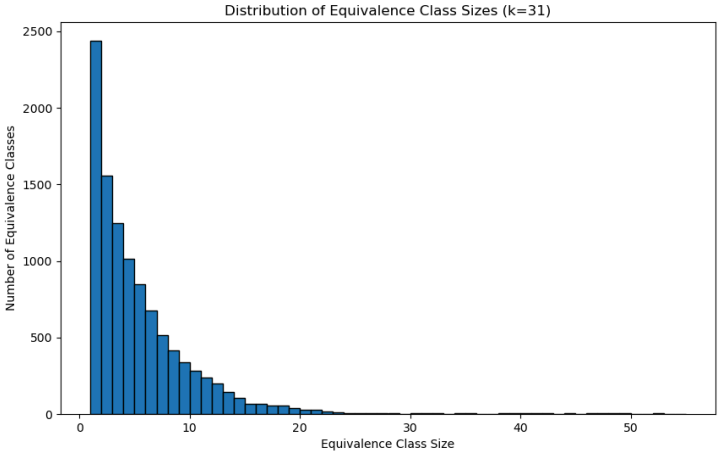
| | counts | number of items in equivalence class | isoforms in equivalence class |
|---|---|---|---|
| 32 | 66897 | 1 | NA |
| 4302 | 61263 | 2 | ENST00000329251,ENST00000496634 |
| 1737 | 15353 | 1 | ENST00000536684 |
| 2162 | 12320 | 7 | ENST00000227157,ENST00000379412,ENST0000039622... |
| 8907 | 12097 | 1 | ENST00000393067 |
| 3871 | 10440 | 5 | ENST00000345732,ENST00000389939,ENST0000053207... |
| 547 | 10038 | 3 | ENST00000321153,ENST00000530398,ENST00000530797 |
| 1989 | 9317 | 2 | ENST00000228140,ENST00000525634 |
| 9590 | 8807 | 2 | ENST00000527673,ENST00000532567 |
| 9885 | 7457 | 8 | ENST00000227378,ENST00000524552,ENST0000052611... |
| 9861 | 7386 | 1 | ENST00000260197 |
| 9589 | 6755 | 3 | ENST00000527673,ENST00000527791,ENST00000532567 |
| 2161 | 6028 | 11 | ENST00000227157,ENST00000379412,ENST0000039622... |
| 9594 | 5954 | 1 | ENST00000527673 |
| 1465 | 5545 | 7 | ENST00000314138,ENST00000524496,ENST0000052656... |

**Output of Top 15 counts for k=31**

Timothy Liu
005994409

# Plots




# Duration

Pseudoalignment (and index building) with k=31 took 3 minutes and 46 seconds on an M1 Air Macbook.

# Statistics

| | Min | Max | Median | Mean | STD |
|---|---|---|---|---|---|
| Counts | 1.0 | 66897 | 10.0 | 122.27 | 1007 |
| Number of Items in Equivalence Class | 1.0 | 54 | 4.0 | 5.079 | 5.32 |