Assignment Tutorial

MSBD5009/COMP5112 Parallel Programming

Assignment 1: Super-mer Generation with MPI

Tutorial Overview

- Problem Description
- Implementation Instruction
- Environment Setup

Problem Description

- Basic Concepts

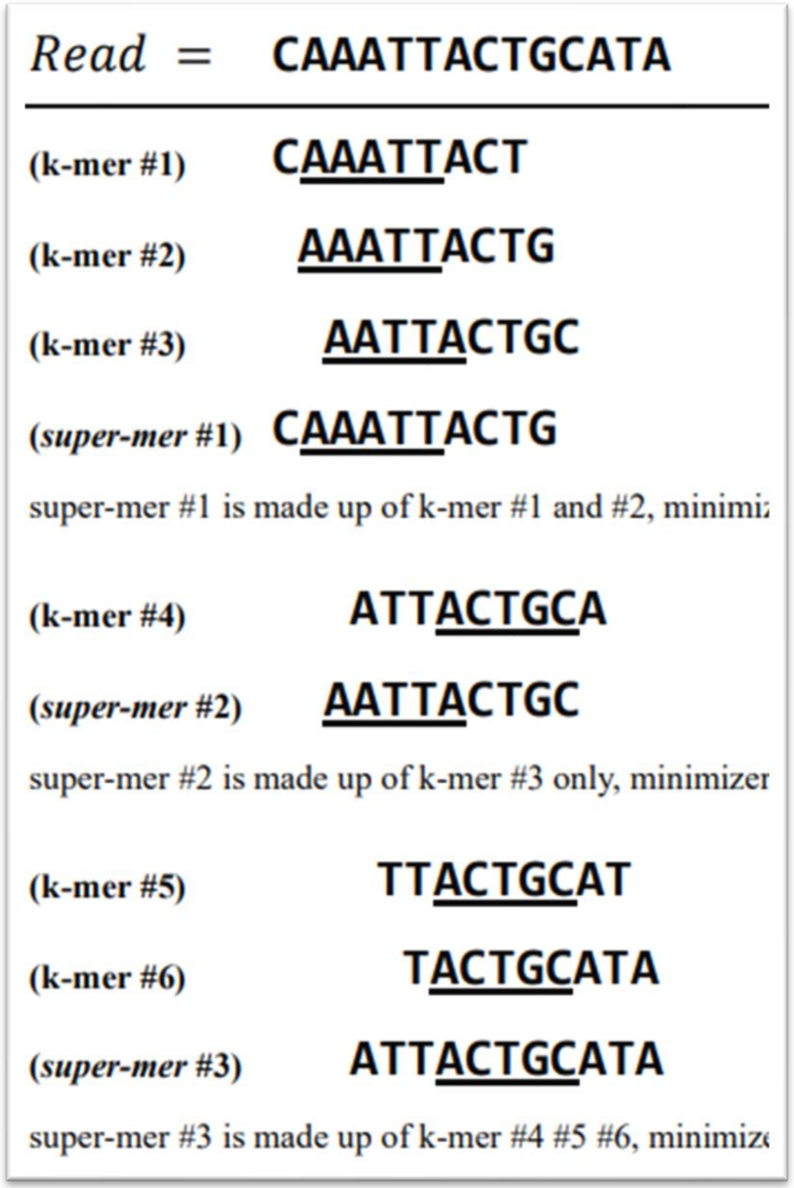
  1. Read
  - A DNA fragment with base A, C, T, G (i.e., a string contains ‘A’, ‘C’, ‘T’, ‘G’ only).
  - `CAAATTACTGCATA`

  2. K-mer
  - A length-k substring on a read. A read of length n contains $n - k + 1$ k-mers.
  - (k=9) `CAAATTACT, AAATTACTG, …, TACTGCATA` are the k-mers of the above read

  3. Minimizer

  4. Super-mer

.



Read = CAAATTACTGCATA

(k-mer #1) CAAATTACT

(k-mer #2) AAATTACTG

(k-mer #3) AATTACTGC

(*super-mer* #1) CAAATTACTG

super-mer #1 is made up of k-mer #1 and #2, minimi

(k-mer #4) ATTACTGCA

(*super-mer* #2) AATTACTGC

super-mer #2 is made up of k-mer #3 only, minimizer

(k-mer #5) TTACTGCAT

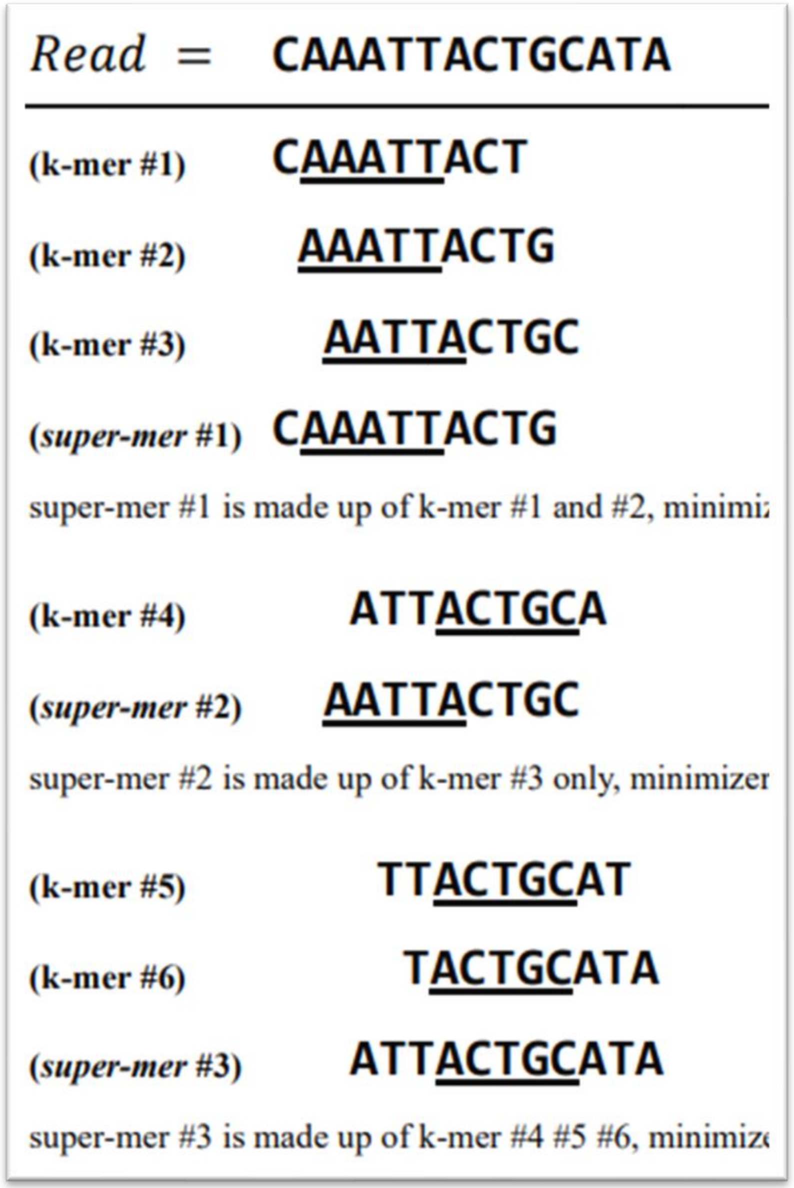
(k-mer #6) TACTGCATA

(*super-mer* #3) ATTACTGCATA

super-mer #3 is made up of k-mer #4 #5 #6, minimiz

Problem Description

- Basic Concepts
  1. **Read**
  2. **K-mer**
  3. **Minimizer**
     - The lexicographically smallest length-p substring of a k-mer.
     - (p=5) The minimizer of CAAATTACT is AAATT.
  4. **Super-mer**
     - A substring of a read generated by merging multiple consecutive k-mers which have the same minimizer value.
     - (k=9, p=5) The first super-mer in the read CAAATTACTGCATA will be CAAATTACTG because the first two k-mers have the same minimizer AAATT.



Read = CAAATTACTGCATA

(k-mer #1) C$\underline{\text{AAATT}}$ACT

(k-mer #2) $\underline{\text{AAATT}}$ACTG

(k-mer #3) $\underline{\text{AATTA}}$CTGC

(*super-mer* #1) C$\underline{\text{AAATT}}$ACTG

super-mer #1 is made up of k-mer #1 and #2, minimiz

(k-mer #4) ATT$\underline{\text{ACTGC}}$A

(*super-mer* #2) $\underline{\text{AATTA}}$CTGC

super-mer #2 is made up of k-mer #3 only, minimizer

(k-mer #5) TT$\underline{\text{ACTGC}}$AT

(k-mer #6) T$\underline{\text{ACTGC}}$ATA

(*super-mer* #3) ATT$\underline{\text{ACTGC}}$ATA

super-mer #3 is made up of k-mer #4 #5 #6, minimiz

Problem Description

- Your Task
  - Input
    - Many reads
    - *Given in CSR format in the program*
  - Output
    - All the super-mers generated from these reads
    - *You need to save all the super-mers to a vector of strings "all_supermers" in Process 0*

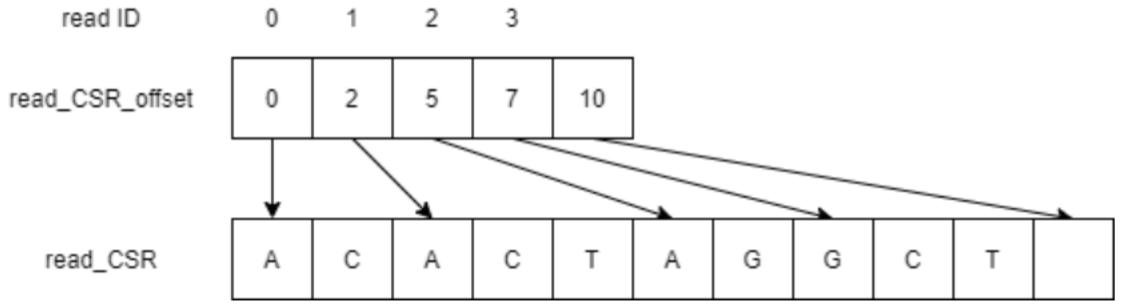
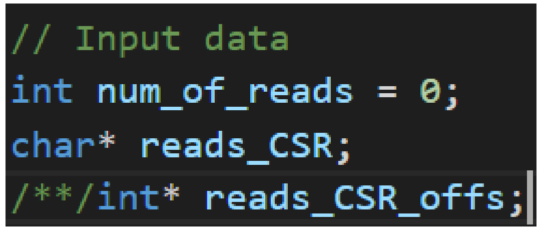
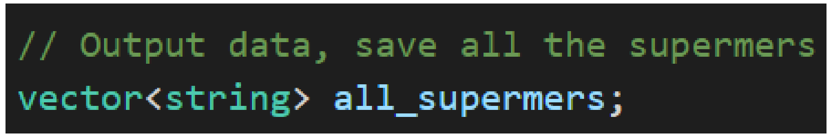


Figure 2: An Example of CSR Format



```
// Input data
int num_of_reads = 0;
char* reads_CSR;
/**/int* reads_CSR_offs;
```



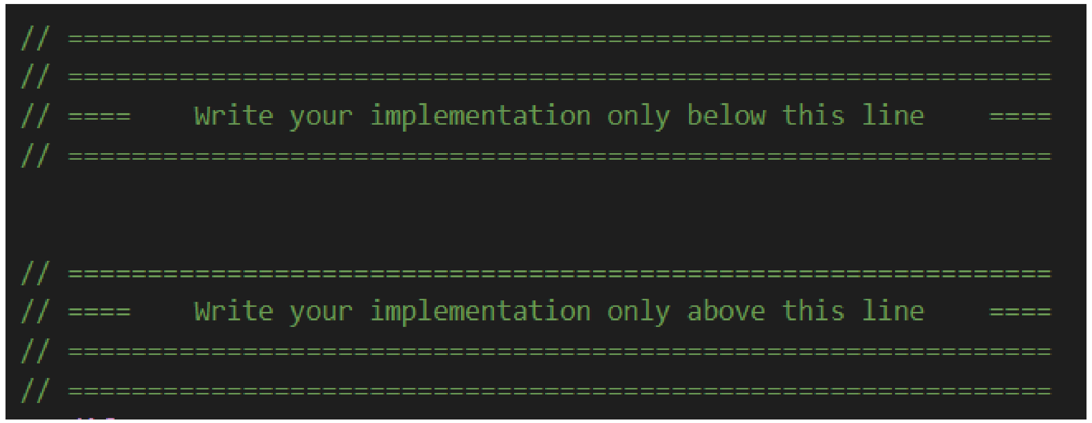
```
// Output data, save all the supermers
vector<string> all_supermers;
```

Implementations

- The code skeleton ***gensuper-mer_mpi.cpp***
  - Already implemented:
    - MPI initialization and finalization
    - Loading reads from the dataset file and converting to CSR format
    - Result correctness checking
    - Outputting super-mers to text file
    - * Function ***read2supermers***(…) which can convert a read to its corresponding super-mers
  - You need to:
    - Scatter the read data to each MPI process
    - Perform the super-mer generation in each process
      - You can refer to the sequential version to know the usage of the function read2supermers(…)
    - Gather all the super-mers to Process 0 and store in the vector "all_supermers“
      - Each string represents a super-mer
      - The order in the vector doesn't matter

Implementations

- Only write your code in the specified area of ***gensuper-mer_mpi.cpp*** and only submit this file to Canvas.



```
// ==============================================================
// ==============================================================
// ====    Write your implementation only below this line    ====
// ==============================================================


// ==============================================================
// ====    Write your implementation only above this line    ====
// ==============================================================
// ==============================================================
```