Assi torial

MSBD5009/COMP5112 ramming

Assignment 1: Super-mer with MPI

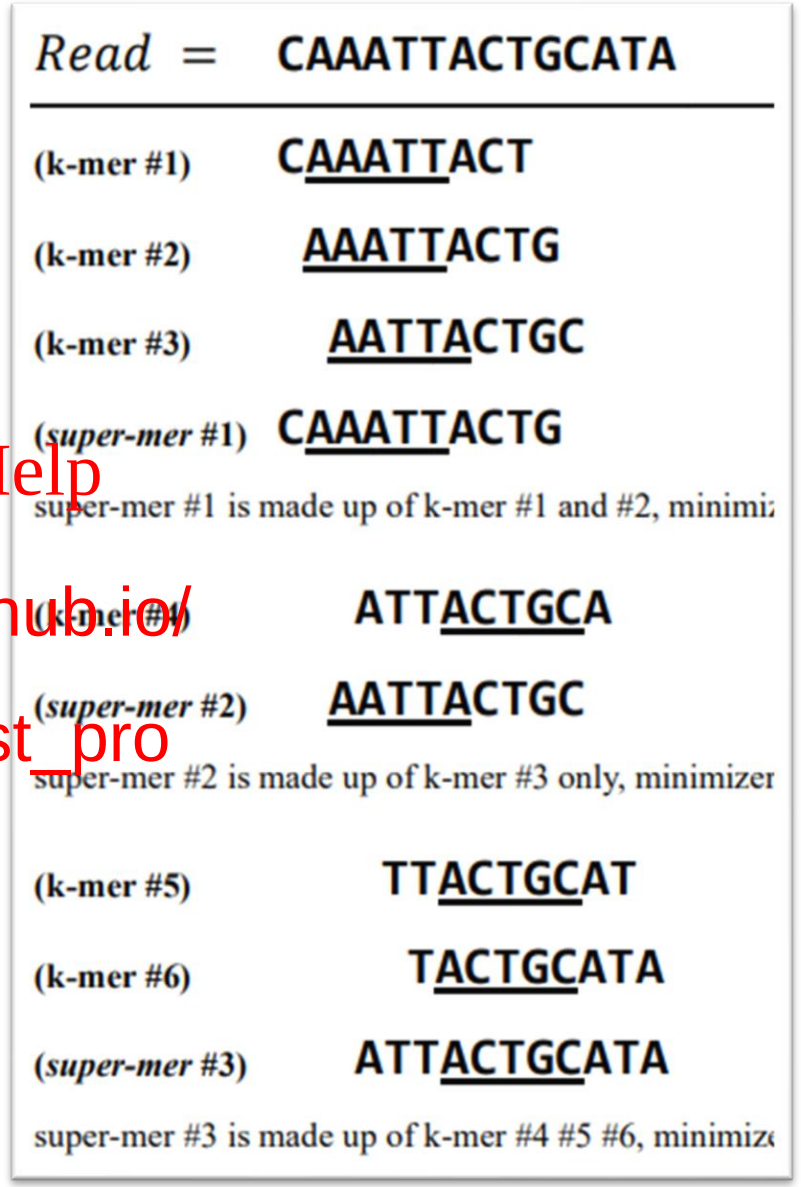
Tutorial Overview

- Problem Description
- Implementation Instruction
- Environment Setup

Problem Description

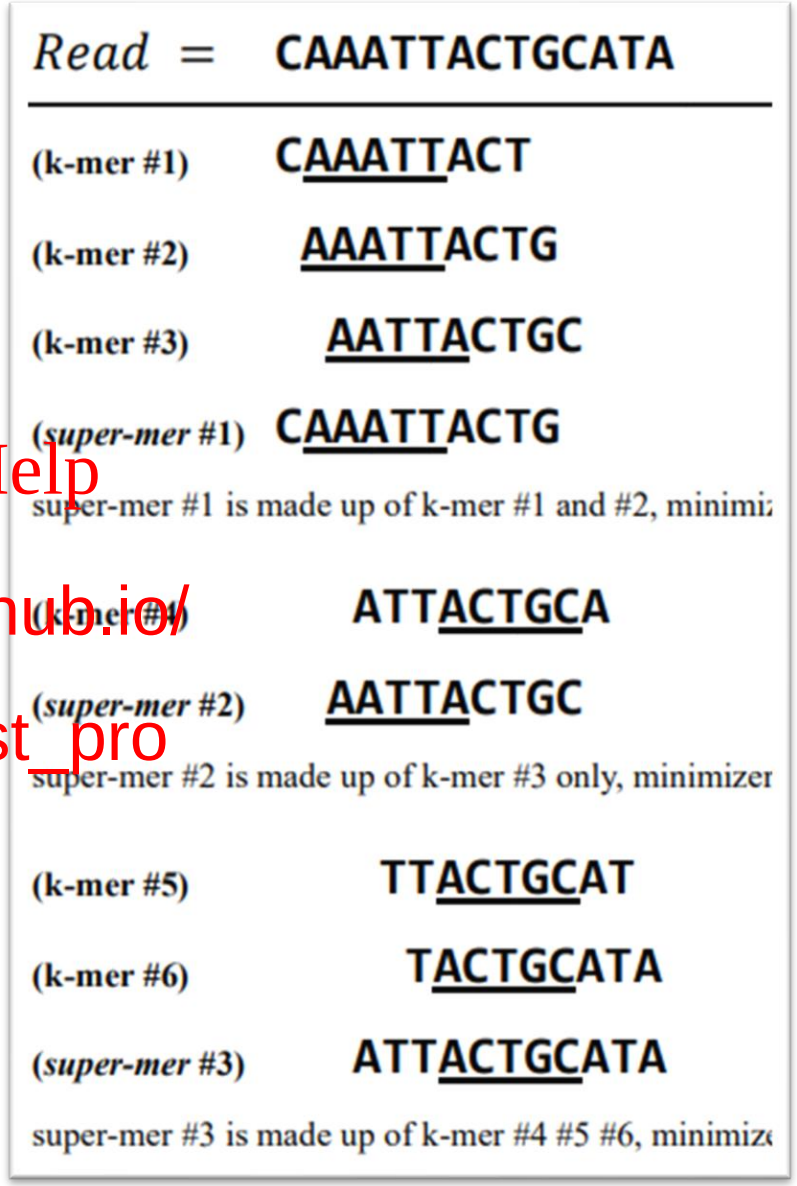
- Basic Concepts
  1. **Read**
     - A DNA fragment with base A, C, T, G (i.e. a string contains ‘A’, ‘C’, ‘T’, ‘G’ only).
     - CAAATTACTGCATA
  2. **K-mer**
     - A length-k substring on a read. A read of n contains $n-k+1$ k-mers.
     - (k=9) CAAATTACT, AAATTACTG, …, TACTGCATA are the k-mers of the above read
  3. **Minimizer**
  4. **Super-mer**

.



Problem Description

- Basic Concepts
  1. **Read**
  2. **K-mer**
  3. **Minimizer**
     - The lexicographically s of a k-mer.
     - (p=5) The minimizer of CAAATTACT is AA
  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.



Problem Description

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

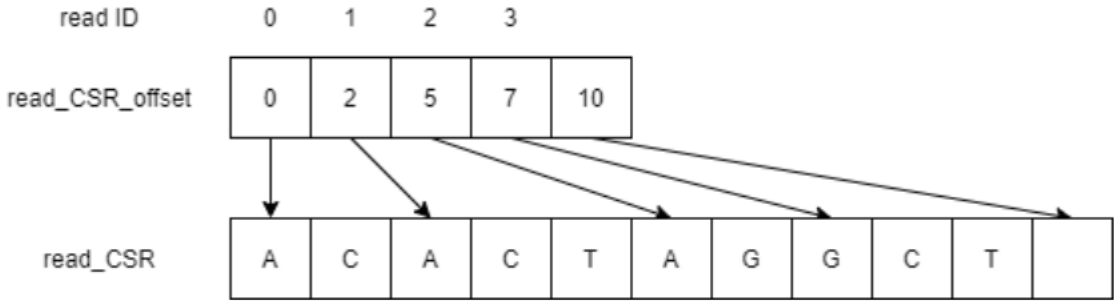


Figure 2: An Example of CSR Format


```
// 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 che
    - Outputting super-mer
    - * Function ***read2supermers***(…) which can 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.