**FastHASH: A new algorithms for fast and comprehensive next generation sequence mapping**

**Hongyi Xin**  
Computer Science Department of School of  
Computer Science, CMU

**Donghyuk Lee**  
Electrical and computer engineering department  
of School of Engineering, CMU

### DNA sequencing & mrFAST

**Background: DNA sequencing**
- **Goal:** People want to know their DNA at a cheap cost
- **Mechanism:** Read short fragments and reconstruct them  
  - Via aligning them against a Reference DNA  
  - String Mapping
- **Difficulties:** Individuals have mutations including  
  - Mismatch, Insertion, and Deletions

**mrFAST: Hash table for fast location look up**
- Hash table -- fast lookup
- String comparison -- detailed mutation information

**String comparison -- Expensive function**

**mrFAST flow chart**

**Problem of mrFAST**
- Too slow: 5 hours 40 min for 1M reads
- String Compare is expensive: over 9 µs each
- Extra memory access further slows down the program

### Adjacency Filtering

**Observation**
- String comparison takes too much time
  - Most string compares are useless
  - Lots of memory accesses

**Mechanism**
- Adjacency Filtering -- Reducing String Comparison
  - If perfect match, consecutive segments should be at consecutive locations!

**Evaluation**
- Filter out less relevant coordinates: -- Reducing Edit-distance calculation

**Cheap Key Selection**

**Observation**
- Load imbalance of Hash Table
  - Longer entries ➔ more computation, higher frequency

**Mechanism**
- Selecting e+1 keys for full coverage: Pigeon Hole Theorem
  - Search for e+1 key is sufficient:  
    - At least one of them has no error
  - Sort them based on their coordinate list's size,  
    - Shorter, the cheaper
  - Select the cheapest e+1 keys
- Benefits of Cheap Key Selection:
  - Reducing the number searching in Adjacency Filtering

**Evaluation**
- Selecting cheapest e+1 keys: Reduce searching

### Methodology

- **Input fragment set:**
  - Fragment length: 108 base-pairs
  - Fragment size: 1 million
  - Fragment composition: first 1 million fragments from the first chromosome of reference DNA
  - Endurance: 3 mismatches or insertions or deletions
- **Processing Platform:**
  - CPU: Intel Sandybridge i7 with 16 GB main memory
  - GPU: Nvidia Tesla 2070C with 6 GB local memory (GDDR5)

### Evaluation & Conclusion

- **CPU Run time Comparison**
  - CPU Run time Comparison
  - CPU GPU Comparison
- **Conclusion**
  - Adjacency Filtering Provides 1.87x speed up
  - Adjacency Filtering + Check Key Selection Provides 14x speed up
- **Explanation**
  - The code does not explore enough parallelism
  - Bad communication to computation ratio