Distributed Genome Preprocessing

Charles W. Kazer

07/30/15

Abstract

- Genome preprocessing – analyzing and manipulating short read data in order to improve downstream assembly
- Hadoop – A distributed computing framework that scales to an arbitrary number of nodes
- Counting *k*-mers, substrings of reads, in order to determine read reliability
- Remove reads with low *k*-mer counts from data
- Testing shows that our approach is scalable and effective

Preprocessing

- Short read data straight from a sequencer is unreliable

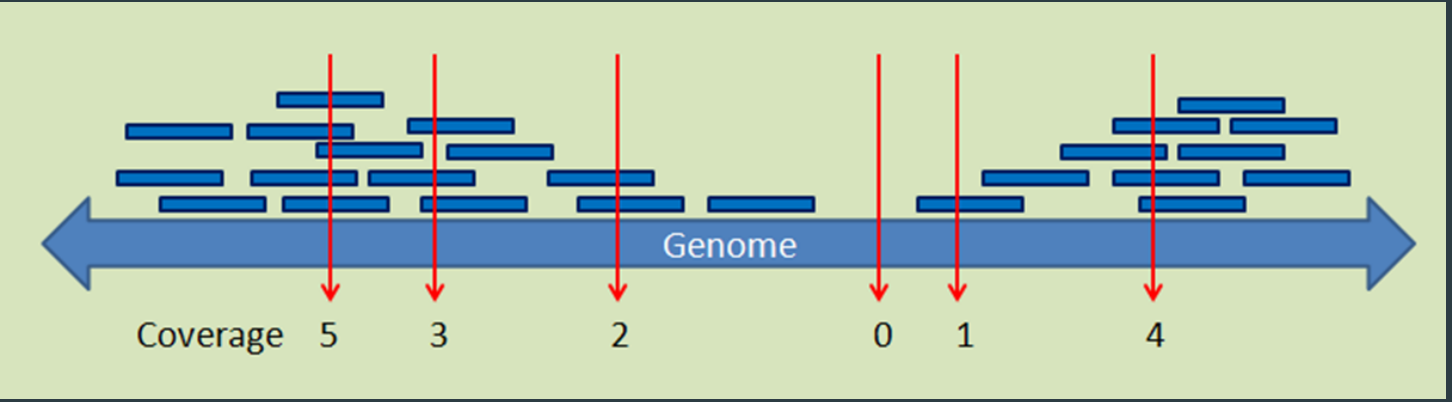
```
@SRR067577.10004404/2
CCATAGATGCCAGAATCTATCCCTGCCCTCGGCGTGAGACCTCTGCTGGGGAACTGGTGTTTGCCGTATCAGACCACAGCAAGGTGTCGTAATGGAGTCTC
+
IIIIIIIIIGIIIGIHHIIGIIIHIHIIGDGEBGBIIIIHIHHGIHBHE<EDIEDDD=DBA@DD@FBEBDDDBEBEDB@FDB>B4?08?;?;>?>B@9D##
@SRR067577.10029246/1
CTGAATCCCTTTTTCCATGGATAGCTTTCTGGTAGACCATAAATGAAAGCATGGCCACAGCATCTTGGCAGCCAGCACAGACCCATGGAACTCTTGGCCAG
+
HHGHGHHHHGHHHDHGHHEB<FBEG<BGGBDDAGFHHHBHHHHGBDD3GDGEBF@HHHBHGDDF<D>DGBDFFEFFBEEHCDF8E38?=>DADA#######
```

Examples of read data. Each read is four lines long. The lines are as follows: Identifier, Read, Optional Description, Quality Score

- Two major approaches: trimming, error correction
- Goal is to improve average quality of reads used

Methods

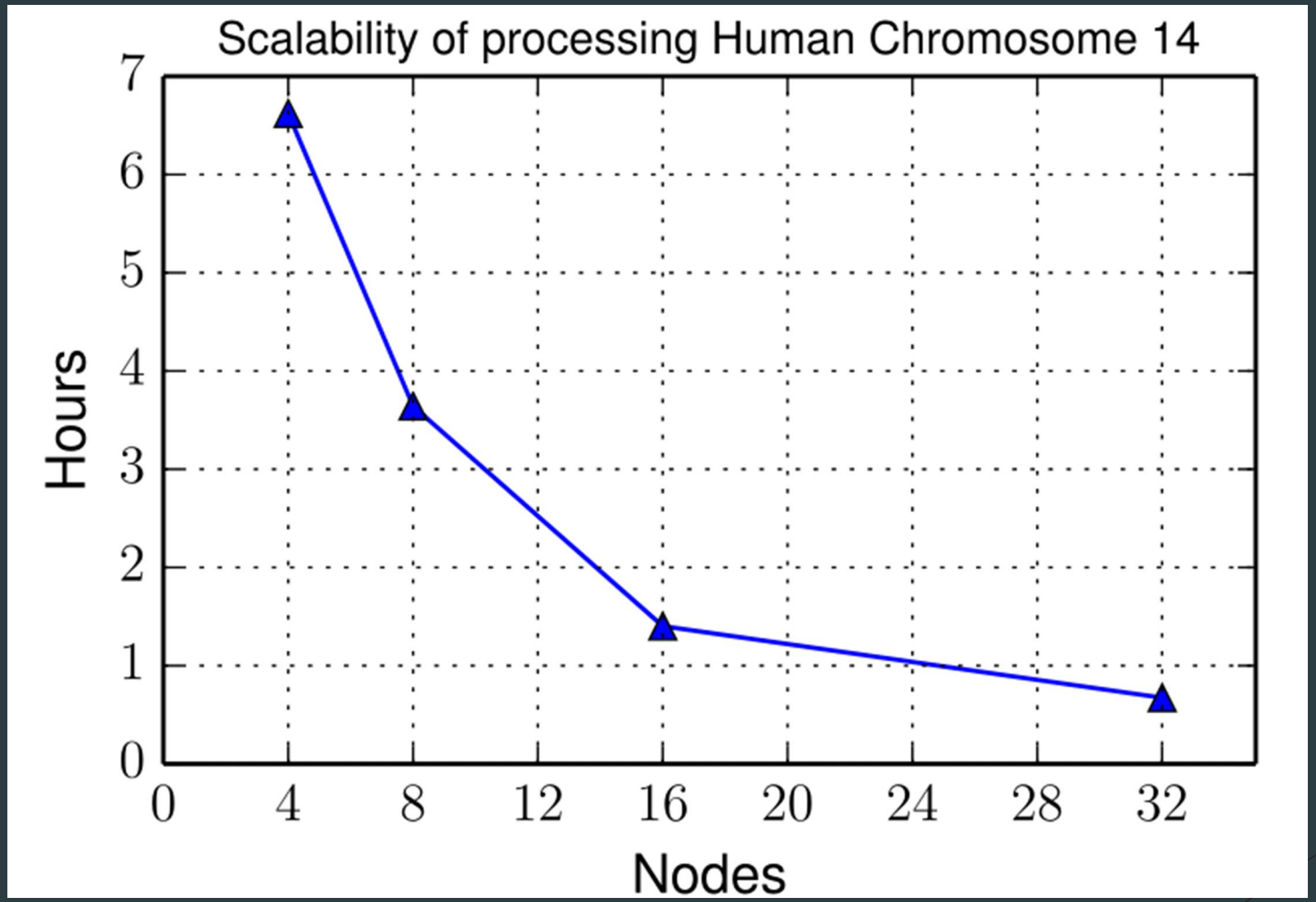
- Count *k*-mers and filter based on *k*-mer frequency



- Four steps:
  1. Count *k*-mers
  2. Reorganize by read identifiers
  3. Filter reads based on their *k*-mer coverage
  4. Apply the filter to the original read data

Results

- Tested scalability by timing process on human chromosome 14 data set and doubling nodes used: 4, 8, 16, 32
- Tested accuracy by using Velvet assembler to determine assembly quality before and after preprocessing.


Scalability of processing Human Chromosome 14
Hours
Nodes
0
1
2
3
4
5
6
7
0
4
8
12
16
20
24
28
32

Assembly of Read Data		
	Unfiltered	Filtered
Nodes in Graph	4419219	2792604
N50	249	599
Max Length	6760	9524

Other Work

- Implemented other trimmers:
  - Average quality filter
  - Sliding window filter
  - *Q*-mer counter
- Started writing simple Spark scripts for *k*-mer analysis

References

1. Sharafat, Ali. "De Novo Assembly." *Stanford Artificial Intelligence Laboratory*. Web. 29 May 2015.
2. Chen et al.: Software for pre-processing Illumina next generation sequencing short read sequences. Source Code for Biology and Medicine 2014 9:8.
3. Zerbino, D. R., and E. Birney. "Velvet: Algorithms for De Novo Short Read Assembly Using De Bruijn Graphs." Genome Research 18.5 (2008): 821-29. Web.
4. Kelley, David R., Michael C. Schatz, and Steven L. Salzberg. "Quake: Quality-aware Detection and Correction of Sequencing Errors." Genome Biology 11.11 (2010): n. pag. Web.
5. Salzberg et al. "GAGE: A Critical Evaluation of Genome Assemblies and Assembly Algorithms." *Genome Research* 22.3 (2012): 557-67. Web.

Acknowledgements

- This material is based upon work supported by the National Science Foundation under award OCI-1263236 with additional support from the Center for Computation & Technology at Louisiana State University.