BIGGIE: A Distributed Pipeline for Genomic Variant Calling

Richard Xia[1], Sara Sheehan[1], Yuchen Zhang[1], Ameet Talwalkar[1], Matei Zaharia[1] Jonathan Terhorst[2], Michael Jordan[1,2], Yun S. Song[1,2], Armando Fox[1], David Patterson[1]

[1] Computer Science Division, UC Berkeley; [2] Department of Statistics, UC Berkeley

Motivation: faster, open-source genome variant calling tools

Impact:

Human genome variation is being used more and more to impact disease diagnosis and treatment, however:

- current tools frequently disagree on variant calls
- different types of variation require specialized tools

Current Tools:

- GATK [2]: slow and difficult to use
- CASAVA [1]: fast, but not free
- samtools mpileup [3]: slow and some accuracy issues

Our Goal:

- fast, distributed variant caller
- separate the genome into regions of high and low complexity and use the right tool for the right region

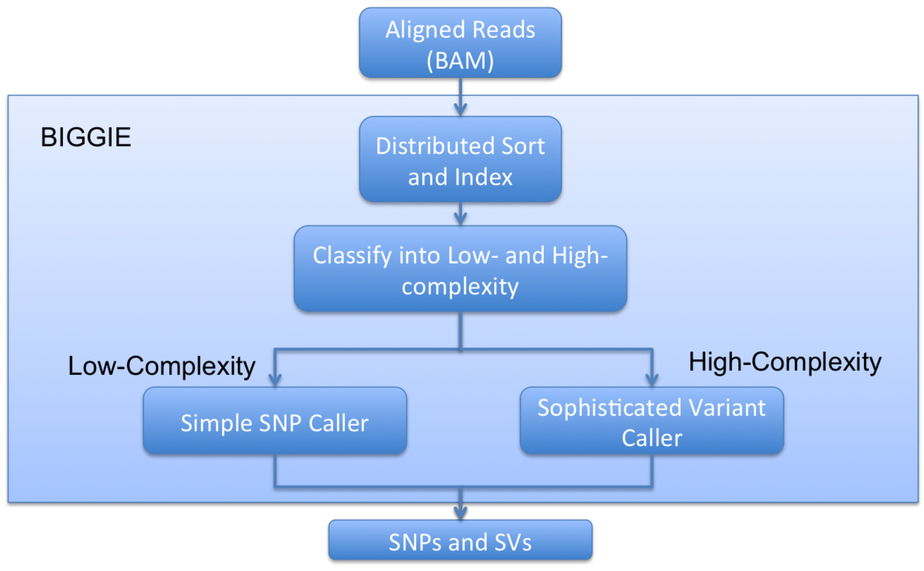

Figure 1: BIGGIE pipeline

Per-base SNP caller

Main idea:

- Distrubted pipeline for variant calling using Spark [4]
- Assign a *complexity* score to each base
- Use a simple SNP caller at bases with a low complexity score
- Use more robust structural variant callers at high complexity bases

Complexity region examples:

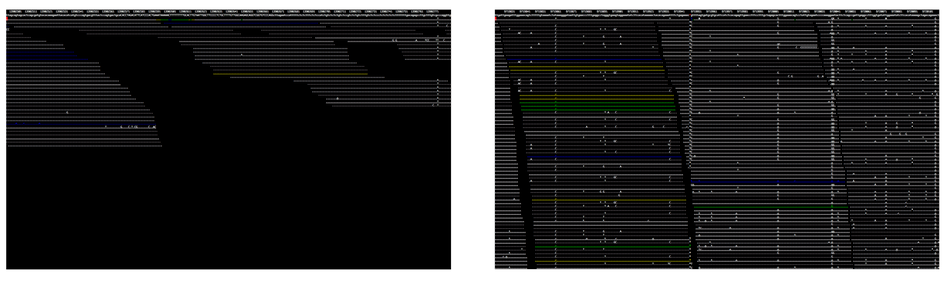

Figure 2: Different variant calling tools should be used for regions of the genome.

Complexity score features:

Name	Weight	Description
Substitution	3	Number of aligned reads showing a substitution with respect to the reference.
Insertion	10	Number of aligned reads showing an insertion with respect to the reference.
Deletion	10	Number of aligned reads showing a deletion with respect to the reference.
Low Quality	3	Number of reads aligned with low map quality (a common indicator of a repetitive region).

Table 1: Relative weight of features for computing complexity.

Results

Simulating data:

- Used reads simulated from the consensus sequence for Venter's genome
- Better approximates the true pattern of SNPs, indels, and structural variants found in a true genome; reads were aligned using BWA and SNAP

Effect of thresholds:

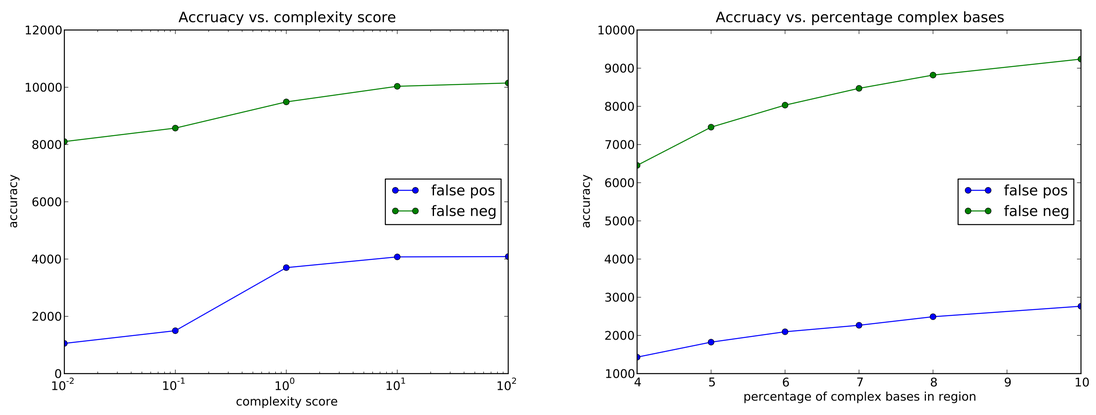

Figure 3: On the left are the per-base results, measuring false negatives only on the regions we called. Both accuracy measures increase as the threshold increases, but the number of correct calls increases as well. We see a similar pattern on the right for the region results, where the number of false positives and false negatives increase with the density of complex bases in high-complexity regions, but the number of true calls increases as well.

Incorporating high complexity regions

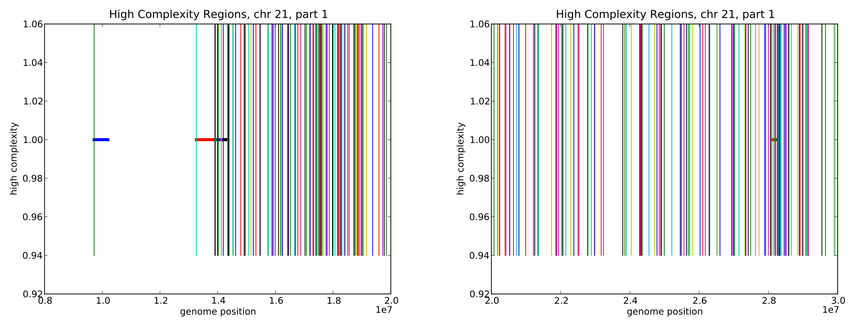

Figure 4: Regions are fairly uniformly distributed, except near the chromosome ends.

- We group bases into a high-complexity region in a greedy fashion, maintaining that the overall high-complexity base density is $> t$
- We filter out regions that are < 500 bases long

Stats, $t = 5\%$	
Number of high complexity regions	3603
Percentage of genome is high complexity regions	16.6%

Results

Timing Results:

Algorithm	Runtime
GATK	35m 17s
mpileup	49m 53s
BIGGIE	4m 38s

Table 2: Timing results for GATK, mpileup, and BIGGIE. The runtime is not significantly impacted by the complexity threshold.

Low vs. High Complexity:

region type	false pos	false neg	correct
low-complexity	1824	7455	38232
high-complexity	2289	2788	13046

Table 3: Our performance degrades in the high complexity regions, which is why a special purpose variant caller should be used.

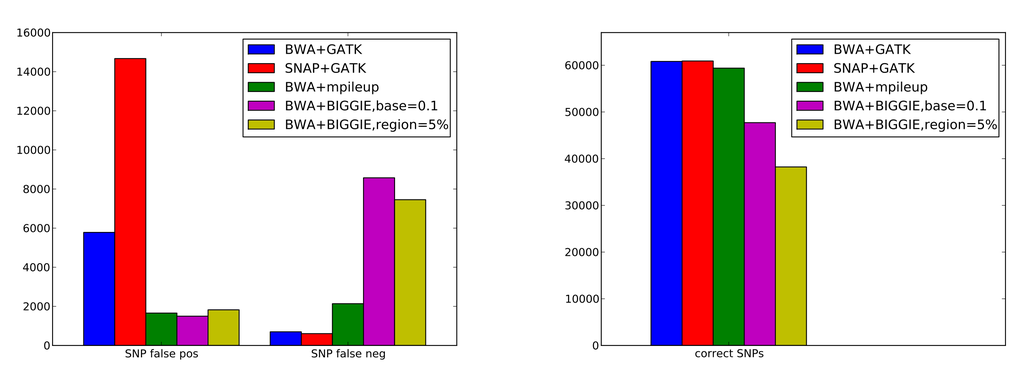

Figure 5: Accuracy comparison of BIGGIE with mpileup and GATK. False positives in BIGGIE are often associated with alignment errors or confusion with a small indel. For each algorithm, a very small percentage of correct SNP bases actually have the incorrect (unphased) genotype.

Future Work: Use the high and low complexity regions to distribute the reads across machines, then call variants using appropriate algorithms.

References

[1] CASAVA. (2012) http://support.illumina.com/sequencing/sequencing_software/casava.ilmn.
[2] DePristo M. *et al*, "A framework for variation discovery and genotyping using next-generation DNA sequencing data." *Nature Genetics* (2011), 43:491-498.
[3] Li H. *et al* and 1000 Genome Project Data Processing Subgroup, "The Sequence alignment/map (SAM) format and SAMtools." *Bioinformatics* (2009), 25: 2078-9.
[4] Zaharia M. *et al*, "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing." *NSDI* (2012).

{rxia, ssheehan, yuczhang}@eecs.berkeley.edu