**Title:** CS-BWAMEM: A fast and scalable read aligner using cloud-computing technology

**Authors and affiliations:**

1Yu-Ting Chen, 1Jason Cong, 2Myron Peto, 2Paul Spellman, and 1Peng Wei

1Computer Science Department, University of California, Los Angeles, CA 90095, USA

2Department of Molecular and Medical Genetics, Oregon Health and Science University, Portland, OR 97239, USA

**Corresponding author:**

Yu-Ting Chen

4732 Boelter Hall, Los Angeles, CA 90095, USA

Tel: 310-994-4108; E-mail: ytchen@cs.ucla.edu

**Abstract (word limit: 70)**

As the depth of whole-genome sequencing keeps increasing, aligning hundreds of millions of short reads on a single server can take more than 10 hours. We introduce cloud-scale BWAMEM (CS-BWAMEM), an ultrafast and scalable aligner, to align reads by leveraging the cloud-computing technology either in private or public clouds. CS-BWAMEM is the first aligner that can complete 30x coverage whole-genome alignment within one hour and whole-exome alignment within ten minutes.

**Introduction**

Aligning the reads to a reference genome is the first step and usually one of the most time-consuming steps in many genomics pipelines, such as the pipelines for variant discovery1 and differential gene expression2. Recent high-throughput sequencing systems, such as Illumina HiSeq X Ten, can deliver more than 18,000 genomes annually. As the data throughput increases more than doubling each year, a fast and scalable aligner is needed to process the ever-increasing data.

The state-of-the-art aligners, such as BWA-SW3, Bowtie 24, and BWA-MEM5, can perform gapped-read alignment while using pair-end reads to improve alignment quality. These aligners usually proceeds in two steps for each read. The first step is called seeding. The full-text minute index6 (FM-index) and Burrows-Wheeler transform7 (BWT) are used to find the candidate locations of a read on the reference genome very efficiently. These ungapped and exact-matched alignments collected from a read are used as seeds in the second step. To allow gaps, such as mismatches, insertions, and deletions, the Smith-Waterman algorithm8 is applied to extend the seeds with gaps. The score of each can be used to filter out poor gap-allowed alignments. However, the seed extension step involves significant computation since dynamic programming is used. Recent aligners, like Bowtie 2 and BWA-MEM, leverage the efficiency of single-instruction multiple-data (SIMD) built in modern processors to accelerate this compute-intensive part.

Though the efficiency of the aligners can be significantly improved by FM-index aided method together with the SIMD acceleration on dynamic programming algorithm, the tremendous number of reads delivered from sequencers nowadays still takes significant amount of time to be aligned using existing aligners. Existing aligners use multi-threaded parallelization to boost performance. However, they can only utilize the computation power from a single server, which limits the degree of parallelism to the number of CPU threads. For the deep-coverage whole-genome sequencing (WGS) data, billions of reads can be generated in one run. The alignment of 30x coverage WGS data can take more than 10 hours if BWA-MEM5 is used on a single server with two state-of-the-art Intel six-core processors. In fact, the alignment process is highly parallelizable. But we haven’t seen the prior work that can efficiently use private or public clouds to fully exploit the huge parallelism arising from hundreds of millions of reads.

In this work we developed a new aligner, CS-BWAMEM, which can leverage the abundant computing resources in private or public clouds. CS-BWAMEM leverages the BWM-MEM algorithm and is developed under MapReduce9 programming model, one of the most successful models in developing distributed software packages for modern datacenters. CS-BWAMEM is built on top of Apache Spark10, which realizes a highly efficient MapReduce system that can cache intermediate data in memory to avoid unnecessary disk I/O accesses. Using Spark, CS-BWAMEM can distribute read alignment tasks to multiple servers within a cluster, providing the capability to accelerate alignment process beyond one single server. Therefore, CS-BWAMEM can provide scalable and ultrafast alignment speed. Users can deploy CS-BWAMEM based on the computing resources they have in a cluster or the runtime target they set up. With such scalability, CS-BWAMEM can finish deep-coverage (30x) whole-genome sequencing within 40 minutes by using a 30-node cluster.

CS-BWAMEM provides two major functions. First, the large FASTQ files stored in the local Linux file system can be automatically partitioned into small file fragments and then uploaded to a distributed file system in a cluster. CS-BWAMEM uses Hadoop distributed filesystem11 (HDFS) as the storage backend. Second, the small FASTQ fragments are fetched from HDFS and aligned in parallel across the allocated servers in the cluster.



**Figure 1** CS-BWAMEM flow overview. CS-BWAMEM is composed of two Map/Reduce stages with six steps.

For each read, CS-BWAMEM proceeds two MapReduce stages, which is composed of six steps, as shown in **Figure 1**. In the first map stage, CS-BWAMEM first uses super maximal exact match (SMEM) method12 to generate the seeds from the read. The generated seeds are then extended by the Smith-Waterman-like dynamic programming algorithm. The estimated insert distance of all the reads in this group is calculated in the first reduce stage. In the second map stage, the pair-end alignments are performed and then the output data are written back to HDFS in the distributed format. Users can also collect the alignment data in the widely used SAM format13.

**Results**

To demonstrate the speed of CS-BWAMEM on real whole-genome data, we compare CS-BWAMEM to BWAMEM and Bowtie 2. In all experiments, we used the reference genome of the phase 1 analysis from 1000 Genomes project.

We first evaluate the performance of CS-BWAMEM on real whole-genome data with 30x coverage. We used 150-by-150 nucleotide (nt) paired-end HiSeq reads obtained from an Autism study, with 359.4 millions of pair-end reads. We compare the alignment time with the state-of-the-art aligners, BWA-MEM and Bowtie2. By leveraging the cloud computing technology, CS-BWAMEM outperforms BWM-MEM and Bowtie2 by 18x and 20.3x, respectively (**Fig. 2a**). For this whole-genome data sample, CS-BWAMEM can complete 30x coverage whole-genome alignment in 32 minutes, which demonstrates that CS-BWA-MEM is especially useful for processing deep-coverage WGS. More importantly, CS-BWAMEM can provide scalable alignment speed. In this case, users can determine their cluster size based on the target alignment speed. **Figure 2b** shows the runtime to align a 30x coverage whole-genome data from five to 30 nodes. The performance scales well when more computing resources are provided.

To further access the alignment performance, we use CS-BWAMEM to align 13 30x coverage pair-end whole-genome data from an Autism study, and 6 pair-end whole-exome research samples (**Supplementary Figure 1**). CS-BWAMEM can complete whole-genome sequencing from 32 to 49 minutes and whole-exome sequencing from 5.8 to 7.5 minutes.

Next, we validate the alignment quality of CS-BWAMEM is with BWA-MEM (**Supplementary Table 1**). We use the read simulator, ART14, to generate 16 millions of simulated Illumina HiSeq 2500 pair-end reads. We generate both 100bp and 150bp reads, which are commonly used in nowadays’ datasets. We can observe that the Q20 reads, i.e. the reads with mapping quality score (MAPQ) larger than or equal to 20, are similar in both aligners. The error rate here stands for the percent of the wrong alignment out of Q20 reads. We can observe that CS-BWAMEM achieves similar alignment quality to BWA-MEM.

To further access the result quality, we compare the alignment results of CS-BWAMEM with BWA-MEM on two real whole-exome samples. One of the sample is collected from primary cancer tumor while the other is a buffy sample. **Supplementary Figure 2** shows distribution of alignment over different MAPQs. Both exome samples show that CS-BWAMEM achieves similar alignment quality. We also compare the variant calling results of using alignment data from both CS-BWAMEM and BWA-MEM (**Supplementary Figure 3**). We apply the alignment results of the two exome samples on variant discovery flow and use Mutect15 for variant calling.



**(a)**

****

**(b)**

**Figure 2.** (**a**). Performance comparison between BWA-MEM, Bowtie 2, and CS-BWAMEM on a 30x coverage whole-genome data. (**b**). The alignment time of CS-BWAMEM m a five-server cluster to a 30-server cluster.

We also successfully deploy CS-BWAMEM on an open cloud, the Google Compute Engine (GCE). **Supplementary Figure 4** demonstrates the alignment time and the corresponding costs of using different numbers of cores in GCE. **Supplementary Figure 5** shows the cost and throughput estimation on both private and public clouds.

CS-BWAMEM is an open-source tool. One can download CS-BWAMEM from: <https://github.com/ytchen0323/cloud-scale-bwamem>.

**References**

1. McKenna, A., *et al.* The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome Res.*, **20**, 1297–303 (2010).
2. Trapnell, C., *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* **28**, 511-515 (2010).
3. Li, H. & Durbin, R. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics* **26**, 589–595 (2010).
4. Langmead, B. & Salzberg, S. Fast gapped-read alignment with bowtie 2. *Nature Method* **9**, 357–359 (2012).
5. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv*:1303.3997v1 (2013).
6. Ferragina, P., & Manzini, G. Opportunistic data structures with applications. *Proc. 41st Annual Symposium on Foundations of Computer Science*, 390-398 (2000)
7. Burrows, M., & Wheeler, D. J. A block-sorting lossless data compression algorithm. *Technical report* **124**, Palo Alto, CA, Digital Equipment Corporation.
8. Smith, T. F. & Waterman, M. S. Identification of common molecular subsequences. *Journal of Molecular Biology* **147**, 195-197 (1981).
9. Dean, J. & Ghemawat, S. MapReduce: simplified data processing on large clusters. *USENIX OSDI*, 10-10 (2004).
10. Zaharia, M., *et al.* Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. *USENIX* *NSDI*, 2-2 (2012).
11. Shvachko, K., *et al.* The Hadoop distributed file system. *MSST*, 1-10 (2010).
12. Li, H. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics* **28**, 1838-1844 (2012).
13. Li, H., *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078-2079 (2009).
14. Huang, W., *et. al.* ART: a next-generation sequencing read simulator. *Bioinformatics* **28**, 593-594 (2012).
15. Cibulskis, K., *et. al.* Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nature Biotechnology* **31**, 213-219 (2013).