P-Dindel: A multi-thread based tool for calling indels from short reads

Mohammad Shabbir Hasan, Liqing Zhang

Department of Computer Science
Virginia Tech, Blacksburg, VA 24061, USA.
E-mail: shabbir5@vt.edu, lqzhang@vt.edu

Abstract. Insertion and deletion (indel) of DNA bases is the second most common forms of genetic variation in human genomes and is linked to various genetic diseases and cancers. With next generation sequencing technology, indels are identified through short read sequence alignment and subsequent indel calling. The open source indel calling program Dindel has relatively high sensitivity yet prohibitive running time. To accelerate indel calling of Dindel, we introduce P-Dindel, a multi-thread based implementation of Dindel. Results show that the proposed algorithm achieves 4X speed up for both diploid samples and pooled samples compared to Dindel, while producing the same result as Dindel.

Keywords: Indel Calling; Variant Calling; Next Generation Sequencing; Dindel; Deep Sequencing.

1 Introduction

Out of different types of genetic variation taking place in the human genome, indel is the second most common form [1] and constitutes almost one fourth of the sequence polymorphisms. Indels have been shown to be linked to a number of genetic diseases such as Cystic fibrosis, Fragile X Syndrome, Bloom Syndrome, acute myeloid leukemia etc. Because of its great influence in human traits and diseases, indel analysis has become a growing field of interest in genetic variation assessment.

With the development of the Next Generation Sequencing (NGS) technology, an increasing number of indels have been identified in humans. Short reads generated from the NGS platform are first aligned to a reference genome. From the alignments, indels are inferred/called by an indel calling tool. Various tools have been developed so far and performance of some of these tools has been evaluated with different criteria including read depth, read length, indel size, indel frequency, number of indels called, comparison to the set of "gold standard" indels [2, 3]. Observations in these studies showed that Dindel has the highest sensitivity at low coverage among these tools for both real and simulated data [2, 3]. However compared to others, the running time of Dindel is too high and increases quadratically with the increase of depth of coverage [2, 4]. In this paper we propose P-Dindel, a multi-thread implementation of Dindel that achieves 4X speed up for both diploid and pooled samples.

2 Methods

Dindel [4] is a software tool developed by the Wellcome Trust Sanger Institute in UK. It is one of the variant calling programs used in the 1000 Genomes project [5]. This open source program utilizes Bayesian network for calling indels from NGS data. Among all the steps of indel calling by Dindel, the most time consuming one is the realignment step where Dindel tests all potential indels identified by the read mapper to detect potential sequencing error. With the increase of sequencing errors, the realignment model scales linearly with the number of reads and increases the computational time. Hence our goal is to parallelize this step to reduce the overall computational time.

The Simultaneous Multithread (SMT) is one of the best parallel multithreading techniques for thread level parallelism. POSIX thread (Pthread) is one of the most popular thread based APIs that use shared memory and support a variety of programming languages. In our proposed algorithm, we use Pthread for several reasons. Firstly, Pthread has a lower execution time than other thread based APIs and obtains higher speed up [6]. Secondly, in Pthread, threads are created only once and can be used many times with explicit synchronization among threads. Thirdly, Pthread ensures fine-tuned user control over parallelization. Table 1 shows the proposed algorithm.

Since the number of candidate indels in each realignment window is the same, the proposed algorithm balances the efficiency of having a large number of windows with the desire to have all of the cores finish the realignment at the same time, without having some exit early for lack of work while others are still processing.

Table 1. Algorithm of P-Dindel

Algorithm 1: P-Dindel (ref, alignment_bam)

Input: Reference Sequence (ref), Alignment file (alignment_bam)
Output: List of indels

1. Extract indels from the input BAM file (alignment_bam), these are candidate indels
2. Create realignment window files and assign equal number of candidate indels to each of the realignment windows
3. $w \leftarrow$ number of realignment windows
4. $c \leftarrow$ number of physical cores available in the system
5. $t \leftarrow c \times$ number of threads assigned to each core // t is the total number of logical cores
6. $r \leftarrow \frac{w}{t}$ // r is the number of realignment windows assigned for each thread
7. Create T, a set of slave threads having t number of threads. Master thread enters the waiting state.
8. for each thread $t_s \in T$
   (a) Get the portion of the realignment windows. //each thread will have r realignment windows.
   (b) for $i = 1$ to r
      (i) Realign mapped reads and their unmapped mates to candidate haplotypes consisting of the indels in that realignment window i.
      (ii) Using Bayesian inference, check for sufficient evidence for an alternative haplotype. This way it separates candidate indels from sequencing errors.
      (iii) After all the realignment is done, thread t_s enters the waiting state.
9. After the last thread is done with the realignment, it signals the master thread before entering the waiting state.
10. Master thread integrates the results from all realignment windows and produces the final result that includes all indels for the input alignment file.

2.1 Dataset

For diploid samples, the dataset consists of alignment profiles from eight humans with coverage ranging from ~3X to ~42X. For pooled samples, however, we used low coverage samples (average coverage = ~3.73X). All of these samples were collected for the 1000 Genomes project [5]. The short reads of these samples were sequenced on Illumina Genome Analyzer platform [7] and mapped to the human reference genome using BWA [8].

2.2 System Configuration

A Dell machine with Linux operating system was used to run both P-Dindel and Dindel for all test cases. The machine was equipped with Intel Core i7-2600 CPU @ 3.40 GHz * 8 processors, 16 GB RAM and Ubuntu 12.04 LTS operating system.

3 Results and Discussion

Total execution time for P-Dindel and Dindel are shown in Fig. 1 for diploid samples and pooled samples. In Fig. 1 (a), the comparison is done based on the execution time related to the coverage of the samples. Since Dindel tests all candidate indels to separate the actual indels from sequencing errors, with the increase in the coverage of the samples, the computation time increases quadratically [4]. However, for P-Dindel, the increase is not as rapid as Dindel as indicated in Fig. 1 (a).

For pooled samples, we compared the execution time of P-Dindel and Dindel related to the number of samples in the pool. From Fig. 1(b), we can see that P-Dindel has much more lower execution time than Dindel when the number of samples in the pool increases gradually. We calculated the speed up and found that the average speed up for diploid samples was 3.47X and for pooled samples, it was 4.11X. Due to communication overhead, it is not possible to get linear speed up with the increase in the number of cores which is the case in all problems related to high performance computing.

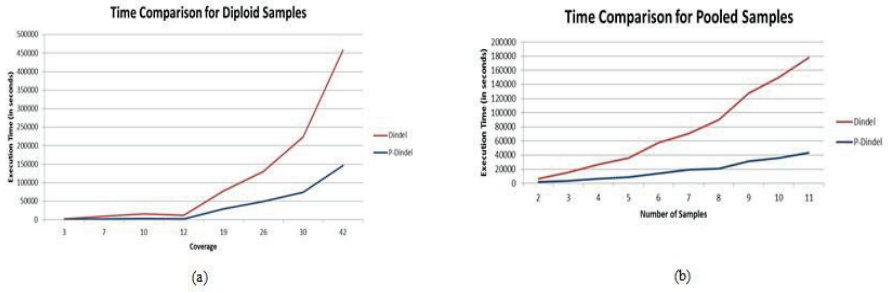

Fig. 1. Comparing execution time of P-Dindel and Dindel for (a) diploid samples and (b) pooled samples

4 Conclusion

In human genome, indel is one of the most common forms of disease causing genetic variants. Hence identifying indels in an efficient manner through NGS data is very important for identification of disease causing variations. A good number of tools have been developed to identify indels. Recent research works showed that among the existing indel calling tools, Dindel is the most promising one. However, the main limitation of Dindel is its running time which increases exponentially with the increase of read coverage. Here we propose a tool named P-Dindel, an improved version of Dindel to reduce the execution time. P-Dindel parallelizes the realignment steps and distributes the realignment windows among the threads that are created based on the cores available in that system. For this parallelization, it uses Pthread that ensures fined-grained user control over parallelization and has been identified as one of the best techniques for parallel multithreading. Using the proposed algorithm, P-Dindel achieved, on average, 4X speed up than Dindel for both diploid and pooled samples. Results show that the speed up improves gradually with the increase of read coverage. This improvement is also noticeable when the number of samples increases for pooled data. Hence P-Dindel can be considered to be a very auspicious tool for discovering indels in human genomes. In future, we plan to extend this approach to design a haplotype based indel caller for more accurate indel prediction.

References

1. J. M. Mullaney, R. E. Mills, W. S. Pittard, and S. E. Devine, "Small insertions and deletions (INDELs) in human genomes," *Human molecular genetics,* vol. 19, pp. R131-R136, 2010.
2. M. S. Hasan, X. Wu, and L. Zhang, "Performance evaluation of indel calling tools using real short-read sequences," *Submitted,* 2015.
3. J. A. Neuman, O. Isakov, and N. Shomron, "Analysis of insertion–deletion from deep-sequencing data: software evaluation for optimal detection," *Briefings in Bioinformatics,* vol. 14, pp. 46-55, 2013.
4. C. A. Albers, G. Lunter, D. G. MacArthur, G. McVean, W. H. Ouwehand, and R. Durbin, "Dindel: accurate indel calls from short-read data," *Genome research,* vol. 21, pp. 961-973, 2011.
5. M. Via García and G. P. Consortium, "An integrated map of genetic variation from 1,092 human genomes," *Nature, 2012, vol. 491, p. 56-65,* 2012.
6. W. Zhong, G. Altun, X. Tian, R. Harrison, P. C. Tai, and Y. Pan, "Parallel protein secondary structure prediction schemes using Pthread and OpenMP over hyper-threading technology," *The Journal of Supercomputing,* vol. 41, pp. 1-16, 2007.
7. D. R. Bentley, S. Balasubramanian, H. P. Swerdlow, G. P. Smith, J. Milton, C. G. Brown, *et al.*, "Accurate whole human genome sequencing using reversible terminator chemistry," *Nature,* vol. 456, pp. 53-59, 2008.
8. H. Li and R. Durbin, "Fast and accurate long-read alignment with Burrows–Wheeler transform," *Bioinformatics,* vol. 26, pp. 589-595, 2010.