Bioinformatics Advance Access published June 4, 2013

Isaac: Ultra-fast whole genome secondary analysis on Illumina sequencing platforms

Come Raczy[1*], Roman Petrovski[1], Christopher T. Saunders[2], Ilya Chorny[2], Semyon Kruglyak[2], Elliott H. Margulies[1], Han-Yu Chuang[2], Morten Källberg[2], Swathi A. Kumar[2], Arnold Liao[2], Kristina M. Little[2], Michael P. Strömberg[2], Stephen W. Tanner[2]

[1]Illumina United Kingdom, Chesterford Research Park, Little Chesterford, Nr Saffron Walden, Essex, CB10 1XL, UK

[2]Illumina, Inc., 5200 Illumina Way, San Diego, CA 92122

Associate Editor: Prof. Martin Bishop

Abstract
An ultrafast DNA sequence aligner (Isaac Genome Alignment Software) that takes advantage of high memory hardware (>48GB) and variant caller (Isaac Variant Caller) have been developed. We demonstrate that our combined pipeline (Isaac) is 4-5 times faster than BWA+GATK on equivalent hardware, with comparable accuracy as measured by trio conflict rates and sensitivity. We further show that Isaac is effective in the detection of disease-causing variants and can easily/economically be run on commodity hardware.

Availability: Isaac has an open source license and can be obtained at https://github.com/sequencing

Contact: craczy@illumina.com

Supplementary Information:
TBD

Contributions:
IC wrote the manuscript and performed the analysis. CR and RP developed the Isaac aligner. CTS developed the Isaac variant caller. Additional authors contributed to the data analysis.

1 INTRODUCTION

Motivated by a growing need for faster turnaround times for whole genome sequencing (WGS) data analysis, we present here a novel alignment and variant calling pipeline that is able to rapidly align WGS data and deliver high quality variant calls on a single server node. The aligner, Isaac Genome Alignment Software, is designed to align next generation sequencing (NGS) data with low error rates (single or paired-ends).Speed improvements come from the fact that the Isaac aligner has been designed to take full advantage of all the computational power available on a single server node. As a result, the Isaac aligner scales well over a broad range of hardware architectures, and alignment performance improves with hardware capabilities (i.e. clock speed, number of cores, IO bandwidth and memory). The typical end-to-end time to align a ~30-40x human data set from BCL or FASTQ files to a sorted and duplicate-marked BAM file is approximately 4 hours on an Amazon High-Memory Quadruple Extra Large Instance and can be as fast as 2 hours on an optimized high end server (see Supplement for specs). Beyond speed and scalability, the Isaac aligner also delivers ease-of-use, flexibility, and robustness. The creation of sorted, duplicate-marked BAM files from BCL or FASTQ files is done in a single operation, alleviating the need to rewrite large BAM files multiple times in a typical workflow. Additional command-line options are available to the expert user to finely control the algorithms inputs, outputs, and computational performance (Section S1 of the supplementary materials describes the details of the Isaac aligner algorithm and its implementation).

The Isaac Variant Caller calls SNPs and small indels using a Bayesian framework to compute probabilities over diploid genotype states. The Isaac Variant Caller uses an internal read realignment routine to improve variant call accuracy near indels and includes a site specific error dependency term (Section S2 of the supplementary materials provides a detailed explanation of the Isaac Variant Caller algorithm and implementation). The Isaac Variant Caller is designed to efficiently genotype and provide output for all variant and non-variant genomic loci as Genome VCF files (gVCF; Saunders et al, manuscript in prep; https://sites.google.com/site/gvcftools/), a convention for efficiently representing whole genome output in VCF format (http://www.1000genomes.org/node/101).

To demonstrate the performance of the Isaac aligner and variant caller pipeline (Isaac), we compare the quality of the variant calls and the time-to-answer of this pipeline to the community standard combination of Burrows-Wheeler Alignment (BWA) (Li and Durbin, 2009; Li and Durbin, 2010) and the Genome Analysis Tool Kit (GATK) (DePristo, et al., 2011; McKenna, et al., 2010). We also demonstrate that Isaac can successfully detect a clinically deleterious variant in a neonatal sample (Saunders, et al., 2012).

2 METHODS

2.1 Software

BWA can be obtained from http://bio-bwa.sourceforge.net/. GATK can be obtained from http://www.broadinstitute.org/gatk/. Isaac can be obtained from https://github.com/sequencing and are subject to the Illumina open source license.

2.2 Alignment and Variant Calling

The details of the alignment and variant calling pipelines are discussed in section S3 of the supplemental material. Briefly the aligner / variant caller combinations are Isaac and BWA+GATK. For Isaac indel realignment is performed by the Isaac aligner while for BWA+GATK, indel realignment is performed post alignment using GATK. For the GATK variant calling, the GATK best practices is used which involves variant calling using the Unified Genotyper followed by filtering with the variant quality score recalibration (VQSR) protocol(McKenna, et al., 2010).

2.3 Data sets

Two data sets were used for the analysis. The first data set, used for the comparison of Isaac and BWA+GATK, is a human family trio selected from the 1000 Genomes project (Genomes Project, 2010). The trio consists of CEPH family members NA12878 (Child), NA12891 (Father), and NA12892 (Mother). This data set was used to evaluate the variant call quality by assessing the number of Mendelian SNP conflicts, the SNP conflict rate, and the sensitivity (% callable bases) of each pipeline.

The second data set is a neonatal sample (UDT173) used for genetic disease diagnosis (Saunders, et al., 2012). This data set was used to demonstrate that Isaac can be effectively used to isolate clinically relevant variants.

In addition to evaluating the quality of the variant calls, the performance in wall clock time of each pipeline on equivalent computer hardware architectures is reported.

The CEPH DNA was obtained from Coriell Institute and sequenced internally on a HiSeq 2000. The neonatal sample was sequenced on a HiSeq 2500. PCR-free sequencing methods were used in for all the samples analyzed (Saunders, et al., 2012).

2.4 Hardware Specifications

Alignment and variant calling was performed on commodity hardware comprised of a single computer node having 65 GB of RAM and containing two 8 core Intel® Xeon® CPU E5-2650 @ 2.00GHz processors. Hyper threading was activated resulting in 32 virtual cores. To run the Isaac aligner a minimum of 48GB of RAM is required whereas BWA requires a minimum of 3GB of RAM.

3 RESULTS

Table 1 depicts the wall clock time for each of the pipelines. Isaac took approximately 7-8 hours as compared with 43-46 hours for BWA+GATK, demonstrating a significant performance enhancement on equivalent computer hardware. One source of this improved performance is that Isaac does not require generation of fastq files prior to alignment. In general, the generation of fastq files adds an additional 2-3 hours to the BWA+GATK workflow.

Table 1. Wall clock times in hours for Alignment, Indel Realignment, and Variant Calling for Isaac and BWA+GATK

	Sample	*Yield (Gb)*	*Alignment*	*Indel Realignment*	*Variant Calling*
Isaac	NA12878	120	4.46	N/A	1.51
	NA12891	119	5.66	N/A	1.50
	NA12892	129	5.68	N/A	1.58
BWA+ GATK	NA12878	120	32.22	3.55	8.37
	NA12891	119	31.33	3.60	8.12
	NA12892	129	34.55	3.76	8.61

Table 2 compares the quality of the resulting variant calls and the sensitivity of the two pipelines. The number of conflicts was slightly larger for Isaac with a slight reduction in sensitivity.

Table 2. Total number of SNP conflicts, SNP conflict rate, and sensitivity (% of non N reference sites called) of Isaac and BWA+GATK

	Conflicts	*Conflict Rate*	*Sensitivity*
Isaac	6318	0.139%	94.5%
BWA+GATK	5315	0.126%	95.8%

Additional alignment and variant metrics are shown in the supplemental section S.4.

To demonstrate Isaac's clinical utility, we analyzed a genome with a previously confirmed novel disease causing mutation in ATP7A, causing Menkes Disease (Saunders, et al., 2012). In order to show that the results of Isaac are capable of being equivalently filtered to identify the correct disease-causing mutation, we generated small variants from the same genome sequence data using Isaac. The variants went through an annotation pipeline (Variant Effect Predictor (VEP), 1000 genomes, Human Gene Mutation Database (HGMD))(Genomes Project, 2010; McLaren, et al., 2010; Stenson, et al., 2003) and produced results that also identified the correct disease-causing variant.

Table 3. Variant filtering results with Isaac

Isaac	*Applied Filter*
13,212	Transcripts with variants
1,136	Transcripts containing two or more autosomal variants, or one variant on chrX or chrY; <5% allele frequency
147	Variants altering the protein coding sequence
16	Variants overlapping a medically relevant gene
6	Variants predicted to be deleterious
5	Variants excluding splice site variants
3	Evolutionarily conserved variants
1	Homozygous/hemizygous variants (*disease-causing variant*)

Note: Filters are applied consecutively.
Variants altering the protein coding sequence are those that are non-synonymous, frame shirt, stop gain/loss, or splice site; Medically relevant genes are those genes with variants in HGMD; Variants predicted to be deleterious are determined by its polyphen score (Ramensky, et al., 2002) and/or SIFT score (Ng and Henikoff, 2003)

REFERENCES

DePristo, M.A., *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature genetics*, **43**, 491-498.

Genomes Project, C. (2010) A map of human genome variation from population-scale sequencing, *Nature*, **467**, 1061-1073.

Li, H. and Durbin, R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform, *Bioinformatics*, **25**, 1754-1760.

Li, H. and Durbin, R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform, *Bioinformatics*, **26**, 589-595.

McKenna, A., *et al.* (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data, *Genome research*, **20**, 1297-1303.

McLaren, W., *et al.* (2010) Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor, *Bioinformatics*, **26**, 2069-2070.

Ng, P.C. and Henikoff, S. (2003) SIFT: Predicting amino acid changes that affect protein function, *Nucleic acids research*, **31**, 3812-3814.

Ramensky, V., Bork, P. and Sunyaev, S. (2002) Human non-synonymous SNPs: server and survey, *Nucleic acids research*, **30**, 3894-3900.

Saunders, C.J., *et al.* (2012) Rapid whole-genome sequencing for genetic disease diagnosis in neonatal intensive care units, *Science translational medicine*, **4**, 154ra135.

Saunders, C.T., *et al.* (2012) Strelka: accurate somatic small-variant calling from sequenced tumor-normal sample pairs, *Bioinformatics*, **28**, 1811-1817.

Stenson, P.D., *et al.* (2003) Human Gene Mutation Database (HGMD): 2003 update, *Human mutation*, **21**, 577-581.