J. Dairy Sci. 102:3216–3229
https://doi.org/10.3168/jds.2018-15172

Calling known variants and identifying new variants while rapidly aligning sequence data

P. M. VanRaden,[1]* D. M. Bickhart,[1]† and J. R. O'Connell[2]
[1]USDA, Agricultural Research Service, Animal Genomics and Improvement Laboratory, Beltsville, MD 20705-2350
[2]University of Maryland School of Medicine, Baltimore 21201

ABSTRACT

Whole-genome sequencing studies can identify causative mutations for subsequent use in genomic evaluations. Speed and accuracy of sequence alignment can be improved by accounting for known variant locations during alignment instead of calling the variants after alignment as in previous programs. The new programs Findmap and Findvar were compared with alignment using Burrows–Wheeler alignment (BWA) or SNAP and variant identification using Genome Analysis ToolKit (GATK) or SAMtools. Findmap stores the reference map and any known variant locations while aligning reads and counting reference and alternate alleles for each DNA source. Findmap also outputs potential new single nucleotide variant, insertion, and deletion alleles. Findvar separates likely true variants from read errors and outputs genotype probabilities. Strategies were tested using cattle, human, and a completely random reference map and simulated or actual data. Most tests simulated 10 bulls, each with 10× simulated sequence reads containing 39 million variants from the 1000 Bull Genomes Project. With 10 processors, clock times for processing 100× data were 105 h for BWA, 25 h for GATK, and 11 h for SAMtools but only about 4 h for SNAP, 3 h for Findmap, and 1 h for Findvar. Alignment programs required about the same total memory; BWA used 46 GB (4.6 GB/processor), whereas >10 processors can share the same memory in SNAP and Findmap, which used 40 and 46 GB, respectively. Findmap correctly mapped 92.9% of reads (compared with 92.6% from SNAP and 90.5% from BWA) and had high accuracy of calling alleles for known variants. For new variants, Findvar found 99.8% of single nucleotide variants, 79% of insertions, and 67% of deletions; GATK found 99.4, 95, and 90%, respectively; and SAMtools found 99.8, 12, and 16%, respectively. False positives (as percentages of true variants) were 11% of single nucleotide variants, 0.4% of insertions, and 0.3% of deletions from Findvar; 12, 8.4, and 2.9%, respectively, from GATK; and 37, 1.3, and 0.4%, respectively, from SAMtools. Advantages of Findmap and Findvar are fast processing, precise alignment, more useful data summaries, more compact output, and fewer steps. Calling known variants during alignment allows more efficient and accurate sequence-based genotyping.
Key words: sequence alignment, variant calling, indel

INTRODUCTION

Whole-genome sequencing studies can identify causal mutations and variants for subsequent use in genomic evaluations, but accurately aligning DNA sequence reads to the reference map and identifying differences require much computation. Many programs are available for alignment and for calling variants such as single nucleotide variants (**SNV**) and small indels, but few alignment programs use already-known information about the variants (Tithi et al., 2015; Yuan et al., 2015). Current alignment programs use only 1 reference map to represent each species (Li et al., 2008).

Reads that contain variants are the most important for estimating genetic differences among individuals but are also harder to align because the alternate alleles do not match the reference map. Reads that contain indels are often misaligned by current algorithms (DePristo et al., 2011) but could be aligned correctly if indel positions and lengths were already known and used by the alignment algorithm. Yuan et al. (2015) used alternate alleles during alignment but only those genotypes already known for the individual (e.g., from an array) rather than all variant locations known for the whole population. The computational cost was greater than that with standard algorithms because a customized reference map was created for each individual. Zheng and Grice (2016) used known SNV to improve mapping quality in repetitive regions with AlignerBoost after alignment if all potential locations were listed by

Received June 7, 2018.
Accepted December 10, 2018.
*Corresponding author: Paul.VanRaden@ars.usda.gov
†Current address: Cell Wall Biology and Utilization Research, US Dairy Forage Research Center, Agricultural Research Service, USDA, 1925 Linden Drive, Madison, WI 53706.

the aligner. Tithi et al. (2015) improved accuracy by including previously known SNV during alignment, but computation was 20 times longer.

For species with many individuals that have already been sequenced, a list of known variants may be available from a central database, such as those for cattle (Daetwyler et al., 2014) or humans (The 1000 Genomes Project Consortium, 2012), and these same variants could be called immediately as additional DNA sources are sequenced. This strategy may allow distributed processing at other separate locations without exchanging all the raw data, only the differences from the map. Strategies to account for known variants during alignment will become more valuable as data sets grow and more populations are sequenced because (1) each additional individual will possess many known variants but few new variants and (2) only the new variants not previously observed need to be identified from the new data.

Alignment and variant calling programs are often tested in human genetics using deeply sequenced actual trios such as in Utah Pedigree 1463 (Pirooznia et al., 2014; Cornish and Guda, 2015). An alternative is to simulate reads from the reference map and variant list so that true locations and true genotypes are known (Li et al., 2008; Kessner and Novembre, 2015). Estimates of genotype probabilities are often improved using imputation after initial variant calling (Daetwyler et al., 2014), whereas some recent programs such as the Genome Analysis ToolKit (**GATK**) HaplotypeCaller (Van der Auwera et al., 2013) or STITCH (Davies et al., 2016) use phase information from individual reads to form haplotypes before calling genotypes. The alignment, variant identification, variant calling, and imputation steps can be combined and tested together (Pabinger et al., 2014), but current research does not include the phasing and imputation steps that were previously tested (VanRaden et al., 2015).

Most cattle sequencing projects have used Burrows–Wheeler alignment (**BWA**; Li and Durbin, 2009) or similar software for alignment (Keel and Snelling, 2018) and SAMtools (Li et al., 2009) for variant calling. Hash-based algorithms such as in SNAP [M. Zaharia, University of California (UC), Berkeley; W. J. Bolosky, Microsoft Research, Redmond, WA; K. Curtis, UC, Berkeley; A. Fox, UC, Berkeley; D. Patterson, UC, Berkeley; S. Shenker, UC, Berkeley; I. Stoica, UC, Berkeley; R. M. Karp, UC, Berkeley; and T. Sittler, UC, San Francisco, unpublished data; arXiv:1111.5572(cs.DS)] and GENALICE MAP software (Lunenberg, 2014) can align sequences much faster. Variant-calling software may have similar accuracy for calling SNV, but differences are larger between the GATK UnifiedGenotyper (Van der Auwera et al., 2013) and SAMtools for calling indels (Baes et al., 2014).

This study (1) examined speed and accuracy of alignment and variant calling by doing both jointly; (2) compared the new programs Findmap and Findvar with previous BWA and SNAP alignment programs and SAMtools and GATK UnifiedGenotyper variant-calling programs; (3) applied the new programs to cattle, human, and completely random reference maps; and (4) tested performance with simulated and actual data. Our goal was to outline new strategies for processing sequence data. Further programming may be needed to integrate these algorithms with existing tools requiring other more complex formats.

MATERIALS AND METHODS

Algorithms

New algorithms to align sequence and call variants were developed, coded, and tested as part of the findmap.f90 software package (https://aipl.arsusda.gov/software/findmap/). A main difference from previous algorithms is the use of previously known variant locations (if available) during alignment. A secondary difference is that output files remain in DNA source by variant order instead of (optionally) transposing to variant by source order. Before processing data, the program Storemap is run once to create hash tables that rapidly access the reference map and (optionally) the known variant locations. Findmap then reads the hash tables, aligns the reads from each source to the reference map, calls any known variants within the reads, outputs any differences of the read from the reference map at the aligned location, and outputs allele counts, genotype probabilities, and genotypes for each DNA source at the known variant locations. Findvar identifies new variants for either single or multiple samples by comparing numbers of alternate alleles with the read depth. This second step is needed because the read depth for a single sample may be too low to separate read errors from true variants.

Storemap. Storemap selects segments of 16 consecutive bases (seeds) from the reference map, converts the seeds to 8-byte integers, and stores them in a hash table for use in alignment. Many alignment programs use similar short seeds to find potential map locations and then extend to the left and right to check for full agreement of reads with the map. The integers are formed by converting adenine (A), cytosine (C), guanine (G), and thymine (T) to 0, 1, 2, and 3, respectively, and then summing across the seed while multiplying each previous sum by 4. While creating the hash table, any

Location	1 . . . 5 . . . . 10 . . . . 15 . . . . 20 . . . . 25 . . . . 30 35
Reference map	C C A T G A C A G A T C T T T A A G A C C G G G T T A G C A C C C T A
Alternate map	C C A T G A C C G A T C T T T A A G A C C A G G T T A G C A C C C T A
Read 1 (SNP)	C C A T G A C C G A T C T T T A A G A C C G G G T T A G C A C C C T A
Read 2 (SNP and errors)	C C A C G A C A G A T C T T T A A G A C C A G G T T A G C A G C C T A
Read 3 (deletion)	C C A T G A C A G A T C T A A G A C C G G G T T A G C A C C C T A A G
Read 4 (insertion)	C C A T G A C A G A T C T T T A A G A C C C G G G T T A G C A C C C T

Figure 1. Reference map, alternate map, and reads containing SNP (green), errors (red), a 2-base deletion at location 14, or a 1-base insertion at location 21.

duplicate map locations that have the same 16-base seed are stored in a linked list so that all duplicate seeds can be checked in succession.

The hashing strategy in Storemap selects simple base patterns that are easy to identify but sufficiently numerous to occur in almost all read segments. The stored patterns are selected to have at least 7 bases between occurrences of the same nucleotide. For example, the pattern ACGTTCGTA is selected because the distance from an A to the next A exceeds 7. In the reference, the 16-base seed ends at the rightmost base of the selected pattern. In sequence reads, the 16-base seeds are formed to end at the left or right of the selected pattern to allow alignment of segments read in either the same or reverse direction as the reference. Locations with at least 7 bases between the same nucleotide are the same in reverse cDNA as in the reference but are detected in the opposite direction (e.g., TACGAACGT is the reverse complement of ACGTTCGTA). To process longer segments, fewer seeds would be needed, which would either decrease memory to store the hash table or increase speed by faster hash access to a less-full table.

Storemap also reads and stores the known SNV and indels and then hashes the alternate map using the same techniques as for the reference map. The alternate map is simply the reference map but with the most common alternate allele (not third alleles) at each SNV replacing the reference allele. Both maps are stored in 2-dimensional arrays by chromosome and location, along with a third array storing variant numbers. Variant types and variant lengths are stored in arrays by sequential variant number. For insertions, the inserted bases are packed into a vector, with their starting positions stored by variant number for rapid access. The alternate map is then hashed, and only new seeds are stored. This added storage is not large because the reference and the alternate maps are the same within most 16-base intervals. Thus, only 1 additional map containing all variants is needed instead of a creating a personalized map for each DNA source containing its known genotypes as in Yuan et al. (2015).

Findmap. Findmap loads the hash table and variant data from Storemap into memory before aligning segments. Reads are aligned to the map by first selecting all seeds in the segment with >7 bases between the same nucleotide. The selected seeds, which may not contain missing (indeterminate) base calls, are then sorted by descending distance between the same nucleotide because longer distances are more unique. The 8-byte integer for the first seed is hashed, and the full segment is then checked to determine whether it matches the map at that location or at any duplicate locations in the map. The error count includes differences from the reference map but excludes differences from known alternate alleles. If the map location contains ≥1 indels, the segment is also compared with the map adjusted for the indels, and the lowest error count is declared a full match if the error count is less than the error rate multiplied by segment length. Thus, errors are counted only if the bases do not match reference alleles, known alternate alleles, or the base shifts after accounting for known indels (Figure 1).

If the segment does not match the map or the variants for the first selected seed, the reverse complement is formed, and its corresponding seed is hashed. The search proceeds until a full match is found for any selected seed or its complement or until no more seeds are present. Seeds with >100 duplicate locations in the reference map are skipped to reduce processing and because they are less likely to identify a unique map location. During the search process, the location of the partial match with the most correct bases is stored so that the best match can be reported if a full match is not found. If both ends of a paired-end read do not align fully to the reference map, the length of a potential indel within the read is calculated from the map location difference for 2 partial matches. The algorithm then finds the indel location and checks whether the full read matches after accounting for the indel.

Parallel processing uses the same memory and hash table to process several segments at the same time. Blocks of 1,000,000 segments are stored in memory and distributed to processors by a parallel "do" loop. Within

this loop, paired-end reads are processed together so that their map locations can be compared. If either end has a unique location, potential duplicate locations for the other end are checked in the linked list to determine whether any are near the unique read within the fragment length size. If neither end has a unique location, the linked lists for both ends are scanned in descending order across the chromosomes; whichever list currently has the higher location is incremented. The first location where both ends fully match is reported.

After finding the first location that matches the map, Findmap can optionally search for the next best location by excluding map locations within the fragment length of the first detected location and then repeating the search for a second location. This multiplies run time by about 4 because the first search often finds the best location immediately, whereas the second search may need to check many alternate locations. A map quality score is then computed for each read pair using the difference in number of read errors at the first and next best location as in Li et al. (2008) and is reported as -10 times the $\log_{10}$ probability of mapping error as in sequence alignment map (SAM) format. The upper limit is set to 99 instead of 255 to save space in the output. The map quality score is not used in downstream processing.

New indels not already in the known variant list are identified by storing the detected map locations in memory for each seed checked within a given segment. If a read does not align fully, the length of an indel within the read is calculated from the difference of 2 map locations for a seed to the left and seed to the right of the indel. If the difference of the 2 map locations is less than the maximum indel length, the algorithm searches for the exact location of the indel between the 2 seeds that minimizes error and checks whether the full read then matches after accounting for the indel. The size, location, and bases inserted or deleted are output to a file of newly detected indels by Findmap and are input by Findvar to check consistency across reads. Findmap also outputs the reference and alternate allele counts for each previously known variant in file *variant.readdepth* and for each individual in file *individual.readdepth* for use in imputation of genotypes, such as by the program Findhap version 4 (VanRaden et al., 2015). Findmap then converts the allele counts into genotype probabilities and best-call genotypes and outputs those into files *individual.genoprobs* and *individual.genotypes*, respectively.

Findvar. Findvar processes potential new variants detected by Findmap to separate true variants from probable read errors. For example, if the read depth is 8 at a particular location with reference allele C, and allele T is observed at least 3 times, allele T is declared to be a true variant rather than 3 independent read errors. The likelihood of read errors is calculated from the assumed error rate. Total read depth at each location of each individual is obtained from the leftmost positions where the segments mapped on each chromosome and then extended to the right by their read length. Potential SNV are counted in a 3-dimensional array by chromosome, location, and allele (A, C, G, or T).

Alternate alleles are declared for a single individual if the sum of posterior probabilities for the heterozygous and homozygous alternate genotypes is greater than the posterior probability of the homozygous reference genotype. This Bayesian method combines the likelihood of observing the data for each genotype with their prior probabilities, which are assumed to be 0.001, 0.0005, and 0.9985 for heterozygous, homozygous alternate, and homozygous reference genotypes, respectively, at each location and potential alternate allele. False-positive variants are avoided in Findvar by assuming that the prior probabilities of true variants are smaller than the prior probabilities of read errors. For real data, the prior probabilities can be adjusted for the transition: transversion ratio to account for mutation likelihood.

Additional multisample alternate alleles are declared or deleted from the total counts and total read depths (sums of individual read depths) after processing all individuals. Variants called from single samples are deleted if the total count of an observed allele is less than the expected count under the null hypothesis, which equals the error rate multiplied by the total read depth divided by 3 (because errors are divided equally among 3 alternate alleles). Variants are added if the total allele count exceeds 1.3 times the expected allele count plus the square root of the number of samples. Indels are also detected from the counts across all samples but with less restrictive limits than for SNV. An indel is accepted when 1 case is observed from a total read depth of <40, >1 case is observed from a total read depth of <100, >2 cases are observed from a total read depth of <500, or >3 cases are observed. Sorting of the aligned data is not required because counts are accumulated in memory.

A deduplication step is done in Findvar before identifying new variants. A small percentage of DNA segments may appear more than once in the FASTQ file, and such duplicates with identical placement on the map should be removed (Kelly et al., 2015). In Findvar, only the first read at a particular chromosome location is kept, and all other reads mapped to exactly the same location can be ignored with a deduplication option.

After combining any new variants and any previously identified variants into file *variants.all*, the allele counts, genotype probabilities, and best-call genotypes are output by Findvar into files *readdepth.all*, *geno-*

probs.all, and *genotypes.all*, respectively. These use the same methods as in Findmap, but the Findvar output includes all instead of only the previously known variants. All these outputs should be considered preliminary and can be further improved by imputation, as is done routinely in the 1000 Bull Genomes Project (Daetwyler et al., 2014).

Simulation

Program Map2seq is part of the findmap.f90 software package and was developed to simulate paired-end reads at random locations in the reference map because existing simulation programs such as wgsim (https://github.com/lh3/wgsim) do not allow use of a prior variant list or a pedigree structure for the sequenced individuals. Map2seq has options controlling segment lengths, read error rates, and missing rates. The reads can also contain alternate alleles that can be heterozygous or homozygous, thus allowing simultaneous testing of alignment and variant calling accuracy. Genotypes for the true variant list were processed sequentially in groups of 4. Every fourth variant was set to homozygous alternate allele, every second was homozygous reference, and the first and third were heterozygous. For the heterozygous variants, a random 50% of reads were switched to alternate allele. Map2seq can also use genotypes generated from pedigree or simulate linkage among adjacent heterozygotes, but those options were not used in this research. Map2seq can simulate parallel streams of FASTQ files at the same time for efficiency with just 1 copy of the reference map and variant list held in memory. Total memory was 28 GB for any number of FASTQ files simulated at once.

Computer Processing

The programs were compiled with Intel ifort (Intel, 2017) and run with Linux operating system. They use unformatted (binary) file exchanges (optionally) and the Math Kernel Library and Message Passing Interface library included with the compiler for efficiency and parallel processing. Tests were performed on 2 machines: an IBM xSeries 3850 server (IBM Corp., Armonk, NY) with 640 GB of memory and four 64-bit Intel Xeon X7560 dies, which provided a total of 64 computing threads running at 2.27 GHz, and an HP 580DL Gen8 server (Hewlett-Packard, Santa Clara, CA) with 256 GB of memory and two 64-bit Intel Xeon CPU E7-8893 v2 dies, which provided a total of 24 computing threads running at 3.40 GHz. Other compilers or operating systems were not tested. Because alignment speed is fast, the whole process can be iterated using previously identified variants to improve alignment and genotype calls in the next iteration or when more DNA samples are sequenced.

Parameter options were compared and optimized primarily using paired-end reads simulated from the cattle map. Some options describe the input data and the memory required for storage, such as for single-end or paired-end reads, maximum read length, number of chromosomes, maximum chromosome length, maximum number of variants, maximum total length of all insert variants, maximum number of unique seeds in the hash table, and maximum total number of duplicated seeds. The programs will report whether the memory reserved is insufficient for the data. Some options such as assumed error rate, DNA fragment length, seed length, and minimum gap size for selecting seeds affect accuracy, speed, memory, and percentage of reads mapped. The detection option will compute much more accurate map quality scores with double the computation. Other options can input and output either readable text files or more compact Fortran binary files for efficiency. Number of processors to use is also optional. The program Findvar can optionally output all previously known variants or retain only those with sufficient frequency to still qualify as variants.

Program Depth2vcf.c also is part of the findmap.f90 software package and can transpose the sample by variant read depth files from Findmap or Findvar into variant by sample genotype files in VCF format if needed for other applications such as imputation. However, program Findhap4 (VanRaden et al., 2015) imputes accurately from even low coverage sequence with no requirement to transpose inputs. We did not yet convert to binary alignment map (**BAM**) file (Li et al., 2009) format because the Findmap simple alignment format is more analogous to CRAM format. Also, the main benefits of BAM format are sorting and indexing, but those are not needed for whole-genome processing.

The findmap.f90 software package with the Fortran code described above, compiled programs for execution, example simulated files, and expected results, is freely available from the USDA at https://aipl.arsusda.gov/software/findmap/. The package also includes programs to simulate reference maps, variants, and FASTQ files for a variety of options.

Software Comparisons

Findmap output files report only the leftmost position of the read and its differences from the reference map, which is similar to reference-based compression strategies (Hsi-Yang Fritz et al., 2011) used in CRAM format and in GENALICE MAP software (Lunenberg,

Table 1. File sizes for sequence reads, aligned reads, and variant calls from alignment and variant-calling software

		BWA and GATK[1]		Findmap and Findvar[2]	
Data files	File extension	Unzipped (MB)	Zipped (MB)	Unzipped (MB)	Zipped (MB)
Sequence reads/1×	.FASTQ	6,000	1,800	6,000	1,800
Aligned reads/1×	.BAM or .found and .lost	3,200	3,200	800	150
Called variants/source	.VCF or .readdepth	1,000	38	79	13

[1]BWA = Burrows–Wheeler alignment (Li and Durbin, 2009); GATK = Genome Analysis ToolKit (Van der Auwera et al., 2013).
[2]https://aipl.arsusda.gov/software/findmap/.

2014). Thus, file sizes from Findmap were greatly reduced compared with widely used standard formats (Table 1). The BAM files were >20 times larger than zipped output from Findmap, and FASTQ input files were >10 times larger than Findmap output. The main difference between Findmap output and that from previous software is that positions of alternate alleles and called indels are identified directly in the aligned data format instead of listing the known variants simply as alignment errors. In addition, Findmap output is split into a .found file for reads with matching paired-end locations and a .lost file for unmatched reads.

Indel notation differed by software. The program SAMtools allows both leftmost and rightmost positions of the indel to be determined by reporting extra bases in the reference and alternate alleles; GATK reports fewer bases and gives only the leftmost position for indels with ambiguous locations; and SAMtools and GATK both report indels 1 base to the left of the leftmost base inserted or deleted, which seems reasonable for insertions but perhaps not for deletions. Findmap and Findvar search from left to right for the first detectable difference of the individual's DNA from the reference genome and thus provide the rightmost indel position; a postprocessing tool can convert indels to the leftmost position if needed. For example, the insertion in Figure 1 could be reported at its leftmost position of 19 instead of its rightmost position of 21.

Alignment accuracy and speed were tested by comparing Findmap results with those from BWA and SNAP. Repetitive sections of the genome were not masked before aligning with Findmap because Findmap stores and counts occurrences of repeated seeds internally. Test results from BWA were obtained with or without masking using the program RepeatMasker (Bedell et al., 2000), and SNAP was run with a seed size of length 22 and a maximum edit distance (option -d) of 12, which gave the best combination of speed and accuracy for 150-base reads. The variants identified by Findvar were compared with those from SAMtools and GATK UnifiedGenotyper. Variants marked as low quality by SAMtools were excluded from counts because >99% were false positives; however, after excluding these, SAMtools still had by far the largest number of false-positive SNV.

Findmap and BWA were compared using simulated paired-end reads of length 150 from fragments of length 1,000 at random locations within the UMD3.1 bovine genome assembly (Zimin et al., 2009). Each base had 1% probability of error and 1% probability of being missing. The 39 million variants from run 5 (July 2015) of the 1000 Bull Genomes Project (Hayes et al., 2014) were included with every other variant set to reference or alternate. Variant calls were output by Findmap only for the 88.2% of pairs for which both ends were located within the fragment length and of opposite orientation.

New variants from Findvar were compared with those from GATK UnifiedGenotyper and from SAMtools after BWA. The previous software packages were run with default parameters and with Picard removal of duplicate reads. Detection accuracy was examined using reads simulated for 10 animals at 10× coverage from the UMD3.1 bovine reference map with almost 40 million variants (38,062,190 SNV, 532,179 insertions, and 1,127,620 deletions) derived from run 5 of the 1000 Bull Genomes Project.

Simulated Bovine Sequences

Bovine sequence reads were simulated using the UMD3.1 bovine reference map and variants from run 5 of the 1000 Bull Genomes Project (http://www.1000bullgenomes.com/). The simulation included the X chromosome map and variants but not the Y, mitochondria, or unmapped contigs. Within the simulated segments, the alternate alleles were set, then 1% of bases were randomly switched to another base to generate 1% error, then 1% of bases were set to missing, and finally a random 50% of segments were switched from forward direction to reverse complement.

A second test included 10 animals with 10× coverage, but each had the same genotype so that the files could also be viewed as 100× coverage for a single animal. Also, 80% of variants were treated as known and

Table 2. Computer resources per DNA source with 10× coverage required by alignment and variant-calling software applied to simulated cattle data

Task	Software[1]	Memory (GB)	Threads (no.)	Processing time (min)
Simulate 10× data	Map2seq	28	10	5
Align 10× reads and call known variants	Findmap	46	10	20
Align 10× reads	SNAP	40	10	22
	BWA	46	10	629
Identify and call variants	SAMtools	—	10	66
	GATK	73	10	150
Identify new variants	Findvar	99	1	8
Impute 39 million variants	Findhap, version 4	—	10	1

[1]BWA = Burrows–Wheeler alignment (Li and Durbin, 2009); Findvar, Findmap, and Map2seq (https://aipl.arsusda.gov/software/findmap/); GATK = Genome Analysis ToolKit (Van der Auwera et al., 2013); SAMtools (Li et al., 2009); SNAP (https://arxiv.org/abs/1111.5572).

20% as new in Findmap, and all programs were tested for ability to identify those 20%, which were chosen as every fifth variant.

The actual bovine sequence tested was from BioProject PRJNA316122, "*Bos taurus* Genome Sequencing" (https://www.ncbi.nlm.nih.gov/bioproject/PRJNA316122). A sample from parental cell line 2122 (run SRR3290632) was used (http://trace.ncbi.nlm.nih.gov/Traces/sra/?run=SRR3290632).

Human Genome Sequences

Human sequence reads were simulated using the hg38 reference map (https://genome.ucsc.edu/cgi-bin/hgGateway) of the University of California, Santa Cruz Human Genome Browser and 28 million common variants in the 00-common_all.vcf file available from the 1000 Human Genomes Project (https://www.ncbi.nlm.nih.gov/variation/docs/human_variation_vcf/) in December 2015. The variants included 25,360,930 SNV, 1,102,402 insertions, and 1,673,964 deletions. The variant file included only the autosomes, whereas alignment and simulation also included the X, Y, and mitochondrial maps.

Random Nonrepetitive Genome

A nonrepetitive reference map was also simulated with 25% each of A, C, G, and T in a purely random pattern and 0.01% of reference bases randomly set to unknown. The simulated variants were also randomly distributed across the 30-chromosome genome of 3 billion bases and included 28 million SNV, 1 million insertions, and 1 million deletions for a total of 30 million variants. Lengths of indels were uniformly distributed between 1 and 10. Tests focused on 150-base reads because those are currently being generated, but read lengths of 50, 100, and 200 to provide 1× coverage were also examined.

RESULTS

Computer resources needed for alignment and variant calling with Findmap and Findvar were compared with previous programs (Table 2). Tests were conducted on simulated cattle sequence, with 80% of variant locations treated as known in Findmap and 20% yet to be discovered. Clock times for aligning 10× coverage for each of 10 individuals using 10 processors (100× total) were 3.3 h for Findmap, 3.7 h for SNAP, and 104.8 h for BWA. Identifying new variants required 1.3 h for Findvar, 11 h for SAMtools, and 25 h for GATK. With 1 processor, Findmap required 20 min per 1× for alignment and calling known variants, whereas SNAP required 22 min and BWA required 629 min per 1× for only alignment.

Memory in Findmap or SNAP can be shared by 10 processors or more; they required 46 and 40 GB, respectively; BWA required 4.6 GB per processor or 46 GB total with 10 processors. The total memory of 46 GB required by Findmap included storing the reference, alternate, and indel data plus the hash table, work space, and read buffers. Thus, little extra memory is needed to account for known variants during alignment when multiple processors share the same memory.

Processing speed improved nonlinearly with number of processors. Times required by Findmap per 1× coverage were 5.1 min with 1 processor, 3.2 min with 2, 2.2 min with 5, and 1.8 min with 10 when tested on an HP580, and similar speedups with more processors were obtained on an IBM3850 (Figure 2). Processing times included reading segments, aligning them to the map, and calling all known variants. Further increasing the number of processors above 10 resulted in little gain. However, running multiple jobs each with 5 or 10 processors further increased speed (e.g., 2 jobs each with 10 processors almost doubled the speed but also doubled the memory). The time needed only for reading Fortran unformatted data analogous to FASTQ format

Table 3. Properties of 3 reference maps and variant files used in software comparisons

Genome and variant file property	Cattle	Human	Random
Reference genome length (MB)	2,661	3,088	3,000
Chromosomes included	29 + X	22 + X,Y,M	30
Single nucleotide variants (thousands)	38,062	25,361	28,000
Insertions (thousands)	1,128	1,102	1,000
Deletions (thousands)	532	1,674	1,000
Average indel length	3.0	4.8	5.5
Maximum indel length	86	50	10
Unique 16-base seeds selected (millions)	355	199	383
Duplicates of 16-base seeds (millions)	158	295	149
Added seeds from alternate alleles (millions)	34	12	29

was 0.6 min per 1×. Thus, alignment and variant calling can be done in about 3 times the clock time needed to read the input data file.

The initial setup in Storemap required 47 GB of memory and 8 min to hash the map and variant data and to output the hash table. The hash table contained 355 million seed locations after processing the reference map, and another 34 million seeds were added after processing the alternate alleles (Table 3). The linked list of duplicate seeds contained 158 million locations. Uniqueness and duplication of seeds were very similar across cattle, human, and random genomes. Before starting alignment, Findmap took 2 min to read the hash table, reference map, and variant list. That start-up time is done just once and not included in the clock times per 1×.

Percentage of correctly mapped reads was 92.9 from Findmap and 90.5 from BWA (Table 4). Percentage of reads where both ends of a pair were correctly mapped was 87.6 from Findmap and 87.2 from BWA. Percentage where both ends were incorrectly mapped was only 1.8 from Findmap compared with 6.2 from BWA. Percentages from SNAP were similar to those from Findmap. Some alignment strategies examine all possible alignment locations and report mapping quality to indicate ambiguous reads that could be mapped to other locations, but Findmap reports only the first location that meets the error limit to improve speed.

Table 4. Alignment status percentages using Burrows–Wheeler alignment (BWA; Li and Durbin, 2009), SNAP,[1] or Findmap[2] with simulated cattle sequence

Alignment status	BWA	SNAP	Findmap
Correctly placed segments overall	90.5	92.6	92.9
Both ends of pair correctly placed	87.2	87.7	87.6
One end correct, one end wrong	6.4	10.0	10.6
Both ends wrong	6.2	2.3	1.8

[1]https://arxiv.org/abs/1111.5572.

[2]https://aipl.arsusda.gov/software/findmap/.

Percentage of correctly aligned reads from Findmap improved moderately from 92.1 to 92.9 when more variant locations were already known. Percentage of both ends of a pair being correctly mapped also improved from 86.8 to 88.4 as more variant locations were known. The presence of unknown variants had little effect on calling accuracy for known variants (Table 5). Alignment locations were less precise when some indel boundaries (locations and lengths) were unknown. The optimal value for the error rate option in Findmap was lower (0.03) when more variants were known and higher (0.05) if none were known because alternate alleles act as errors if not already known but are removed from the error count if known. Higher or lower error rates could be optimal for reads with more or fewer errors than the 0.01 simulated average.

For previously known variant sites, Findmap was very accurate at calling individual alleles within reads. Of the SNV calls with 80% of previous variants known, 99.8% of the reference alleles and 99.8% of the alternate alleles were called correctly. Of the insertion calls, 98.5% of the normal and 99.8% of the inserted base calls were correct. Of the deletion calls, 97.7% of the normal and 99.8% of the deleted base calls were correct. Of the reads where paired-end locations matched (not shown), calls were obtained for 98.2% of SNV, 97.9% of insertions, and 97.0% of deletions. Call rates for indels were slightly lower because about 10% of segments with an indel actually had multiple indels and only the indel nearest to the center of the segment was called.

For new variants, numbers identified and false-positive rates differed greatly by software, especially for indels (Table 6). For homozygous alternate variants, Findvar found 99.8% of SNV, 78.6% of insertions, and 66.6% of deletions; GATK found 99.4, 95.3, and 90.0%, respectively; and SAMtools found 99.8, 11.9, and 15.5%, respectively, with low-quality variants excluded. For heterozygotes, Findvar found 99.1, 75.2, and 62.2%, respectively; GATK found 99.0, 92.5, and 88.0%, respectively; and SAMtools found 98.2, 7.2, and 8.0%, respectively, with low-quality variants excluded.

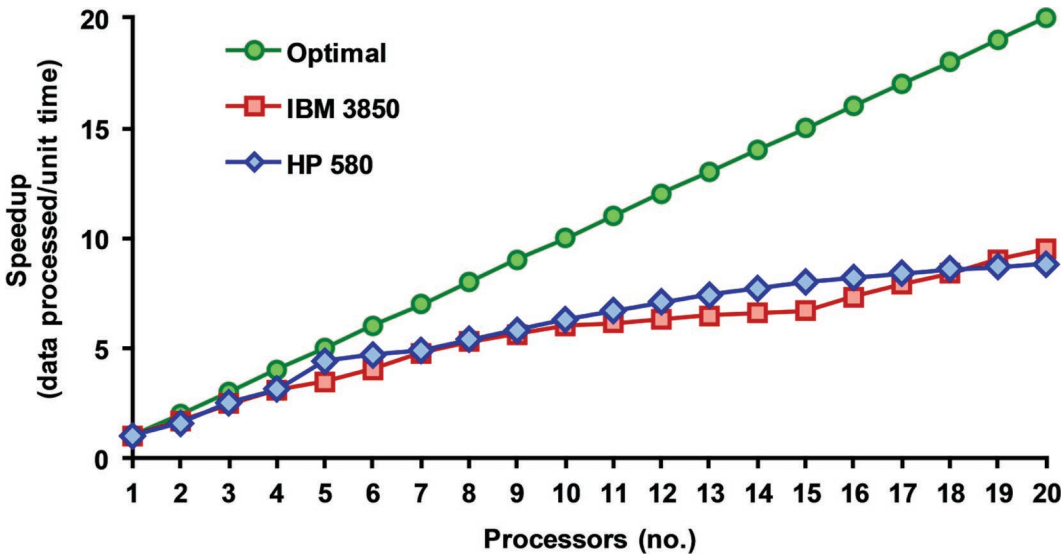

Figure 2. Ratio of actual to optimal increase in processing speed (speedup) for Findmap (https://aipl.arsusda.gov/software/findmap/) parallel processing of simulated bovine sequence data using up to 20 processors on an IBM3850 (IBM Corp., Armonk, NY) with 64 processors or an HP580 (Hewlett-Packard, Santa Clara, CA) with 24 processors.

False positives as a percentage of true variants were 10.6, 0.4, and 0.3%, respectively, from Findvar; 12.4, 8.4, and 2.9%, respectively, from GATK; and 37.3, 1.3, and 0.4%, respectively, from SAMtools with low-quality variants excluded. Total read depth after aligning paired reads was 85.9 from Findmap/Findvar, 96.1 from BWA/GATK, and 84.4 from BWA/SAMtools compared with 100× simulated coverage (not shown). Higher read depth is an advantage if map locations are correct but is a disadvantage if more of the incorrectly placed reads are used in calling genotypes.

With Findmap, longer read lengths were optimal for most measures of success when comparing 50-, 100-, 150-, and 200-base reads (Table 7). Speed was also fastest for longer segments because correct locations were found with the fewest hash table lookups. Short reads with errors may not have an exact match in the hash table, whereas long reads with more indels may not

Table 5. Effect of previously known variants on Findmap[1] alignment and variant calling for a cattle genome for 1 individual with 10× simulated coverage

	Previously known variants (%)			
Alignment and call rate statistics	0	67	80	100
Segments mapped (%)	97.9	98.2	98.3	98.4
Segments correctly mapped (%)	92.1	92.7	92.8	92.9
Segments with both ends consistent (%)	86.8	87.9	88.0	88.4
Segments with both ends correctly located (%)	86.1	87.0	87.3	87.8
Mapped locations off by 1 to 30 bases (%)	0.25	0.21	0.21	0.17
Error rate in matched segments (%)	1.60	1.26	1.19	1.06
New variants correctly identified				
Homozygous SNV[2] (%)	97.4	97.8	97.5	—
Heterozygous SNV (%)	57.9	63.4	74.0	—
Homozygous insertions (%)	75.7	71.6	70.4	—
Heterozygous insertions (%)	49.1	53.1	52.7	—
Homozygous deletions (%)	61.5	57.9	57.3	—
Heterozygous deletions (%)	37.2	40.0	39.6	—
Previous variant alleles correct				
Normal/alternate SNV alleles (%)	—	99.7/99.7	99.8/99.8	99.7/99.7
Normal/alternate insertion alleles (%)	—	97.7/99.5	98.5/99.8	98.2/99.5
Normal/alternate deletion alleles (%)	—	94.7/98.7	97.7/99.8	97.1/98.6
Clock time per 1× using 10 processors (min)	2.1	1.8	1.8	1.6

[1]https://aipl.arsusda.gov/software/findmap/.

[2]SNV = single nucleotide variant.

Table 6. Identification success percentages for homozygous alternate (BB) or heterozygous (AB) variants using Findvar,[1] Genome Analysis ToolKit (GATK; Van der Auwera et al., 2013), and SAMtools (Li et al., 2009) from 10× simulated sequences for 10 cattle

Variant genotype	Variants simulated (no.)	Variants identified	Findvar	GATK	SAMtools	
					Low-quality variants included	Low-quality variants excluded
SNV[2] BB	1,905,427	Found/true	99.8	99.4	99.9	99.8
SNV AB	3,809,342	Found/true	99.1	99.0	98.4	98.2
Insert BB	26,622	Found/true	78.6	95.3	12.3	11.9
Insert AB	53,086	Found/true	75.2	92.5	8.5	7.2
Delete BB	56,292	Found/true	66.6	90.0	16.0	15.5
Delete AB	112,425	Found/true	62.2	88.0	9.3	8.0
SNV	28,546,642	False positive	10.6	12.4	85.6	37.3
Insertion	399,134	False positive	0.4	8.4	2.0	1.3
Deletion	845,715	False positive	0.3	2.9	0.6	0.4
Overall	—	Call rate	86.4	95.6	84.4	83.8

[1]https://aipl.arsusda.gov/software/findmap/.
[2]SNV = single nucleotide variant.

match across the whole segment. Optimal seed length was shorter for shorter reads but caused slower processing because shorter seeds were less unique. Longer reads had fewer exact matches on the first try because of more indels per read. Findmap currently accounts for only 2 indels per read and thus may not perform well for very long reads, which were not tested in this research.

Map quality scores were converted to probabilities of correct location and compared with actual location status for simulated cattle reads of length 150 with 1% error rate per base. For paired ends with matching locations and orientation, computed probabilities and actual percentages of correct locations both averaged 99.8%; correlation of the predicted and actual correct status was 0.60. For reads where the pair's mapped locations did not match, computed probabilities averaged 43.3% and actual percentages of correct location averaged 43.5%; correlation of the predicted and actual correct status was 0.91. Thus, map quality could accurately detect which of the reads were mapped correctly, especially for the nonmatching pairs.

For a perfectly random reference map with 25% chance of A, C, G, or T for each subsequent base, 1% missing, 1% error, and 80% of the 30 million variants known, 99.8% of the 150-base segments were mapped with Findmap (Table 8). Of those, 99.6% were mapped correctly, and another 0.21% were within 30 bases of

Table 7. Effect of read length on Findmap[1] alignment and variant calling for a cattle genome for 1 individual with 10× simulated coverage using paired-end reads and 80% of true variant locations known

Alignment and call rate statistics	Read length (bases)			
	50	100	150	200
Segments mapped (%)	85.3	95.4	98.3	99.1
Segments correctly mapped (%)	73.8	88.4	92.8	95.0
Segments with both ends consistent (%)	57.8	80.5	88.0	91.7
Segments with both ends correctly located (%)	57.3	79.9	87.3	91.0
Mapped locations off by 1 to 30 bases (%)	0.14	0.17	0.21	0.22
Error rate in matched segments (%)	1.03	1.14	1.19	1.17
New variants correctly identified				
Homozygous SNV[2] (%)	80.0	94.8	97.5	98.6
Heterozygous SNV (%)	41.7	62.3	74.0	70.7
Homozygous insertions (%)	6.9	59.2	65.3	74.8
Heterozygous insertions (%)	3.2	39.6	43.0	59.4
Homozygous deletions (%)	5.0	46.8	52.4	62.5
Heterozygous deletions (%)	2.4	28.7	31.8	46.3
Previous variant alleles correct				
Normal/alternate SNV alleles (%)	99.7/99.8	99.8/99.8	99.8/99.8	99.8/99.8
Normal/alternate insertion alleles (%)	95.8/99.8	97.9/99.8	98.5/99.8	98.8/99.9
Normal/alternate deletion alleles (%)	95.6/99.8	97.1/99.8	97.7/99.8	97.9/99.8
Clock time per 1× using 10 processors (min)	3.6	2.1	1.8	1.6

[1]https://aipl.arsusda.gov/software/findmap/.
[2]SNV = single nucleotide variant.

Table 8. Comparison of Findmap[1] alignment and variant calling from cattle, human, or random (nonrepetitive) reference maps for 1 individual with 10× simulated coverage using paired-end reads of 150 bases and 80% of true variant locations known

Alignment and call rate statistics	Cattle	Human	Random
Segments mapped (%)	98.3	99.1	99.8
Segments correctly mapped (%)	92.8	94.2	99.6
Segments with both ends consistent (%)	88.0	91.0	99.6
Segments with both ends correctly located (%)	87.3	89.9	99.2
Mapped locations off by 1 to 30 bases (%)	0.21	0.15	0.21
Error rate in matched segments (%)	1.19	1.16	1.09
New variants correctly identified			
Homozygous SNV[2] (%)	97.5	98.1	99.6
Heterozygous SNV (%)	74.0	76.1	81.0
Homozygous insertions (%)	65.3	43.6	93.3
Heterozygous insertions (%)	43.0	26.6	84.7
Homozygous deletions (%)	52.4	50.2	88.0
Heterozygous deletions (%)	31.8	31.8	74.2
Clock time per 1× using 10 processors (min)	1.8	2.2	1.5

[1]https://aipl.arsusda.gov/software/findmap/.

[2]SNV = single nucleotide variant.

the correct location. Map locations were incorrect for only 0.03% (not shown). If all variants were known, 99.98% of segments were mapped, 99.84% were correctly mapped, and the other 0.14% were within 30 bases of the correct location (not shown). Segments not mapped or mapped incorrectly often contained ≥2 different indels within the same segment. This test demonstrated that mapping problems with real genomes are mainly the result of repetitive DNA or undetected indels.

More time was required to align to cattle or human maps than to the perfectly random map, primarily because linked lists of repeated seeds had to be checked when neither paired end was uniquely mapped. The increased processing time was caused by the more complex actual genomes, which contain repetitive sequence and regions with more G and C content compared with A and T content. The slower speed of BWA could be improved by masking repetitive regions of the genome. After removing repeated sections of the cattle map as identified by RepeatMasker (Bedell et al., 2000), alignment with BWA took 4.4 instead of 14.1 h/1× coverage, but only 45% of segments were correctly aligned instead of 91% because many reads were not mapped. The speedup in computation did not justify such a large loss of data.

Cattle, human, and random genome results for alignment, variant calling, and variant identification are compared in Table 8, and almost all results were best for the perfectly random genome. The human genome had somewhat more accurate alignment than the cattle genome, whereas processing time was around 20% faster for the cattle genome. Success of calling previous variants (not shown) and identifying new variants differed by variant type. Calls of previous variants were less accurate for deletions in cattle and for insertions in human genomes. Identification of new insertions was more accurate for simulated cattle than for human data, and new indels were much easier to identify in the perfectly random genome than in the human or cattle genomes.

An actual cattle sequence with a read length of 125 (not shown) had lower percentages of mapped segments (93 vs. 97%) and matching paired ends (75 vs. 85%) than did a simulated sequence processed with Findmap. Processing time was more than twice as long (42 vs. 20 min/1× per processor), and deduplication removed more reads (5.2 vs. 3.7%). Possible explanations for these differences are clustering of errors within actual reads instead of only random errors within simulated reads, chimeric reads in actual but not simulated data, longer actual insertions or deletions, and poor map quality or repetitive sections excluded from the reference map.

DISCUSSION

Many other alignment and variant calling software packages are available but were not tested in this study; all do alignment and variant calling as 2 separate steps rather than combined steps. Some other public alignment programs are slightly more accurate or a few times faster than BWA, whereas SNAP and Findmap were both much faster than BWA and had slightly improved accuracies. Both use a rapidly accessed hash table in memory that is shared among several parallel processors. GENALICE MAP (Lunenberg, 2014) may be the most similar to Findmap/Findvar and is advertised to be even more efficient; however, it uses private rather than public code. Accuracy was further

improved by about 1% in Findmap by calling alleles for the previously identified variants during alignment and detecting only new variants in Findvar.

In human data, SAMtools identified 45% as many SNV as did GATK but only 12% as many indels (Pabinger et al., 2014), which was similar to the low detection rate observed for indels in this study using SAMtools. Pirooznia et al. (2014) compared SNV identified by SAMtools and GATK and found 96% in common. For SNV that differed, the true-positive rate was 95% for SNV identified only by GATK but was 70% for SNV identified only by SAMtools. The GATK UnifiedGenotyper and HaplotypeCaller had very similar performance, with a slight advantage for HaplotypeCaller but requiring about 10 times more computation. All programs had more difficulty identifying indels than SNV (Cornish and Guda, 2015).

By comparison, Findmap had low error rates of about 1% for calling alleles at previously known indel and SNV sites within simulated human data (Table 8). Findmap also reduced the false-positive rate for calling new indels by requiring a fully matching seed to both the left and the right of the indel within the same read. That strategy also reduces detection rate with moderate coverage, but more such reads would improve detection rate with higher coverage. Thus, the program could serve as a rapid genotyping tool for large populations, for realignment of previous data to a new reference map, or even for clinical sequencing. Given the approximately 30-fold reduction in alignment time of Findmap compared with BWA, generating accurate variant calls could be far more rapid and would eliminate the bottleneck of computational processing required in current variant-calling pipelines.

In cattle data, Baes et al. (2014) compared variant callers using multisample calling with 65 sequenced cattle and obtained about 90% as many SNV from SAMtools as from GATK but 74% as many indels from SAMtools. However, if variants from single-sample calling were combined, slightly more variants were identified by SAMtools than by GATK. The ratio of transitions to transversions was used as a measure of success for SNV; SAMtools had slightly higher ratios than did GATK for both single-sample and multisample identification. With 234 sequenced cattle, Daetwyler et al. (2014) found that concordance of genotypes from the BovineHD SNP array and from sequence SNV by SAMtools was >90% before imputation and improved to 97% after updating genotype probabilities using Beagle. They verified that mutations predicted to be harmful had much lower frequency than did neutral mutations as expected. The 1000 Bull Genomes Project used SAMtools for run 1 to run 6 but switched in 2018 to GATK for run 7.

The reference maps for cattle, both previous (Zimin et al., 2009) and new (Rosen et al., 2018), are from a Hereford cow, whereas most sequencing applications are for other breeds such as Holstein or Angus. Results for all breeds should be more accurate and less sensitive to the choice of reference animal when known differences among and within breeds are accounted for during alignment. As data sets grow, alignment of data for 1 breed could perhaps ignore alternate alleles that are unique to a different breed, and lower limits on allele frequency may be needed to prevent all 3 billion bases and all 3 nonreference nucleotides from eventually being listed as variants.

Researchers in human genetics are also now exploring use of known variants and linkage during alignment in addition to the reference map in a topic known as genome graphs (Paten et al., 2017). Other new variant identification strategies such as the GATK HaplotypeCaller or support vector machines (O'Fallon et al., 2013) require extra inputs such as lists of true variants, false-positive variants, or estimates of machine- or allele-specific bias that can improve call rate but often with even longer run times. Researchers in animal genetics often need different computing strategies than those developed in human genetics (Biscarini et al., 2018), primarily because of differing goals and limited budgets. Animal genetics focuses on genomic prediction, low-density genotyping, lower coverage sequencing, and deep pedigrees, whereas human genetics often focuses on disease treatment, higher density genotyping, higher coverage sequencing, unrelated individuals, and discovering genetic origins. However, algorithms in genomics often apply to many species because DNA inheritance is similar.

Findmap allows newly identified variants to be reused as known variant priors in the next iteration, thus fulfilling the strategy imagined by Li et al. (2008), who stated, "It would be possible in an iterative scheme to update the reference with an estimate of the new sample sequence from the first mapping and then remap to the updated reference." Previous research has assumed that alignment and variant calling must be separate steps; according to DePristo et al. (2011), "Mapping reads to the reference genome is a first critical computational challenge whose cost necessitates that each read be aligned independently, guaranteeing that many reads spanning indels will be misaligned." Because fewer reads with alternate alleles align correctly, ratios of alternate to reference alleles may be reduced, whereas Findmap improves alignment accuracy by simultaneously calling known variant alleles. More testing is needed with actual instead of simulated data; strategies to process longer reads or larger variants were not tested.

Imputation from a low-coverage sequence can be more accurate using raw allele counts derived from sequence alignment mismatches instead of estimated genotype probabilities (VanRaden et al., 2015). Rare variants usually are not reported in VCF files unless at least 1 individual appears to have the alternate allele. Thus, when combining VCF files across DNA sources with low coverage, a raw read count for the reference allele may be difficult to derive from most sources. This problem can be overcome by listing previously known variants so that allele counts are generated at each known site even if the DNA source or sources processed together in 1 file do not contain the alternate allele but only reference or missing alleles.

CONCLUSIONS

Research using large data sets requires efficient computation. The new program Findmap reads the previous variant list, calls variant alleles, and sums allele counts for each DNA source while aligning sequence. Advantages are faster processing, more precise alignment, more useful data summaries, more compact output, and fewer steps. Using information from all previous variants requires little extra memory and speeds the processing of new data because processor time is not spent on rediscovering known variants. Indels are identified while aligning individual DNA sequence reads to the reference map. Findmap can process files in FASTQ format very quickly to generate allele counts for previously known variants or to reprocess data after new variants are discovered or a new reference map becomes available. The new strategy allows aligning sequence data to the genomes of all previously sequenced individuals or breeds but reporting the locations back to the common reference map. Alignment using 1 processor is 50 times faster with Findmap than with BWA and 30 times faster using 10 processors. Findmap can use known variants during alignment to correctly map 2% more segments than BWA. The program can provide variant calls more quickly than SAMtools or GATK for individual sequencing, large population genotyping, or reprocessing data with an updated reference map. Output files are simpler and 3 to 10 times smaller than standard formats. Accuracy can be improved by accounting for known DNA variants while aligning sequence data.

ACKNOWLEDGMENTS

We thank Suzanne Hubbard and Gary Fok of USDA's Animal Genomics and Improvement Laboratory, Agricultural Research Service (Beltsville, MD) for assistance with technical editing and program automation, respectively. P. M. VanRaden and D. M. Bickhart were funded solely by USDA Agricultural Research Service appropriated projects 8042-31000-101-00, "Improving Genetic Predictions in Dairy Animals Using Phenotypic and Genomic Information," and 8042-31000-002-00, "Improving Dairy Animals by Increasing Accuracy of Genomic Prediction, Evaluating New Traits, and Redefining Selection Goals"; J. R. O'Connell was funded by the University of Maryland School of Medicine (Baltimore) and by USDA Agricultural Research Service specific cooperative agreement 58-1245-4-070, "Improvement of Algorithms and Software to Process Very Large Genomic Datasets." Data visualization was aided by Daniel's XL Toolbox add-in for Excel, version 7.2.13, by Daniel Kraus, Würzburg, Germany (www.xltoolbox.net). Mention of trade names or commercial products in this article is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the USDA.

REFERENCES

Baes, C. F., M. A. Dolezal, J. E. Koltes, B. Bapst, E. Fritz-Waters, S. Jansen, C. Flury, H. Signer-Hasler, C. Stricker, R. Fernando, R. Fries, J. Moll, D. J. Garrick, J. M. Reecy, and B. Gredler. 2014. Evaluation of variant identification methods for whole genome sequencing data in dairy cattle. BMC Genomics 15:948.

Bedell, J. A., I. Korf, and W. Gish. 2000. MaskerAid: A performance enhancement to RepeatMasker. Bioinformatics 16:1040–1041.

Biscarini, F., P. Cozzi, and P. Orozco-ter Wengel. 2018. Lessons learnt on the analysis of large sequence data in animal genomics. Anim. Genet. 49:147–158.

Cornish, A., and C. Guda. 2015. A comparison of variant calling pipelines using Genome in a Bottle as a reference. BioMed Res. Int. 2015:456479.

Daetwyler, H. D., A. Capitan, H. Pausch, P. Stothard, R. van Binsbergen, R. F. Brøndum, X. Liao, A. Djari, S. C. Rodriguez, C. Grohs, D. Esquerré, O. Bouchez, M.-N. Rossignol, C. Klopp, D. Rocha, S. Fritz, A. Eggen, P. J. Bowman, D. Coote, A. M. Chamberlain, C. Anderson, C. P. Van Tassell, I. Hulsegge, M. E. Goddard, B. Guldbrandtsen, M. S. Lund, R. F. Veerkamp, D. A. Boichard, R. Fries, and B. J. Hayes. 2014. Whole-genome sequencing of 234 bulls facilitates mapping of monogenic and complex traits in cattle. Nat. Genet. 46:858–865.

Davies, R. W., J. Flint, S. Myers, and R. Mott. 2016. Rapid genotype imputation from sequence without reference panels. Nat. Genet. 48:965–969.

DePristo, M. A., E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M. A. Rivas, M. Hanna, A. McKenna, T. M. Fennell, A. M. Kernytsky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly. 2011. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat. Genet. 43:491–498.

Hsi-Yang Fritz, M., R. Leinonen, G. Cochrane, and E. Birney. 2011. Efficient storage of high throughput DNA sequencing data using reference-based compression. Genome Res. 21:734–740.

Hayes, B. J., I. M. MacLeod, H. D. Daetwyler, P. J. Bowman, A. J. Chamberlian, C. J. Vander Jagt, A. Capitan, H. Pausch, P. Stothard, X. Liao, C. Schrooten, E. Mullaart, R. Fries, B. Guldbrandtsen, M. S. Lund, D. A. Boichard, R. F. Veerkamp, C. P. Van Tassell, B. Gredler, T. Druet, A. Bagnato, J. Vilkki, D. J. deKoning, E. Santus, and M. E. Goddard. 2014. Genomic prediction from whole genome sequence in livestock: The 1000 bull genomes project. Commun. 183 in Proc. 10th World Congr. Genet.

Appl. Livest. Prod., Vancouver, BC, Canada. Am. Soc. Anim. Sci., Champaign, IL.

Intel. 2017. Intel Fortran Compiler 18.0 Developer Guide and Reference. Accessed Jan. 30, 2019. https://software.intel.com/en-us/fortran-compiler-developer-guide-and-reference.

Keel, B., and W. Snelling. 2018. Comparison of Burrows-Wheeler transform-based mapping algorithms used in high-throughput whole-genome sequencing: Application to Illumina data for livestock genomes. Front. Genom. 9:35.

Kelly, B. J., J. R. Fitch, Y. Hu, D. J. Corsmeier, H. Zhong, A. N. Wetzel, R. D. Nordquist, D. L. Newsom, and P. White. 2015. Churchill: An ultra-fast, deterministic, highly scalable and balanced parallelization strategy for the discovery of human genetic variation in clinical and population-scale genomics. Genome Biol. 16:6.

Kessner, D., and J. Novembre. 2015. Power analysis of artificial selection experiments using efficient whole genome simulation of quantitative traits. Genetics 199:991–1005.

Li, H., and R. Durbin. 2009. Fast and accurate short read alignment with Burrows–Wheeler transform. Bioinformatics 25:1754–1760.

Li, H., B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, and R. Durbin. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079.

Li, H., J. Ruan, and R. Durbin. 2008. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 18:1851–1858.

Lunenberg, J. 2014. Ultra-fast, accurate and cost-effective NGS read alignment validated for complex whole plant genomes. Abstract P1032 in Proc. Plant Anim. Genome Conf. XXII, San Diego, CA. Accessed Jan. 18, 2018. https://pag.confex.com/pag/xxii/webprogram/Paper9842.html.

O'Fallon, B. D., W. Wooderchak-Donahue, and D. K. Crockett. 2013. A support vector machine for identification of single-nucleotide polymorphisms from next-generation sequencing data. Bioinformatics 29:1361–1366.

Pabinger, S., A. Dander, M. Fischer, R. Snajder, M. Sperk, M. Efremova, B. Krabichler, M. R. Speicher, J. Zschocke, and Z. Trajanoski. 2014. A survey of tools for variant analysis of next-generation genome sequencing data. Brief. Bioinform. 15:256–278.

Paten, B., A. M. Novak, J. M. Eizenga, and E. Garrison. 2017. Genome graphs and the evolution of genome inference. Genome Res. 27:665–676.

Pirooznia, M., M. Kramer, J. Parla, F. S. Goes, J. B. Potash, W. R. McCombie, and P. P. Zandi. 2014. Validation and assessment of variant calling pipelines for next-generation sequencing. Hum. Genomics 8:14.

Rosen, B. D., D. M. Bickhart, R. D. Schnabel, S. Koren, C. G. Elsik, A. Zimin, C. Dreischer, S. Schultheiss, R. Hall, S. G. Schroeder, C. P. Van Tassell, T. P. L. Smith, and J. F. Medrano. 2018. Modernizing the bovine reference genome assembly. Page 802 in Proc. World Congr. Genet. Appl. Livest. Prod., Auckland, New Zealand. Accessed Jan. 30, 2019. http://www.wcgalp.org/system/files/proceedings/2018/modernizing-bovine-reference-genome-assembly.pdf.

The 1000 Genomes Project Consortium. 2012. An integrated map of genetic variation from 1,092 human genomes. Nature 491:56–65.

Tithi, S. S., L. S. Heath, and L. Zhang. 2015. SNPwise: A SNP-aware short read aligner. Pages 187–192 in Proc. 7th Int. Conf. Bioinform. Comput. Biol. (BICoB 2015), Honolulu, HI. F. Saeed and N. Haspel, ed. International Society for Computers and Their Applications, Winona, MN.

Van der Auwera, G. A., M. O. Carneiro, C. Hartl, R. Poplin, G. del Angel, A. Levy-Moonshine, T. Jordan, K. Shakir, D. Roazen, J. Thibault, E. Banks, K. V. Garimella, D. Altshuler, S. Gabriel, and M. A. DePristo. 2013. From FastQ data to high-confidence variant calls: The genome analysis toolkit best practices pipeline. Curr. Protoc. Bioinformatics 43:11.10.1–11.10.33.

VanRaden, P. M., C. Sun, and J. R. O'Connell. 2015. Fast imputation using medium or low-coverage sequence data. BMC Genet. 16:82.

Yuan, S., H. R. Johnston, G. Zhang, Y. Li, Y.-J. Hu, and Z. S. Qin. 2015. One size doesn't fit all—RefEditor: Building personalized diploid reference genome to improve read mapping and genotype calling in next generation sequencing studies. PLOS Comput. Biol. 11:e1004448.

Zheng, Q., and E. A. Grice. 2016. AlignerBoost: A generalized software toolkit for boosting next-gen sequencing mapping accuracy using a Bayesian-based mapping quality framework. PLOS Comput. Biol. 12:e1005096.

Zimin, A. V., A. L. Delcher, L. Florea, D. R. Kelley, M. C. Schatz, D. Puiu, F. Hanrahan, G. Pertea, C. P. Van Tassell, T. S. Sonstegard, G. Marçais, M. Roberts, P. Subramanian, J. A. Yorke, and S. L. Salzberg. 2009. A whole-genome assembly of the domestic cow, *Bos taurus*. Genome Biol. 10:R42.