BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 11 2014, pages 1508–1513
doi:10.1093/bioinformatics/btu071

Genome analysis

Advance Access publication February 12, 2014

Supercomputing for the parallelization of whole genome analysis

Megan J. Puckelwartz[1], Lorenzo L. Pesce[2], Viswateja Nelakuditi[3], Lisa Dellefave-Castillo[1], Jessica R. Golbus[1], Sharlene M. Day[4], Thomas P. Cappola[5], Gerald W. Dorn, II[6], Ian T. Foster[2] and Elizabeth M. McNally[1,3,*]

[1]Department of Medicine, [2]Computation Institute and Argonne National Laboratory, 9700 S. Cass Ave. Argonne, IL 60439, USA, [3]Department of Human Genetics, The University of Chicago, 5841 S. Maryland Ave Chicago, IL 60637, USA, [4]Department of Internal Medicine, The University of Michigan, 1150 W Medical Center Dr. Ann Arbor, MI 48109, USA, [5]Perelman School of Medicine, Penn Cardiovascular Institute and Department of Medicine, University of Pennsylvania, 3400 Civic Center Blvd. Philadelphia, PA 19104, USA and [6]Washington University School of Medicine, 660 S. Euclid Ave. St. Louis, MO 63110, USA

Associate Editor: John Hancock

ABSTRACT

Motivation: The declining cost of generating DNA sequence is promoting an increase in whole genome sequencing, especially as applied to the human genome. Whole genome analysis requires the alignment and comparison of raw sequence data, and results in a computational bottleneck because of limited ability to analyze multiple genomes simultaneously.

Results: We now adapted a Cray XE6 supercomputer to achieve the parallelization required for concurrent multiple genome analysis. This approach not only markedly speeds computational time but also results in increased usable sequence per genome. Relying on publically available software, the Cray XE6 has the capacity to align and call variants on 240 whole genomes in ~50 h. Multisample variant calling is also accelerated.

Availability and implementation: The MegaSeq workflow is designed to harness the size and memory of the Cray XE6, housed at Argonne National Laboratory, for whole genome analysis in a platform designed to better match current and emerging sequencing volume.

Contact: emcnally@uchicago.edu

Received on December 2, 2013; revised on January 26, 2014; accepted on January 27, 2014

1 INTRODUCTION

With the advent of massively parallel DNA sequencing, the rate at which human genome variation can be determined is limited less by sequence generation but instead by the computational tools required to analyze these data. With current sequencing technology using short sequence reads of ~100 bp, whole genome analysis (WGA) requires the cleaning, aligning and interpreting of a billion sequence reads per single genome. With focus on scalability, we sought to improve the timeline required to process whole genome sequencing (WGS) by optimizing extraction, alignment, processing and variant calling. We reasoned that supercomputing capacity was better suited to parallelize WGA and allow for the rapid simultaneous analysis of multiple genomes.

Beagle is a Cray XE6 supercomputer housed at Argonne National Laboratory and administered by the Computation Institute at the University of Chicago. Beagle has ~726 compute nodes each with 32 GB of memory. Each node has 24 cores, 2.1 GHz cores on two AMD "Magny-Cours" processors. The XE6 can work in both Extreme Scalability Mode for scalable applications and Cluster Compatibility Mode for use with programs that are designed for smaller machines or clusters, such as the freely available genomics tools that are now routinely implemented for WGA (Li and Durbin, 2009; McKenna *et al*., 2010). While parallelization is possible on smaller systems, both memory and computational core number limit the capacity for simultaneous computation. Beagle uses a parallel computation environment and a parallel file system (Lustre) based on shared storage. Having both a parallel computation environment and external disk storage based on a parallel file system ensures that each node (and core) is able to access all data at any time without waiting for transfer of data across nodes. In this system, nodes have no local storage, and therefore no disk-to-disk transfer is required. On clusters without a shared file system, data transfer across nodes during analysis can be a time-intensive process. Here, we describe a workflow referred to as MegaSeq that uses the MapReduce (Dean and Ghemawat, 2008) approach to take advantage of supercomputing size and memory.

2 MATERIALS AND METHODS/RESULTS

2.1 Workflow-alignment

A summary of the MegaSeq workflow is shown in Figure 1. We adapted Beagle for WGA using a test dataset of 61 genomes that had been determined by Illumina Inc. Illumina provided WGS from 100 bp paired end reads as bam files for data transfer. FASTQ files were extracted from bam files using the Picard tool, SamToFastq (http://picard.sourceforge.net), and extraction was performed by readgroup (Table 1). The readgroup tag provides information on sample identity, library of origin and sequencing machine lane (see SAM Format Specification, *samtools.sourceforge.net/SAM1.pdf)*. The extraction step is unnecessary if FASTQ sequence data are directly available. In the present cohort, each genome was represented by ~3–4 readgroups, creating a natural division of the reads. Alignment was performed with the Burrows–Wheeler Aligner (BWA) on 2n nodes, with n equal to the number of readgroups (Li and Durbin, 2009). Each readgroup alignment was concurrent on 24 cores (Table 1). Alignment displayed a linear speedup and therefore

*To whom correspondence should be addressed.

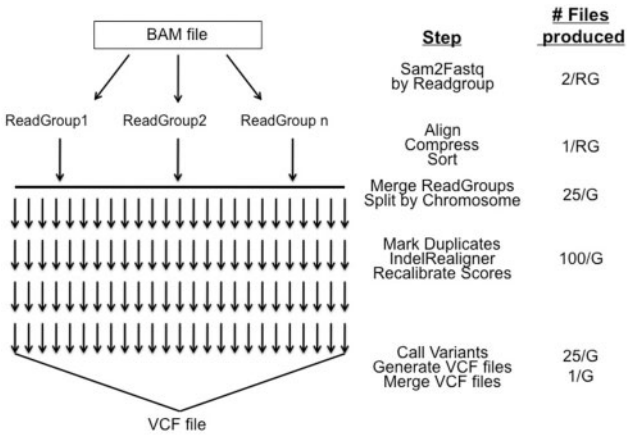

Fig. 1. Schematic of MegaSeq workflow. Each step and the number of files produced per readgroup (RG) or genome (G) are listed. To improve speed, a MapReduce approach was used with sequences being split into smaller groups. Sequences were initially split by readgroup, which roughly represents each lane of sequence. After alignment, readgroups were merged and the aligned genome was split by chromosome. After variants were called per chromosome, VCF files were merged, creating one file per individual human genome

scaled with the number of cores used. Scaling is limited to the number of cores per node available because it is based on shared-memory threads. After reads were aligned, alignment files were converted from sai to sam files using the bwa samse (for single-end reads) or sampe (for paired-end reads). To improve performance, we used the bwa 0.5.9-tpx.patch (ftp.conveysupport.com/outgoing/bwa). The tpx patch creates a threaded parallel bwa samse/pe, allowing for the use of multiple cores on a node. The speedup of this step was considerably less efficient than alignment, resulting in imperfect scaling, but is still considerably faster than standard bwa samse/pe.

SAM files were then compressed to bam files and sorted using samtools (Li *et al.*, 2009). Readgroups were merged and each genome was split by chromosome to better use Beagle's size and memory.

2.2 Workflow-clean-up

The alignments were then "cleaned" to improve reliability of variant calling. Potential PCR artifacts were marked with the Picard MarkDuplicates tool (http://picard.sourceforge.net). Notably, alignment of genomic information to the referent genome is often imperfect, especially in areas around small base pair insertions and deletions (indels). Therefore, indels within an individual genome may misalign. Because alignments to the referent genome are performed using each short sequence read individually, multiple alignment information is not available to help identify mismatches. Two tools available from the Broad Institute's Genome Analysis Tool Kit (GATK), RealignerTargetCreator and IndelRealigner, were applied to help facilitate the identification of Indels. When used together, these tools use the full alignment context to determine whether an Indel exists (DePristo *et al.*, 2011). Finally, quality scores for sequence reads were recalibrated using BaseRecalibrator and PrintReads from GATK to more closely match the actual probability of mismatching the referent genome and to correct for any variation in quality between machine cycle and sequence context (DePristo *et al.*, 2011). At this stage, the data are still split by chromosome. The efficiency of BaseRecalibrator is proportional to the number of base pairs provided. Because smaller datasets may have poor recalibration, splitting should be avoided with smaller datasets. On smaller datasets, the CRAY XE6 has the capability to perform recalibration across the genome in a reasonable time frame. Regardless of dataset size, recalibration results should always be reviewed by the end-user.

Table 1 details the programs and flags used for each operation. Each step was performed on 25 cores concurrently, with approximately six nodes dedicated to each genome (Table 1). On Beagle, running multiple genomes simultaneously is more efficient than running a single genome because of the structure of both Beagle and the genome. For the cleaning steps, each genome was run on 25 cores (24 chromosomes plus the mitochondrial genome). Four chromosomes were analyzed on each node, and therefore each genome needed six nodes, plus one core. During testing of the pipeline, we noted that node failure occurred as a result of java errors, often associated with memory allocation. To improve memory usage and avoid node failure, we hard coded the chromosomes into groups that are always sent out together. This scheme allows us to send the largest and most memory intensive chromosomes with smaller, less taxing ones resulting in fewer memory issues. For java run programs we used 28 GB memory/number of jobs per node. We also found that for java programs, using two threads for garbage collection better managed memory issues allowing us to pack more jobs per node. For each clean-up step, threading was used, where available (Table 1).

2.3 Workflow – variant calling

Variants were called using the HaplotypeCaller from GATK. HaplotypeCaller calls both single nucleotide polymorphisms and indels using *de novo* assembly of haplotypes in the active region. Haplotypes are evaluated using an affine gap penalty pair hidden Markov model (DePristo *et al.*, 2011). Table 1 provides flags used for each step. MegaSeq identifies both single nucleotide variants (SNVs) and Indels simultaneously on ~6 nodes with 25× concurrency per genome. After variants were called and exported in variant call format (VCF), we used the GATK tool VariantFiltration to filter variants (DePristo *et al.*, 2011). Variants were removed from the analysis using these criteria: biallelic balance >0.75; quality score <30; depth of coverage >360; strand bias more than −0.01 and mapping quality zero reads ≥10. Variants were then annotated using the default parameters of snpEff. snpEff is a fast variant annotation and effect prediction tool that is integrated with GATK (Cingolani *et al.*, 2012).

2.4 Testing the workflow

The above workflow was tested using data from 61 human genomes. The starting FASTQ file size of each genome was ~300 GB, requiring ~18 TB of space to process all individuals simultaneously. Reads were aligned to NCBI reference genome 37.1 (hg19). We compared alignment output of MegaSeq with that produced by Illumina using the proprietary alignment/variant calling software Eland/Casava because this software is designed to efficiently perform WGA (Cox, 2007). MegaSeq alignment using BWA resulted in greater coverage with a mean coverage of 40.0× compared with 37.2× for ELAND/Casava's alignment across all 61 genomes (paired *t*-test, $P < 0.0004$, Fig. 2a). The mean percent of the non-N reference genome covered was also greater with MegaSeq compared with Eland/Casava (98.7 versus 98.0) using MegaSeq versus Illumina (paired *t*-test, $P < 0.0001$, Fig. 2b). In total, 285 896 445 variants were called by MegaSeq across the 61 genomes identifying ~4.5 million variants per individual. To compare variants between MegaSeq and Illumina's software, only variants with a quality score ≥30 were included. The mean number of SNVs called per genome differed between MegaSeq and ELAND/Casava data (3.670×10^6 versus 3.736×10^6, MegaSeq and ELAND/Casava respectively, paired *t*-test, $P < 0.0001$, Fig. 2c). ELAND/Casava also called more indels (536 853 versus 618 779, MegaSeq and ELAND/Casava, respectively, paired *t*-test, $P < 0.0001$) (Fig. 2d). There was 88% concordance between MegaSeq and ELAND/Casava SNV calls and only 64.7% concordance for indels (Fig. 3a).

Table 1. Computational approach to genome analysis using the massively parallel Cray XE6 supercomputer

Step	Program	Module/Call	Input	Parameters	Output	Number per genome	Number of active cores/ node	Number nodes	Concurrency
Extract fastq	Picard 1.98	SamToFastq	bam	default	fastq	2(#RGs)	1	1	1
Alignment	BWA 0.5.9	bwa aln	fastq	-qtrim 15	sai file	2(#RGs)	24	2(#RGs)	24(#RGs)
Convert sai to sam	BWA tpx 0.5.9	bwa sampe	sai	-T –X –P (BWA)	sam	2(#RGs)	~24	#RGs	24(#RGs)
Compression	samtools 0.1.18	view	sam	-bo (samtools)	bam	2(#RGs)	3	#RGs	1(#RGs)
Split	samtools 0.1.18	view/merge	bam	-h (preserve readgroup)	bam	25	24	1	25
Sort	Picard 1.98	SortSam	bam	-coordinate	bam	25	4 + GC	~6	25
Mark duplicates	Picard 1.98	Mark Duplicates	bam	-REMOVE_DUPLICATES false	bam	25	4 + GC	~6	25 (each with 3 threads)
Reorder	Picard 1.98	ReorderSam	bam	default	bam	25	4 + GC	~6	25 (GC[a])
Identify indel re-alignment targets	GATK 2.7-1	GATK	bam	-T RealignerTargetCreator -L <chromosome ID>	intervals	25	4 + GC	~6	25 (GC)
Realign targeted intervals	GATK 2.7-1	GATK	bam	-T IndelRealigner -targetIntervals <intervals> -LOD 5 -L <chromosome ID>	bam	25	4 + GC	~6	25 (GC)
Base recalibrator	GATK 2.7-1	GATK	bam	-T BaseRecalibrator -cov ReadGroupCovariate -cov QualityScoreCovariate -cov CycleCovariate -cov ContextCovariate -knownSites dbSNP_135	csv	25	4 + GC	~6	25 (GC)
Print reads	GATK 2.7-1	GATK	bam	-T PrintReads -baq RECALCULATE -baqGOP 30 -BQSR <csv file>	bam	25	4 + GC	~6	25 (GC)
Call variants	GATK 2.7-1	GATK	bam	-T Haplotype Caller -L <chromosome ID> -D dbSNP_135.hg19.vcf -A AlleleBalance -A Coverage -A HomopolymerRun -A FisherStrand -A HaplotypeScore -A HardyWeinberg -A ReadPosRankSumTest -A QualByDepth -A MappingQualityRankSumTest -A VariantType -A MappingQualityZero -minPruning 10 -stand_call_conv 30.0 -stand_emit_conv 10.0	vcf	25			
Filter variants	GATK 2.7-1	GATK	bam	-T VariantFiltration -L <chromosome ID> --clusterWindowSize 10 --filterExpression "(AB?: 0) > 0.75 || -QUAL < 30.0 || DP > 360 || SB > −0.1 || MQ0 ≥ 10"	vcf	25	4 + GC	~6	25 (GC)
Annotate variants	snpEff 2.0.5	snpEff	vcf	default	vcf	25	4	~6	

Note: RG = readgroup. [a]GC = 2 threads used for java Garbage Collection.

To estimate validity of the calls, we examined the normalized density of quality scores. Concordant SNVs, called by both MegaSeq and ELAND/Casava, had higher quality scores compared with non-concordant SNVs (Fig. 3b, MegaSeq, light purple; ELAND/CASAVA, dark purple). Non-concordant SNVs had lower quality scores, especially in the ELAND/Casava call set (Fig. 3b, MegaSeq, red; ELAND/Casava, blue). This same pattern was evident for indels (Fig. 3c). Non-concordant SNVs identified by MegaSeq have quality scores more closely resembling the concordant calls. These data indicate that the ELAND/Casava call set contained more low quality variants than the MegaSeq call set. We next compared depth of sequence reads called by MegaSeq or ELAND/Casava. Depth is a by-product of alignment, with higher depth indicating a more reliable variant call. Concordant SNVs called by both MegaSeq and ELAND/Casava have similar normalized depths, with the highest density of calls occurring between 35–40×. The non-concordant SNVs have markedly different depth distributions (Fig. 3d). SNVs identified solely by ELAND/Casava had the highest density of calls at ~20× depth. In contrast, those SNVs identified solely by MegaSeq had a depth density that more closely matched the concordant SNV distribution, with the majority of calls occurring between 30–35× depth. These data

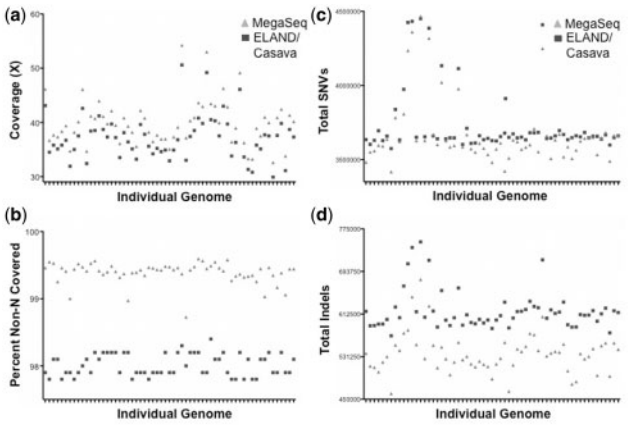

Fig. 2. MegaSeq identifies more usable sequence and fewer SNVs and Indels per genome. WGA from 61 individual genomes was compared between MegaSeq using BWA/GATK on the Cray XE6 and ELAND/CASAVA from Illumina because ELAND/CASAVA is aimed at efficiency (Cox, 2007). (**a**) The mean coverage for each genome was higher with MegaSeq (light gray triangles) compared with ELAND/Casava (black squares) (40.0 × for MegaSeq and 37.2× for ELAND/Casava [paired *t*-test, $P<0.0004$)]. (**b**) The percentage of non-N genome covered by MegaSeq (light gray triangles) was greater than ELAND/Casava (black squares) for each genome (98.7 and 98.0, MegaSeq and ELAND/Casava, respectively [paired *t*-test, $P<0.0001$)]. (**c**) The total number of SNVs identified per genome with MegaSeq (light gray triangles) was less than with ELAND/Casava (black squares) [3.670×10^6 and 3.736×10^6 for MegaSeq and ELAND/Casava, respectively, paired *t*-test, $P<0.0001$)]. (**d**) MegaSeq (light gray triangles) identified fewer indels compared with ELAND/Casava (black squares) in each genome [mean number of indels 536 853 and 618 779 for MegaSeq and ELAND/Casava, respectively, paired *t*-test, $P<0.0001$)]

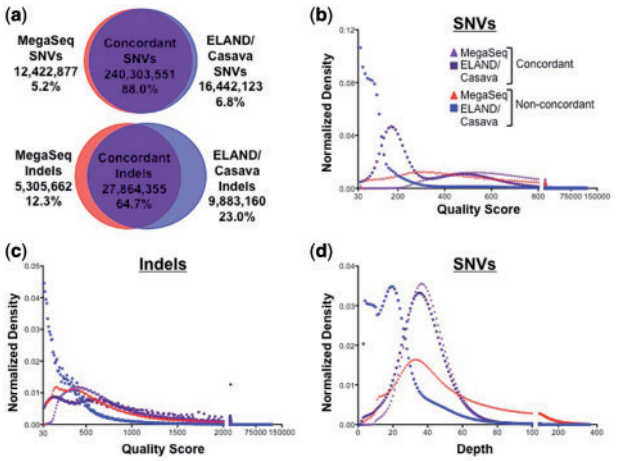

Fig. 3. Concordance between MegaSeq and Illumina. We compared the output from MegaSeq with that provided by Illumina, which uses ELAND/CASAVA because these algorithms are optimized for speed. (**a**) A total of 88.0% of SNVs were identified by both MegaSeq and ELAND/Casava. Over 12 million and 16 million SNVs were only identified by MegaSeq (red) and ELAND/Casava (blue), respectively. In all, 64.7% of Indels were concordant between MegaSeq (red) and ELAND/Casava (blue). (**b**) The non-concordant variants found by MegaSeq have higher quality scores than the non-concordant variants scored by ELAND/Casava. Normalized quality score densities are similar for concordant SNVs identified by both MegaSeq (light purple triangles) and ELAND/Casava (dark purple squares). Non-concordant SNVs found only by MegaSeq (red triangles) had higher quality scores than non-concordant SNVs identified only by ELAND/Casava (blue squares). (**c**) Similarly, non-concordant indels found only by Megaseq had higher quality scores than those found only by ELAND/Casava. (**d**) MegaSeq non-concordant SNVs have higher depth than ELAND/Casava non-concordant SNVs

indicate that MegaSeq calls a greater number of high confidence SNVs, based on both quality score and depth.

2.5 Speed of analysis

Computational time is a major bottleneck in WGA. We calculated the central processing unit (CPU) time that would be required on a single 2.1 GHz processor as 1701 h (0.20 years) for a single genome. This time can vary based on clock speed, memory speed and disk speed. These calculations are bound by disk and memory much more than by CPU clock speed. The SamToFastq step requires ~48 CPU h, but is only necessary when genomes are delivered as the compressed pre-aligned bam format and, as such, will not be a consideration for many users. Alignment scales perfectly and therefore cannot be easily accelerated. We suggest that these single genome times may reflect the maximum speed for the Beagle supercomputer.

To highlight the power of a parallel system, we calculated hypothetical run times on a single core, a 3 (24-core) node cluster and compared them with the CRAY XE6 run times. Using CRAY XE6 execution time as a reference, we predicted total CPU time required for WGA of 240 genomes on a single core to be ~47 years. A total of ~11.8 years of CPU time is required to complete the workflow for 61 genomes. A biological computing cluster with 3 nodes can accelerate this time to ~7.2 months. Using the MegaSeq workflow on Beagle, 61 genomes were analyzed in 50.3 real time hours (Fig. 4a). Based on the size and number of nodes available, Beagle has the capacity to perform WGA on 240 genomes in the same time frame (50.3 h) by using additional nodes. Additional genomes would need to be run consecutively as the number of nodes in use is exhausted.

Multisample variant calling can also be performed using the MegaSeq workflow. Multisample calling reduces the false discovery rate, but is computationally intensive. We performed multisample calling on 61 genomes using HaplotypeCaller from GATK with the flags noted in Table 1. To take advantage of the size of Beagle, we split the genomes by chromosome, then further split chromosomes into overlapping subunits that varied based on total chromosome size resulting in 2400 total subunits. For 61 genomes, ~600 nodes were used with four jobs per node, requiring ~40 000 CPU hours. Calls were completed in ~16 h, real time. A biological computing cluster with three nodes would require ~4.4 months real time to perform multisample calling on 61 genomes using HaplotypeCaller. We estimate that ~154 224 CPU h (~17.8 years) would be required to complete 240 genomes. We estimate that a 3 node cluster would take ~1.3 years to complete multisample calling on 240 genomes. In real time using 600 nodes with four jobs per node, we estimate Beagle can complete multisample calling using HaplotypeCaller on 240 genomes in ~2.5 days, making Beagle an excellent resource for multisample calling. After calling the resulting 2400 vcf files were merged and overlapping calls were removed.

2.6 Space management

Space constraints are another major hurdle in large volume WGA. We estimate that approximately 1 TB of space per genome was needed to complete the MegaSeq analysis (Fig. 4b). Although it is possible to discard

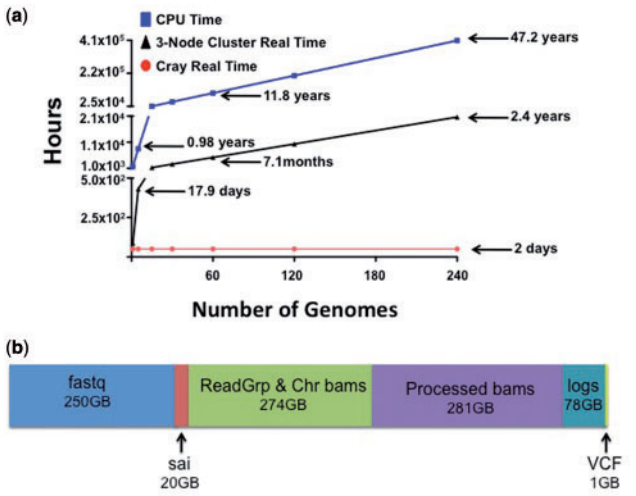

Fig. 4. Time and space constraints. (**a**) Graph illustrating amount of both CPU time and real time needed to analyze genomes. CPU time (blue squares) scales linearly with the number of genomes analyzed. We calculated the real time needed by a hypothetical 3-node cluster (black triangles) and estimated that 240 genomes would take ~2.4 years to analyze. The total real time required on a 740 node Cray XE6 massively parallel supercomputer (red circles) is ~2 days for 240 genomes. (**b**) Schematic illustrating space requirements for WGA. Each genome requires ~1 TB of space. The final vcf file for each genome is ~1 GB

output at each successive step, there is still a large volume of data that were active at any single step. For instance, at the SamToFastq step, 18 TB was required for the FASTQ files. During the cleanup steps, each genome required ~85 GB of space; ~5 TB of space was needed to process each step for 61 genomes. Because each step requires computational time, discarding data from earlier steps can be costly if there is an analysis failure. These space requirements are generally not met by small computing clusters or by larger clusters with many users. The final merged VCF file for each individual genome required ~1 GB of space.

3 CONCLUSION

The development of next-generation sequencing technology has transformed the genetic analysis paradigm, from examination of the coding region of a handful of genes, to the current methods of interrogation of the entire coding region of the genome and, finally, to analysis of the entire 3 billion base pair genome. While cost and time are no longer major barriers to whole genome sequencing, data analysis and storage are major bottlenecks in using whole genome data (Metzker, 2010). Currently, whole genome sequencing using the Illumina HiSeq 2000 platform at moderate coverage (30–50×) yields >100 GB of data. This can be an overwhelming amount of data to process and store, and therefore many have turned to exome sequencing. Exome sequencing focuses on evaluating variation in the coding portion of the genome (~1–2%), and therefore provides smaller and less complex data to manage than whole genome sequencing. Exome sequencing is useful, as ~85% of recognized disease-causing mutations are located in protein coding regions of the genome (Majewski *et al.*, 2011). However, this finding reflects a bias in our ability to evaluate non-coding variation.

Recently, great strides have been made to decipher the other 98% of genome sequence. The ENCODE project (Encyclopedia of DNA Elements) has assigned biochemical functions to 80% of the genome (Bernstein *et al.*, 2012). These annotations will prove valuable tools in evaluating non-coding variation. Only whole genome sequencing can be used to interrogate non-coding variation, although the complexity of WGA has limited this possibility. Whole genome sequencing may also be better suited than exome sequencing to assess structural variation in genomes. Structural variants are emerging as important factors in human disease, making them an important factor in weighing the benefits of whole genome sequencing in relation to the challenges of computation (Snyder *et al.*, 2010; Spielmann and Mundlos, 2013). Thus, whole genome sequencing may be the method of choice for many researchers if not for the tremendous computational bottleneck.

The deluge of genetic data is appropriate for high-performance computing and large-scale storage options (Koboldt *et al.*, 2010). Sequence analysis includes read alignment to a reference genome, alignment clean-up and variant calling. A number of resources are freely available for analysis of genome sequencing, including BWA, GATK and snpEff. These tools can be used to align and call variants from a single genome by most laboratories, even those with limited computational experience and resources. However, high-throughput analysis of many genomes is significantly accelerated by parallelization and better meets the needs of the genetics community.

A common approach for analysis has relied on computing clusters, and more recently, cloud-based computing. By transitioning WGS to a supercomputing environment, we achieved high reliability with accelerated speed. One of the more cumbersome problems with clusters and cloud-based computing involves long wait times for data transfer between nodes (Zhao *et al.*, 2013). The Cray XE6 supercomputing environment described here eliminates these wait times by using a parallel file system (Lustre) without creating a resource conflict bottleneck. A parallel file system, like Lustre, removes the need for tracking of data location, leaving only the issues of cache, RAM and disk hierarchy (Eijkhout, 2013). The demonstration that whole genome sequences can be aligned, cleaned and interpreted in parallel was achieved by using BWA/GATK, robust, publicly available software packages, in the Cray XE6 environment. Notably, this method uses the same software packages commonly used in computing clusters, but takes advantage of the Cray platform to parallelize the analysis. The ability to apply multisample variant calling significantly improves reliability and begins to extend analysis to beyond what is possible in a cluster environment. The application of the Cray XE6 has the capacity to analyze, in parallel, as many as 240 genomes in ~50 h. This is a platform-dependent workflow that serves as proof of principle that genome analysis is greatly accelerated when performed on a supercomputer. More importantly, this work demonstrates that the publically available software currently in use for genome analysis is amenable to the supercomputing environment and can be installed as is on a CRAY XE6 and likely other systems, although we have not tested those systems. Currently, MegaSeq is available on the Beagle supercomputer at the University of Chicago.

The MegaSeq workflow backbone is based on bash shell instruction, and the submission subscripts are based on Portable Batch System (PBS) commands and are adaptable to other batch systems including Sun grid engine or SLURM, because the parallel logical structure of the workflow is compatible. Disk, memory and CPU usage will likely require optimization because of differences in machine design, which may affect bottlenecks and stability. The workflow should port directly with only minimal modifications to any Cray XE6, CRAY XC30 and related systems. The Beagle supercomputer is an NIH supported resource and provides an opportunity for large-scale genome projects. This computing application provides a format where human WGS can be rapidly analyzed relieving major constraint for better defining the range and utility of human genome variation.

Funding: This work was supported in part by National Institutes of Health through resources provided by the Computation Institute and the Biological Sciences Division of the University of Chicago and Argonne National Laboratory [S10 RR029030-01], and NIH AR052646, NIH HL61322, NIH NS072027, and the Doris Duke Charitable Foundation.

Conflict of Interest: Spouse receives patent royalties related to DNA sequencing (EMM).

REFERENCES

Bernstein,B.E. *et al.* (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature*, **489**, 57–74.

Cingolani,P. *et al.* (2012) A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly (Austin)*, **6**, 80–92.

Cox,A. (2007) *ELAND: Efficient Large-Scale Alignment of Nucleotide Databases*. Illumina, San Diego, CA.

Dean,J. and Ghemawat,S. (2008) Mapreduce: simplified data processing on large clusters. *Commun. ACM*, **51**, 107–113.

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

Eijkhout,V. (2013) Introduction to High Performance Scientific Computing. lulu.com (December 2013, date last accessed).

Koboldt,D.C. *et al.* (2010) Challenges of sequencing human genomes. *Brief. Bioinform.*, **11**, 484–498.

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

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Majewski,J. *et al.* (2011) What can exome sequencing do for you? *J. Med. Genet.*, **48**, 580–589.

McKenna,A. *et al.* (2010) The Genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.

Metzker,M.L. (2010) Sequencing technologies - the next generation. *Nat. Rev. Genet.*, **11**, 31–46.

Snyder,M. *et al.* (2010) Personal genome sequencing: current approaches and challenges. *Genes Dev.*, **24**, 423–431.

Spielmann,M. and Mundlos,S. (2013) Structural variations, the regulatory landscape of the genome and their alteration in human disease. *Bioessays*, **35**, 533–543.

Zhao,S. *et al.* (2013) Rainbow: a TOOL for large-scale whole-genome sequencing data analysis using cloud computing. *BMC Genomics*, **14**, 425.