PEMapper and PECaller provide a simplified approach to whole-genome sequencing

H. Richard Johnston, Emory University
Pankaj Chopra, Emory University
Thomas Wingo, Emory University
Viren Patel, Emory University
Michael Epstein, Emory University
Jennifer Mulle, Emory University
Stephen Warren, Emory University
Michael Zwick, Emory University
David Cutler, Emory University

Journal Title: Proceedings of the National Academy of Sciences
Volume: Volume 114, Number 10
Publisher: National Academy of Sciences | 2017-03-07, Pages E1923-E1932
Type of Work: Article | Final Publisher PDF
Publisher DOI: 10.1073/pnas.1618065114
Permanent URL: https://pid.emory.edu/ark:/25593/s4wjz

Final published version: http://dx.doi.org/10.1073/pnas.1618065114

Copyright information:
© 2017, National Academy of Sciences.

Accessed January 8, 2018 8:06 AM EST
PEMapper and PECaller provide a simplified approach to whole-genome sequencing

H. Richard Johnston,a,b,c Pankaj Chopra,a Thomas S. Wingo,a,b,c,d Viren Patel,c International Consortium on Brain and Behavior in 22q11.2 Deletion Syndrome1 Michael P. Epstein,a Jennifer G. Mulle,a,e Stephen T. Warren,a,b,c,d Michael E. Zwick,a,d and David J. Cutler,a,d

aDepartment of Human Genetics, Emory University School of Medicine, Atlanta, GA 30322; bDepartment of Biostatistics and Bioinformatics, Emory University Rollins School of Public Health, Atlanta, GA 30322; cDivision of Neurology, Atlanta Veterans Affairs Medical Center, Atlanta, GA 30322; dDepartment of Neurology, Emory University School of Medicine, Atlanta, GA 30322; eDepartment of Epidemiology, Emory University Rollins School of Public Health, Atlanta, GA 30322; Department of Pediatrics, Emory University School of Medicine, Atlanta, GA 30322; and Department of Biochemistry, Emory University School of Medicine, Atlanta, GA 30322

Contributed by Stephen T. Warren, January 25, 2017 (sent for review October 31, 2016; reviewed by John Blangero and Andrew G. Clark)

The analysis of human whole-genome sequencing data presents significant computational challenges. The sheer size of datasets places an enormous burden on computational, disk array, and network resources. Here, we present an integrated computational package, PEMapper/PECaller, that was designed specifically to minimize the burden on networks and disk arrays, create output files that are minimal in size, and run in a highly computationally efficient way, with the single goal of enabling whole-genome sequencing at scale. In addition to improved computational efficiency, we implement a statistical framework that allows for a base by base error correction, with the single goal of enabling whole-genome sequencing at scale

Whole-genome sequencing (WGS) using short reads on the Illumina platform is an increasingly cost-effective approach for identifying genetic variation, with growing potential for both research and clinical applications (1–4). A critical challenge is in the development of efficient algorithms capable of rapidly and accurately identifying variable sites from among the enormous collection of sequence reads (5). Given the large size of eukaryotic genomes, even modest false-positive or negative error rates can act as barriers to the success of genetic studies and would inhibit the utility of such studies for both research and clinical applications.

The de facto standard methodology for mapping and calling variants is the so-called Burrows–Wheeler aligner (BWA)/Genome Analysis Toolkit (GATK) best practices pipeline (6), which was devised and validated for whole-exome experiments and has greatly facilitated whole-exome studies for identification of disease-causing variants (7–9). Although this pipeline can be used successfully at whole-genome scales, there are barriers to its use, particularly as the number of samples increases. BWA (10), Bowtie (11), and most other commonly used read mapping software packages are designed to run in low-memory footprints [i.e., less than 8 or 16 gigabytes (GB) random-access memory (RAM)]. Because whole-genome datasets are large (necessarily greater than 100 GB uncompressed for 30x coverage), these read mappers must continuously read and write large quantities of data to and from the disk. Sorting reads, in particular, is highly disk input/output (I/O)-intensive. Although a high-performance disk array can provide the needed I/O performance for a single instance of BWA/GATK processing (6), no disk array can possibly accommodate the I/O performance required to run multiple GATK instances simultaneously on parallel processors. Moreover, even if the disk array itself could meet the demand, the network/fiber interconnects between the array and the computational nodes quickly become saturated. Simply put, although BWA/GATK best practices do an excellent job in a nonclustered environment, the “network cost” in a clustered environment significantly limits its performance for large WGS datasets.

GATK best practices have additional limitations. First, output files can be quite large. Binary sequence alignment/map (BAM) files, required to store sequence alignment data, are almost always larger than the initial fastq files of nucleotide sequences, and Haplotype Caller (HC) output can be nearly one-half the size of the BAM files. Thus, total storage requirements to run the pipeline can approach 300 GB compressed per sample for WGS data. Second, variant calling begins with individual samples (not collections of samples; i.e., joint calling), and as a result, the distinction between sites called as homozgyous reference genotypes and those called as missing (insufficient evidence to make a call) is not always maintained. Third, the GATK best practices joint genotyping caller, required to generate the highest-quality genotype calls, does not scale well to whole-genome data. As currently implemented, the joint caller simply will not run on whole-genome size files in sample collections larger than 10–20 human genomes, even on computers with 512 GB RAM. Crashing with more than 20 samples seriously limits the utility of GATK for large-scale sequencing. Finally, the entire GATK best practices pipeline relies on and uses enormous quantities of “previous knowledge” about the position and frequency of SNPs and indels (insertion/deletion variants). Using known positions of variants is both a strength, in

Significance

PEMapper and PECaller are paired software programs that simplify mapping and variant calling for whole-genome datasets. Whole-genome sequencing data are fast becoming the most natural dataset for all genetic studies. Analysis tools for data at this scale are essential. This manuscript describes tools, which solve the challenges of data analysis at whole-genome scale, using an approach involving 16-mer mapping and SNP calling based on a Pólya–Eggenberger distribution for SNP genotypes. We show that our software package is faster (cheaper to run), uses much less disk space (cheaper to store results), requires no previous knowledge of existing genetic variation (easier to deploy to nonhuman species), and achieves calling results that are as good as Genome Analysis Toolkit best practices.


Reviewers: J.B., University of Texas Rio Grande Valley School of Medicine; and A.G.C., Cornell University.

The authors declare no conflict of interest.

1A complete list of the International Consortium on Brain and Behavior in 22q11.2 Deletion Syndrome can be found in SI Appendix.

2To whom correspondence may be addressed: swarren@emory.edu, mzwick@emory.edu, or djcutler@emory.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1618065114/-/DCSupplemental.

www.pnas.org/cgi/doi/10.1073/pnas.1618065114

PNAS | Published online February 21, 2017 | E1923–E1932

GENETICS
that it leverages outside knowledge to improve performance, and a weakness, in the sense that it makes its application to nonhuman systems difficult and may also create biases in variant calling.

Here, we describe two software programs intended to overcome the limitations of GATK best practices called PEMapper and PECaller. PEMapper solves the inherent limitations of the BWA/GATK pipeline by performing all of the necessary read sorting, storing, and mapping procedures in RAM. Human genome indices are preloaded, and final output is written only once (never reloaded, resorted, etc.). These technical changes lead to substantial performance gains as detailed below. PEMapper requires a large RAM allocation (typically nearly 200 GB for the sequence of a whole human genome) but in exchange, does not overburden the network or disk subsystems. Modern computational clusters, such as those found at many universities or available from cloud providers [i.e., Amazon Web Services (AWS)], are well-equipped to run many simultaneous instances of PEMapper in parallel to expedite experiments. Additionally, output from PEMapper/PECaller comes in much smaller files, decreasing the long-term storage requirements for WGS data (Table 1).

Unlike PEMapper, with innovations that are strictly in implementation, PECaller represents an intellectual departure from several other genotype-calling models (6, 12). First, variant detection occurs simultaneously (joint calling in the initial stage) in all samples from the same experiment. Joint calling from the outset is important, because it ensures that the distinction between missing data (data with insufficient evidence for any genotype) and homozygous reference data is recognized from the inception. In addition, it allows the imposition of a population genetics-inspired prior on the data and the ability to fit sophisticated models of read error to help distinguish bases with high error rates from those that actually harbor variants. The population genetics prior accounts for the fact that most sites are expected to be invariant but conditional on the site containing a variant; the variant is expected to be in Hardy–Weinberg equilibrium. Second, another innovation of the PECaller method involves the underlying statistical model used to describe the data. Formally, we assume that read depths are drawn from a Polya–Eggenberger (PE; Dirichlet multinomial) distribution, not the more conventional multinomial assumption. Using a PE distribution allows us to model a nucleotide base as having both a relatively high “error rate” but also, importantly, a large variance in that use. The PE distribution helps us reduce false-positive variant calling, while at the same time, enabling us to call true heterozygotes, even when the relative fraction of the two alleles is highly uneven (another common occurrence). The fact that PEMapper/PECaller does not use any information about “known” SNPs or indels makes it far better suited for nonhuman systems or human diseases, such as cancer, with large numbers of de novo mutations. We show that PEMapper/PECaller, despite not using any such information, performs as well or better than GATK best practices in all aspects of variant discovery and calling.

### Results

**Performance of PEMapper/PECaller.** The simplest measure of variation, named theta (13), counts the number of heterozygotes called per sample per base. Theta is estimated to be somewhere between 0.0008 and 0.001 in humans (14, 15). Fig. 1 shows theta for each of 97 sequenced human genome samples that passed quality control (QC) (Methods). Most remarkable is the extremely consistent levels of variation called between samples, with individuals ranging from 0.0008351 to 0.0008624. The overall variation levels are consistent with previous estimates.

**False-Positive Calling.** Our analysis provides ample evidence that this called variation contains very few false-positive findings (nonvariant sites called variant in error). Sequence changes from A-\(\rightarrow\)G, G-\(\rightarrow\)A, C-\(\rightarrow\)T, and T-\(\rightarrow\)C are called “transversions.” All other changes are called “transitions.” There are twice as many transversions possible as transitions. Many mutational mechanisms favor transitions over transversions (oxidative deamination, etc.). Selection also likely favors transitions over transversions (much more likely to be silent in exons; similar binding for transcription factors; e.g., wobble binding). However, random genotype calling error likely results in increased transversions (because there are twice as many ways to get a transversion as a transition when you make an error). Thus, real data ought to be enriched for transitions over transversions, and false data ought to be enriched for transversions. Picking nucleotides at random would give a 0.5:1 transition to transition (Ts/Tv) ratio. It is widely believed that the overall Ts/Tv ratio is \(\sim 2.0\) in humans (genome.sph.umich.edu/wiki/SNP_Call_Set_Properties).

For every sample in this study, the Ts/Tv ratio was between 2.042:1 and 2.051:1 (Fig. 2). Looking at the entire collection of variants, the ratio was 2.073:1. This overall ratio can be used to estimate the fraction of false-positive variant calls. If we assume that the “true” ratio is 2.12:1, a value determined from all variants called by both PEMapper/PECaller and GATK (see below), and we assume that false-positive variant calls have a ratio of 0.5:1 (as expected by chance), then an observed ratio of 2.073:1 implies that, over all 97

![Theta Genome](image-url)
samples, ~3% of the variants were false positives. On a per sample basis, less than 1 in 3,000–5,000 called variants per sample were false positives. The data quality from PEMapper/PECaller compares favorably with that of other next-generation sequencing (NGS) analytical tools (16).

**Exonic Variation.** In general, there ought to be far less variation in whole-exome sequencing (18). Of the sequencing call was 99.

Overall, more than 98.4% of the singletons have a Ts/Tv ratio of 2.105–1, better than the PEMapper/PECaller average of 2.073–1 and very close to Ts/Tv ratio of the overlap set between GATK and PEMapper/PECaller. Therefore, singletons, despite the additional potential for false-positive calls, actually seem to be as reliable or more reliable than the set of all sites.

dbSNP 146 contains all variants currently reported in the ExAC (7) dataset as well as all variants discovered by 1000 Genomes (19). An exonic variant not found in dbSNP 146 is almost surely either a false-positive call or a variant that is exceedingly rare in the general population. Exonic sites that change the amino acid (replacement sites) and are not found in dbSNP should be the category of variation most enriched for false-positive calls. For the entire set of replacement SNPs, the Ts/Tv ratio is 2.173. Replacement SNPs in dbSNP are 2.254, whereas those not in dbSNP are 1.762. For singleton replacement SNPs, the Ts/Tv ratio is 2.328. Singleton replacement SNPs in dbSNP are 2.562, whereas those not in dbSNP are 1.846. This set of singleton replacement sites that are not found in dbSNP is the set that ought to be most enriched for false positives. Despite this enrichment, replacement sites that are not in dbSNP have a Ts/Tv ratio only ~10% lower than SNPs overall, suggestive that, although this set may be the most enriched for false positives of any possible set, it is still comprised largely of true-positive calls.

**Completeness and Accuracy.** Overall, more than 98.4% of the nonrepeat-masked genome had high-quality calls. As expected, more than 99% of these sites were called homozygous reference in all 97 samples. At sites called variant in at least one sample, our overall data completeness was 99%. Most of these samples (93) were also genotyped on Illumina 2.5M arrays. These arrays provide over 140 million genotypes that can be compared with the sequence-called genotype. Over these 140 million genotype calls, PECaller data were 99.85% complete and agreed with array call 99.76% of the time. Partitioning these numbers by array-called genotype, we note that, if the genotyping array called “homozygote reference,” the sequencing call was 99.95% complete and agreed 99.94% of the time. If the array called a “heterozygote,” the sequencing was 99.81% complete and agreed 99.23% of the time. Finally, if the array called a “homozygote nonreference,” the sequencing was 99.88% complete and agreed with the array 99.56% of the time.

Lack of agreement between sequencing- and array-based calls can be caused by errors in either the array or the sequencing call. One can show that, if the arrays are 99.8% accurate, regardless of true genotype, the agreement level above is consistent with sequencing being 99.9% accurate overall (i.e., if arrays are only 99.8% accurate, most of the disagreements between array and sequencing are caused by array errors).
GATK Exonic Variation. GATK finds the value of theta in the exomes of these samples to be between 0.00041 and 0.00043, averaging 0.000423 in PASS variants. Using both PASS and Tranche99.9, theta in exomes averages 0.000458. The Ts/Tv ratio in exons averages 3.086 in PASS variants and 1.88 in Tranche99.9 variants. The silent to replacement site ratio averages 1.131 in PASS sites and 0.613 in Tranche99.9 sites, again suggesting that Tranche99.9 variants are not high quality. The individual samples averaged ~19,000 exonic variants identified by GATK PASS.

GATK Vs. PEMapper/PECaller. To a great extent, PEMapper/PECaller and GATK generally make the same genotype calls at variant sites in the same samples. This level of overlap is a remarkable achievement for PEMapper/PECaller given the impressive accuracy and extensive use of training set data for GATK (20, 21). Over all 97 samples, PEMapper called 6,588,872 SNPs (SNPs with exactly two alleles) (Fig. 1), with an overall Ts/Tv ratio of 2.07–1. In category PASS, there are 6,338,222 SNPs, with a Ts/Tv ratio of 2.09–1; of these SNPs, 6,241,660 (98.4%) were also called by PECaller. In Tranche99.9, there were 424,564 SNPs, with a Ts/Tv ratio of 1.25–1. Of those SNPs, “only” 145,573 variants were called in common with PECaller, and those SNPs had a much better Ts/Tv ratio than Tranche99.9 overall (1.72–1). The PASS GATK calls not made by PECaller (96,562) had a Ts/Tv ratio of 1.25–1. The Tranche99.9 GATK calls not made by PECaller had a Ts/Tv (266,521) ratio of 1.06–1. Finally, PECaller SNPs not called by GATK (197,660) had a Ts/Tv ratio of 1.31–1 (Fig. 2 and Table 2). Overall, these results mean that PEMapper/PECaller calls slightly more variants than GATK PASS and slightly fewer than GATK total (PASS + Tranche99.9). SNPs called by GATK but not called by PEMapper/PECaller look to be of worse quality than SNPs called by PEMapper/PECaller but not called by GATK. The performance of Tranche99.9 SNPs in all ways suggests that they should probably not be used for analysis, because they are likely to have significant numbers of false positives.

Using the Illumina 2.5M Array as the gold standard, we were able to compare the completeness and accuracy of both PEMapper/PECaller and the GATK pipeline. Across the
board, PEMapper/PECaller outperformed GATK, albeit only slightly (Table 3). If the array called homozygous reference, PEMapper/PECaller was 99.95% complete and 99.94% agreement with the array compared with GATK, with 98.98% complete and 99.83% agreement. If the array called heterozygous, PEMapper/PECaller was 99.81% complete and 99.23% agreement with the array compared with GATK, with 99.31% complete and 99.78% agreement. If the array called homozygous nonreference, PEMapper/PECaller was 99.88% complete and 99.56% agreement with the array compared with GATK, with 99.68% complete and 99.15% agreement. Overall, PEMapper/PECaller was 99.85% complete and 99.76% agreement with the array compared with GATK, with 99.82% complete and 99.74% agreement with array.

Essentially, both callers are primarily “limited” by microarray-based errors. This limitation means that it may be that both callers are nearly always getting the right answer when the array is correct and that, when the array is in error, they differ in differing ways. To a first approximation, the difference between the two can be summarized, because GATK is slightly more likely than PEMapper to fail to report a site called variant by the array. The sites that GATK excludes but PEMapper calls are slightly more likely than average to disagree between PEMapper and the array. There is certainly no evidence that GATK is doing a substantially better job than PEMapper. We also point out that all of this is despite the fact that GATK is using knowledge about the position of high-frequency variants to help align sequences and set thresholds for calling. PEMapper/PECaller uses none of this information, and it is mapping and calling variants “naively” and yet, achieves the same overall results.

In a slightly different comparison experiment, we know that, with the Illumina arrays, GATK, and PECaller, we have three separate sets of calls. Dropping any call that is missing in the array, GATK, or PECaller, there are ~140 million genotypes called in common between the arrays and either GATK or PECaller and over 633 million variant calls that can be compared between GATK and PECaller. For each of three, we can assume that one of three is the “gold standard” for accuracy and ask what the error rate is at variant sites relative to this gold standard. These results are shown in Table 4. Several conclusions can be drawn. First, all three are excellent and in close agreement. Second, GATK looks to be a slight outlier. If GATK is set as the gold standard, both the array and PE Caller seem to have approximately a 1% error rate at heterozygous sites and very low error rates at homozygous sites. Conversely, comparing GATK with the array gold standard, heterozygotes seem to have an excellent error rate, but homozygous nonreference calls have an abnormally high error rate. The simplest explanation of both of these observations is that GATK is slightly “overcalling” heterozygotes at the expense of homozygous calls but only very slightly, because overall calling is truly excellent.

### Table 2. Comparison of the number of variants called and the Ts/Tv ratio for those variants between PEMapper/PECaller and GATK

<table>
<thead>
<tr>
<th>Category</th>
<th>No. of variants called</th>
<th>Ts/Tv ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEMapper/PECaller</td>
<td>6,588,872</td>
<td>2.07:1</td>
</tr>
<tr>
<td>GATK PASS</td>
<td>6,338,222</td>
<td>2.09:1</td>
</tr>
<tr>
<td>GATK 99.9</td>
<td>424,564</td>
<td>1.21:1</td>
</tr>
<tr>
<td>PEMapper/PECaller and GATK 99.9</td>
<td>145,373</td>
<td>1.72:1</td>
</tr>
<tr>
<td>GATK PASS but not PEMapper/PECaller</td>
<td>96,562</td>
<td>1.25:1</td>
</tr>
<tr>
<td>GATK 99.9 but not PEMapper/PECaller</td>
<td>266,521</td>
<td>1.06:1</td>
</tr>
<tr>
<td>PEMapper/PECaller but not GATK</td>
<td>197,660</td>
<td>1.31:1</td>
</tr>
</tbody>
</table>

Variants not called by PEMapper/PECaller (but called by GATK) are of worse quality than those not called by GATK (but called by PEMapper/PECaller).

### Insertion and Deletion Comparisons

Calling of small insertions and deletions remains a very difficult category of variation to evaluate, largely because, unlike SNPs, there is no obvious external gold standard (such as genotyping arrays) with which to compare. Various investigators have attempted to create artificial gold standards [e.g., Chong et al. (22) use sets of “artificial” indels created by the investigators at known locations], but given that GATK calling is highly dependent on using known training set data to improve its accuracy, it is not clear whether such testing paints an appropriate picture of GATK accuracy. It is certain that GATK does a very poor job (less than 20% accurately detected) at determining the exact position and presence of small indels when those indels have not previously been seen in its training set, but given that GATK uses such extensive training sets, it is certainly better than 20% accuracy overall. What is absolutely clear is that GATK and PEMapper/PECaller often call the same small indels. When they do, they usually agree on genotype (homozygous reference vs. heterozygous indel vs. homozygous indel) for individual samples. When the two calling algorithms disagree concerning the presence or absence of an indel, it is very difficult to interpret. Precise calling of copy number variants (CNVs) requires specialized software and is not part of either the PEMapper/PECaller or GATK package (22, 23).

PECaller called 406,015 small deletions, of which 84% (342,094) were called in exactly the same position by GATK. PECaller also called 212,272 insertions, of which 84% (178,478) were called by GATK. Therefore, a large percentage of the indels called by PEMapper/PECaller overlap perfectly with GATK. In the other direction, GATK called many more indels than PECaller. A total of 37% of deletions called by GATK and 57% of GATK-called insertions were not called by PECaller. This difference was not primarily because of the fact that the Smith–Waterman mapping parameters in PEMapper were set to drop any read with a large (larger than ~10 bp) indel, because even for the smallest of indels, there was often considerable disagreement (30% of the one-base deletions called by GATK were not called by PEMapper, and 55% of the one-base insertions were not called). It should be noted that this comparison required the indel to be called in exactly the same position (i.e., not even one base different from one another). In even slightly repetitive sequence, precise indel position is often unknowable, and it is hardly surprising that indels called by one algorithm are sometimes given slightly different positions by another; even allowing for such, there is a considerable number of small indels called by each algorithm not called by the other, with vastly more indels called by GATK than called by PEMapper/PECaller.

However, when both algorithms called an indel at the same position, there is great agreement between them in individual genotype calls. Among individuals called homozygous reference by PECaller at sites containing a deletion, GATK called those same individuals homozygous reference 99% of the time; individuals called heterozygous deletions by PE Caller 99% of the time were also called heterozygous deletions by GATK. Individuals called

### Table 3. Comparison of calling completeness and accuracy compared with the Illumina 2.5M array gold standard for PEMapper/PECaller and GATK

<table>
<thead>
<tr>
<th>Variant call type</th>
<th>PEMapper/PECaller, %</th>
<th>GATK, %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Completeness</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Ref/ref</td>
<td>99.95</td>
<td>99.94</td>
</tr>
<tr>
<td>Ref/alt</td>
<td>99.81</td>
<td>99.23</td>
</tr>
<tr>
<td>Alt/alt</td>
<td>99.88</td>
<td>99.56</td>
</tr>
<tr>
<td>Overall</td>
<td>99.85</td>
<td>99.76</td>
</tr>
</tbody>
</table>

PEMapper/PECaller performs slightly better than GATK. Alt/alt, two alternate alleles; Ref/alt, one reference and one alternate allele; Ref/ref, two reference alleles.
Table 4. Comparison of error rates using three possible gold standards (Illumina array with 140 million calls, 630 million PECaller calls, and 630 million GATK calls)

<table>
<thead>
<tr>
<th>Variant type</th>
<th>PECaller as gold standard</th>
<th>GATK as gold standard</th>
<th>PECaller as gold standard</th>
<th>GATK as gold standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref/Ref</td>
<td>0.00061</td>
<td>0.00174</td>
<td>0.00224</td>
<td>0.00157</td>
</tr>
<tr>
<td>Ref/Alt</td>
<td>0.00766</td>
<td>0.00217</td>
<td>0.00351</td>
<td>0.00712</td>
</tr>
<tr>
<td>Alt/Alt</td>
<td>0.00439</td>
<td>0.00849</td>
<td>0.00123</td>
<td>0.00739</td>
</tr>
<tr>
<td>All</td>
<td>0.00235</td>
<td>0.00261</td>
<td>0.00235</td>
<td>0.00300</td>
</tr>
</tbody>
</table>

When Illumina array calls are the gold standard, PECaller has much less error in homozygous reference and homozygous alternate calls, while having more in heterozygous calls. Overall, PECaller has slightly less error. Using all three, it is possible to discern that GATK is overcalling heterozygotes at the expense of homozygous calls. Alt/Alt, two alternate alleles; Ref/Alt, one reference and one alternate allele; Ref/Ref, two reference alleles.

Discussion

The future of genomics is WGS on greater than thousands of genomes. Analysing that many genomes at once, both efficiently and accurately, is a tremendous computational challenge. The GATK best practices pipeline is the de facto standard for analysis of sequencing data because it does an excellent job and has proven its utility in vast numbers of exome studies. Although a user may be well-advised to continue using the GATK pipeline for exome analysis or small numbers of whole genomes (24), we show here that PE Caller/PECaller is the decidedly better option for large-scale mapping and calling of genomes (25). PE Caller/PECaller is significantly more efficient than GATK, requiring fewer computational resources and less storage space and thus, costing less (5). PE Caller/PECaller manages to do this while providing nearly identical (or better) calling quality than GATK. PE Caller/PECaller also does not rely on any more outside information than a reference genome, making it applicable to both human and nonhuman sequencing studies.

PE Mapper/PECaller completely overcomes the technical challenges of GATK best practices. It runs well in a shared disk environment. Batch calling can occur in batches of hundreds to thousands of whole genomes easily [although computation time scales as $\log(N)$ of batch size]. All sites are output together with a confidence score, so that the missing vs. homozygous reference distinction is always maintained trivially. This distinction is important, because it allows straightforward implementation of genome-wide association study (GWAS) style QC procedures—e.g., sites can be filtered on call rate and Hardy–Weinberg. The most natural way to handle these data is simply to convert them to PLINK format, QC, and analyze them like any other GWAS, except that these data just happen to include all of the rare and common variants in the set.

Overall, GATK best practices and PE Caller/PECaller make identical calls at almost every site. When they differ from one another, there is evidence that neither is very reliable. GATK best practices achieves its excellent results in large part by incorporating preexisting knowledge into the pipeline. Reads are realigned based on preexisting knowledge of SNPs and indels. Variants are classified, filtered, or dropped based on extensive training sets of known human variants. PE Caller/PE Caller achieves essentially the same result based on no specific prior knowledge but an intelligent genotyping model that uses nothing more than the observed data at hand. In principle, the PE Caller variants could be similarly filtered/truncated/etc., but we show there is no obvious need. By not using any preexisting knowledge, PE Caller/PE Caller is far easier to use in nonhuman systems.

PE Caller/PE Caller is not only much simpler to use than GATK best practices, but also, it produces data that are of the same or very slightly higher quality. It is clear that either calling platform is more than adequate to support modern genetic studies (26), but PE Caller/PE Caller is far easier to run, uses less computational time and storage, and behaves far better in a shared disk environment. These benefits will enable researchers to analyze large numbers of whole-genome sequences both faster and more efficiently. Using PE Mapper/PE Caller to map and call large-scale genome sequencing will also further precision medicine efforts (27). Large studies using whole-genome sequences are now much easier to complete computationally using PE Mapper/PE Caller by reducing the currently most challenging bottleneck from experiments of this type.
Methods

The PEMapper/PECaller assumes that a reference target sequence is available, but no other information is needed. All mapping and genotype calling occur relative to this reference sequence. The PEMapper pipeline is composed of a series of loosely interconnected programs. The first of three prepares a hashed index of the target sequence. The remaining two programs form a pipeline, with the output of PEMapper forming the input of PECaller. PEMapper is computationally intensive but extremely gentle on disk and network subsystem. To make this possible, the underlying philosophy behind the PEMapper is that memory use should be killed for speed and limited I/O. As a result, PEMapper uses ~45 bytes of memory per base in the reference sequence plus approximately 1 GB of memory per computational core. Therefore, a whole-human genome sequence on a 64-core workstation typically uses ~200 GB RAM. The source code is freely available at https://github.com/willribolag/pecaller.

The first of three programs in the PEMapper/PECaller is called index_target. Following BLAT (28), Maq, and several other published algorithms, the target region is decomposed into 16-nt reads. The positions of all overlapping 16-mers in the target are stored. This program needs to be run only once for each target region examined. Unlike GATK best practices, no additional information on “known SNPs,” “indels,” or training sets is required or used.

The next stage called PEMapper, which also builds on approaches similar to BWA, contains a small innovation to help enable indel mapping. Reminiscent of several other algorithms, the 16-mers are allowed to have up to one sequence mismatch from the target. Thus, when mapping a 100-base read with a 16-base index, the maximum number of mismatches allowed as long as these errors are evenly distributed along the read. However, the algorithm also allows the 16-mers some “wobbly” room, so that, relative to each other, they can map a few bases away from their expected location (up to eight bases for a 16-mer). Finally, only one-half of the 16-mers need to map in the correct order, orientation, and distance apart from one another. Positions that satisfy these requirements are taken as “potential mapping” positions.

PEMapper takes this list of putative mapping locations for each read and performs a Smith–Waterman alignment in each potential location to determine the optimal position and alignment score. At this stage, reads are rejected if the final Smith–Waterman alignment score is less than a user-defined percentage of the maximum score possible for the given read length (29). For results described below, we required 90% of the maximum alignment score and used the following alignment penalties: match = −1, mismatch = −1.5, gap open = −2, and gap extend = −0.56. The primary output of PEMapper is the “pileup” statistic for each base in the target. PEMapper pileup output files include the number of reads where an A, C, G, or T nucleotide was seen together with the number of times that base appeared deleted or the expected proportion of reads coming from channels different from the channel associated with the homozygote allele (i.e., the expected proportion of error reads) is set at 1/4, or 0.25, whichever is larger for each channel; thus, at initialization, we assume between 0.25 and 0.5% error reads in every channel. The remainder of the reads is expected to come from the “correct” channel. For heterozygote genotype calls, the error channels are set similarly, except for the “deletion” channel, which is expected to have 5% of the reads, indicating a prior assumption that ~5% of true heterozygote reads will map incorrectly as deletions. If the heterozygote genotype does not involve the reference allele, the remaining reads are expected to come equally to both of the appropriate channels. However, if the heterozygote includes the reference allele, we assume that 52% of the remaining reads map to the reference allele and that 48% map to the non-reference allele. This assumption incorporates our notion that, some portion of the reads is not map or worse are indel reads.

To meet the challenge of mapping indel variation, we made the following assumptions: for deletion homozygotes, we again assume a 0.3–1% read proportion in all of the channels that do not involve the reference allele or the deletion; however, we expect the remaining reads to divide 80% deletion and 20% reference, indicating our assumption that a substantial fraction of deletion reads mismap as reference, even when the deletion is homozygous. When the deletion is heterozygous, we assume the nonerror channels to divide 60% and 40% between the reference channel and the deletion channel, respectively. Insertions after the current base are again assumed to have 0.3–1% reads in the error channels. For homozygotes, 80% of the remaining reads are expected to include the reference base and have an insertion afterward, whereas 20% of the reads will only include the reference allele. For heterozygous insertions, 40% of the remaining reads are expected to include both the reference allele and an insertion, and 60% are expected to include only the reference allele.

Bayesian Genotype Calling with a Population Genetics Prior

We assume that m samples have been sequenced. Each of those m samples can be any of 21 possible genotypes. Thus, there are a total of (21)m possible genotype configurations of those m samples. Let $\mathbb{Q}_m$ be such a configuration; $\mathbb{Q}_m$ is an m-dimensional vector, where element $\mathbb{Q}_{mj}$ is an integer between 1 and 21, and indicates the genotype of sample $i$. Genotypes of all of the samples are assumed to be independent, and therefore, the likelihood of configuration $\mathbb{Q}_m$ is

$$L(\mathbb{Q}_m | n_1, n_2, \ldots, n_m) = \prod_{i=1}^{m} \text{PE}(n_i | \mathbb{Q}_{mi})$$

where $\theta$ is a user-supplied parameter corresponding to 4$\mu_N$, $N_j$ is the effective size of the population from which the samples were drawn, and $\mu$ is the per site and per generation mutation rate (13). For humans, it is generally assumed to be ~0.001 (14). All other configurations have at least one sample with at

[Note: The rest of the text is not fully transcribed due to the limitations of the text extraction process.]
least one allele different from the reference allele. Let \( n_{a,r} \) be the number of nonreference alleles of type \( r \), \( 0 < r < 6 \), found in configuration \( \mathbf{c}_r \). The prior probability of configuration \( \mathbf{c}_r \) is assumed to be

\[
\text{Prior}(\mathbf{c}_r) = \left(1 - \text{Prior}(\mathbf{c}_r')\right) \frac{\text{HW}(\mathbf{c}_r) \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j)}{\text{HW}(\mathbf{c}_j')}}{\text{HW}(\mathbf{c}_r') \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j')}{\text{HW}(\mathbf{c}_j)}}
\]

where \( \text{HW}(\mathbf{c}_r) \) is the Hardy–Weinberg exact \( P \) value \( 33 \) associated with configuration \( \mathbf{c}_r \) and the sum in the denominator is taken over all \( 21! - 1 \) genotype configurations (Methods, Computational Efficiencies).

Overall, this prior can be summarized as follows. The population from which these samples are drawn is assumed to be of constant size and neutral, and the reference allele is assumed to be the ancestral allele at every site. The prior probability that a site is segregating is the one derived by Watterson \( 13 \) for an infinite site neutral model. Conditional on the site segregating, the assumption is that the site is in Hardy–Weinberg equilibrium, and the derived allele frequency was drawn from an infinite site neutral model. Thus, the prior probability is a combination of two terms, one of which derives from the Hardy–Weinberg \( P \) value and the other that derives from the number of different alleles seen to be segregating. Finally, we should note that we have tacitly assumed that all of the sequenced samples are randomly drawn from the underlying population (i.e., not intentionally picked to be relatives of one another). Alternatively, the user may provide a standard linkage phase (PLINK pedigree format) \( 34 \) to specify the relationship between samples. When this option is invoked, Hardy–Weinberg is calculated only among unrelated individuals (i.e., founders), and for every configuration, \( \mathbf{c}_r \), the minimum number of de novo mutations, \( D_n(\mathbf{c}_r) \), is calculated for the configuration. Each de novo mutation is assumed to occur with user specified probability \( \mu \), and the prior probability of the configuration is modified to

\[
\text{Prior}(\mathbf{c}_r) = \left(1 - \text{Prior}(\mathbf{c}_r')\right) \frac{\text{HW}(\mathbf{c}_r) \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j)}{\text{HW}(\mathbf{c}_j')}}{\text{HW}(\mathbf{c}_r') \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j')}{\text{HW}(\mathbf{c}_j)}} \times \left(1 - \mu^m\right)
\]

The posterior probability of configuration \( \mathbf{c}_r \) is

\[
\text{Post}(\mathbf{c}_r | n_1, n_2, \ldots, n_m) = \frac{\text{Prior}(\mathbf{c}_r | L(\mathbf{c}_r, n_1, n_2, \ldots, n_m)) \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j)}{\text{HW}(\mathbf{c}_j')}}{\sum_{\mathbf{c}_r'} \text{Prior}(\mathbf{c}_r' | L(\mathbf{c}_r', n_1, n_2, \ldots, n_m)) \prod_{j=1}^{k} \frac{\text{HW}(\mathbf{c}_j)}{\text{HW}(\mathbf{c}_j')}}}
\]

where the sum is taken over all possible genotype configurations (but see below). If \( 0 < g_i < 22 \) is the genotype of individual \( i \), then

\[
\text{Pr}(g_i = j) = \sum_{j} \left(\text{IC}_i = j\right) \text{Post}(\mathbf{c}_r | n_1, n_2, \ldots, n_m),
\]

where \( \text{IC}_i = j \) is an indicator function that equals one whenever element \( i \) of configuration \( \mathbf{c}_r \) is equal to \( j \) and zero otherwise. Thus, we take the probability that the genotype of individual \( i \) is \( j \) to be the sum of the posterior probabilities of the genotype configurations in which we call sample \( i \) genotype \( j \). The PE Caller finds samples \( i \) genotype \( j \) whenever \( \text{Pr}(g_i = j) \) is greater than some user-defined threshold, and otherwise, the genotype is called "N" for undetermined. For all of the results presented here, the threshold was set at 0.95.

**Estimating PE Parameters and Repeating.** Because of local sequence context, the repetitive nature of many organisms’ sequence, and specific issues with sequencing chemistry as a function of base composition, not all bases have the same error characteristics. Some bases may appear to have a very high fraction of reads containing errors, whereas other bases have almost none. Some heterozygotes may exhibit nearly 50:50 ratios of the two alleles; others can be highly asymmetrical. To account for this fact, we wish to estimate the PE parameters independently at every base. There are three technical challenges. First and most importantly, the genotypes of the samples are not known with certainty; hence, we do not know with certainty which observations are associated with which underlying PE distribution. Second, for technical reasons (one lane “worked better” than another, etc.), some samples may have many more reads than other samples, and we do not want these high-read samples to dominate our estimates disproportionately. Third, because it is necessary to estimate parameters repeatedly, the algorithm must be computationally efficient. With this requirement in mind, we chose moment-based estimators of our parameters \( 35 \).

In principle, we would like to estimate the PE coefficients for genotype \( j, \alpha_j \), by averaging over the observed number of reads seen in every sample that has genotype \( j \); however, we do not know this with certainty. Therefore, let \( L \) be a 6D vector, where element \( L_{i,k} = n_{i,j}/n \) contains the fraction of individual \( i \)’s reads that were observed in channel \( k \). Let

\[
W_i = \sum_{j=1}^{m} \text{Pr}(g_i = j),
\]

\[
M_{i,j} = \frac{\sum_{j=1}^{m} \text{Pr}(g_i = j)}{W_i},
\]

and

\[
V_{i,j} = \frac{\sum_{j=1}^{m} L_{i,j}^2 \text{Pr}(g_i = j)}{W_i} - M_{i,j}^2.
\]

Thus, \( M_{i,j} \) and \( V_{i,j} \) are the “weighted” mean and variance, respectively, in read fraction from channel \( k \) among individuals with genotype \( j \), where both moments are weighted by our confidence that the individual truly is genotype \( j \). Usually, most genotypes will have little weight (i.e., few if any samples are called that genotype), and even when samples are called that genotype, sometimes there is little to no variation seen in read fractions (i.e., 100% of the reads come from one channel in all of the samples called that genotype). Let \( Y_j \) be the number of channels for genotype \( j \) that have nonzero observed variance in read fraction. Thus,

\[
Y_j = \sum_{k=1}^{6} \left(1 - \frac{V_{i,j}}{V_{i,j}^2} > 0\right),
\]

where \( R_{i,j} > 0 \) is an indicator that genotype \( j \) has nonzero variance in channel \( k \). For any genotype with \( W_i < 1.5 \) (i.e., less than two samples called that genotype) or \( Y_j < 2 \) (i.e., less than two channels with variance in read fraction), all PE parameters are returned to their initialization values. Otherwise, let channel \( z \) be the channel with nonzero variance \( (V_{i,j} > 0) \) but minimal mean \( (M_{i,j} < M_{i,j}) \) for all other \( k \) with nonzero variance estimate

\[
S_{i,j} = \left(\prod_{k, \text{nonzero} \times z} \left(\frac{M_{i,j}}{M_{i,j}} \left(1 - \frac{M_{i,j}}{M_{i,j}}\right) - 1\right)^{1/n}\right)
\]

and

\[
\alpha_{i,j} = \max\left(M_{i,j}, S_{i,j}, 1\right).
\]

5 can be thought of as a “leave one out” moment estimate of the “precision” of the PE distribution, and \( M_{i,j} \) is a first-moment estimate of the mean read fraction in each channel \( 35 \). Notice that all channels with a small expected read fraction are rounded up to one (see below). After the PE parameters for all of the genotype models are estimated, the process repeats, and genotypes are recalled until genotype calls no longer change or a maximum of five iterations is reached.

**Computational Efficiencies.** The sample space of configurations is impossibly large. For anything other than a trivially small number of samples, the sums over the configuration sample space cannot be done. Nevertheless, the prior distribution is remarkably “flat,” and this property can be used to great advantage. If two configurations, \( \mathbf{c}_r \) and \( \mathbf{c}_s \), differ by only a single sample’s genotype, then we know that the ratio of their prior probabilities is bounded by

\[
\frac{\text{Prior}(\mathbf{c}_r)}{\text{Prior}(\mathbf{c}_s)} > \frac{4m}{\alpha(\mathbf{c}_r, \mathbf{c}_s)}
\]

To see this idea, note that the largest difference in prior probabilities occurs when configuration \( \mathbf{c}_r \) has a single homozygote of an allele not seen in configuration \( \mathbf{c}_s \). The difference in Hardy–Weinberg \( P \) values associated with this is less than \( 1/2m \) \( 33 \), and the difference caused by the number of alleles segregating is \( \theta + 2 \). Thus, if

\[
L(\mathbf{c}_r, n_1, n_2, \ldots, n_m) \ll \frac{4m}{\alpha(\mathbf{c}_r, \mathbf{c}_s)}
\]

then

\[
\text{Post}(\mathbf{c}_r, n_1, n_2, \ldots, n_m) \ll \text{Post}(\mathbf{c}_s, n_1, n_2, \ldots, n_m).
\]

The immediate implication of this is that dropping configuration \( \mathbf{c}_r \) from the sum will have little effect on the posterior probabilities of any of the likely
configurations of the genotypes, and a simple, nearly linear time algorithm to enumerate all of the likely configurations and ignore the unlikelihoods is possible.

We build the list of likely configurations by moving through the samples one at a time. Initially, we start with a set of 21 configurations that correspond to all of the possible genotypes for sample 1. We calculate the likelihood of all 21 one-sample configurations and then, remove any configuration with likelihood less than $10^{-6}$ times the largest likelihood. Additionally, we always save the configuration associated with all samples being homozygote reference, because a priori this configuration is the most likely configuration of samples. Next, to each of the remaining configurations, we add all 21 possible genotypes for the second sample, thereby increasing the number of sample configurations by a factor of 21. However, we again immediately remove all configurations with likelihood less than $10^{-6}$ times the largest likelihood. We repeat until we have gone through all $m$ samples. In principle, each step could increase the number of likely configurations by a factor of 21, but in practice, it almost never increases the number by more than a factor of 2 (i.e., there are almost never more than two likely genotypes for one sample); most of the time, it does not increase the number of configurations at all (i.e., most of the time, there is only one likely genotype for a sample). Even when $m$ is in the hundreds, most bases have only a handful of likely configurations, and seldom is the total number of likely configurations more than a few thousand.

**PECaller takes advantage of two other computational efficiency tricks. First, Hamiltonian Monte Carlo (HMC) searches for a posterior probability density function of the numbers of heterozygous and homozygous genotypes. Those values can be calculated ahead of time and stored in lookup tables, greatly aiding that computation. Second, PE distributions contain several gamma functions, and although gamma functions can be computationally expensive to calculate, in a special case, they are cheap. If $n$ is an integer, $\Gamma(n) = (n−1)!$ factorial; and for $n = \frac{1}{2}$, $\Gamma(n) = \sqrt{\frac{\pi}{2}}$. The PE coefficient read distributions can be calculated exactly in terms of factorials, and it is easy to precalculate and store all factorial values less than, say, 10,000. It should be noted as well that all likelihood calculations occur computationally as natural logs and are raised to an exponential only when necessary for posterior probability determinations. Thus, as a practical matter, the natural log of factorials is computed and stored.**

**Final placement of reads with Smith-Waterman alignment.** PECaller can be set to disregard highly repetitive sequences. By default, during the initial placement of reads, PEMapper ignores any 16-bp that maps to over 100 different locations in the genome. Thus, to even attempt Smith–Waterman alignment, at least one-half of the 16-mers in a read must map to less than 100 places in the genome. Any read more repetitive than this is dropped. Similarly, PECaller can be given a file in bed format that constitutes the "target" region to be called. This file can be used, for example, to specify the exome only for exome studies or the nonrepeat-masked regions of the human genome for WGS studies. Because variation in repeat-masked regions is both extremely difficult to interpret and highly prone to error/mismapping, all of the results described will be for the unique portion of the genome (i.e., nonrepeat masked).

**Bisulfite Sequencing and Other User Options.** A possible application of next-generation sequencing is to determine the pattern of methylation in a given region sequenced. One way of doing this is to first treat the DNA with bisulfite, which converts Cs to Ts, unless the C has been methylated. Bisulfite treatment cannot pose unique challenges for mapping short sequence reads. The PEMapper/PECaller methods are well suited to bisulfite-treated DNA. Bisulfite-methylated DNA will be treated as if they are the same nucleotide. In principle, each step could increase the number of likely configurations by a factor of 21, but in practice, it almost never increases the number by more than a factor of 2 (i.e., there are almost never more than two likely genotypes for one sample); most of the time, it does not increase the number of configurations at all (i.e., most of the time, there is only one likely genotype for a sample). Even when $m$ is in the hundreds, most bases have only a handful of likely configurations, and seldom is the total number of likely configurations more than a few thousand.

**PECaller locally.** To do so, one gathers the pileup and indel files for each sample, which gives the numbers of C and T alleles mapping at any base. We repeat until we have gone through all $m$ samples. In principle, each step could increase the number of likely configurations by a factor of 21, but in practice, it almost never increases the number by more than a factor of 2 (i.e., there are almost never more than two likely genotypes for one sample); most of the time, it does not increase the number of configurations at all (i.e., most of the time, there is only one likely genotype for a sample). Even when $m$ is in the hundreds, most bases have only a handful of likely configurations, and seldom is the total number of likely configurations more than a few thousand.

**End User Instructions.** Running the PEMapper/PECaller pipeline is very straightforward for an end user. One begins with fastq files from WGS (the number does not matter; however, many represent the complete sequencing of the sample of interest). If the end user has opted to use the Globus Genomics pipeline on AWS, the fastq files are uploaded to the PEMapper/PECaller workflow, and the user receives three important files in return: a pileup file, a summary file, and an indel file. If the end user is running PEMapper locally, he or she must have a copy of the reference genome and load that into memory before running PECaller with the map_directory_array.pl script. In either case, the user will run PECaller locally. To do so, one gathers the pileup and indel files for each sample
to be processed in a single folder. The script call directory.pl is used to launch PEcall. That script generates an .snp file (containing all SNPs but no indels in an unsorted list) as output. Then, merge_nodes.snp.pl is run to merge the indels into the list of SNPs. This script produces a merged .snp file (containing SNPs and indels in a sorted list). This file can be converted simply to a PLINK pedigree format and represents the primary output of the pipeline. Several additional scripts permit easy quality control assessments of the data. The first script, snap_trans_counter.pl, generates a file with Vc/Vr ratio information about the samples. At this point, the web-based annotation program, SeqAnt (https://sequant.genetics.emory.edu/) (17), can be used to annotate the merged .snp file. Finally, a second script, snap_trans_silent_rep.pl, takes the output from SeqAnt and generates a file with silent/replacement information about the samples.

**GATK Methods.** The initial steps of GATK, BWA and HC, were similarly run on AWS r3.xlarge instances with 32 CPUs and 244 GB RAM for each sample. Globus Genomics (https://www.globus.org) was also contracted to facilitate the running of GATK. A GATK workflow is available through them that runs, in order, BWA v0.7.12-r1039, sambamba v0.5.4, and GATK v3.5-0-g36282e4h. The reference genome used was hg38 downloaded from the Broad Institute. This workflow leverages batch submission, such that multiple samples can be submitted for mapping simultaneously. The sequencing files (fastq format) are also converted simply to a PLINK pedigree format and represent the primary output of GATK. A GATK workflow is available through them that runs, in order, BWA v0.7.12-r1039, sambamba v0.5.4, and GATK v3.5-0-g36282e4h. The reference genome used was hg38 downloaded from the Broad Institute. This workflow leverages batch submission, such that multiple samples can be submitted for mapping simultaneously. The sequencing files (fastq format) were uploaded to AWS through Globus Genomics, and the GATK output (BAM and VCF files) was subsequently returned to the user’s local machine.

**Joint genotyping and variant recalibration were done in GATK v3.6 locally in batches of 10 samples because of the intensive computational load. The joint genotyping and variant recalibration tools were run on nodes with 64 cores and 512 GB RAM. All mapping and genotype calling were relative to same reference hg38 genome in PEGmapper/PEcaller, with SNP sets, etc. taken from the hg38 resource bundle provided by GATK. All repeat-masked regions of the genome were dropped. The Unified caller would not run on the entire 97 sample dataset, even on compute nodes with 512 GB RAM free (it always eventually reported an “out of heap space” error whether run on the whole genome or each chromosome separately). We attempted to run the unified caller on subsequently smaller batches of data: it would complete in a batch size of 10 genomes but failed at a batch size of 20. Results below are from nine batches of 10 samples each and one batch of 7 samples.**

**ACKNOWLEDGMENTS.** We thank the members of the laboratories of M.E.Z. and D.J.C for comments on the manuscript, Cheryl T. Strauss for editing, and the Emory-Georgia Research Alliance Genome Center supported in part by Public Health Service Grant UL1 RR025008 from the Clinical and Translational Science Award Program, the NIH, and the National Center for Research Resources (formerly the National Center for Research Resources’ High Performance Computing Cluster) was used for this project. This work was supported by NIH/National Institute of Mental Health Grants U54 HD082015 and U01 MH101720, which are part of the International Consortium on Brain and Behavior in 22q11.2 Deletion Syndrome, and the Simons Foundation Autism Research Initiative (M.E.Z.).