Bioinformatics, 32(15), 2016, 2243–2247
doi: 10.1093/bioinformatics/btw139
Advance Access Publication Date: 11 March 2016
Discovery Note

OXFORD

Sequence analysis

On genomic repeats and reproducibility

Can Firtina and Can Alkan*

Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on October 26, 2015; revised on January 16, 2016; accepted on March 7, 2016

Abstract

Results: Here, we present a comprehensive analysis on the reproducibility of computational characterization of genomic variants using high throughput sequencing data. We reanalyzed the same datasets twice, using the same tools with the same parameters, where we only altered the order of reads in the input (i.e. FASTQ file). Reshuffling caused the reads from repetitive regions being mapped to different locations in the second alignment, and we observed similar results when we only applied a scatter/gather approach for read mapping—without prior shuffling. Our results show that, some of the most common variation discovery algorithms do not handle the ambiguous read mappings accurately when random locations are selected. In addition, we also observed that even when the exact same alignment is used, the GATK HaplotypeCaller generates slightly different call sets, which we pinpoint to the variant filtration step. We conclude that, algorithms at each step of genomic variation discovery and characterization need to treat ambiguous mappings in a deterministic fashion to ensure full replication of results.

Availability and Implementation: Code, scripts and the generated VCF files are available at DOI:10.5281/zenodo.32611.

Contact: calkan@cs.bilkent.edu.tr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The advancements in high throughput sequencing (HTS) technologies have increased the demand on producing genome sequence data for many research questions and prompted pilot projects to test its power in clinical settings (Biesecker *et al.*, 2009). Any 'medical test' to be reliably used in the clinic has to be proven to be both accurate and reproducible. However, the fast-evolving nature of HTS technologies make it difficult to achieve full reproducibility.

We recently showed that resequencing the same DNA library with the same model HTS instrument twice and analyzing the data with the same algorithms may lead to different variation call sets (Kavak *et al.*, 2015).

Aside from the potential problems in the 'wet lab' side, there may be additional complications in the 'dry lab' analysis due to alignment errors and ambiguities due to genomic repeats. The repetitive nature of the human genome causes ambiguity in read mapping when the read length is short (Treangen and Salzberg, 2012). A 100 bp read generated by the Illumina platform may align to hundreds of genome locations with similar edit distance. The BWA-MEM (Li, 2013) mapper's approach to handle such ambiguity is randomly selecting one location and assigning the mapping quality to zero to inform the variant calling algorithms that the alignment may not be accurate.

Although many algorithms exist for HTS data analysis, only a handful of computational pipelines for read mapping and variant calling may considered a 'standard' such as those that are commonly used in large scale genome projects such as the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2015).

Recently, the Genome in a Bottle Project (Zook *et al.*, 2014) was started to set standards for accurate HTS data analysis for both research and clinical uses by addressing the differences in detection performances of different algorithms and different sequencing platforms.

In this study, we investigated whether some of the commonly used variant discovery algorithms make use of this mapping quality information, and how they react to genomic repeats. Briefly, we aligned two whole genome shotgun (WGS) datasets, one low and one high coverage genome sequenced as part of the 1000 Genomes Project (The 1000

Genomes Project Consortium, 2015) to the human reference genome (GRCh37) twice using the same parameters. In the second mapping, we shuffled the order of reads to make sure that the same random numbers are not used for the same reads. We then generated two single nucleotide variant (SNV) and indel call sets each from each genome.

We observed substantial differences in the call sets generated by all of the variant discovery tools we tested except VariationHunter/CommonLAW. However, VariationHunter explicitly requires a deterministic read mapper, therefore we removed it from further comparisons. GATK's HaplotypeCaller showed discordancies of 1.06–1.7% in SNV/indel call sets, where Freebayes showed the most concordancy (up to 99.2%). Genome STRiP showed the greatest discrepancy in structural variation calls (up to 25%). Our results raise questions about reproducibility of callsets generated with several commonly used genomic variation discovery tools.

2 Methods

2.1 Data acquisition

We downloaded both whole genome and whole exome sequencing datasets from the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2015) FTP server.

2.2 Read mapping, shuffling and BAM file processing

We used Bowtie2 (Langmead *et al.*, 2009), RazerS3 (Weese *et al.*, 2012), BWA-MEM (Li, 2013) and mrFAST (Alkan *et al.*, 2009) (Supplementary Table S2) to align the reads generated by the Illumina platform to human reference genome (GRCh37) using default options. For testing the effects of read order, we randomly shuffled the reads in the FASTQ file using an in-house program, while keeping the relative order of read pairs intact. The reason for reshuffling the reads is the following. In our small scale test, we noticed that BWA-MEM uses the same pseudorandom number generator seed in all mapping experiments. This causes the same ambiguously mapping read to be randomly assigned to the same position when the read order is kept. However, when we shuffle the reads, the random number that corresponds to the read changes, causing it to be placed to another random location. Note that, the DNA molecules are hybridized randomly to the oligos on the flow cell, thus, our read randomization simulates the randomness in cluster generation. Next, we used SAMtools (Li *et al.*, 2009) to merge, sort and index BAM files, and Picard to remove PCR duplicates (MarkDuplicates). We then followed the GATK's 'best practices' guide (Van der Auwera *et al.*, 2013) to realign around indels (RealignerTargetCreator and IndelRealigner) and recalibrate base quality values (BaseRecalibrator). We used the resulting BAM files for SNV, indel and structural variation (SV) calling. The names and version numbers of the tools we used are listed in Supplementary Table S2.

2.3 SNVs and indels

We used GATK HaplotypeCaller (DePristo *et al.*, 2011), SAMtools (Li *et al.*, 2009), Freebayes (Garrison and Marth, 2012) and Platypus (Rimmer *et al.*, 2014) to characterize SNV and indels. We followed the developers' recommendations and default parameters for all variant calling tools, including potential false positive filters. Specifically, we used both Variant Quality Score Recalibrator and SnpCluster methods to filter out false positives in GATK call sets, and for other tools we required a variant quality of at least 30. For GATK, we used the GATK Resource Bundle version 2.8 as the reference genome and its annotations, and variant score recalibration training material.

2.4 Structural variation

For structural variation discovery using the BWA-generated BAM files, we tested the reproducibility of the calls produced by DELLY (Rausch *et al.*, 2012), LUMPY (Layer *et al.*, 2014), Genome STRiP (Handsaker *et al.*, 2015) and VariationHunter/CommonLAW (Hormozdiari *et al.*, 2009, 2010, 2011). We note that VariationHunter explicitly remaps reads to the reference genome using mrFAST, which is a deterministic mapper, therefore we removed it from further comparisons. We used default parameters for each tool and followed recommendations in relative documentations.

2.5 Variant annotation and comparison

We downloaded the coordinates for segmental duplications, genes, coding exons and common repeats from the University of California Santa Cruz (UCSC) Genome Browser (Kent *et al.*, 2002). We then used the BEDtools suite (Quinlan and Hall, 2010) and standard UNIX tools to calculate the discrepancies among the call sets and their underlying sequence annotations.

2.6 Code and script availability

We released our FASTQ read shuffling tool, shell scripts to map reads and call variants and the VCF files generated for this study at the Zenodo data archival site. The DOI for this submission is 10.5281/zenodo.32611.

3 Results

Data and tools. We downloaded two WGS datasets, one at low coverage (~5X, HG00096) and one at high coverage (~44X, HG02107), and 12 whole exome shotgun sequence (WES) datasets with coverage ranging from 120X to 656X from the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2015) (Supplementary Table S1). We tested the behaviors of three different read mappers, four SNV/indel callers and three SV characterization algorithms (Supplementary Table S2, Section 2).

Small scale test for ambiguous mapping. We first sub sampled 1 million reads from HG00096, and mapped it to the human reference genome (GRCh37) using Bowtie2, RazerS3, mrFAST and BWA-MEM (Li, 2013). Next, we randomly shuffled the reads in the FASTQ file (Section 2) and remapped the reordered reads to GRCh37 using the same tools. The read order randomization simulates the random nature of DNA hybridization on the flow cell. We confirmed that mrFAST and Bowtie2 generated the same alignments, as described in their respective documentations, where BWA-MEM mapped several reads to different locations due to placing such reads to random locations (Supplementary Table S3). However, although Bowtie2 was not affected by read order, it reported different locations when the read names are changed (Heng Li, personal communication).

Read mapping in parallel. Due to the large number of reads generated by HTS platforms, it is a common practice to use scatter/gather operations (or, its implementation using the MapReduce framework) to distribute the work load to large number of CPUs in a cluster. This approach leverages the embarrassingly parallel nature of read mapping, where the FASTQ files that typically contain >50 million reads are divided into 'chunks' with just 1–2 million reads per file, the reads in each chunk are mapped separately, and the resulting BAM files are combined. Reasoning from our observation of different random placements of ambiguous reads when the reads are shuffled, we employed the scatter/gather method to map 1 million reads twice, using different chunk sizes. In this experiment, we

divided the reads into chunks of 50 000 and 100 000 read pairs, mapped them using BWA-MEM and observed mapping discordance ratios similar to that of random shuffling (2.1%, Supplementary Table S4). We also observed less pronounced differences in read mapping when different number of threads are used for the same FASTQ file (0.05%, Supplementary Table S5).

WGS analysis. We then repeated the same mapping strategy to the full versions of all datasets we downloaded, but we mapped using only BWA-MEM, since we observed the other mappers to be deterministic based on the small scale test. We also investigated BWA-MEM's behavior of random placements using the HG00096 genome, and interestingly, although BWA-MEM reported zero mapping qualities for most of the discrepant read mappings (∼97%), it also assigned high MAPQ values (≥30) for a fraction of them (∼0.75%; Supplementary Table S6).

Single nucleotide variants and indels. We used GATK's HaplotypeCaller, Freebayes, Platypus and SAMtools to characterize SNVs and indels within the HG00096 and HG02107 genomes using recommended parameters for each tool (Section 2). We did not evaluate GATK UnifiedGenotyper since it is deprecated by its developers. We then compared each call set generated by the same tools using the reads in original versus shuffled order using BEDtools (Quinlan and Hall, 2010), and found up to 1.70% of variants to be called in one alignment of the same data but not in the other (Table 1). Next, we investigated the underlying sequence context of the SNVs and indels differently detected using the same tools with two different alignments (i.e. original versus shuffled order). As expected, 72–80% of the discrepant calls were found within common repeats and segmental duplications (Supplementary Tables S7–S10). In most genomic analysis studies duplications and repeats are removed from analyses; however, in this study we observed discrepancies in functionally important regions (i.e. coding exons). For example, 253–1249 SNVs that were called from one alignment but not another map to coding exons (Supplementary Table S7). Furthermore, 1543 of the 1884 (81.9%) discordant exonic SNVs predicted by GATK HaplotypeCaller (either original or shuffled order, non-reduntant total) **did not** intersect with any common repeats or segmental duplications. Freebayes, Platypus and SAMtools predictions were more reproducible, as >98.5% of the calls were identical, and the number of exonic discrepant SNV calls were substantially lower than that of GATK's (Supplementary Table S8–S10).

Structural variation. Next, we analyzed the deletion calls predicted using DELLY, LUMPY and Genome STRiP. All three SV detection tools we tested showed 3.5–25.01% difference in call sets using the original versus shuffled order read datasets (Table 1). Similarly, the discrepancies were mostly found within repeats and duplications, however, only a couple of deletion calls intersected with coding exons (Supplementary Tables S11, S15 and S16).

Using DELLY, we predicted ∼3% of deletion, ∼4% of tandem duplication, ∼6% of inversion and ∼3.6% of translocation calls to be specific to a single alignment, and >91% of these differences intersected with common repeats. Owing to the difficulties in predicting these types of SVs, more discrepant calls intersected with functionally important regions (i.e. genes and coding exons; Supplementary Tables S12–S14).

Reusing the same alignments. More interestingly, when we ran GATK's HaplotypeCaller on the same BAM file twice we observed discrepant calls similar to using two different BAM files generated from original versus shuffled read order (Supplementary Table S17). Other tools produced no discrepancies (Supplementary Tables S18–S26). Detailed analysis of these discordancies revealed that 21 497 of the 21 510 (>99.9%) 'second-run specific' HaplotypeCaller calls were initially found in the first run, however, filtered in the variant

Table 1. Summary of SNV, small indel and deletion calls

Tool	HG00096				
	Original		Shuffled		*Diff (%)*
	All	*Private*	*All*	*Private*	
HaplotypeCaller	2 279 678	10 898	2 294 808	26 028	1.06
Freebayes	2 400 545	992	2 400 595	1042	0.08
SAMtools	2 277 691	2683	2 277 674	2666	0.24
Platypus	2 022 412	2342	2 022 294	2224	0.23
DELLY	1325	37	1323	35	5.29
LUMPY	1366	12	1363	9	1.55
Genome STRiP	1218	25	1212	25	4.04
	HG02107				
	Original		Shuffled		*Diff (%)*
	All	*Private*	*All*	*Private*	
HaplotypeCaller	4 654 338	54 051	4 625 648	25 361	1.70
Freebayes	5 174 644	4715	5 189 285	19 356	0.46
SAMtools	5 355 604	9838	5 355 053	9287	0.36
Platypus	4 642 336	6200	4 642 300	6164	0.27
DELLY	13 517	831	13 505	819	11.51
LUMPY	9786	182	9853	249	4.49
Genome STRiP	3452	482	3477	508	25.01

We list the number of SNV, small indel and deletion calls in the genomes of HG00096 and HG02107 characterized by different tools using the reads in the original (i.e. as released by 1000 Genomes Project) and shuffled order. Calls that are specific to one order of reads are listed as Private. The difference percentage is calculated as the total number of Private calls divided by the number of calls in the union set (i.e. $\frac{|(O\backslash S) \cup (S\backslash O)|}{|O \cup S|}$; O, original; S, shuffled).

[a]Deletions > 100 bp only.

quality score recalibration (VQSR) step. Similarly, 10 631 of the 10 646 'first-run specific' HaplotypeCaller calls were eliminated by VQSR in the second run. We then performed a line-by-line analysis in such calls and found that the *VQSLOD* score was calculated differently, although the training data were the same in both runs. We speculate that this is due to the random sampling of the training data to reduce computational burden (This random subsampling can be seen in the GATK code VariantDataManager.java at https://github.com/broadgsa/gatk-protected/(commit ID: 8ea4dcab8-d78e7a7d573fcdc519bd0947a875c06, line 255).). We then confirmed our observation by rerunning the VQSR filter on one of the VCF files five times. Each iteration of the VQSR filtering generated a different set of VQSLOD values, causing different variants to be filtered. However this effect seems to be diminished when multiple samples are used simultaneously.

Exome analysis. Finally, we tested the effect of discordant call sets generated by GATK even with the same alignment files using 12 WES datasets from the 1000 Genomes Project (Supplementary Table S1). We followed the same alignment, post-processing for the WES datasets. We then generated two call sets each using HaplotypeCaller on the same BAM files, followed with VQSR filtering. In this experiment, we used the multisample calling options. HaplotypeCaller produced discordant calls at 1–3% rate (Supplementary Tables S17 and S27).

4 Discussion

In this article, we documented the effects of different approaches to handle ambiguities in read mapping due to genomic repeats. We focused on more widely used computational tools for read mapping and variant calling and observed that random placement of ambiguously mapping reads have an effect on called variants. Although discordancies within repeats are less of a concern due to their relatively negligible effects to phenotype, we also discovered hundreds to thousands variants differently detected within coding exons. HaplotypeCaller showed the most discrepancies, where the discordant calls were less pronounced in Freebayes and Platypus results. Using the same alignments twice, we found that the callers themselves are deterministic, however, they return different call sets when the same data is remapped. Interestingly, we observed differences in call sets generated using HaplotypeCaller even when the same alignments and variant filtration training datasets were provided. Although we could not fully characterize the reasons of this observation with GATK, since HaplotypeCaller algorithm is yet unpublished, we observed that the differences were mainly due to differences in calculation of the *VQSLOD* score by the VQSR filter (Section 3). Therefore, a second source of randomness we observed is within the training step of the VQSR filter, which is specific to GATK.

Recommendations. We, point out that randomized algorithms may achieve better accuracy in practice, albeit without 100% reproducibility. Full reproducibility could only be achieved through using deterministic methods. Therefore, for full reproducibility, we recommend to opt for a deterministic read mapper, such as RazerS3 mrFAST, etc., and a deterministic variant caller, such as Platypus or Freebayes for SNV and indels. We note that all SV calling algorithms we surveyed in this article are deterministic algorithms; therefore, the SV call sets can be fully reproducible when they are used together with a deterministic mapper. Another approach may be more strict filtering of variants that map to repeats and duplications, however, this may result in lower detection power in functionally important duplicated genes such as the MHC and KIR loci. It may be possible to work around the GATK's VQSLOD calculation problem outlined above either by analyzing multiple samples simultaneously, or by setting the maxNumTrainingData parameter and other downsampling parameters to high values, however, we recommend disabling these randomizations by default to be a better practice for uninformed users. In our tests, changing only the maxNumTrainingData parameter did not fully resolve the variant filtration problem, which points that there may be other downsampling and/or randomization step within the VQSR filter.

Conclusion. Mapping short reads to repetitive regions accurately still remains an open problem (Treangen and Salzberg, 2012). RazerS3 and mrFAST use edit distance and paired-end span distance to deterministically assign a single 'best' map location to ambiguously mapping reads, where BWA-MEM selects a random map location all mapping properties are calculated the same. BWA-MEM assigns a zero mapping quality to such randomly selected alignments. This approach is still valid since it informs the downstream analysis tools for problematic alignments, however, as we have documented in this article, several variant discovery tools do not fully utilize this information. Complete analysis of the reasons for these discrepancies may warrant code inspection and full disclosure of every algorithmic detail.

The differences in call sets we observed in this study have similar accuracy when compared to 1000 Genomes data (Supplementary Tables S28 and S29). In addition a recent study did not find any significant difference between deterministic and non-deterministic mappers in terms of accuracy (Cornish and Guda, 2015). It is still expected to have differences between different algorithms and/or parameters but obtaining different results should not be due to the order of *independently generated* reads in the input file. We may simply count these discordancies as false positives and negatives, and such discordancies may not have any adverse effects in practice, however, we argue that computational predictions should not be affected by luck, and inaccuracies in computational results should be deterministic so they can be better understood and characterized. We are in exciting times in biological research thanks to the development of HTS technologies. However, under the shining lights of the discoveries we make in this 'big biology' revolution, it can be easy to overlook that the methods matter. No genomic variant characterization algorithm achieves 100% accuracy yet, even with simulation data, but it is only possible to analyze and understand the shortcomings of deterministic algorithms, and impossible to fully understand how an algorithm performs if it makes random choices.

Acknowledgements

The authors thank H. Ozercan, A. Gundogdu, A. Senol and Y. Ozkaya for the initial observation of the effects of reshuffling reads in alignment results using BWA-MEM. They also thank M. Somel, O. Gokcumen, E. Cicek, O. Tastan and K. Meltz-Steinberg for their valuable comments during the preparation of this article.

Funding

Funding for this project was provided by a Marie Curie Career Integration Grant [303772] and an European Molecular Biology Organization (EMBO) Installation Grant [IG-2521] to C.A.

Conflict of Interest: none declared.

References

Alkan,C. *et al*. (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet*., **41**, 1061–1067.

Biesecker,L.G. *et al*. (2009) The ClinSeq project: piloting large-scale genome sequencing for research in genomic medicine. *Genome Res*., **19**, 1665–1674.

Cornish,A. and Guda,C. (2015) A comparison of variant calling pipelines using genome in a bottle as a reference. *Biomed. Res. Int*., ID 456479.

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

Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short-read sequencing. arXiv preprint arXiv:1207.3907v2.

Handsaker,R.E. *et al*. (2015) Large multiallelic copy number variations in humans. *Nat. Genet*., **47**, 296–303.

Hormozdiari,F. *et al*. (2009) Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res*., **19**, 1270–1278.

Hormozdiari,F. *et al*. (2010) Next-generation VariationHunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics*, **26**, i350–i357.

Hormozdiari,F. *et al*. (2011) Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res*., **21**, 2203–2212.

Kavak,P. *et al*. (2015) Robustness of massively parallel sequencing platforms. *PLoS One*, **10**, e0138259.

Kent,W.J. *et al*. (2002) The human genome browser at UCSC. *Genome Res*., **12**, 996–1006.

Langmead,B. *et al*. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*., **10**, R25.

Layer,R.M. *et al*. (2014) LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol*., **15**, R84.

Li,H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*.

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

Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.

Rausch,T. *et al*. (2012) DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**, i333–i339.

Rimmer,A. *et al*. (2014) Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat. Genet*., **46**, 912–918.

The 1000 Genomes Project Consortium. (2015) A global reference for human genetic variation. *Nature*, **526**, 68–74.

Treangen,T.J. and Salzberg,S.L. (2012) Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet*., **13**, 36–46.

Van der Auwera,G.A. *et al*. (2013) From FASTQ data to high confidence variant calls: the genome analysis toolkit best practices pipeline. *Curr. Protoc. Bioinformatics*, **11**, 11.10.1–11.10.33.

Weese,D. *et al*. (2012) RazerS 3: faster, fully sensitive read mapping. *Bioinformatics*, **28**, 2592–2599.

Zook,J.M. *et al*. (2014) Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol*., **32**, 246–251.