9-24-2015

Assessment of Alignment Algorithms, Variant Discovery and Genotype Calling Strategies in Exome Sequencing Data

Anthony Corbett
amc6866@rit.edu

R·I·T

Assessment of Alignment Algorithms, Variant Discovery and Genotype Calling Strategies in Exome Sequencing Data

Anthony Corbett

Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Bioinformatics

Thomas H. Gosnell School of Life Sciences
College of Science

Rochester Institute of Technology
Rochester, NY

September 24, 2015

Abstract

Advances in next generation sequencing (NGS) technologies, in the past half decade, have enabled many novel genomic applications and have generated unprecedented amounts of new knowledge that is quickly changing how biomedical research is being conducted, as well as, how we view human diseases and diversity. As the methods, algorithms and software used to process NGS data are constantly being developed and improved, performing analysis and determining the validity of the results become complex. Moreover, as sequencing moves from being a research tool into a clinical diagnostic tool understanding the performance and limitations of bioinformatics pipelines and the results they produce becomes imperative. This thesis aims to assess the performance of nine bioinformatics pipelines for sequence read alignment, variant calling and genotyping in a Mendelian inherited disease, parent-trio exome sequencing design. A well-characterized reference variant call set from the National Institute of Standards and Technology and the Genome in a Bottle Consortium is be used for producing and comparing the analytical performance of each pipeline on the GRCh37 and GRCh38 human references.

Committee Approval

Michael V. Osier, Ph.D. Date
Associate Professor
Thomas H. Gosnell School of Life Sciences
Rochester Institute of Technology
Thesis Advisor

John M. Ashton, Ph.D. Date
Research Assistant Professor
Department of Microbiology and Immunology
School of Medicine and Dentistry
University of Rochester Medical Center
Committee Member

Steven R. Gill, Ph.D. Date
Associate Professor
Department of Microbiology and Immunology
School of Medicine and Dentistry
University of Rochester Medical Center
Committee Member

Alex R. Paciorkowski, M.D. Date
Assistant Professor
Department of Neurology, Pediatrics, and Biomedical Genetics
School of Medicine and Dentistry
University of Rochester Medical Center
Committee Member

Contents

List of Figures

List of Tables

1 Introduction

With the emergence of next generation sequencing (NGS) technologies in the recent decade, major shifts have taken place in many fields to leverage the advantages that they offer. These advantages include a reduction in cost and time to generate sequence data as well as an increase in the amount of sequence data generated; resulting in an increase in sensitivity and resolution in investigating biological systems and processes. These advances have led to a whole host of applications including whole genome sequencing; targeted sequencing of the coding regions of the genome (exome); determining genetic variation like single nucleotide polymorphisms (SNPs) and insertions and deletions (indels), as well as larger structural variations between individuals in a population. Metagenomics, or the exploration of heterogeneous populations of organisms through shotgun sequencing, allows for the identification and presence of individual microbial species and elucidation of functions that these communities provide for their host environment. In addition, NGS technology has replaced the microarray for measuring gene expression and has even allowed for the discovery of novel transcripts leading to new or revised gene models and alternate isoforms of expressed genes. Finally, new methods like Chip-Seq detect the DNA binding locations of proteins such as transcription factors. As the technology matures, even more novel applications are being developed to allow unprecedented amounts of new knowledge that is quickly changing how biomedical research is being conducted, as well as, how we view human diseases and diversity.

The higher throughput, along with the more comprehensive and unbiased manner in which to examine the genome, has generated a lot of interest in using next-generation sequencing to re-sequence many whole exomes and whole genomes on a much larger scale and in many different contexts than ever before. This has led to re-sequencing efforts in cancer genomics (Watson et al., 2013), for the discovery of de novo mutations associated with Mendelian diseases (Bamshad, Ng, et al., 2011), and for the study of human population level studies (H. Li and R. Durbin, 2011). Large scale projects like the 1000 Genomes Project (1000 Genomes Project Consortium, 2012), the Copy Number Variation Project,

and medically relevant projects like the Cancer Genome Atlas, Exome Sequencing Project (https://esp.gs.washington.edu/), the Centers for Mendelian Genomics (Bamshad, Shendure, et al., 2012), and the Exome Aggregation Consortium (*Exome Aggregation Consortium (ExAC)* 2015) have all generated and assembled extraordinary amount of genomic and genetic variation data. However, as high-throughput sequencing begins to inform clinical decisions, it becomes critical to assess the accuracy of variant calls and understand biases and sources of error in sequencing and data processing methods.

Deciphering the genetic components that affect human phenotypes, including simple single nucleotide variants (SNVs) and more complex variants, such as multiple nucleotides variants (MNV), insertions, deletions, copy number variants, and large structural changes (inversions) require accurate methods for read alignment and for discovering genetic variation. While major progress in algorithm development has been made and many tools have been developed and are continually being improved, including SAMtools (H. Li, 2011), the Genome Analysis Toolkit (DePristo et al., 2011), and Platypus (Rimmer et al., 2014), it has remained a challenge to process these data into high-quality variant calls, especially for larger insertion/deletion and other large structural variants (Fang et al., 2014; Ghoneim et al., 2014; MacArthur et al., 2012; Meynert et al., 2014; Narzisi and Schatz, 2015; J. O'Rawe et al., 2013; Rieber et al., 2013; Ross et al., 2013). With every advancement in algorithmic processing of these data, genotype accuracy evaluations must be performed. Moreover, the human reference continues to be improved and methods must be re-evaluated given new and corrected sequence information; with the release of the latest human reference (GRCh38, December 2013) this is a major transitioning time for most existing software tools and pipelines.

Accurate read alignments are fundamental in the variant calling processing (Ruffalo et al., 2011) and moreover, the combination of aligners and variant callers are essential in both sensitivity and specificity of the variant calling process (Cheng et al., 2014). For the use of next generation sequencing in a clinical diagnostic test setting the College of Ameri-

can Pathologists (CAP) developed 18 laboratory accreditation checklist requirements (Aziz et al., 2015) for both analytical wet bench processes (library creation and sample prep, and sequencing) as well as for bioinformatics analysis (alignment, variant calling, annotation, prioritization), and the final patient report. Developing a comprehensive diagnostic pipeline made of multiple bioinformatics steps requires the integration of many algorithms and software. Determining which algorithms and software components that should make up a pipeline has to be tested empirically because each has their own strengths and weaknesses in diagnostic performance in different applications. In addition, providing a benchmark performance assessment against which the pipeline can be measured for continual optimization during test development, validation, and re-validation after upgrade or changes are challenging and must also be done empirically for many different types of diagnostic applications. These performance characteristics, such as analytical sensitivity and specificity, as well as the pipelines reproducibility and repeatability, need to be established and continually monitored for deviations in quality management programs. The CAP standards for exome sequencing suggest using a well characterized reference sample to test analytic validity as well as developing metrics that define pass/fail criteria for determining high-quality and optimal performing exome sequencing; such as average coverage and percent bases that meet a set of minimum coverage threshold.

This thesis project will set out to compare and assess the effects of different algorithms and tools used at all steps of the variant discovery process, specifically in exome sequencing of a parent-offspring trio study design used to study Mendelian disorders. This includes the effects of different read mapping algorithms their trade offs in the downstream effects in variant discovery and/or quality. In addition, the multiple variant calling algorithms will be used to test the effect of different variant callers ability to correctly assign genotypes to each sample in combination with each read mapper. Quality metrics will be generated and compared to assess possible performance differences between the different pipeline combinations. Finally, a set of well characterized variant calls from a reference sample, NA12878, will be used to

compute the analytical performance of each pipeline.

2 Background

The typical next generation sequencing technology achieves its high-throughput by massive parallel sequencing of DNA fragments. The results are on the order of millions to billions of short sequences, called reads. Since the first NGS instruments became available, the sequencing output has doubled every few months, greatly outpacing Moores law and Kryders law for computing performance and hard drive storage capacity, respectively (Kahn, 2011). For example, in 2007 Illuminas Genome Analyzer produced, on average, 40 million reads with an average read length of 36 base pairs(bp) totaling more than 1 Gbases of output. By 2014, Illuminas HiSeq 2500 is capable of producing 2 billion single end reads or 4 billion pairs of reads with an average read length of 125 bp totaling a maximum output of 400 Gbases, on a single flowcell, using the v4 HiSeq SBS kit, 125x coverage of the human genome. Considering such increases in the amount of sequence data to analyze, as well as sequencing bias and error rates of NGS instruments, the analysis of high-throughput NGS data is very complex and time consuming.

Downstream applications that make use of next generation sequencing data usually begin in one of two ways. First, is a de novo approach in which the reads are assembled to produce larger contiguous sequences. This approach is typically used to assemble small or novel genomes or to find novel genes being expressed as mRNA transcripts. The second approach makes use of a reference sequence, usually the genome from the same, or highly related, organism from which the sequence data was produced. The goal is to determine the corresponding location in the reference to which each read belongs. This process is called read mapping and involves alignment of each read and the reference sequence to determine a homologous location or locations. The process of mapping millions of reads has proven to be a computationally complex and intensive process. Many algorithms used to align capillary

reads, like BLAT (Kent, 2002), become overwhelmed by the amount of data to be processed. To cope with the large amount of short reads new algorithms and tools have been developed.

Most alignment algorithms for NGS data create a secondary data structure, called an index, in order to achieve the speed up needed to process the large number of reads. The index can be created for either the reference sequence or the reads and is usually based on a hash table or prefix/suffix tree type data structure.

Hash table indexing has its roots in BLAST (Altschul, Gish, et al., 1990) and its seed and extend paradigm. In this paradigm, substrings of the query, of a given length `k` (`k`-mer), are created and placed in the hash table. Then the reference sequence is scanned for exact `k`-mers, called seeds, which are looked up in the hash table. These candidate seeds are extended and joined and finally refined using the Smith-Waterman alignment algorithm (Smith and Waterman, 1981). Hash table based alignment algorithms for NGS data face two major issues when implementing the seed and extend paradigm. First, the size of `k` impacts both the sensitivity and run time performance in an inverse relationship; for millions of small read fragments that are produced by NGS technologies a smaller `k` will report a large number of candidate seeds, thus drastically increasing the run time. Each candidate will need to be extended with an expensive Smith-Waterman alignment, in which the majority of hits will turn out to be false positives. Secondly, exact `k`-mer matches are not sensitive to sequencing errors that are inherent in next generation sequencing technologies or to small nucleotide polymorphisms (SNPs), which produce differences between the reads and the reference sequence. To overcome these issues, hash table based alignment algorithms usually modify the seeding step for finding and reporting candidate seeds. One such modification to improve on exact match `k`-mers is to use multiple spaced seeds (Ma et al., 2002). Spaced seeds allow some positions in the `k`-mer to deviate from the reference sequence which increases sensitivity when the read has differences from the reference sequence. Illuminas ELAND alignment program (Cox, 2007) was the first NGS tool to include spaced seeds. A second modification to the seeding phase is to use a filter criterion to reduce the number of candidate

hits that will be passed on to the expensive extension phase. One such filter, the q-gram filter (Rasmussen et al., 2006), is based on the idea that a window of fixed length between the query and reference sequence will contain a certain number of common subsequences, T, of length q, while allowing a number of mismatches or gaps. A cutoff can be applied to the number of q-hits, thus filtering out those candidate locations that have poor matches. Two common hash table based alignment programs that utilize these techniques are SHRiMP (Rumble et al., 2009 and David et al., 2011) and Novoalign (*Novocraft* 2015).

Another hash-based aligner, Stampy (Lunter and Goodson, 2011), was created to achieve good sensitivity but still be quick. Stampy's hash table represents the location of selected 15-mers in the reference genome and employs a novel data structure, which results in improved search times and efficient use of the available memory. The algorithm first identifies candidate mapping locations for each read by searching for every overlapping 15-mer in the read, as well as, their neighbors at one mismatch removed. The candidate mapping locations are filtered for sufficient sequence similarity to the read. Then the read is aligned, to each qualifying location in the reference, using fast gapped aligner that respects quality scores and considers short indels of up to 15 bp. Finally, read pairs are realigned using a fully Bayesian probabilistic aligner that considers indels up to, by default, 30 bp. In detail, Stampy uses an approximate Bayesian model to estimate the mapping quality, which is the probability that a read [pair] is mapped incorrectly. Using a probabilistic model rather than an alignment score thresholds, Stampy's sensitivity is improved and also allows a consistent treatment of read pairs spanning large indels and structural variation. The model considers three scenarios: (1) that the correct candidate locus was not considered due to an excess of errors or variants in the read; (2) that the best-matching location is incorrect despite the correct locus having been considered, either because an exact repeat was chosen or because read errors cause a near-repeat to match better; and (3) that the original sequence is not represented in the reference. Finally, a likelihood ratio test is performed to identify cases where a read sequence is not represented in the reference. This test is performed by assessing

whether the inferred sequence similarity is sufficiently unlikely to have occurred randomly, assuming a random reference sequence.

The second category of alignment algorithms for NGS data are based on some representation of a prefix or suffix trie. The common implementations are the suffix tree, suffix array, and the fast minute (FM) index. The FM-index, based on the Burrows Wheeler (Burrows and Wheeler, 1994) transformation of the reference sequence, is the most common as it allows for optimal lookup time and optimal space; most implementations of the FM index can hold the human genome in 2-8 Gb of memory and exhibit a traversal time complexity that is linear with respect to the query length for determining exact matches. The use of a FM-index has advantages over the hash table based index because repetitive sequences collapse onto the same path in the tree, therefore, alignment of identical copies of a substring only need to be done once, whereas with the hash table each appearance of the substring must be interrogated individually. The most popular of early FM-index based alignment algorithms were Bowtie (Langmead, Trapnell, et al., 2009) and BWA (H. Li and R. Durbin, 2009). These aligners, optimized for small 36-100bp read lengths only find alignments with little, 1 or 2, to no mismatches and require the whole length of the read to be aligned globally. These constraints allow these tools to be very fast, however, as read lengths increase it is more likely that a read will cover more than 2 SNPs or sequencing errors in base calling. Moreover, these tools will not successfully align reads that contain gaps (indels) of lengths greater than 1bp or other larger structural variants or misassemblies in the reference genome. Due to these shortcomings, newer generations of these tools have been published, Bowtie 2 (Langmead and Salzberg, 2012) and BWA-SW (H. Li and R. Durbin, 2010), respectively. Bowtie 2 still uses the index to find exact matches, but instead of aligning the whole read against the index it follows the seed and extend paradigm by extracting unique seed substrings of the query and uses the efficiency of the index to lookup corresponding reference locations and then uses SIMD-accelerated dynamic programming to extend the seeds in a gapped fashion. By using this two phase approach Bowtie 2 is able to achieve an improved

sensitivity and still take advantage of the speed and memory footprint of the FM-index. BWA-SW, also understanding the inherent limitations of exact match mapping for longer read lengths, adds in a Smith-Waterman like dynamic programming step to increase mapping sensitivity while applying heuristics to speed up computing possible local alignments. To achieve this, BWA-SW creates FM-indices for both the reference sequence and each read and does dynamic programming between these two structures, thus producing seeds with mismatches and gaps. The reference is represented as a prefix trie and the read sequence as a directed acyclic word graph (DAWG), which is computed from the reference prefix trie. By creating a DAWG, smaller, repetitive sequences in the read that contain positive scoring alignments that overlap in larger alignments are ignored.

Another, more recent, aligner that combines seed and extension paradigm with a single FM-index for both the forward and reverse reference sequence, called a FMD-index, is BWAMEM (H. Li, 2013). Instead of using a hash table to produce seeds for the alignment step, it uses the FMD-index to find supermaximal exact matches (SMEM) of the read to the reference and uses those as alignment candidates. A SMEM is a maximal exact match (MEM) that cannot be extended in either direction of the match that is not contained in other MEMs at the read position (H. Li, 2012). As BWAMEM produces SMEM seeds from the FMD-index, it will greedily group those that are collinear and are close to each other into chains. By the end of seeding, the shortest chains (38bp shorter than the longest chain) or those mostly contained within larger chains (50%) are filtered out. This aims to reduce the number of possible unsuccessful seed extensions in the next step. During extension, the seeds are ranked according to the length of the seed weighted by the length of the chain to which the seed belongs. A seed is then dropped if it is already contained within an alignment found before (by a higher ranked seed for this read) or the seed is extended with a banded-affine-gap-penalty dynamic programming. During dynamic programming two heuristic are used. First, BWAMEM will stop the extension if the difference between the best extension score, computed so far, and the best possible alignment score is larger than a threshold.

This is similar to the X-dropoff heuristic in BLAST (Altschul, Madden, et al., 1997) and prevents extension through poorly aligned regions with good flanking alignment. To reduce reference bias near the end of the read alignment and to choose between local and end-to-end alignments, a second heuristic is used. During extension, if the best score reaching the end and the best local alignment score is below a threshold, the local alignment will be rejected. These unique features of BWAMEM give it good run time performance and high accuracy.

Since read mapping is at the heart of all workflows of any downstream application of next generation sequencing, it is important to understand the characteristics of different alignment programs within specific aspects such as time requirements, computational resources, and/or mapping sensitivity and accuracy that might have potential impacts on downstream analysis (i.e. variant discovery and RNA-seq or Chip-Seq). There have been a few published comparisons of different alignment programs with respect to resource and mapping metrics (H. Li and N. Homer, 2010, Ruffalo et al., 2011, Lindner and Friedel, 2012). However, as new versions of alignment programs have become available, improvements have been made in terms of both running time and accuracy.

After alignment of reads to a reference sequence the next procedure is to determine where the aligned reads give evidence of differences, or variation, from the reference and then to assign to each sample the correct genotype at each one of these loci. There are two common approaches to determining where samples differ from the reference.

The first approach is to map reads to a reference (as described above) and perform either a systematic scan of each genomic position generating the pileup of read bases that exist at the position or to identify the possible haplotypes that are well supported by the read data (Garrison and Marth, 2012). The strengths of this approach is access to a large portion of the human genome, including repeat regions that can be informed by paired-end read information, and a high sensitivity (DePristo et al., 2011; H. Li, Ruan, et al., 2008). However, there are a few weaknesses of mapping-based callers. First, mapping-based approaches can create many spurious SNV calls around indels and larger variants. Second,

there can be systematic misalignments in and near highly divergent regions that can create false evidence for SNV calls. Third, the statistical models built into the mapping-based approaches rely on the individual base quality scores that may not be well calibrated by the base calling software of the sequencing platform. While many of these weakness can be mitigated in part by realignment around indels and recalibrating the base qualities by sharing information across multiple reads mapped to the same location (Nils Homer and Nelson, 2010; McKenna et al., 2010) this is not without computation overhead and may not improve alignments around other variant types. Recent application on de novo assembly techniques show promise of more accurately detecting longer indels, GATK HaplotypeCaller, SOAPindel (S. Li et al., 2013) and Cortex (Iqbal et al., 2012), as well as micro-assembly to detect larger structural rearrangements and breakpoints, TIGRA (Chen et al., 2014) and Scalpel (Narzisi, J. A. O'Rawe, et al., 2014).

The Genome Analysis Toolkit (GATK) was developed early in the 1000 Genomes Project to provide a unified analytic framework to discover and genetic variation across different types of study designs and sequencing technologies (DePristo et al., 2011). With its ease of use, general good performance, and best practice workflow (Van der Auwera et al., 2013), it has become a very well used tool in many studies. With version 3.0, a new model of variant calling was introduced, namely the reference confidence model. This model is used to generate a genomic variant call format (gVCF) file per sample. These gVCFs contain a record of the genotype likelihoods and annotations for every single site in the genome or exome, whether or not there is evidence of variation. This allows incremental joint genotype calling on multiple samples without the computational overhead of re-running the variant calling step over again for each sample that has already been previously called.

The reference confidence model works by looking at the read alignments at each position in the genome and determining if there is either a non-reference variant, using the standard calling mechanism, and if not then the chance that some unseen non-reference allele is at this position by performing two calculations. First, it will estimate the confidence that no

SNP exists at the particular site by comparing the number of reads with the reference base versus the number of reads with any non-reference base and, secondly, it will estimate the confidence that an indel, smaller than a certain size X, could not exist at this position by estimating the change that such an indel would not have been seen confidently based on the number of reads that provide evidence against such an indel existing in the data. A genotype likelihood is produced for the position based on the least confident of these two calculations.

Along with the release of version 3.3 (October 23, 2014), it is recommended to use the HaplotypeCaller without exceptions. It is stated to be equivalent to that of the UnifiedGenotyper in its ability to call SNPs, but is superior in calling indels. The following section describes the algorithms used by the HaplotypeCaller in variant detection and producing genotype calls.

The first step is to identify regions of interest, called active regions. These regions are based on evidence of considerable variation relative to the reference, on which future steps will operate. To identify these regions the algorithm first computes an activity score for each position in the reference producing a raw activity profile over the whole genome. This per-position score is the probability that the position contains a variant as computed using the reference confidence model, described above, applied to the original alignments. Then, a smoothing algorithm is applied to the raw profile to average the raw activity profile. Next, areas of local maxima, where the smoothed activity profile rises above a threshold, are used to define intervals. These intervals, given that they are within a preset size constraint, result in the final active regions. This step reduces the total amount of the reference that must be looked at in the remaining steps by removing portions of the genome which are not likely to contain any variation beyond a level of expected background noise in the sequencing data.

The second step is to determine the possible haplotypes that exist in the sequenced individuals. This is done by re-assembly of each active region. Re-assembly begins by creating a directed DeBruijn graph based on the reference sequence of the active region as the simple starting graph. This initial graph is constructed from consecutive `k`-mers

overlapping by k-1 bases. This graph can be thought of as modeling the sequential nature of adjacent bases. Then using the reads that map to the given active region, it threads each read's k-mers through the graph by trying to match it to a path on the graph. When adding two consecutive k-mers that belong to two nodes which are already connected in the graph, the edges weight is increased by one. Where the nodes in the graph of two consecutive k-mers differ from the existing path through the graph a new edge is created between the nodes. As each read is threaded through the graph in turn, edge weights will accumulate along the paths that are best supported by the read data. Once this process is finished the graph will be pruned. Sections of the graph that are supported by few reads are considered random sequencing errors and are removed. Next, the program traverses all possible paths in the graph and builds haplotype sequences and computes their likelihood scores. The likelihood score of a unique path through the graph (a potential haplotype) is the product of the transition probabilities of the path edges; where the transition probability of an edge is calculated as the number of reads supporting the edge divided by the sum of all the support of all edges that come from the same source node. The top N (default 128) scoring plausible haplotypes are used to identify potential variant sites by performing a Smith-Waterman alignment of each haplotype to the original reference sequence yielding the final set of potential variants that will be modeled in the next step.

The third step determines how much evidence there is in the read data to support each haplotype. This is done by taking each read and aligning it against each haplotype, including the reference, using the PairHMM algorithm (Richard Durbin et al., 1998). PairHMM is a pairwise alignment algorithm that produces likelihood scores using a Hidden Markov Model. These scores express the likelihood of observing the read given the haplotype and takes base quality into consideration. This produces a matrix of likelihoods of the haplotypes on a per-read basis that is then used to compute the likelihood of individual alleles by marginalization. In short, the total likelihood of the data for the given allele is the product of the per-read haplotype likelihoods that best supports that allele (the highest likelihood). For sites where

there is sufficient evidence for at least one of the non-reference alleles being considered then a variant at that site is called.

Now that we have per allele likelihoods the only thing left is to assign the genotypes to each sample using these likelihoods by applying Bayes theorem to calculate the likelihood of each possible genotype given each samples data, and selecting the most likely and assigning it to the sample. For example, if we see G and T at a site, the possible genotypes are GG, GT, and TT. To determine the most likely of these genotypes, given the data, Bayes theorem is used as in equation 1.

$$P\left(G \mid D\right) = \frac{P(G)P(D \mid G)}{\sum_i P(G_i)P(D \mid G_i)} \tag{1}$$

The denominator is a constant across all genotypes and is equal to the probability of generating the observed read data $P(D)$. In the numerator are two terms: $P(G)$, the prior probability of genotype G, and $P(D \mid G)$, the conditional probability of the observing the read data given the genotype. By default, GATK tools use a flat prior making the probability of each genotype equally as likely. The conditional probability of the data given the genotype is calculated per equation 2.

$$P(D \mid G) = \prod_j \left(\frac{P(D_j \mid H_1)}{2} + \frac{P(D_j \mid H_2)}{2} \right) \tag{2}$$

where H_1 and H_2 are the individual haplotype alleles of genotype G and $P(D_j \mid H_1)$ and $P(D_j \mid H_2)$ are the per-read likelihood for each allele calculated in the previous step. While the equations shown above assume a diploid genome, GATKs implementation is more general and can handle any number of haplotypes (multi-allelic variants).

3 Methods

3.1 Datasets

The human exome data for the HapMap CEPH trio (NA12878, NA12891 and NA12892) were downloaded from the NCBI Sequence Read Archive (SRA) (`http://www.ncbi.nlm.nih.gov/sra`), accession numbers SRR098401, SRR098359, ERR034529 respectively. The resource b37 resource bundle was downloaded from the Broad Institute (`ftp://ftp.broadinstitute.org/bundle/2.8/b37/`). Sure Select Human All Exon v2 target capture region bed file was downloaded from Agilent SureDesign (`http://earray.chem/agilent.com/suredesign` using ELID: S0293689). The GRCh38 human reference without alternative (ALT) loci sequences was downloaded from NCBI (`ftp://ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/vertebrates_mammals/Homo_sapiens/GRCh38/seqs_for_alignment_pipelines/GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.gz`). The NIST Genome in a Bottle (GiaB) high confidence callset version 2.19 was downloaded from NCBI (`ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/NA12878_HG001/NISTv2.19/`), along with a bed file that includes regions in which it is believed that the GiaB genotype calls are highly accurate, including homozygous reference calls if no snp or indel is called. This callset is an integration of 1 Complete Genomics, 1 SOLiD WGS, 1 454 WGS, 1 Ion WGS, 1 Ion Exome, 2 Illumina exomes, and 9 Illumina WGS datasets (Zook et al., 2014). The bed file excludes regions/variant locations that are uncertain due to low coverage, genotypes called in ¡ 2 datasets, locations with unresolved discordant genotypes, locations where most datasets have evidence of bias (systematic sequencing errors, local alignment problems, mapping problems, or abnormal allele balance), variants inside possible deletions, known segmental duplications, and structural variants reported in dbVar for NA12878.

3.2 Lifting over resources to GRCh38

The b37 resource bundle files, excluding dbSNP, as well as the Sure Select target capture regions bed file and the GiaB callset were all lifted over from hg19/GRCh37 to GRCh38 using crossmap (version 0.1.8, `http://crossmap.sourceforge.net/`) and the UCSC hg19ToHg38 chain file (`http://hgdownload.soe.ucsc.edu/goldenPath/hg19/liftOver/hg19ToHg38.over.chain.gz`).

3.3 Pipelines

The workflow used in this study follows the suggested best practices from the Broad Institute in order to produce high quality alignment files and accurate variant calls. First, FastQC was run on the downloaded exome data to check for data quality or contamination. Raw reads are trimmed for low quality sequence and adapter removal prior to alignment to b37 and GRCh38 references. This preprocessing step ensures all input data is of high quality and helps reduce any possible bias in how the different aligners may handle hard clipping and handle base uncertainty. During alignment, PCR de-duplication and sorting was performed at the same time to reduce disk I/O and processing time. Next, indel realignment and base quality recalibration was performed in order to reduce misalignment around insertions and deletions and to normalize the base qualities based on empirical error rates estimated from the mapped reads.

There were three read alignment tools and three variant callers used in this analysis; the combination of which produced nine distinct alignment-variant caller pipelines. All nine pipelines were run with the same input trio exomes against b37 and the latest human reference, GRCh38 without alternative loci sequences. In all pipelines the trio were called jointly together. The tools used in the analysis were chosen by a combination of criteria. A tool had to be used in a previously published comparison and commonly cited in the literature. In addition, the tool had to be under current maintenance (bug fixes) or on-going feature development or improvements. Tools that were chosen for the alignment step

Table 1: The three aligners and three variant callers used to make up the 9 pipelines in this analysis and their respective versions.

Type	Software	Version
Aligner	bwamem	0.7.12
Aligner	novoalign	3.02.12
Aligner	stampy	1.0.25
Caller	GATK HaplotypeCaller	3.4
Caller	platypus	0.8.1
Caller	samtools	1.2

Table 2: Table of variables used in generalized commands

Variable	Values
REFERENCE	b37 or GRCh38 reference fasta file
REF	b37 or GRCh38
SAMPLE	SRR098401, SRR098359, or ERR034529
ALIGNER	bwamem, novoalign, or stampy
CALLER	gatk, platypus, or samtools
REFERENCE_INDEX	specific sequence index file for an aligner
DBSNP	b37 or GRCh38 specific dbSNP vcf file
OneKG_INDELS	b37 or GRCh38 liftover specific 1000 genomes phase 1 indels vcf file
GOLD_INDELS	b37 or GRCh38 liftover specific Mills gold standard indels vcf file

were bwamem (H. Li, 2013), novoalign (*Novocraft* 2015), and stampy (Lunter and Goodson, 2011) and the three variant callers were GATK HaplotypeCaller, platypus (Rimmer et al., 2014), and samtools (H. Li, 2011). Table 1 has the details on the versions of each tool. All commands given here follow a generalized form with variable pieces of information in <> brackets; Table 2 has the possible values for each of the variables.

3.4 Preprocessing

Raw data was downloaded from SRA (SRR098401, SRR098359, ERR034529) and converted to fastq format using the fastq-dump utility from the SRA toolkit, with the `--split-files`

option enabled. Low quality sequence and adapters were trimmed using cutadapt (version 1.8). Low quality sequence, below a threshold of 16, was trimmed from both the 5' and 3' ends of each read. For adapter trimming, if a perfect match was found within the first 9 base pairs (bp), 1 mismatch in a 10-19 bp or 2 mismatches in a 20 to 25 bp overlap the bases were discarded. Reads shorter than 50 bp were removed, along with their mate pair, from further analysis.

```
(1) # Read 1
    cutadapt --quality-cutoff 16,16 --error-rate 0.1
      --match-read-wildcards
      --adapter PE2_rc=AGATCGGAAGAGCACACGTCTGAAC
      --adapter FlowCell2_rc=TCGTATGCCGTCTTCTGCTTGAAAA
      --adapter PCRPrimer2_rc=AGATCGGAAGAGCGGTTCAGCAGGA
      --times 2 --minimum-length 50
      --paired-output <SAMPLE>_2.tmp.fastq.gz
      --output <SAMPLE>_1.tmp.fastq.gz
      <SAMPLE>_1.fastq.gz <SAMPLE>_2.fastq.gz


(2) # Read 2
    cutadapt --quality-cutoff 16,16 --error-rate 0.1
      --match-read-wildcards
      --adapter RE1_rc=AGATCGGAAGAGCGTCGTGTAGGGA
      --adapter FlowCell1_rc=GTGTAGATCTCGGTGGTCGCCGTAT
      --adapter PCRPrimer1_rc=AGATCGGAAGAGCGTCGTGTAGGGA
      --times 2 --minimum-length 50
      --paired-output <SAMPLE>_1.trimmed.fastq.gz
      --output <SAMPLE>_2.trimmed.fastq.gz
      <SAMPLE>_2.tmp.fastq.gz <SAMPLE>_1.tmp.fastq.gz
```

3.5 Read Alignment

All alignment tools used default arguments except when their respective user guide had specific recommendations for processing exome data (novoalign). If a multi-threading option was available it was used, however, since this parameters were changed based on available cluster resources and queue wait time it is omitted here. Read de-duplication, conversion from SAM to BAM format and sorting was done inline for computational efficiency, using

samblaster and samtools respectively. A generalized form of each alignment command are as follows.

3.5.1 Bwamem

```
(1) bwa mem -M -t 12 <REFERENCE> <SAMPLE>_1.trimmed.fastq <SAMPLE>_2.trimmed.fastq |
      samblaster |
      samtools view -bS - |
      samtools sort -o <SAMPLE>.bwamem.<REF>.sorted.bam
```

3.5.2 Novoalign

```
(1) novoalign -k -t15,3 --hlimit 8 --softclip 20 -i PE 250,50
      -d <REFERENCE_INDEX>
      -f <SAMPLE>_1.trimmed.fastq <SAMPLE>_2.trimmed.fastq |
      samblaster |
      samtools view -Sb - |
      samtools sort -o <SAMPLE>.novoalign.<REF>.sorted.bam
```

3.5.3 Stampy

```
(1) bwa sampe -P <REFERENCE> \
      <(bwa aln <REFERENCE> <SAMPLE>_1.trimmed.fastq) \
      <(bwa aln <REFERENCE> <SAMPLE>_2.trimmed.fastq) \
      <SAMPLE>_1.trimmed.fastq <SAMPLE>_2.trimmed.fastq |
      samtools view -Sb - > <SAMPLE>.bwa.<REF>.bam

(2) stampy -g <REFERENCE_INDEX> -h <REFERENCE_INDEX>
      --bamkeepgoodreads -M <SAMPLE>.bwa.<REF>.bam |
      samblaster |
      samtools view -bS - |
      samtools sort -o <SAMPLE>.stampy.<REF>.sorted.bam
```

3.6 Post Alignment

In order to provide the variant calling step with the highest quality input BAM files the Broad best practice recommendations were applied to the raw BAM files from each aligner. This post processing involved fixing alignment errors around insertions and deletions and to

recalibrate base qualities based on the empirical base mismatch (error) rate seen in the raw BAM file.

3.6.1 Indel Realignment

```
(1) java -jar <GATK_JAR> \
      -T RealignerTargetCreator \
      -R <REFERENCE> \
      -I <SAMPLE>.<ALIGNER>.<REF>.sorted.bam \
      -known <OneKG_INDELS> \
      -known <GOLD_INDELS> \
      -o target_intervals.<REF>.list


(2) java -jar <GATK_JAR> \
      -T IndelRealigner \
      -R <REFERENCE> \
      -I <SAMPLE>.<ALIGNER>.<REF>.sorted.bam \
      -targetIntervals target_intervals.<REF>.list \
      -known <OneKG_INDELS> \
      -known <GOLD_INDELS> \
      -o <SAMPLE>.<ALIGNER>.<REF>.realigned.bam
```

3.6.2 Base Quality Recalibration

```
(1) java -jar <GATK_JAR> \
      -T BaseRecalibrator \
      -R <REFERENCE> \
      -I <SAMPLE>.<ALIGNER>.<REF>.realigned.bam \
      -knownSites <DBSNP> \
      -knownSites <OneKG_INDELS> \
      -knownSites <GOLD_INDELS> \
      -o recal_data.table

(2) java -jar <GATK_JAR> \
      -T PrintReads \
      -R <REFERENCE> \
      -I <SAMPLE>.<ALIGNER>.<REF>.realigned.bam \
      -BQSR recal_data.table \
      -o <SAMPLE>.<ALIGNER>.<REF>.recal.bam
```

3.7 Post Alignment Calculations

To compare the alignments for each aligner the Picard's `CollectAlignmentSummaryMetrics` tool was used. For assessing the depth of coverage for the exome target capture regions the Picard's `CalculateHsMetrics` tool was used. Bedtools, in combination with cumulative sum calculations in R, was also used to further assess and plot the fraction of bases covered as depth increased.

```
(1) java -jar <PICARD_JAR> CollectAlignmentSummaryMetrics \
      REFERENCE_SEQUENCE=<REFERENCE> \
      INPUT=<SAMPLE>.<ALIGNER>.<REF>.recal.bam \
      OUTPUT=<SAMPLE>.<ALIGNER>.<REF>.recal.alignment_summary_metrics


(2) java -jar <PICARD_JAR> CalculateHsMetrics \
      BAIT_INTERVALS=SureSelect_regions.<REF>.interval_list \
      TARGET_INTERVALS=SureSelect_regions.<REF>.interval_list \
      REFERENCE_SEQUENCE=<REFERENCE> \
      INPUT=<SAMPLE>.<ALIGNER>.<REF>.recal.bam \
      OUTPUT=<SAMPLE>.<ALIGNER>.<REF>.HsMetrics.txt \
      PER_TARGET_COVERAGE=<SAMPLE>.<ALIGNER>.<REF>.HsMetrics.targets.txt


(3) bedtools coverage -hist
      -abam <SAMPLE>.<ALIGNER>.<REF>.recal.bam
      -b SureSelect_regions.<REF>.bed |
      grep ^all > <SAMPLE>.<ALIGNER>.<REF>.coverage.txt
```

```
(4) #Cumulative Sum R code and plot
    <SAMPLE>.<ALIGNER>.<REF>.cov <- read.table(<SAMPLE>.<ALIGNER>.<REF>.coverage.txt)
    <SAMPLE>.<ALIGNER>.<REF>.cov_cumul <- 1-cumsum(<SAMPLE>.<ALIGNER>.<REF>.cov[,5])

    plot(<SAMPLE>.<ALIGNER>.<REF>.cov[2:401, 2],
         <SAMPLE>.<ALIGNER>.<REF>.cov_cumul[[1]][1:400],
         type='n', ylim=c(0,1.0))

    abline(v = 20, col = "gray60")
    abline(v = 50, col = "gray60")
    abline(v = 80, col = "gray60")
    abline(v = 100, col = "gray60")
    abline(h = 0.50, col = "gray60")
    abline(h = 0.90, col = "gray60")
    axis(1, at=c(20,50,80), labels=c(20,50,80))
    axis(2, at=c(0.90), labels=c(0.90))
    axis(2, at=c(0.50), labels=c(0.50))

    points(<SAMPLE>.<ALIGNER>.<REF>.cov,
           <SAMPLE>.<ALIGNER>.<REF>.cov_cumul[1:400],
           type='l', lwd=3)
```

3.8 Variant Calling

Default arguments were used for each variant caller expect as specified. GATK HaplotypeCaller was run in `GVCF` mode on each sample independently and `GenotypeGVCFs` program was used to jointly call variants in all samples. Platypus was run with `--assemble=1` to enable reassembly using a de Bruijn graph. Samtools mpileup used default arguments, but bcftools was used to call variants using its multi-allelic calling model (`-m`), which is the recommendation over the older samtools calling model. Specifically, the commands used are as follows.

3.8.1 GATK HaplotypeCaller

```
(1) java -jar <GATK_JAR> \
      -T HaplotypeCaller \
      -R <REFERENCE> \
      -I <SAMPLE>.<ALIGNER>.<REF>.recal.bam \
      --emitRefConfidence GVCF \
      --dbsnp <DBSNP> \
      -o <SAMPLE>.<ALIGNER>.<REF>.hc_raw.g.vcf.gz

(2) java -jar <GATK_JAR> \
      -T GenotypeGVCFs \
      -R <REFERENCE> \
      --variant SRR098401.<ALIGNER>.<REF>.hc_raw.g.vcf.gz \
      --variant SRR098359.<ALIGNER>.<REF>.hc_raw.g.vcf.gz \
      --variant ERR034529.<ALIGNER>.<REF>.hc_raw.g.vcf.gz \
      --dbsnp <DBSNP> \
      -o <ALIGNER>.<REF>.recal.gatk.vcf.gz
```

3.8.2 Platypus

```
(1) platypus callVariants
      --assemble=1
      --refFile=<REFERENCE>
      --bamFiles=input_bams.list # aligner and ref specific bam files
      --output=<ALIGNER>.<REF>.recal.platypus.vcf.gz
```

3.8.3 Samtools

```
(1) samtools mpileup -go <ALIGNER>.<REF>.recal.samtools.bcf
      -f <REFERENCE> \
      SRR098401.<ALIGNER>.<REF>.recal.bam \
      SRR098359.<ALIGNER>.<REF>.recal.bam  \
      ERR034529.<ALIGNER>.<REF>.recal.bam

(2) bcftools call -vmO z \
      -o <ALIGNER>.<REF>.recal.samtools.vcf.gz \
      <ALIGNER>.<REF>.recal.samtools.bcf
```

3.9 Post Variant Calling

The GiaB callset is based only on NA12878, however, in this analysis the variants were called jointly. Therefore, the NA12878 sample (SRR098401) was selected out of original vcf file prior to comparison. This was done using GATK's `SelectVariants` tool while discarding non-variant sites for this sample (`-env`) as well as trimming unused alternative alleles (`--trimAlternatives`) and keeping the original depth and allele count vcf fields (`--keepOriginalDP` and `--keepOriginalAC`). In addition, the comparison to the GiaB callset should be restricted to only those sites that are targeted in the exome capture. Therefore, `tabix`, from the htslib library (version 1.2.1), was used to subset the NA12878 vcf file to only those variants that are within regions that were confidently called in the NIST GiaB and are within the target capture regions.

```
(1) java -jar <GATK_JAR> \
      -T SelectVariants \
      -R <REFERENCE> \
      -V <ALIGNER>.<REF>.recal.<CALLER>.vcf.gz \
      -o SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.vcf.gz \
      -sn SRR098401 -env -trimAlternates -keepOriginalAC -keepOriginalDP

(2) tabix -h -R SureSelect_Exome_NIST_callable_regions.<REF>.bed \
      SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.vcf.gz \
      > SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.callable.vcf
```

3.10 GiaB Comparison

Once each pipeline's callset has been subset to only NA12878's variants which are within the target capture regions and confidently callable in the GiaB callset the comparison can be performed. In order to accurately compare genotypes between the pipeline calls and the GiaB callset Real Time Genomics' (RTG-tools) `vcfeval` program was used (version 3.5, `http://realtimegenomics.com/products/rtg-tools/`). `Vcfeval` is capable of dealing with differences in representation that can arise in complex situations or when comparing variants produced by different callers with a high degree of accuracy and in an efficient way.

The comparison between a pipeline's callset and GiaB callset results in three subset vcf files: true positives, false positives, and false negatives; see section 3.10.3 below for definitions of these subsets. In order to continue with downstream analysis of the classified variants the three subsets were combined using GATK's `CombineVariants` tool and re-annotated using GATK's `VariantAnnotator` tool to add a VartiantType annotation to each variant, as well as INFO flag fields if the variant was present in any of the following dataset: dbSNP, HapMap, OMNI 2.5 genotypes for 1000 Genomes samples, 1000 Genomes phase1 indels, and Mills gold standard indels. Finally, to facilitate statistical and data analysis in the R programming environment, the combined classified variants were converted to a tab separated file using vfclib's `vcf2tsv` tool (`https://github.com/ekg/vcflib#vcf2tsv`). The commands used are as follows.

3.10.1 RTG-tools vcfeval

```
(1) rtg vcfeval --all-records
      -b NIST_v2.19_callable.<REF>.exome.vcf.gz
      -c SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.callable.vcf.gz
      -T 3 -t <REFERENCE_INDEX> -o eval
```

3.10.2 Post Comparison and Annotation

```
(1) java -jar <GATK_JAR>
      -T CombineVariants
      -R <REFERENCE>
      --out SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.eval.vcf.gz
      --variant:tp tp.vcf.gz
      --variant:fp fp.vcf.gz
      --variant:fn fn.vcf.gz
      -genotypeMergeOptions PRIORITIZE
      --rod_priority_list tp,fp,fn
```

```
(2) java -jar <GATK_JAR> \
      -T VariantAnnotator \
      -R <REFERENCE> \
      -I SRR098401.<ALIGNER>.<REF>.recal.bam \
      -o SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.eval.annot.vcf.gz  \
      -V SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.eval.vcf.gz  \
      -A VariantType \
      --comp:H3 <HAPMAP> \
      --comp:Omni <OMNI> \
      --comp:1000G <OneKG_SNPS> \
      --comp:Mills <GOLD_INDELS> \
      --dbsnp <DBSNP> \
      --alwaysAppendDbsnpId


(3) vcf2tsv -g SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.eval.annot.vcf.gz \
      > SRR098401.<ALIGNER>.<REF>.recal.<CALLER>.eval.annot.tsv
```

3.10.3 Statistical Calculations

Variants from each pipeline were classified by rtg-tools `vcfeval` according to the definitions given below. To better assess each pipeline's performance the variants were split into two categories, SNP and indels. For SNP calculations, any variant which was annotated `SNP`, `MULTIALLELIC_SNP`, `MNP`, or `MULTIALLELIC_MNP` by GATK's VariantAnnotator were included. For indel calculations, any variants annotated as `INSERTION` or `DELETION` were included. For each of the nine pipelines, and the two respective categories, their sensitivity, precision, and F score was calculated according to Equations 3, 4 and 5

True Positive (TP). A variant that exists in the GiaB callset and was detected by the pipeline.

False Positive (FP). A variant that does not exist in the GiaB callset and was detected by the pipeline.

True Negative (TN). A variant that does not exist in the GiaB callset and was not detected

by the pipeline.

False Negative (FN). A variant that exists in the GiaB callset and was not detected by the pipeline.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Fscore = \frac{2TP}{2TP + FP + FN} \tag{5}$$

4 Results and Discussion

4.1 Alignment Statistics

One of the goals of this analysis is to determine the affect of different alignment algorithms on variant calling on both the commonly used b37 reference and the newer GRCh38 reference. Differences in alignment would first be seen from alignment statistics of each aligner prior to any variant calling. Tables 3 and 4 show the performance of each of the three aligners on the b37 and GRCh38 human references respectively. For both of the human references, bwamem and stampy align similar number of reads, however, novoalign is unique in that it aligns 5.5 to 6% less reads than the other two aligners; with 954,159 and 1,025,734 less reads aligned compared to the aligner with the most alignments in each reference. There are a few possible reasons for this trend. First, both of the references could be missing the same sequence from their assemblies, though this is unlikely as the nature of exome capture is such that the sequence to be captured is identified a priori in order for the pull down experiments to be performed. While b37 is missing patch sequences from the GRCh37 assembly and this new

Table 3: NA12878 (SRR098401) alignment metrics for each alignment tool against the b37 human reference.

Aligner	Total Reads	Aligned Reads	% Aligned	% Pairs Aligned	Strand Bias
bwamem	170176638	169966666	99.88	99.98	0.50
novoalign	170176638	160425076	94.27	99.78	0.50
stampy	170176638	169964473	99.88	99.99	0.50

Table 4: NA12878 (SRR098401) alignment metrics for each alignment tool against the GRCh38 human reference.

Aligner	Total Reads	Aligned Reads	% Aligned	% Pairs Aligned	Strand Bias
bwamem	170176638	170131240	99.97	99.99	0.50
novoalign	170176638	159883353	93.95	99.88	0.50
stampy	170176638	170140694	99.98	99.99	0.50

sequence does exist in the GRCh38 assembly, one would then expect the number of aligned reads to increase on the GRCh38 reference but this is not the case. One explanation of why this might be is that some unique sequence in the primary assembly of GRCh37/b37 has been moved to ALT sequences in the new assembly model and in this analysis ALT sequences are not being used. However, one would then expect the alignments for all aligners to be lower in the newer GRCh38 reference. Again, this is not the case as both bwamem and stampy alignments increase slighty in GRCh38. One final explanation for the lower read alignments by novoalign could be that it is more specific with regards to not mapping reads from homologous sequence onto targeted exons, for example from pseudogenes, but bwamem and stampy are mismapping these highly similar sequences. In either case, whether the additional read alignments from bwamem and stampy are misalignments (false positives) or if novoalign is missing true alignments (false negatives) is one affect that should be investigated in downstream variant calling results.

4.2 Exome Coverage Metrics

An additional metric for comparing alignments within and between different versions of the reference is to consider the affect alignment algorithms have on exome target region coverage.

Tables 5 and 6 show the affect of each of the three aligners on the target region coverage on both b37 and GRCh38 respectively. The size, in total bases, of the target capture regions for GRCh38 is 162.74 kilobases smaller then for b37. This is because some primary sequence in the b37 assembly was moved to alternate loci in the GRCh38 assembly and these alternate sequences were not included in the alignment stage. Current versions of the alignment tools do not understand the new assembly model with alternative loci and including them would cause non-unique alignments. Regardless, and in opposition to the differences seen in number of total read alignments, all aligners have the same percentage of uniquely mapped reads and this is similar across both versions of the reference. In addition, the average depth of coverage is roughly equivalent across aligner and references. Moreover, the percentage of target regions with zero coverage is also roughly the same and is most likely an affect of sequence complexity, composition, or low performing capture bait sequences and not due to any of the alignment tools specifically. This low percentage of zero coverage will introduce a baseline level of false negatives, missed variants, in all pipelines.

While average coverage is a good overall metric, exome capture has a high level of non-uniform coverage, therefore it is important to understand the fraction of bases covered at a certain depth; for example, 80% of bases are covered at greater than or equal to 20x coverage. Figures 1 and 2 show this fraction of bases as a function of increasing depth for each aligner on the two different references. Congruent with the alignment statistics, bwamem and stampy have similar results and novoalign shows a slight reduction in coverage. Overall, all aligners, regardless of the reference used show that greater than 80% of all bases are covered at 20x, greater than 60% of all bases are covered at 50x, and only around 30% bases are covered at a depth greater than 100x. The long tail of these graphs indeed demonstrate the wide variation in coverage typical of exome sequencing.

Table 5: NA12878 (SRR098401) target region coverage metrics for each alignment tool against the b37 human reference.

Aligner	Target Bases	% UQ Aligned	Mean Depth	% Zero Coverage	2x	10x	20x	50x	100x
bwamem	46205186	94.33	80.94	3.22	95.35	88.75	81.88	60.25	28.54
novoalign	46205186	93.94	80.90	3.21	95.36	88.75	81.88	60.24	28.53
stampy	46205186	93.90	80.23	2.51	95.77	88.73	81.83	60.19	28.50

Table 6: NA12878 (SRR098401) target coverage metrics for each alignment tool against the GRCh38 human reference.

Aligner	Target Bases	% UQ Aligned	Mean Depth	% Zero Coverage	2x	10x	20x	50x	100x
bwamem	46042451	93.74	80.97	2.94	95.69	89.06	82.17	60.47	28.66
novoalign	46042451	93.55	80.94	2.93	95.70	89.06	82.17	60.46	28.65
stampy	46042451	93.50	80.40	2.38	96.00	89.03	82.11	60.41	28.62

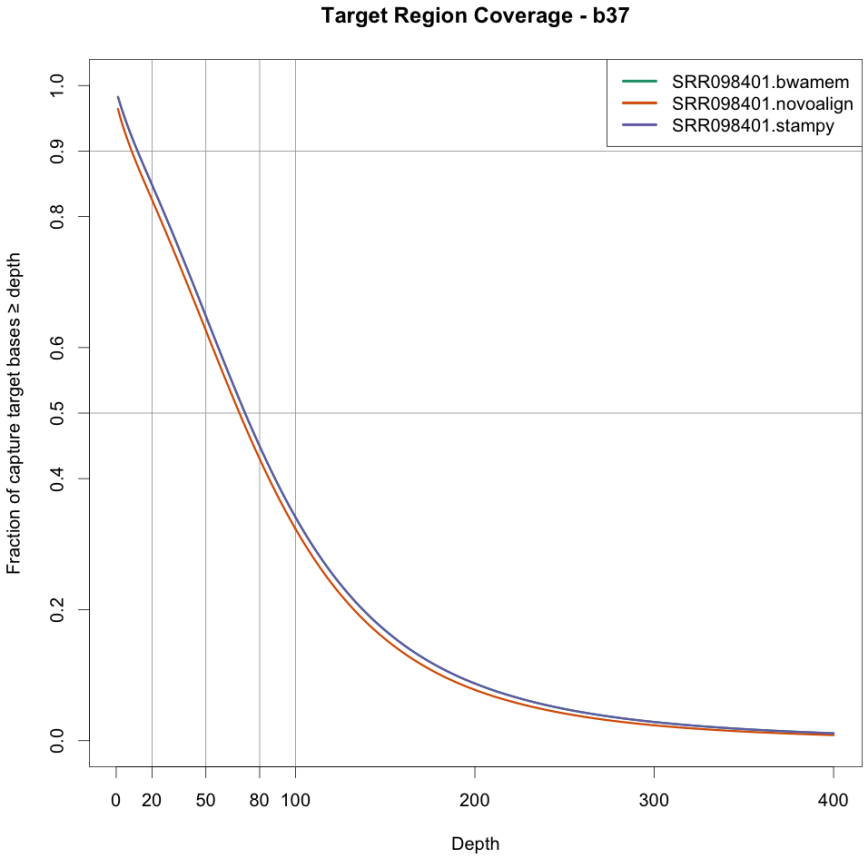

Figure 1: Exome target region coverage as the fraction of bases coverage by depth on the b37 reference.

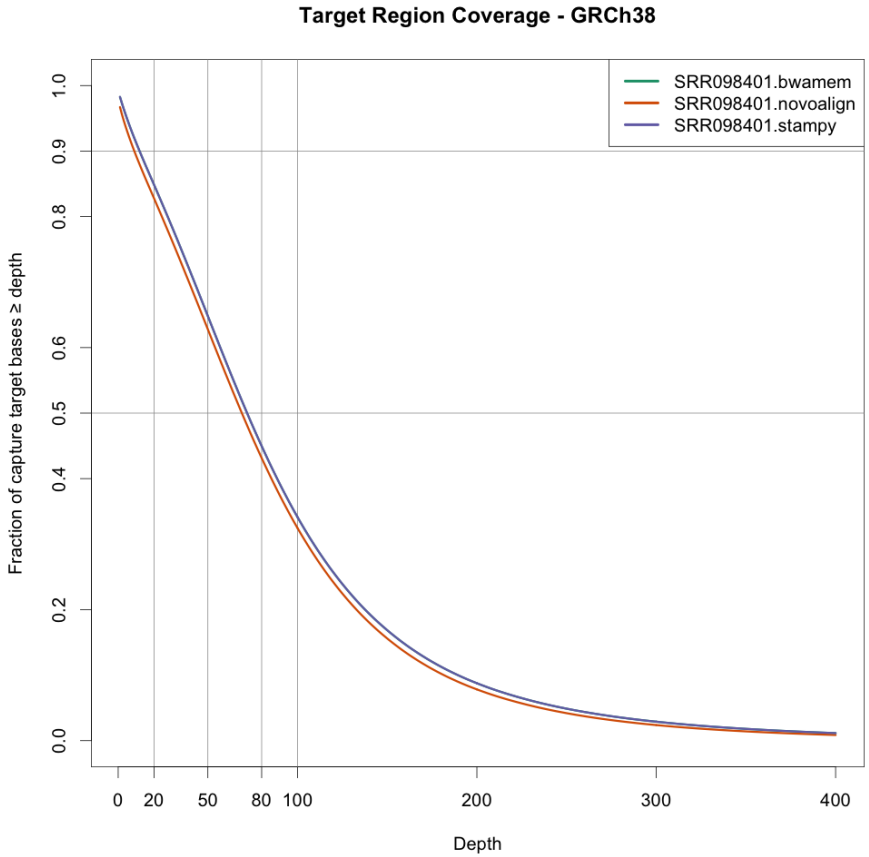

Figure 2: Exome target region coverage as the fraction of bases coverage by depth on the GRCh38 reference.

4.3 GiaB Comparison Results

Variant callers may differ in their relative performance of variant calling for different classes of variants, SNP versus indels. Therefore, in order to better understand the performance differences of the nine pipelines SNP metrics were computed separately from metrics for insertions and deletions. Figures 5 (b37) and 6 (GRCh38) give a general overview of SNP and indel statistics for each of the nine pipelines.

4.3.1 SNP Sensitivity and Precision

Tables 7 and 8, b37 and GRCh38 references respectively, show the number of raw SNP true positives (TP), false positives (FP), and false negatives (FN), as well as, the sensitivity, precision, F score, and transition/transversion (TiTv) ratio for all nine pipelines. The TiTv ratio is one of the most sensitive, albeit indirect, metrics for evaluating SNP accuracy. Across the whole human genome the approximate TiTV ratio is 2.1, however, this ratio varies for different regions of the genome and between populations and individuals. For the coding regions of the genome (exome) the TiTv ratio is approximately 3. Looking at the TiTv ratios for the nine pipelines it can be seen that those pipelines with the highest precision, thus the lowest false positives, are closest to 3. In general, all nine aligner and caller combinations have high TiTv ratios across both the b37 and GRCh38 references. While there is some variation across pipelines, there is a relatively low number of false positives SNPs calls in all nine pipelines. The top panels of Figures 7 and 8, show the relative frequencies of SNP TP and FP (precision) calls of each pipeline grouped into panels by variant caller. There is a relatively larger variation in precision between the different variant callers and little to no affect of individual aligner used within each pipeline. On average, across the nine pipelines, platypus has the highest SNP precision (99.4%), followed by samtools (98.3%) and GATK HaplotypeCaller (97.7%). There is only an average difference of 1.7% between the highest and lowest precision and the actual number of false positive variants ranges from 123 (novoalign-platypus) to 595 (stampy-gatk).

The ability for a pipeline to correctly identify as many true variants as possible, namely sensitivity, is paramount in causal variant discovery and any diagnostic assay's accuracy and utility. Comparing the sensitivity of the nine pipelines shows a different ranking of pipelines compared to their precision, see the top panels of Figures 9 and 10. In general, as with the precision, there is little variation between the different aligners, where as the variant caller has the largest impact. On average, samtools has the highest SNP sensitivity (97.2%), followed by GATK HaplotypeCaller (95.3%) and then platypus (95.0%). The number of true positives range from 22903 (novoalign-samtools) to 22028 (novoalign-platypus). Overall, the SNP sensitivity between b37 and GRCh38 is the same, even though in GRCh38 fewer SNPs are called overall due to a lower number of target bases without the ALT sequences. There is a baseline level of false negatives common to all pipelines because of the small percentage of total target capture regions with no coverage.

In summary, the total precision and sensitivity of all variant callers are well tuned for SNP calling regardless of the aligner. However, while the total sensitivity gives a good global overview of each pipeline it is known that the depth of coverage is one of the most critical variables that impacts sensitivity. Figure 3 plots each pipeline's sensitivity as a function of the depth at the variant position. The depth here is the sum of the individual coverage of all samples used in joint calling (combining data across samples is what provides power to calling variants jointly). The three samtools pipelines are the most sensitive when jointly calling of lower coverage SNPs and increases rapidly to the point where the majority (90%) of variants are detected by 100x coverage. Whereas, GATK HaplotypeCaller and platypus require approximately 150x and 200x total coverage, respectively, to reach 90% sensitivity. Figure 4 shows the same sensitivity as a function of depth but on the GRCh38 reference in which there is little difference in the results compared to the b37 reference.

Table 7: Raw SNP statistics for the nine pipelines against the b37 human reference.

Aligner	Caller	TP	FP	FN	Sensitivity	Precision	F score	TiTv
bwamem	gatk	22522	519	1087	95.40	97.75	96.56	2.82
bwamem	platypus	22053	135	1149	95.05	99.39	97.17	3.01
bwamem	samtools	22899	393	653	97.23	98.31	97.77	2.87
novoalign	gatk	22519	465	1092	95.38	97.98	96.66	2.84
novoalign	platypus	22028	123	1169	94.96	99.44	97.15	3.01
novoalign	samtools	22903	318	652	97.23	98.63	97.93	2.88
stampy	gatk	22525	595	1084	95.41	97.43	96.41	2.81
stampy	platypus	22086	134	1145	95.07	99.40	97.19	3.00
stampy	samtools	22897	475	645	97.26	97.97	97.61	2.86

Table 8: Raw SNP statistics for the nine pipelines against the GRCh38 human reference.

Aligner	Caller	TP	FP	FN	Sensitivity	Precision	F score	TiTv
bwamem	gatk	22020	628	1109	95.21	97.23	96.21	2.84
bwamem	platypus	21550	298	1163	94.88	98.64	96.72	3.02
bwamem	samtools	22389	470	683	97.04	97.94	97.49	2.90
novoalign	gatk	21999	563	1131	95.11	97.50	96.29	2.87
novoalign	platypus	21517	294	1188	94.77	98.65	96.67	3.03
novoalign	samtools	22384	438	687	97.02	98.08	97.55	2.90
stampy	gatk	22016	641	1115	95.18	97.17	96.16	2.84
stampy	platypus	21579	302	1158	94.91	98.62	96.73	3.01
stampy	samtools	22385	503	677	97.06	97.80	97.43	2.89

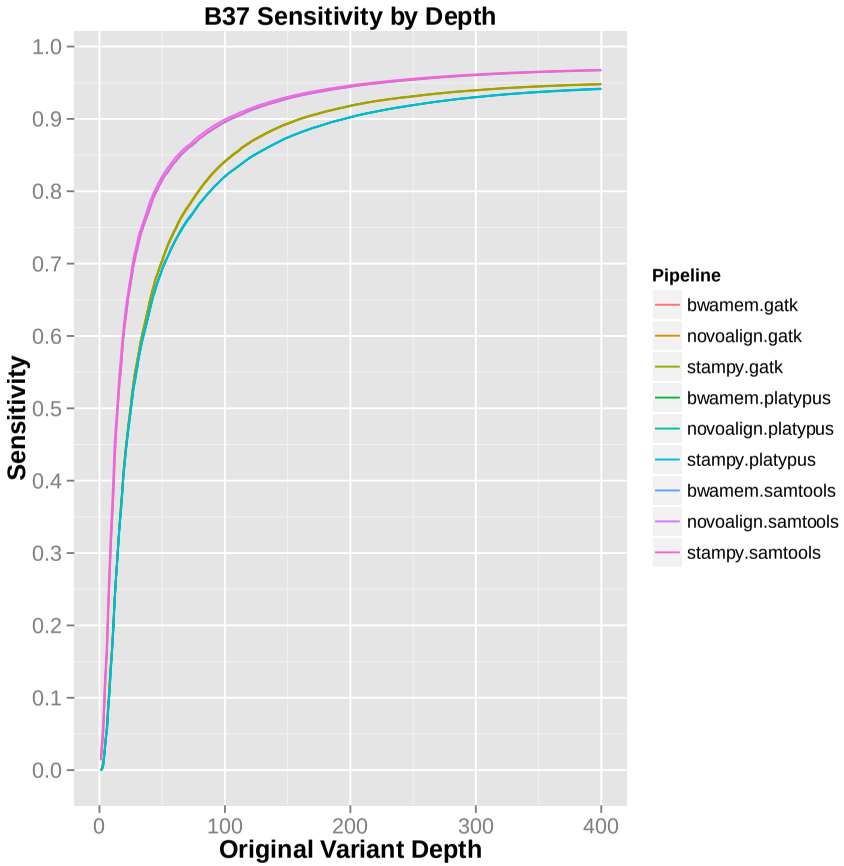

Figure 3: b37 SNP sensitivity as a function of original depth across all samples.

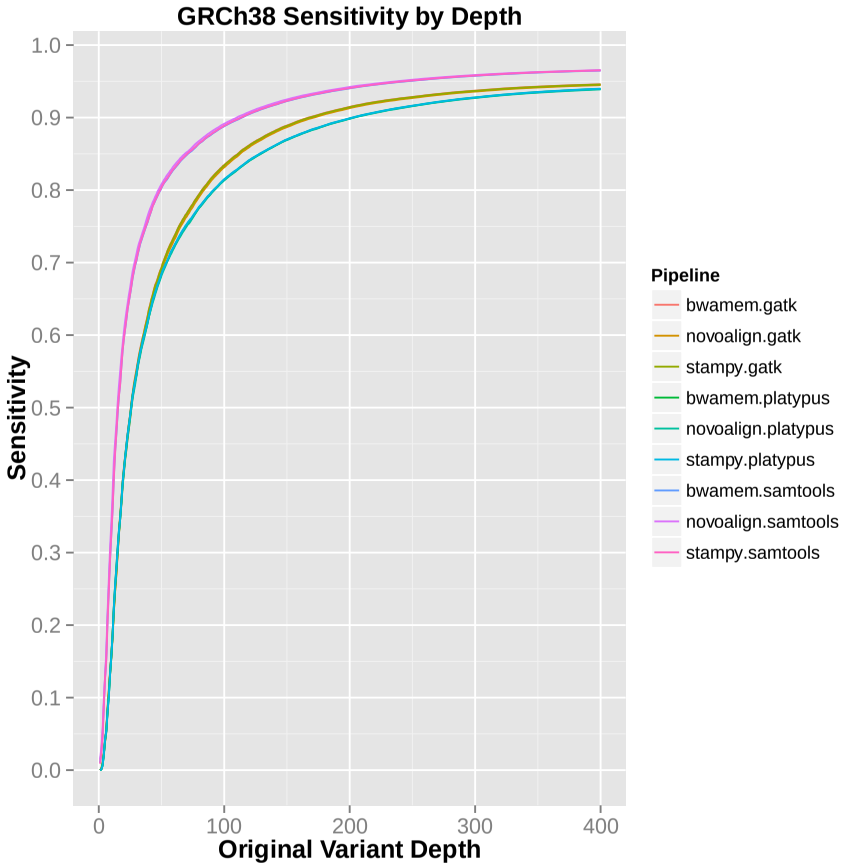

Figure 4: GRCh38 SNP sensitivity as a function of original depth across all samples.

4.3.2 Indel Sensitivity and Precision

Insertions and deletions pose different challenges for aligners and variant callers than SNPs and therefore the evaluation of their performance across the different pipelines should be separated. Table 9 shows the true positive, false positive, and false negative counts, stratified by insertions and deleted, as well as their respective sensitivity and precision and total F score. GATK HaplotypeCaller is clearly the most sensitive overall for both insertions and deletions, however, to differing degrees. The HaploTypeCaller is more sensitive to deletions than to insertions, but in return has many more false positive deletions and therefore is more precise in calling insertions. The bottom panels of Figures 7 and 9 show the proportion of TP to FP and FN for indels group by variant caller. Platypus in general is highly precise and is less sensitive than GATK HaplotypeCaller and performs evenly across both insertions and deletions for all three of the different aligners. Samtools is the worst performing variant caller and has the most FP and FN than the other two variant callers for both insertions and deletions. Samtools also shows the most variability between different aligners for both sensitivity and precision, thus showing the advantage of the reassembly step in the HaplotypeCaller and platypus variant callers, which mainly abrogates the effects of the different aligners. The affect of aligners on the indel calls is much less noticeable that the general affect of the variant callers, but Novoalign tends to be slightly more precise than bwamem across the different variant callers. Overall, the HaplotypeCaller has the top three F scores, followed by platypus, making it the best choice for calling indels.

Table 10 shows the same insertion and deleted statistics for each pipeline against the GRCh38 reference. From Figures 7 and 8 it is clear that the indel precision of the nine pipelines is similar between the two references, however there is a reduced sensitivity which affects insertions to a much larger degree than deletions (Figure 10).

Table 9: Raw indel statistics for the nine pipelines against the b37 human reference.

		Insertions					Deletions					Total		
Aligner	Caller	TP	FP	FN	Sensitivity	Precision	TP	FP	FN	Sensitivity	Precision	Sensitivity	Precision	F Score
bwamem	gatk	548	18	65	89.40	96.82	571	49	55	91.21	92.10	90.31	94.35	92.29
bwamem	platypus	560	9	103	84.46	98.42	560	7	97	85.24	98.77	84.85	98.59	91.21
bwamem	samtools	484	163	118	80.40	74.81	501	46	106	82.54	91.59	81.47	82.50	81.98
novoalign	gatk	547	13	66	89.23	97.68	573	42	53	91.53	93.17	90.40	95.32	92.79
novoalign	platypus	549	6	93	85.51	98.92	556	7	98	85.02	98.76	85.26	98.84	91.55
novoalign	samtools	494	148	108	82.06	76.95	501	41	108	82.27	92.44	82.16	84.04	83.09
stampy	gatk	548	21	64	89.54	96.31	573	45	53	91.53	92.72	90.55	94.44	92.45
stampy	platypus	635	11	87	87.95	98.30	572	9	98	85.37	98.45	86.71	98.37	92.17
stampy	samtools	508	373	94	84.39	57.66	500	100	105	82.64	83.33	83.51	68.06	75.00

Table 10: Raw indel statistics for the nine pipelines against the GRCh38 human reference.

		Insertions					Deletions					Total		
Aligner	Caller	TP	FP	FN	Sensitivity	Precision	TP	FP	FN	Sensitivity	Precision	Sensitivity	Precision	F Score
bwamem	gatk	475	15	134	78.00	96.94	543	48	54	90.95	91.88	84.41	94.17	89.02
bwamem	platypus	490	10	171	74.13	98.00	534	11	95	84.90	97.98	79.38	97.99	87.71
bwamem	samtools	413	155	188	68.72	72.71	472	49	109	81.24	90.60	74.87	81.27	77.94
novoalign	gatk	474	15	136	77.70	96.93	544	43	53	91.12	92.67	84.34	94.61	89.18
novoalign	platypus	478	8	160	74.92	98.35	530	11	97	84.53	97.97	79.68	98.15	87.96
novoalign	samtools	423	141	178	70.38	75.00	473	45	108	81.41	91.31	75.80	82.81	79.15
stampy	gatk	475	16	134	78.00	96.74	545	43	51	91.44	92.69	84.65	94.53	89.32
stampy	platypus	561	13	155	78.35	97.74	547	13	96	85.07	97.68	81.53	97.71	88.89
stampy	samtools	437	358	164	72.71	54.97	473	98	104	81.98	82.84	77.25	66.62	71.54

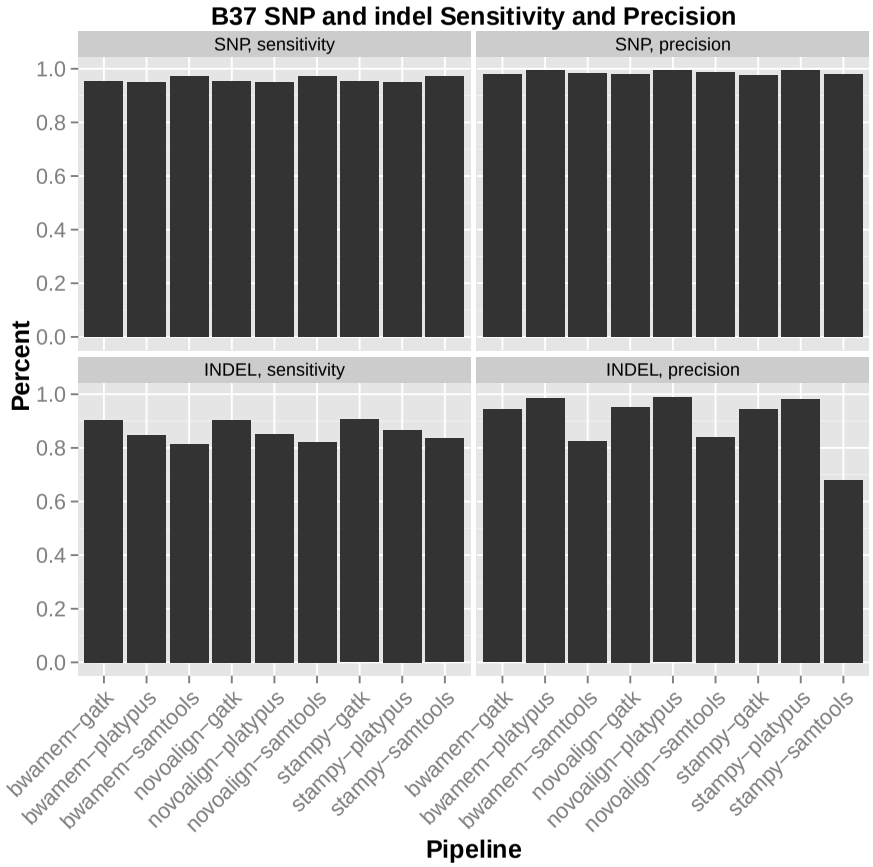

Figure 5: b37 SNP and indel sensitivity and precision for each of the nine pipelines.

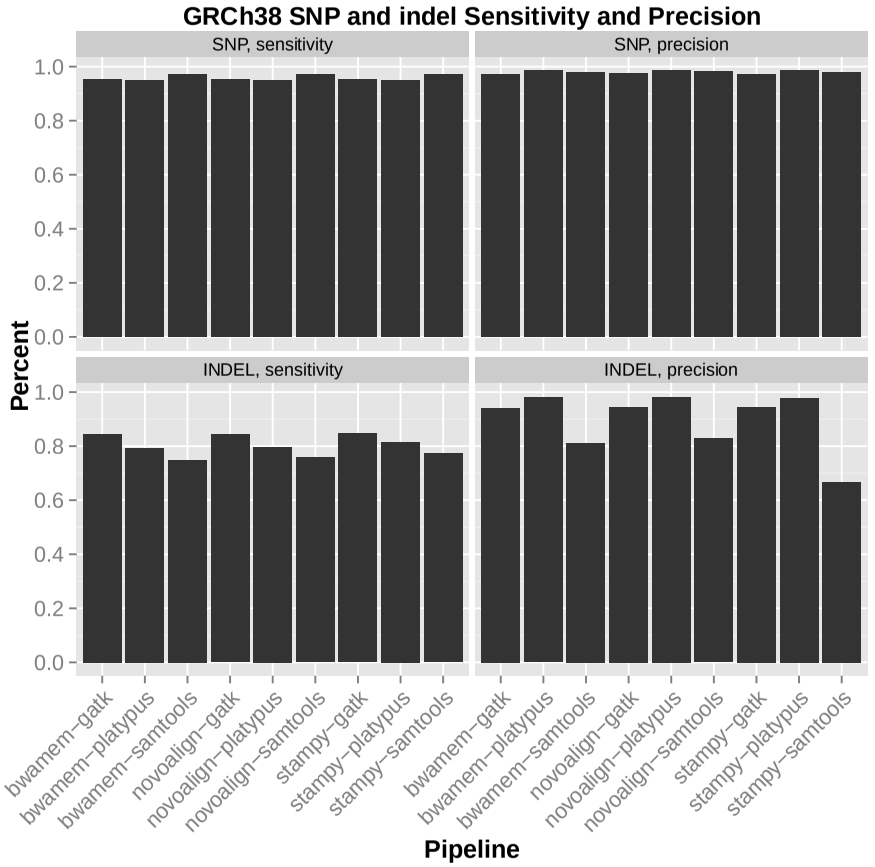

Figure 6: GRCh38 SNP and indel sensitivity and precision for each of the nine pipelines.

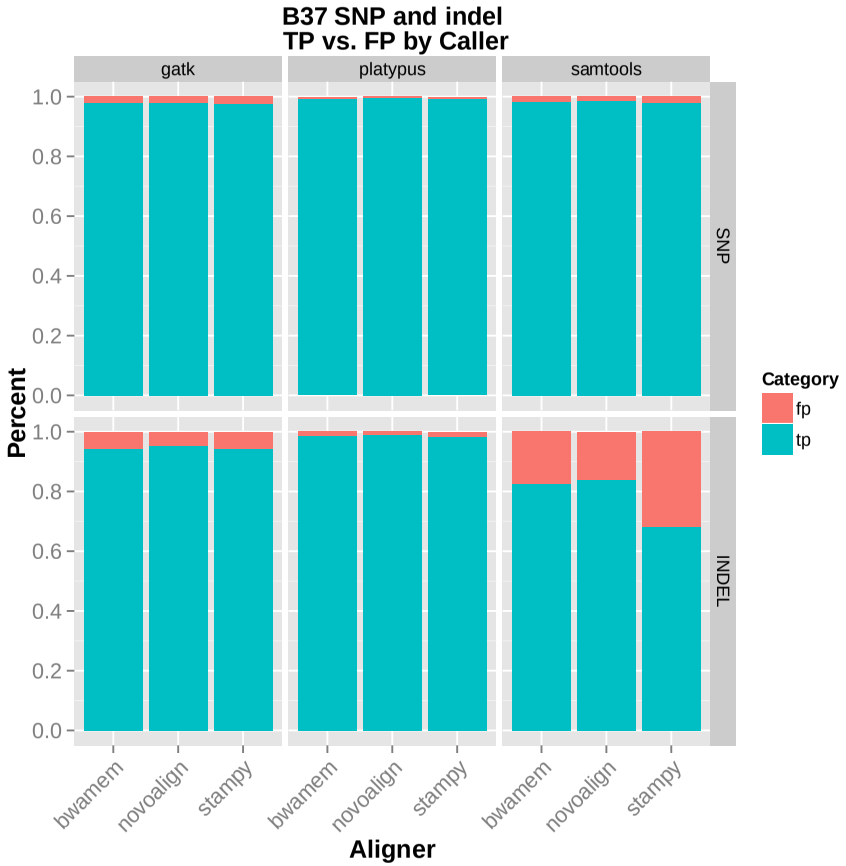

Figure 7: b37 SNP and indel proportion of TP and FP (precision) for each of the aligners grouped by variant caller.

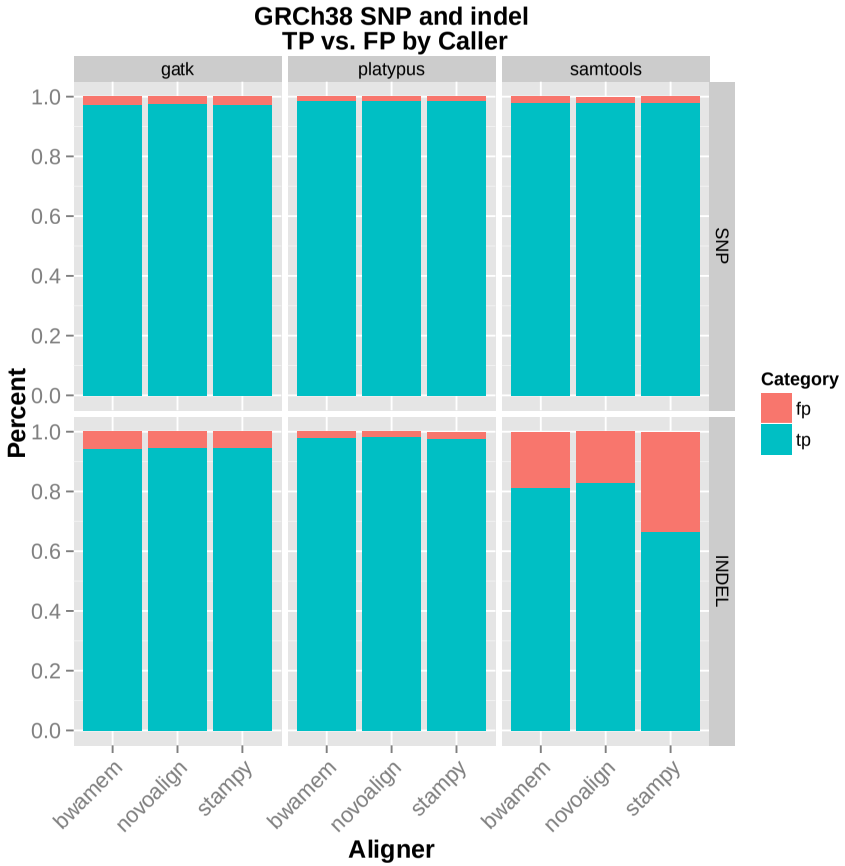

Figure 8: GRCh38 SNP and indel proportion of TP and FP (precision) for each of the aligners grouped by variant caller.

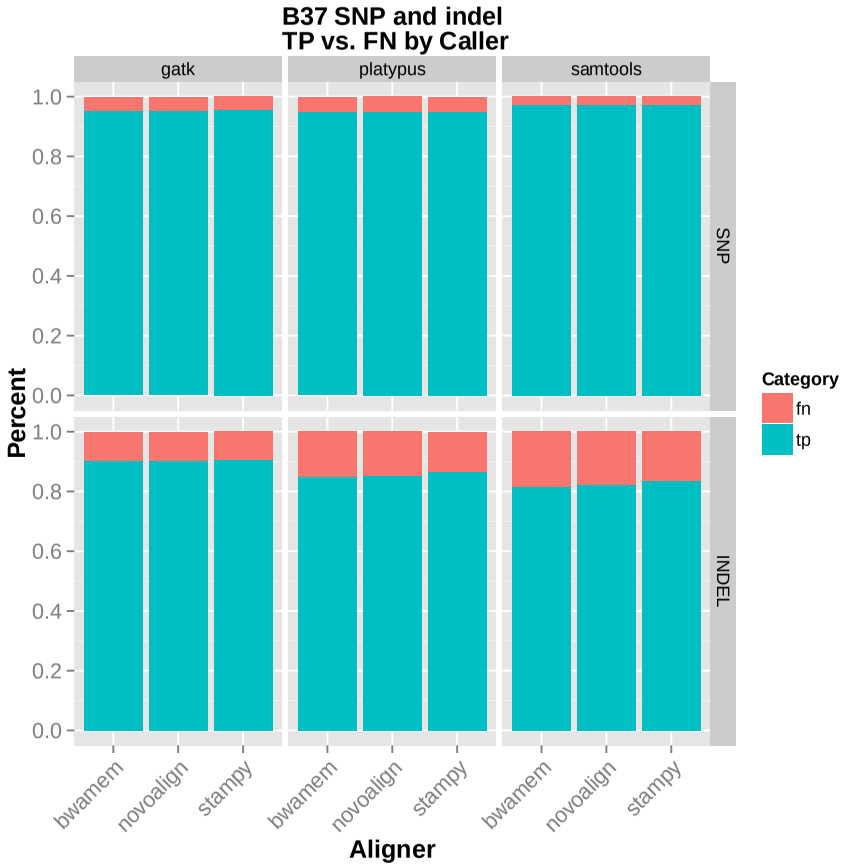

Figure 9: b37 SNP and indel proportion of TP and FN (sensitivity)for each of the aligners grouped by variant caller.

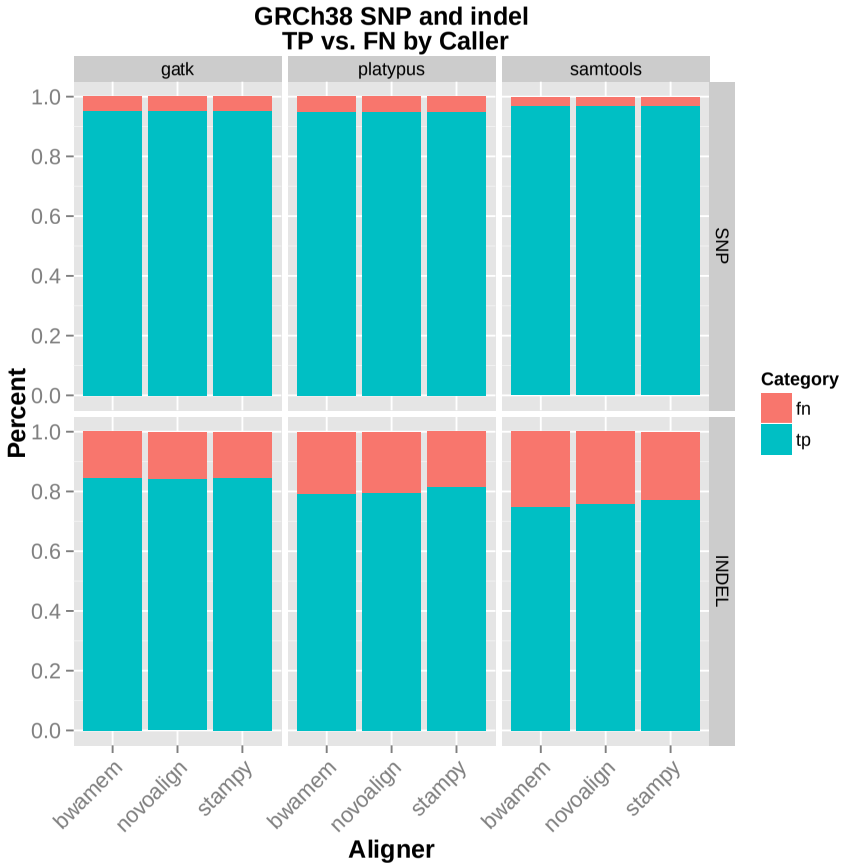

Figure 10: GRCh38 SNP and indel proportion of TP and FN (sensitivity)for each of the aligners grouped by variant caller.

5 Conclusions

In this analysis it was found that all three variant callers are tuned reasonably well to calling SNPs in exome data regardless of which alignment tool was used. Based on the overall SNP F score, novoalign along with samtools performs the best when calling SNPs, though all pipelines are with a 1.5% difference to this best pipeline. In the context of indels, there is considerable reductions overall in sensitivity and, to a lesser extent, precision for both insertions and deletions. Except for GATK HaplotypeCaller, there is more variability in the indel sensitivity based on both the aligner and variant caller used to call indels. Both of the variant callers which perform reassembly using a de Bruijn graph approach to construct haplotypes perform better than samtools which has no reassembly step. GATK HaplotypeCaller performing best without regard to the aligner used. Samtools did not perform well overall in calling indels. Based on the overall Indel F score, novoalign along with GATK HaplotypeCaller performs the best when calling insertions and deletions.

The use of the latest human reference, GRCh38, still seems premature based on the results of this analysis. While calling SNPs using the GRCh38 reference was comparable there was a reduction in indel sensitivity. It may be the case that this will improve as more aligners and variant callers adjust to the use of ALT sequences in the GRCh38 assembly.

While sequencing technologies and algorithms will continue to improve in accuracy over time, it is expected that next generation sequencing will always produce some level of false positive and negative results. Therefore, understanding error rates and their affect on the results and its interpretation is important in moving next generation sequencing into a clinical testing setting.

False positive variant calls lower the precision (also known as the positive predictive value) of the assay and therefore will increase the number of possible confirmatory tests that must be performed in order to obtain a more accurate assessment regarding which variants are truly present in the individual. In addition, large numbers of false positives place a burden on those professionals tasked with conducting interpretation to determine the clinical relevance

and pathogenicity of each variant. This process can be time consuming, therefore, reducing the number of false positive calls, without affecting the sensitivity, is paramount to the success of a sequencing assay in a clinical setting.

Knowing which variants the individual does not have is as equally important as knowing which variants are present. The negative predictive value of the assay is the ratio of the number of true negatives over the sum of the true and false negative variants, thus, as the number of false negative variants increases, the lower the negative predictive value of the assay. Being able to reduce the number of false negatives, and increasing the sensitivity, will determine how well the assay can be used as a decision tool to rule out possible variants that may be suspected to be underlying a certain phenotype or condition. If the negative predictive value is too low, the overall utility or relevance as a diagnostic tool in a clinical setting is reduced and an additional method of assessing the presence of a variant may have to be run to be certain that an individual does not have variants of interest.

In general, the false negative error rate of a next generation sequencing test is difficult to ascertain. Testing involves multiple, complicated wet lab and computational steps which can all be sources of type II errors; including not insufficient read coverage in target regions due to wet laboratory processes and sample handling or local sequence complexity, low base and variant quality scores and forward and reverse strand bias. Therefore, it is important for quality control and the overall interpretability of results of next generation sequencing tests to be able to determine the possibility of false negatives for variants of interest and to attribute them to these different sources of missing-ness to be reported along side true positive variants.

While it is not currently possible to define false positive and negative error rates exhaustively, this assessment has shown that by using a sample with known examples of the variants type indicated for detection, in this case single nucleotide variants as well as small insertions and deletions, the error rates can be analytically assessed and to a relatively high degree of sensitivity and precision. Given that the focus of this assessment was on producing a set of

baseline error rates across a combination of tools that can make up bioinformatics pipelines it should be noted that in a clinical test validation setting the findings of these pipelines may be tested using alternative methods to further ascertain true error rates. Moreover, given the sequencing datasets used in this assessment was from a population level study and was thus of reduce coverage it is hypothesized that the sensitivity and precision would improve when using a sequencing read dataset of high breadth and depth of coverage which are produced under more strict requirement such as those in a clinical testing environment.

In addition, findings in this assessment provide a small degree of evidence for the use of certain pipelines in the initial stages of developing a validated pipeline for clinic sequencing for finding causal and candidate gene identification in inherited diseases by exome sequencing, however, the College of American Pathologists Laboratory standards checklist requirements state that laboratories should analyze a sufficient number of sequence read datasets with de novo, dominant, and recessive variants to ensure reproducibility and to provide a more robust analytic and diagnostic sensitivity and specificity. Therefore, the current assessment does not provide enough information to be able to suggest any of the tested pipelines are ready to be used in a clinical setting; though the computational groundwork for such further validation has been completed in the testing framework developed as part of this assessment.

It is acknowledged that the results obtained from this assessment may not be applicable in other testing scenarios. This is because the experimental design tested in this assessment, of Mendelian inherited diseases by analyzing a parent-offspring trio, may not provide the same level of evidence or necessary testing of these tools to provide evidence for other study designs or hypothesis. These include detecting somatic mutation detection in cancer, detecting mosaicism, or prenatal testing from maternal blood samples. In these cases there is heterogeneity in the cellular population of the sample and very low frequency of alleles for which the limits of detection must be established and which is not assessed here.

This study could be extended in multiple directions assuming that a highly validated variant call set is available in all situations. The immediate would be to include many

more and diverse trio samples for different populations and with many different types of variants, including copy number variations and larger structural variants, such as inversions and translocations. Next, additional metrics could assess in the parent-proband trio design. These include the ability to detect de novo mutations and the accuracy and quality of heterozygous calls in the proband based on the parent genotypes, as well as, testing for deviations from Hardy Weinberg equilibrium and other incorrect or conflicting inheritance patterns. Additionally, assessing the ability to accurately phase genotypes and construct haplotypes and diplotypes from raw sequencing reads is important for imputation of missing SNPs in genome wide association studies and understanding the relationship between DNA content on homologous chromosomes and phenotypes. Finally, as mentioned above, additional experimental designs could be assessed, such as cases where there is a high degree of heterogeneity of cellular composition and differing degrees of allele frequencies. One way to assess this would be to use standard reference samples in which multiple individual samples with known allele frequencies are mixed in differing proportions to assess the limit of detection for low frequency alleles.

6 Future Work

Future work includes automating the developed comparison pipeline to be used in testing regressions and enhancements to existing tools and methods as well as to benchmark newly published tools. As more reference materials are published by NIST and the Genome in a Bottle Consortium this pipeline could include these samples and different study designs for comparison. In addition, this comparison could be extended to include automated filtering that is not specific to any single variant caller. This filtering could be based on either a support vector machine algorithm or a Gaussian mixture model, similar to GATK's Variant Quality Score Recalibrator. More downstream analysis like functional annotations with variant impact on genes and pathways as well as population based metadata, like minor

allele frequencies, would be beneficial in downstream interpretation. Finally, more work could be done to optimize the parallelization on CIRC's BlueHive cluster with the goal of testing the feasibility of whole genome comparisons on that resource.

References

Altschul, Stephen F., Warren Gish, et al. (1990). "Basic local alignment search tool." In: *Journal of Molecular Biology* 215.3, pp. 403–410. DOI: 10.1016/s0022-2836(05)80360-2.

Altschul, Stephen F., T L. Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." In: *Nucleic Acids Res.* 25.17, pp. 3389–402.

Aziz, Nazneen et al. (2015). "College of American Pathologists' Laboratory Standards for Next-Generation Sequencing Clinical Tests." In: *Archives of Pathology & Laboratory Medicine* 139.4, pp. 481–493. DOI: 10.5858/arpa.2014-0250-cp.

Bamshad, Michael J., Sarah B. Ng, et al. (2011). "Exome sequencing as a tool for Mendelian disease gene discovery." In: *Nat Rev Genet* 12.11, pp. 745–755. DOI: 10.1038/nrg3031.

Bamshad, Michael J., Jay A. Shendure, et al. (2012). "The Centers for Mendelian Genomics: A new large-scale initiative to identify the genes underlying rare Mendelian conditions." In: *Am. J. Med. Genet.* 158A.7, pp. 1523–1525. DOI: 10.1002/ajmg.a.35470.

Burrows, M. and D.J. Wheeler (1994). *A block-sorting lossless data compression algorithm*. Tech. rep. 124. Digital Equipment Corp.

Chen, K. et al. (2014). "TIGRA: A targeted iterative graph routing assembler for breakpoint assembly." In: *Genome Research* 24.2, pp. 310–317. DOI: 10.1101/gr.162883.113.

Cheng, A. Y. et al. (2014). "Assessing single nucleotide variant detection and genotype calling on whole-genome sequenced individuals." In: *Bioinformatics* 30.12, pp. 1707–1713. DOI: 10.1093/bioinformatics/btu067.

Cox, A. J. (2007). *ELAND: efficient large-scale alignment of nucleotide databases*. Tech. rep. San Diego, CA: Illumina.

David, M. et al. (2011). "SHRiMP2: Sensitive yet Practical Short Read Mapping." In: *Bioinformatics* 27.7, pp. 1011–1012. DOI: 10.1093/bioinformatics/btr046.

DePristo, Mark A et al. (2011). "A framework for variation discovery and genotyping using next-generation DNA sequencing data." In: *Nature Genetics* 43.5, pp. 491–498. DOI: 10.1038/ng.806.

Durbin, Richard et al. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, UK: Cambridge University Press.

Exome Aggregation Consortium (ExAC) (2015). [Online; accessed 21-February-2015]. Cambridge, MA.

Fang, Han et al. (2014). "Reducing INDEL calling errors in whole genome and exome sequencing data." In: *Genome Medicine* 6.10. DOI: 10.1186/s13073-014-0089-z.

Garrison, E. and G. Marth (2012). "Haplotype-based variant detection from short-read sequencing." In: *ArXiv e-prints*. arXiv: 1207.3907 [q-bio.GN].

Ghoneim, Dalia H et al. (2014). "Comparison of insertion/deletion calling algorithms on human next-generation sequencing data." In: *BMC Research Notes* 7.1, p. 864. DOI: 10.1186/1756-0500-7-864.

Homer, Nils and Stanley F Nelson (2010). "Improved variant discovery through local re-alignment of short-read next-generation sequencing data using SRMA." In: *Genome Biol* 11.10, R99. DOI: 10.1186/gb-2010-11-10-r99.

Iqbal, Zamin et al. (2012). "De novo assembly and genotyping of variants using colored de Bruijn graphs." In: *Nature Genetics* 44.2, pp. 226–232. DOI: 10.1038/ng.1028.

Kahn, S. D. (2011). "On the Future of Genomic Data." In: *Science* 331.6018, pp. 728–729. DOI: 10.1126/science.1197891.

Kent, W. J. (2002). "BLAT—The BLAST-Like Alignment Tool." In: *Genome Research* 12.4, pp. 656–664. DOI: 10.1101/gr.229202.

Langmead, Ben and Steven L Salzberg (2012). "Fast gapped-read alignment with Bowtie 2." In: *Nature Methods* 9.4, pp. 357–359. DOI: 10.1038/nmeth.1923.

Langmead, Ben, Cole Trapnell, et al. (2009). "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." In: *Genome Biol* 10.3, R25. DOI: 10.1186/gb-2009-10-3-r25.

Li, H. (2011). "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data." In: *Bioinformatics* 27.21, pp. 2987–2993. DOI: 10.1093/bioinformatics/btr509.

Li, H. (2012). "Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly." In: *Bioinformatics* 28.14, pp. 1838–1844. DOI: 10.1093/bioinformatics/bts280.

Li, H. (2013). "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM." In: *ArXiv e-prints*. arXiv: 1303.3997 [q-bio.GN].

Li, H. and R. Durbin (2009). "Fast and accurate short read alignment with Burrows-Wheeler transform." In: *Bioinformatics* 25.14, pp. 1754–1760. DOI: 10.1093/bioinformatics/btp324.

Li, H. and R. Durbin (2010). "Fast and accurate long-read alignment with Burrows-Wheeler transform." In: *Bioinformatics* 26.5, pp. 589–595. DOI: 10.1093/bioinformatics/btp698.

Li, H. and R. Durbin (2011). "Inference of human population history from individual whole-genome sequences." In: *Nature* 475.7357, pp. 493–496. DOI: 10.1038/nature10231.

Li, H. and N. Homer (2010). "A survey of sequence alignment algorithms for next-generation sequencing." In: *Briefings in Bioinformatics* 11.5, pp. 473–483. DOI: 10.1093/bib/bbq015.

Li, H., J. Ruan, et al. (2008). "Mapping short DNA sequencing reads and calling variants using mapping quality scores." In: *Genome Research* 18.11, pp. 1851–1858. DOI: 10.1101/gr.078212.108.

Li, S. et al. (2013). "SOAPindel: Efficient identification of indels from short paired reads." In: *Genome Research* 23.1, pp. 195–200. DOI: 10.1101/gr.132480.111.

Lindner, Robert and Caroline C. Friedel (2012). "A Comprehensive Evaluation of Alignment Algorithms in the Context of RNA-Seq." In: *PLoS ONE* 7.12. Ed. by Steven L. Salzberg, e52403. DOI: 10.1371/journal.pone.0052403.

Lunter, G. and M. Goodson (2011). "Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads." In: *Genome Research* 21.6, pp. 936–939. DOI: 10.1101/gr.111120.110.

Ma, B. et al. (2002). "PatternHunter: faster and more sensitive homology search." In: *Bioinformatics* 18.3, pp. 440–445. DOI: 10.1093/bioinformatics/18.3.440.

MacArthur, D. G. et al. (2012). "A Systematic Survey of Loss-of-Function Variants in Human Protein-Coding Genes." In: *Science* 335.6070, pp. 823–828. DOI: 10.1126/science.1215040.

McKenna, A. et al. (2010). "The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data." In: *Genome Research* 20.9, pp. 1297–1303. DOI: 10.1101/gr.107524.110.

Meynert, Alison M et al. (2014). "Variant detection sensitivity and biases in whole genome and exome sequencing." In: *BMC Bioinformatics* 15.1, p. 247. DOI: 10.1186/1471-2105-15-247.

Narzisi, Giuseppe, Jason A O'Rawe, et al. (2014). "Accurate de novo and transmitted indel detection in exome-capture data using microassembly." In: *Nature Methods* 11.10, pp. 1033–1036. DOI: 10.1038/nmeth.3069.

Narzisi, Giuseppe and Michael C. Schatz (2015). "The Challenge of Small-Scale Repeats for Indel Discovery." In: *Front. Bioeng. Biotechnol.* 3. DOI: 10.3389/fbioe.2015.00008.

Novocraft (2015). [Online; accessed 21-February-2015].

O'Rawe, Jason et al. (2013). "Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing." In: *Genome Medicine* 5.3, p. 28. DOI: 10.1186/gm432.

Rasmussen, Kim R. et al. (2006). "Efficient q -Gram Filters for Finding All ε-Matches over a Given Length." In: *Journal of Computational Biology* 13.2, pp. 296–308. DOI: 10.1089/cmb.2006.13.296.

Rieber, Nora et al. (2013). "Coverage Bias and Sensitivity of Variant Calling for Four Whole-genome Sequencing Technologies." In: *PLoS ONE* 8.6. Ed. by Oliver Hofmann, e66621. DOI: 10.1371/journal.pone.0066621.

Rimmer, Andy et al. (2014). "Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications." In: *Nature Genetics* 46.8, pp. 912–918. DOI: 10.1038/ng.3036.

Ross, Michael G et al. (2013). "Characterizing and measuring bias in sequence data." In: *Genome Biol* 14.5, R51. DOI: 10.1186/gb-2013-14-5-r51.

Ruffalo, M. et al. (2011). "Comparative analysis of algorithms for next-generation sequencing read alignment." In: *Bioinformatics* 27.20, pp. 2790–2796. DOI: 10.1093/bioinformatics/btr477.

Rumble, Stephen M. et al. (2009). "SHRiMP: Accurate Mapping of Short Color-space Reads." In: *PLoS Comput Biol* 5.5. Ed. by Wyeth W. Wasserman, e1000386. DOI: 10.1371/journal.pcbi.1000386.

Smith, T.F. and M.S. Waterman (1981). "Identification of common molecular subsequences." In: *Journal of Molecular Biology* 147.1, pp. 195–197. DOI: 10.1016/0022-2836(81)90087-5.

Van der Auwera, Geraldine A. et al. (2013). "From FastQ Data to High-Confidence Variant Calls: The Genome Analysis Toolkit Best Practices Pipeline." In: *Current Protocols in Bioinformatics*. John Wiley & Sons, Inc. ISBN: 9780471250951. DOI: 10.1002/0471250953.bi1110s43.

Watson, Ian R. et al. (2013). "Emerging patterns of somatic mutations in cancer." In: *Nat Rev Genet* 14.10, pp. 703–718. DOI: 10.1038/nrg3539.

Zook, Justin M et al. (2014). "Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls." In: *Nat Biotechnol* 32.3, pp. 246–251. DOI: 10.1038/nbt.2835.