Chapter 11

Variant Calling From Next Generation Sequence Data

Nancy F. Hansen

Abstract

The use of next generation nucleotide sequencing to discover and genotype small sequence variants has led to numerous insights into the molecular causes of various diseases. This chapter describes the use of freely available software to align next generation sequencing reads to a reference and then to use the resulting alignments to call, annotate, view, and filter small sequence variants. The suggested variant calling workflow includes read alignment with novoalign, the removal of polymerase chain reaction duplicate sequences with samtools or bamUtils, and the detection of variants with Freebayes or bam2mpg software. ANNOVAR is then used to annotate the predicted variants using gene models, population frequencies, and predicted mutation severity, producing variant files which can be viewed and filtered with the variant display tool VarSifter.

Key words Variant detection, Next generation sequencing, Genotyping, Whole exome, Annotation

1 Introduction

In the years following the completion of the human genome reference sequence [1], a small set of new technologies became available to researchers hoping to extend DNA sequencing to a large number of human genomes, as well as to other unsequenced organisms. These "next generation" technologies were not notable for reading long stretches of DNA, and in fact, the sequence reads they produced were far shorter (anywhere from 25 to 150 bases) than the read lengths obtained using the capillary electrophoresis-based methods used for the Human Genome Project. What made these new sequencing technologies revolutionary was their ability to produce millions of reads in a single machine run, with lower cost and effort than was previously needed to produce thousands [2].

As a result, the development of next generation DNA sequencing has dramatically altered the process of discovering single nucleotide variants (SNVs) and small deletion and insertion variants (DIVs) from nucleotide sequencing data. By allowing large-scale, accurate interrogation of short single-molecule sequences from

Ewy Mathé and Sean Davis (eds.), *Statistical Genomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1418, DOI 10.1007/978-1-4939-3578-9_11,

large numbers of samples, next generation sequencing has replaced older, more expensive methods involving subcloning or polymerase chain reaction (PCR) amplification followed by gel electrophoresis. As a result, there has been unprecedented characterization of genetic variation in both healthy and diseased tissues from humans and other species [3, 4].

1.1 Statistical Methods in Variant Analysis

The conversion of many millions of short segments of DNA sequence, each on the order of 150 nucleotide bases in length and accompanied by per-base probabilistic quality scores, into accurate chromosomal sequence differences from a reference genome involves large-scale data processing using complex algorithms [5]. Since the sequencing process itself involves random sampling of DNA molecules, the analysis of the data will inevitably make use of statistical methods, both to determine the original genomic locus of each of the bases that have been read by the sequencer (mapping and alignment) and to infer the alleles that were actually present in the sample (genotyping) as well as whether those alleles differ from a user's selected reference sequence (variant calling).

The most widely used read alignment programs use a "seed and extend" approach for the first stage of alignment to the reference. Short, highly similar matches ("seeds") are determined between a single read and various reference locations, using highly efficient algorithms which make use of suffix arrays, suffix trees, hashes, or other types of indexes. One of these types of indexes, created using the "Burrows-Wheeler Transform," is employed by several popular aligners for next generation reads [6–8]. Once seeds are determined, they are generally extended into full read alignments using a dynamic programming algorithm like the Smith–Waterman or Needleman–-Wunsch algorithm [9, 10], and then, if possible, the paired read is aligned in the same vicinity of the reference as the first read's match.

Variant calling generally employs a Bayesian approach, considering each locus of the genome one at a time. The program Freebayes groups short-range haplotypes into single loci [11], while bam2mpg considers each single base, insertion, or deletion variant within the read alignments separately [12]. At each locus, a Bayes model specifying the probability of the observed allele counts within the sequencing reads (i.e., the likelihood of the data) is incorporated into a maximum likelihood approach or the application of Bayes' theorem to predict the presence or absence of a variant and the exact genotype at that locus. This probabilistic framework also allows for the assignment of meaningful phred-scaled variant and genotype scores [13].

1.2 Choice of Analysis Software

There are numerous software packages available for nearly every step in calling small variants from next generation sequencing data, and each has its own particular advantages and drawbacks. Perhaps

the best-known software pipeline for small variant calling is the Broad Institute's "best practices" pipeline, which consists of the BWA aligner [6] and the Genome Analysis Toolkit (GATK) [14]. The GATK pipeline produces variant calls that have been shown to be highly accurate, and offers versatility with respect to handling a variety of experimental designs, including pooling samples without identifying barcodes before sequencing, as well as sequencing samples to only a low depth of coverage. It also employs sophisticated filtering methods by training models to recognize features characteristic of false positives. Unfortunately, running the GATK pipeline requires extensive skill in software installation and configuration, a substantial investment of computational time and memory, and experimentation with the user's datasets in order to make optimal choices for software parameters and settings. In addition, the filtering steps require access to large sets of known variants, which are not always available for the species or genome build of interest.

Alternatively, if the sequence reads to be analyzed are from single samples, rather than unidentified pools, and these samples have been sequenced to sufficient depth of coverage to yield high quality variant calls (generally 25× or greater), then simpler software pipelines exist which are easier to implement and faster to run, without a demonstrable decrease in accuracy. One such alternative pipeline for the detection of small variants from next generation sequencing data is described here. For the alignment of reads to the reference sequence, novoalign is used, and for calling variants, the user can choose one of two Bayesian genotypers: Freebayes [11] or bam2mpg [12]. Variants are then annotated with the program ANNOVAR, filtered using quality scores and genomic location, and viewed with the VarSifter variant viewer.

1.3 Variant Filtering and Accuracy

Any method for calling variants will have reduced accuracy in regions of the genome that are prone to misalignment, due either to repetitive sequences or inaccuracies in the genome reference build, but appropriate use of filters can lead to highly accurate variant calls. A recent study [15] found that for a variety of variant calling pipelines combined with the application of a small handful of filters, variant call sets from high-coverage sequencing data can have less than one error in 100,000 bases, on average, and that the different software pipelines show high agreement in these call sets after filtering. The filters applied included the removal of variant calls overlapping low-complexity regions (LCRs) and the removal of variants in regions with significantly large read depth. In addition, the study found that the inclusion of "decoy" sequences representing regions missing from the genome reference prevented misalignments that result in false variant calls. This provides additional evidence that the use of quality score and genomic location filters can effectively replace GATK's approach using machine learning models to recognize features of true variants.

1.4 Software Pipelines and Reproducibility

Once software tools are chosen, the user needs to run them all in the appropriate sequence, sometimes merging multiple datasets from the same sample or merging multiple samples' variant sets for comparison. Using "pipeline" software to simplify sample processing enhances the reproducibility of analyses, and allows bioinformaticians to track and check job outcomes, log software versions, and avoid errors through automation.

Several publicly available pipelines exist for this purpose. The Galaxy Project [16] offers a website (https://usegalaxy.org) as well as the ability to install the Galaxy software on a local computer cluster. The "blue collar bioinformatics" pipeline available at https://github.com/chapmanb/bcbio-nextgen also offers the ability to run on a local cluster or on the cloud, but installing and running the pipeline still require considerable expertise in computer programming. In this chapter, we will not detail the use of these pipeline packages, but instead, describe a protocol leading the user through the separate steps that might be executed within them. In this way, the reader can expect to gain a better understanding of the software being used and the steps being carried out.

2 Materials

This protocol outlines the steps required to analyze sequence reads, either single- or paired-end, from Illumina GAII or HiSeq sequence analyzers, in order to create files in VCF format containing genotypes of SNVs and small insertions and deletion variants. The required materials to follow this protocol are:

1. A high-performance computer or computer cluster running the Linux operating system (see below for hardware requirements).
2. Sequence reads in FASTQ format for samples of interest. Alternatively, Subheading 3.6.1 gives instructions for downloading a sample dataset containing sequence from the National Center for Biotechnology Information (NCBI).
3. Installed software for sequence analysis:
   - SRA Toolkit—http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software (optional, for use in obtaining sample data from the NCBI Sequence Read Archive)
   - Samtools—http://www.htslib.org
   - bamUtil—http://genome.sph.umich.edu/wiki/BamUtil (only required for PCR duplicate removal if the installed version of samtools is 1.0 or later)
   - Novoalign—http://www.novocraft.com/products/novoalign/

- Variant and genotype calling software (one of the following):
  - Freebayes—https://github.com/ekg/freebayes
  - Bam2mpg—http://research.nhgri.nih.gov/software/bam2mpg/
- tabix—http://sourceforge.net/projects/samtools/files/tabix/
- ANNOVAR—http://www.openbioinformatics.org/annovar/
- VarSifter—https://github.com/teerjk/VarSifter

Familiarity with running programs within a Linux environment and experience with a scripting language such as Perl, Python, or Ruby are helpful, although not strictly necessary. While it is possible to perform these analyses on a single computer with at least eight gigabytes of physical memory available for processing, samples are processed faster using a high-performance computer cluster equipped with job scheduling software, so that jobs can be parallelized to decrease throughput time for each sample. In addition, considerable amounts of disk space are required. For extended analyses, 50 or more gigabytes per mammalian sample for whole exomes, and on the order of a terabyte per sample for mammalian whole genomes will provide a generous amount of room, especially when running these programs for the first time, but eventually, deletion and compression of unnecessary files used early in the process can reduce these disk requirements considerably.

In addition, example processing scripts and updates regarding software usage changes are maintained in the github repository available at http://github.com/nhansen/MMBVariantCalling.

3 Methods

Figure 1 shows the steps to be followed in this single-sample variant calling pipeline. First, sequence reads are aligned to a set of reference sequences which define the loci that will be interrogated for variants. The alignment software (novoalign) produces alignment files in "BAM" format [17], detailing each read's mapped location within the reference, as well as the base-by-base alignment of reads against the reference sequence itself. Pairs of library inserts with common aligned reference endpoints are assumed to come from multiple PCR copies of the same original library fragment ("PCR duplicates"), so in order to obtain a more accurate sampling of the chromosomes present in the DNA sample, reads representing the extra copies are removed from the dataset. Next, a variant caller is run on the alignment file to predict reference loci where at least one chromosome contains sequence different from the reference

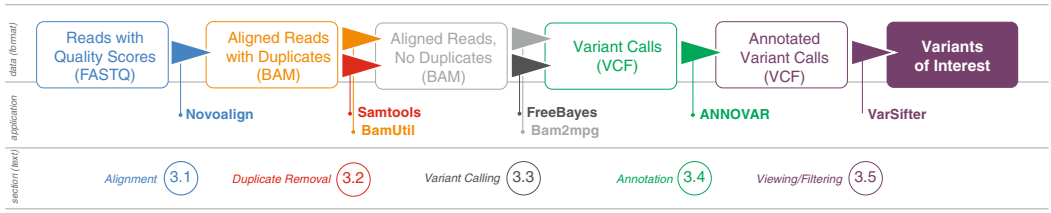

Fig. 1 Diagram of the variant calling workflow. *Circled numbers* beneath each step refer to the section of the chapter in which they are described

Table 1
Approximate memory and CPU requirements for programs run in this protocol

	Whole exome		Whole genome	
Program	**Memory**	**CPU time[a]**	**Memory**	**CPU time[a]**
Novoalign[b]	8 Gb	200 h	8 Gb	2000 h
Duplicate removal	< 1 Gb	40 min	< 1 Gb	6 h
Freebayes	< 100 Mb	5 h	<200 Mb	3 h
Bam2mpg[c]	8 Gb	18 h	8 Gb	300 h
ANNOVAR	<1 Gb	5 min	< 1 Gb	3 h

Whole exome statistics were measured using the paired-end dataset with accession SRR1611184 available from the NCBI Sequence Read Archive

[a] All reported CPU times are approximate

[b] Novoalign processing times are total time. Licensed versions of novoalign, when run on *n* CPUs, have total elapsed times *n* times faster than reported CPU time

[c] On whole genome sequences, bam2mpg should be run on separate regions of up to ten million bases in size and results recombined to conserve memory

sequence. These differences are generally SNVs and DIVs, but can also be complex substitutions which change multiple adjacent bases in a way that is difficult to ascribe to simple mutation events. All discovered variant alleles are typically reported, along with the genotypes of the sample being analyzed, in VCF format [18]. VCF files are then annotated to classify variants according to their locations within genes (5′ or 3′ untranslated region, coding sequence, or intronic sequence) and their effect on the translated protein (synonymous, nonsynonymous, stop-gain, stop-loss, frameshift, or splicing) for protein-coding genes [19]. Finally, annotated variants in VCF format can be viewed and filtered using specialized viewing software [20].

When executing each of the steps detailed in the sections that follow, attention should be paid to the estimated required computer resources (both physical memory and CPU time) provided in Table 1, and longer running jobs should be divided into multiple shorter jobs in the ways described within some sections of the text.

3.1 Alignment of Reads

3.1.1 FASTQ Files

Generally, unaligned next generation sequencing data are stored in FASTQ format. FASTQ-formatted files have four lines per sequence read, containing

1. The read name preceded by the character "@".
2. The sequence of the read, directed from 5′ to 3′.
3. A line beginning with the character "+".
4. A line containing ASCII-encoded quality scores for each base.

If a sample's library or libraries were sequenced in a paired-end run, there should be two FASTQ files for each sample lane, containing entries for read 1 or read 2 of each pair, and each file's read entries should appear in the same order as the other's.

Information on obtaining FASTQ-formatted sequence data from the NCBI's Sequence Read Archive (SRA) can be found in Subheading 3.6.1.

3.1.2 Running Novoalign

Before running novoalign on sequences from a new species, an index of the reference genome for that species must first be created using the "novoindex" command. Steps for obtaining a reference file and preparing a genome index for novoalign are described in Subheadings 3.6.2 through 3.6.4. Once these steps have been completed, the resulting index file, which has the file extension ".ndx", can be used to align any set of FASTQ-formatted reads against that particular genome reference sequence.

Novoalign can be run as licensed or unlicensed software. The program will search for its license file in all directories in the user's Linux path, as well as the directory in which the executable itself resides. Obtaining a paid license for novoalign allows the user to read FASTQ files that have been compressed with the "gzip" utility, and also allows novoalign to be run on multiple threads of a single computer, which can dramatically decrease the storage requirements and elapsed run time to process whole exome and whole genome sequences. Alternatively, FASTQ files can be split into multiple smaller files, and run separately on different processors, after which the resulting BAM files can be merged.

For example, if the FASTQ files are named "SRR1611184_1.fastq.gz" and "SRR1611184_2.fastq.gz", novoalign is run with the command:

```
# Run novoalign, piping output to samtools to make a sorted BAM file:
novoalign -F STDFQ -o SAM -d hg38.ndx -f SRR1611184_1.fastq.gz
  SRR1611184_2.fastq.gz | samtools view -uS - |
  samtools sort - NA12878
```

which will produce a BAM-formatted file with the name "NA12878.bam", containing reads aligned to the hg38 reference.

When using an unlicensed copy of novoalign, or when analyzing a large dataset such as the one obtained from whole genome sequencing, it is usually better to split large FASTQ files into multiple smaller ones and then run novoalign on each smaller FASTQ file separately. A script "split_fastq.pl" is provided for this purpose in the scripts subdirectory of the github repository at http://github.com/nhansen/MMBVariantCalling.

```
# Split fastq file obtained from NCBI into 40 separate smaller files
# --nogzip option will create uncompressed files and should only
# be used when running novoalign without the paid license
split_fastq.pl SRR1611184_1.fastq.gz 40 --nogzip --dir split_40/
split_fastq.pl SRR1611184_2.fastq.gz 40 --nogzip --dir split_40/
```

Once the read sequences for a sample are in multiple FASTQ files, novoalign can be run on each pair of FASTQ files as described above, and the resulting BAM files can be merged using samtools:

```
samtools merge NA12878.merge.bam NA12878.1.bam NA12878.2.bam [...]
```

Note that in the samtools merge command, the output file precedes the BAM files you are merging in the list of arguments.

3.2 Removal of PCR Duplicates

Since variant calling software will generally assume read pairs represent an unbiased random sampling of DNA from cells, it is important to remove extra copies of PCR-amplified DNA molecules, or "PCR duplicates," from the BAM file before calling variants. Prior to the release of samtools version 1.0, the most common way to remove duplicate reads was to use samtools's "rmdup" command, but samtools versions 1.0 and later no longer have a working rmdup command, so "option 2" below describes the use of the bamUtil command "dedup" as an alternative.

3.2.1 Duplicate Removal Option 1

If you are running a version of samtools prior to version 1.0 (e.g., version 0.1.19), run the available rmdup command:

```
# remove PCR duplicates using samtools
samtools rmdup NA12878.merge.bam NA12878.final
```

This will create a new BAM file with duplicates removed called "NA12878.final.bam". If libraries were sequenced with only a single end (not paired), one should include the "-s" option to samtools rmdup, and note that duplicate removal will generally remove more reads than would be the case for paired ends.

3.2.2 Duplicate Removal Option 2

If earlier versions of samtools are not available, one can use the bamUtil package, available at http://genome.sph.umich.edu/wiki/BamUtil, to remove PCR duplicates. Once bamUtil has been installed, users can run:

```
# remove PCR duplicates using bamUtil
bam dedup --in NA12878.merge.bam --out NA12878.final.bam --rmDups
```

After either Option 1 or Option 2 has been completed, the final BAM file "NA12878.final.bam" should be indexed for fast access to genomic regions of interest.

```
# Index BAM file for fast retrieval of genomic regions
samtools index NA12878.final.bam
```

3.3 Variant and Genotype Calling

Both Freebayes and bam2mpg can be used to obtain accurate variant calls from duplicate-free BAM files. Table 1 shows that Freebayes requires far less CPU time and physical memory than bam2mpg, but Freebayes also provides less informative genotype quality scores for sites where the program predicts no variation. This latter point is critical when the goal is to determine whether a sample is truly free of variation in a region without a variant call, or whether more sequence coverage is needed to "test" a sample for a particular variant. Instructions are given below for creating BED files of confidently covered regions with bam2mpg.

3.3.1 Variant Calling with Freebayes

Freebayes [11] is a haplotype-based caller for producing VCF-formatted single nucleotide and small deletion/insertion variant calls from BAM-formatted sequences. Run Freebayes on your final BAM file with the command:

```
freebayes --genotype-qualities -f hg38.mfa NA12878.final.bam >
  NA12878.vcf
```

Freebayes variants represent local haplotypes, so two side-by-side SNVs will be represented in the VCF file by a phased "multi-nucleotide polymorphism," or MNP. Phased MNPs can be annotated more accurately than side-by-side SNVs.

3.3.2 Variant Calling with bam2mpg

Another accurate, yet easy to use, variant caller is bam2mpg [12]. In addition to calling variants in VCF format, bam2mpg also assigns a genotype quality score to genomic sites that are called homozygous reference. In this way, it is possible to create BED-formatted files delineating regions that have adequate sequence coverage to determine with high probability that no variant is present at a locus. This can be valuable in calculating accurate population variant allele frequencies for rare variants.

To generate VCF files NA12878.snv.vcf and NA12878.div.vcf with bam2mpg, run:

```
bam2mpg --bam_filter '-q31' --qual_filter 20 --only_nonref
  --snv_vcf NA12878.snv.vcf --div_vcf NA12878.div.vcf hg19.mfa
  NA12878.final.bam
```

To obtain a BED file of regions determined to have adequate depth of coverage for calling diploid variants, one must not use the

"–only_nonref" option when running bam2mpg. This will create larger files than the "variant only" VCF files (on the order of 5 gigabytes for a whole exome sample, or 15 gigabytes for a whole genome sample), but the resulting "MPG" formatted output file can be used as input to the script "mpg2bed.pl" to create a BED-formatted file of adequately covered regions:

```
bam2mpg --bam_filter '-q31' --qual_filter 20 --mpg NA12878.mpg.out.gz
  hg19.mfa NA12878.final.bam
mpg2bed.pl --minscore 10 sample.mpg.out.gz > NA12878.mpg.bed
```

3.3.3 Compression and Random Access with VCF Files

Especially if they contain invariant loci, VCF files can be large, so to save disk space, they should be compressed with the block compression tool "bgzip," which is available as part of the "tabix" package. Since bgzip retains block structure in its compression of VCF files, it is also possible to index files of variants, and quickly access entries in arbitrary parts of the genome with a handful of simple commands:

```
bgzip NA12878.vcf # create the compressed file sample.vcf.gz
tabix -p vcf NA12878.vcf.gz # index the VCF file
tabix NA12878.vcf.gz chr13:32889617-32973809 # retrieve BRCA2 variants
```

3.4 Annotation of VCF Files

The genomic position and exact sequence change represented by a variant is rarely informative all by itself. To evaluate variants and gain insights into their impact on a gene's function, it is helpful to view these variants in the context of various gene annotations. The program ANNOVAR [19] can be used to compare variant locations to annotation sets downloaded from the ANNOVAR website and the University of California at Santa Cruz's (UCSC) Genome Browser [21]. Specifically, lists of "gene-based" (describing variants' impact on gene models from NCBI's RefSeq, UCSC's known gene, or the ENSEMBL gene set) and "filter-based" (annotating specific variants with a wide variety of allele frequency, pathogenicity, and conservation-related measures) can be downloaded and used to add annotations to VCF files for viewing and filtering.

For example, to add gene annotation from the National Center of Biotechnology's "RefGene" database to a variant VCF file, first download the "refGene" database from the ANNOVAR website:

```
# Download RefSeq gene annotations
annotate_variation.pl --buildver hg38 -downdb -webfrom annovar refGene
  humandb/
```

Annotating variants with allele frequencies from different populations can also be helpful in determining whether a particular variant might be pathogenic. So also download the Exome Aggregation Consortium's "ExAC" database:

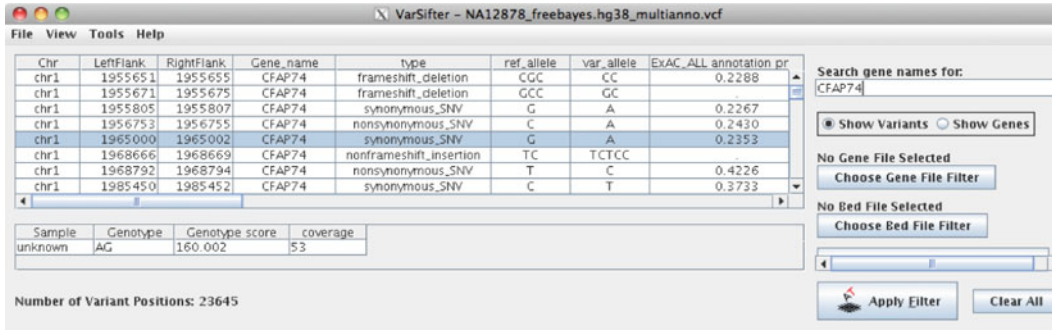

Fig. 2 VarSifter display of NA12878's CFAP74 gene variants

```
# Download ExAC variant annotations
annotate_variation.pl --buildver hg38 -downdb -webfrom annovar exac03
  humandb/
```

Finally, run ANNOVAR's table_annovar.pl script using these databases to add INFO fields to the variant VCF file:

```
# Add RefGene and ExAC annotations to the VCF file
table_annovar.pl --vcfinput NA12878.rmdup.freebayes.vcf.gz humandb/
  -buildver hg38 -remove -protocol refGene,exac03
  -out NA12878_freebayes -operation g,f humandb/
```

This will create a VCF file called NA12878_freebayes.hg38_multianno.vcf, in which all of the Freebayes variants are annotated with their RefSeq locations (on a "gene" basis, as indicated by the -operation "g" value) and with population frequencies from the Exome Aggregation Consortium's ExAC database on a "filter" basis, suitable for viewing and filtering with VarSifter.

3.5 Viewing and Filtering Variants

Figure 2 shows the graphical interface of the viewing tool VarSifter [20]. Using checkboxes and custom queries, VarSifter makes it possible to filter annotated VCF files based on variants' impact on protein sequence, population allele frequency (e.g., from large sequencing projects such as ExomeGO), consistency with known Mendelian inheritance schemes, or potential to be pathogenic from databases like HGMD [22].

To open a VCF file in VarSifter, type:

```
java -Xmx1G -jar VarSifter<version>.jar
```

where "version" is the version number of VarSifter installed, and the memory requirement specified after "-Xmx" may need to be increased for larger VCF files. Open the VCF file to be viewed, and select the name of the INFO fields containing the mutation type and gene name (for RefSeq annotations added by ANNOVAR, these fields have identifiers "ExonicFunc.RefGene" and "Gene.RefGene," respectively). VarSifter then allows the user to select which INFO annotations in the VCF file to display.

To filter variants, check boxes in the "Include" section at the upper right of the screen, and select the "Apply Filter" button. In addition, regular expressions for particular genes can be entered into "Search gene names for" box, and the "Custom query" option under the "View" menu allows users to create their own filters based on genotypes, variant scores, and numerous other variant properties.

To filter based on genomic region, download the BED file of low-complexity repeat locations used in reference [15]:

```
wget https://github.com/lh3/varcmp/blob/master/scripts/LCR-hs38.bed.gz?raw=true
```

An option to filter based on an uncompressed BED file's genomic locations using the "Choose BED File Filter" button allows the user to view only the variants which lie within the specified regions. The file of low copy repeats, above, can be used to find variants which are likely false positives.

3.6 Supplementary Protocols

3.6.1 Obtaining Sequence Data from NCBI

The NCBI SRA acts as a repository for sequence data. After a user locates a sequence dataset of interest, NCBI recommends the use of its "SRA Toolkit" to download data. With the SRA Toolkit and Aspera Connect/ascp installed, a sample dataset for this method can be downloaded using the "prefetch" command, and converted to FASTQ format using fastq-dump:

```
# Download the .sra file and necessary NCBI reference sequences into
# your home directory's "ncbi" subdirectory
prefetch SRR1611184
# Convert .sra file into fastq format, separating read 1 and read 2
# into two files, and gzip'ing to save space
fastq-dump -I --gzip --split-3 SRR1611184
```

If a novoalign license has not been purchased, the "–gzip" option should be omitted from the fastq-dump call, and larger, uncompressed FASTQ files will be generated.

3.6.2 Downloading a Genome's Reference Sequence

The most convenient way to obtain a reference sequence for an organism is often to download FASTA files from the University of California, Santa Cruz's Genome Browser. Since these files can be moderately large (e.g., 1 gigabyte for a gzipped file of the sequences from NCBI's Build GRChr38 of the human genome), it is best to use the rsync command:

```
# Find reference sequences for your genome build of interest at
# ftp://hgdownload.cse.ucsc.edu/goldenPath
# rsync places gzipped fasta file in current directory
rsync -a -P
  rsync://hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz ./
```

3.6.3 Removing Alternate Haplotypes and Masking Pseudo-Autosomal Regions in the Reference

Newer builds of the human reference sequence (e.g., hg19, hg38) contain "alt" sequences containing alternate haplotypes of regions of the human genome that are highly polymorphic in the population. Although some aligners, like bwa-mem, will consider these alternate haplotype sequences differently in their alignment algorithms, variant and genotype callers typically don't know how to handle these alignments yet, so the value in aligning to alternate haplotypes is still unclear, and entries for alternate haplotypes should usually be removed from the reference FASTA file before formatting an alignment index. If a user is particularly interested in one of the regions for which common alternate haplotypes exist, he or she should explore possibilities for custom, region-specific analyses, which are not covered in this protocol.

In addition to removing alternate haplotypes from the reference sequence, it is also helpful to mask the pseudo-autosomal regions, or "PARs," of the organism's sex chromosomes. These PARs are identical for the X and the Y chromosomes, and like the alternate haplotypes above, they can confuse alignment software because they appear to be exact repeats. For human reference builds, the coordinates of these regions are available from the UCSC Genome Browser.

A simple Perl script for removing alternate haplotype sequences from a reference assembly and masking the PARs on chromosome Y for human is included in the "scripts" subdirectory of the github repository at http://github.com/nhansen/MMBVariantCalling, and can be run by typing:

```
# Specify output fasta as an "mfa", or "multi-FASTA" file:
prepare_reference_fasta.pl --build hg38 --input hg38.fa.gz
  -output hg38.mfa
```

3.6.4 Formatting a Novoalign Database

Before running novoalign, it is necessary to first have a FASTA-formatted database of reference (i.e., chromosome) sequences, against which the program will align the reads. Once a particular build of the organism's genome has been selected (e.g., hg19, also known as "GRCh37"), a FASTA file of chromosome sequences can be downloaded from a resource like the UCSC Genome Browser. If a reference database has not been created ahead of time, it can be created using the "novoindex" command of novoalign:

```
# Format an hg38 index for novoalign (requires 8Gb of memory
# for a human genome)
novoindex -n hg38 hg38.ndx hg38.mfa
```

4 Notes

- The program novoalign is available with a free license. However, with a paid license, it can be run using multiple threads, which can considerably speed up processing time, especially for whole

genome sequences. Without a paid license, it is also necessary to use uncompressed FASTQ files as input to novoalign, which necessitates a greater use of disk space.

- When running novoalign on paired-end reads, the reads in the FASTQ file for read number one must be in the same order as their pairs in the FASTQ file for read number two. Most paired FASTQ files are correctly ordered, but a mismatch in ordering will result in an error of the form "Error: Read headers do not match at Record #...".
- If reads are greater than 100 bases in length, novoalign can be run with the option "-t 400" to decrease run time without significant loss of alignment accuracy.
- The program Freebayes should be installed using "git clone" with the recursive option, which will download necessary external repositories that are not included in the distributed tarball.

Acknowledgments

Artwork for Fig. 1 was provided by DXYN Studios. This work was supported by the Intramural Research Program of the National Human Genome Research Institute, National Institutes of Health. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Human Genome Research Institute or the National Institutes of Health.

References

1. International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome. Nature 409 (6822):860. http://dx.doi.org/10.1038/35057062
2. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, Hall KP, Evers DJ, Barnes CL, Bignell HR, Boutell JM, Bryant J, Carter RJ, Keira Cheetham R, Cox AJ, Ellis DJ, Flatbush MR, Gormley NA, Humphray SJ, Irving LJ, Karbelashvili MS, Kirk SM, Li H, Liu X, Maisinger KS, Murray LJ, Obradovic B, Ost T, Parkinson ML, Pratt MR, Rasolonjatovo IMJ, Reed MT, Rigatti R, Rodighiero C, Ross MT, Sabot A, Sankar SV, Scally A, Schroth GP, Smith ME, Smith VP, Spiridou A, Torrance PE, Tzonev SS, Vermaas EH, Walter K, Wu X, Zhang L, Alam MD, Anastasi C, Aniebo IC, Bailey DMD, Bancarz IR, Banerjee S, Barbour SG, Baybayan PA, Benoit VA, Benson KF, Bevis C, Black PJ, Boodhun A, Brennan JS, Bridgham JA, Brown RC, Brown AA, Buermann DH, Bundu AA, Burrows JC, Carter NP, Castillo N, Chiara M, Catenazzi E, Chang S, Neil Cooley R, Crake NR, Dada OO, Diakoumakos KD, Dominguez-Fernandez B, Earnshaw DJ, Egbujor UC, Elmore DW, Etchin SS, Ewan MR, Fedurco M, Fraser LJ, Fuentes Fajardo KV, Scott Furey W, George D, Gietzen KJ, Goddard CP, Golda GS, Granieri PA, Green DE, Gustafson DL, Hansen NF, Harnish K, Haudenschild CD, Heyer NI, Hims MM, Ho JT, Horgan AM, Hoschler K, Hurwitz S, Ivanov DV, Johnson MQ, JamesT, Huw Jones TA, Kang GD, Kerelska TH, Kersey AD, Khrebtukova I, Kindwall AP, Kingsbury Z, Kokko-Gonzales PI, Kumar A, Laurent MA, Lawley CT, Lee SE, Lee X, Liao AK, Loch JA, Lok M, Luo S, Mammen RM, Martin JW, McCauley PG, McNitt P, Mehta P, Moon KW, Mullens JW, Newington T, Ning Z, Ling Ng B, Novo SM, O'Neill MJ, Osborne MA, Osnowski A, Ostadan O, Paraschos LL, Pickering L, Pike AC, Pike AC, Chris Pinkard D, Pliskin DP, Podhasky J, Quijano VJ, Raczy

C, Rae VH, Rawlings SR, Chiva Rodriguez A, Roe PM, Rogers J, Rogert Bacigalupo MC, Romanov N, Romieu A, Roth RK, Rourke NJ, Ruediger ST, Rusman E, Sanches-Kuiper RM, Schenker MR, Seoane JM, Shaw RJ, Shiver MK, Short SW, Sizto NL, Sluis JP, Smith MA, Ernest Sohna Sohna J, Spence EJ, Stevens K, Sutton N, Szajkowski L, Tregidgo CL, Turcatti G, Vandevondele S, Verhovsky Y, Virk SM, Wakelin S, Walcott GC, Wang J, Worsley GJ, Yan J, Yau L, Zuerlein M, Rogers J, Mullikin JC, Hurles ME, McCooke NJ, West JS, Oaks FL, Lundberg PL, Klenerman D, Durbin R, Smith AJ (2008) Accurate whole human genome sequencing using reversible terminator chemistry. Nature 456(7218):53. doi:10. 1038/nature07517

3. The 1000 Genomes Project Consortium (2012) An integrated map of genetic variation from 1,092 human genomes. Nature 491 (7422):56. http://dx.doi.org/10.1038/nature11632
4. Hoadley KA, Yau C, Wolf DM, Cherniack AD, Tamborero D, Ng S, Leiserson MD, Niu B, McLellan MD, Uzunangelov V, Zhang J, Kandoth C, Akbani R, Shen H, Omberg L, Chu A, Margolin AA, van't Veer LJ, N. Lopez-Bigas, Laird PW, Raphael BJ, Ding L, Robertson AG, Byers LA, Mills GB, Weinstein JN, Waes CV, Chen Z, Collisson EA, Benz CC, Perou CM, Stuart JM (2014) Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. Cell 158(4):929. doi:http://dx.doi.org/10.1016/j.cell.2014.06.049. http://www.sciencedirect.com/science/article/pii/S0092867414008769
5. Nielsen R, Paul JS, Albrechtsen A, Song YS (2011) Genotype and SNP calling from next-generation sequencing data. Nat Rev Genet 12 (6):443. doi:10.1038/nrg2986
6. Li H, Durbin R (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics 26(5):589. doi:10.1093/bioinformatics/btp698
7. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10(3):R25. doi:10.1186/gb-2009-10-3-r25
8. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J (2009) SOAPZ: an improved ultrafast tool for shot real alignment. Bioinformatics 25(15):1966. doi:10.1093/bioinformatics/btp336
9. Smith TF, Waterman MS (1981) Identification of common molecular subsequences. J Mol Biol 147(1):195
10. Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443
11. Garrison E, Marth G (2012) Haplotype-based variant detection from short-read sequencing. arXiv:1207.3907V2 [q-bio.GN]. http://arxiv.org/abs/1207.3907
12. Teer JK, Bonnycastle LL, Chines PS, Hansen NF, Aoyama N, Swift AJ, Abaan HO, Albert TJ, Margulies EH, Green ED, Collins FS, Mullikin JC, Biesecker LG (2010) Systematic comparison of three genomic enrichment methods for massively parallel DNA sequencing. Genome Res 20(10):1420. doi:10.1101/gr.106716.110
13. Ewing B, Green P (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res 8(3):186
14. Van der Auwera GA, Carneiro MO, Hartl C, Poplin R, Del Angel G, A. Levy-Moonshine, Jordan T, Shakir K, Roazen D, Thibault J, Banks E, Garimella KV, Altshuler D, Gabriel S, DePristo MA (2013) From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics 11(1110):11.10.1. doi:10.1002/0471250953. bi1110s43
15. Li H (2014) Toward better understanding of artifacts in variant calling from high-coverage samples. Bioinformatics 30(20):2843. doi:10.1093/bioinformatics/ btu356
16. Goecks J, Nekrutenko A, Taylor J (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11(8):R86. doi:10.1186/gb-2010-11-8-r86
17. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/map format and SAM tools. Bioinformatics 25 (16):2078. doi:10.1093/ bioinformatics/ btp352
18. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R (2011) The variant call format and VCF tools. Bioinformatics 27(15):2156. doi:10.1093/bioinformatics/btr330
19. Wang K, Li M, Hakonarson H (2010) ANNOVAR: functional annotation of genetic variants from high-through put sequencing data. Nucleic Acids Res 38(16):e164. doi:10.1093/nar/gkq603
20. Teer JK, Green ED, Mullikin JC, Biesecker LG (2012) Var Sifter: visualizing and analyzing exome-scale sequence variation data on a

desktop computer. Bioinformatics 28(4):599. doi:10.1093/bioinformatics/btr711

21. Karolchik D, Hinrichs AS, Furey TS, Roskin KM, Sugnet CW, Haussler D, Kent WJ (2004) The USSC Table Browser data retrieval tool. Nucleic Acids Res 32(Database issue): D493. doi:10. 1093/nar/gkh103

22. Stenson PD, Mort M, Ball EV, Shaw K, Phillips A, Cooper DN (2014) The Human Gene Mutation Database: building a comprehensive mutation repository for clinical and molecular genetics, diagnostic testing and personalized genomic medicine. Hum Genet 133 (1):1. doi:10.1007/s00439-013-1358-4