HHS Public Access

Author manuscript

Nat Protoc. Author manuscript; available in PMC 2017 July 12.

Published in final edited form as:

Nat Protoc. 2016 December ; 11(12): 2529–2548. doi:10.1038/nprot.2016.150.

Indel variant analysis of short-read sequencing data with Scalpel

Han Fang[1,2,3], **Ewa A Bergmann**[4], **Kanika Arora**[4], **Vladimir Vacic**[4], **Michael C Zody**[4], **Ivan Iossifov**[1], **Jason A O'Rawe**[2,3], **Yiyang Wu**[2,3], **Laura T Jimenez Barron**[2,5], **Julie Rosenbaum**[1], **Michael Ronemus**[1], **Yoon-ha Lee**[1], **Zihua Wang**[1], **Esra Dikoglu**[4], **Vaidehi Jobanputra**[4,6], **Gholson J Lyon**[2,3], **Michael Wigler**[1], **Michael C Schatz**[1,7], and **Giuseppe Narzisi**[1,4]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA

[2]Stanley Institute for Cognitive Genomics, Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA

[3]Stony Brook University, Stony Brook, New York, USA

[4]New York Genome Center, New York, New York, USA

[5]Centro de Ciencias Genomicas, Universidad Nacional Autonoma de Mexico, Cuernavaca, Mexico

[6]Columbia University Medical Center, New York, New York, USA

[7]Department of Computer Science, Johns Hopkins University, Baltimore, Maryland, USA

Abstract

As the second most common type of variation in the human genome, insertions and deletions (indels) have been linked to many diseases, but the discovery of indels of more than a few bases in size from short-read sequencing data remains challenging. Scalpel (http://scalpel.sourceforge.net) is an open-source software for reliable indel detection based on the microassembly technique. It has been successfully used to discover mutations in novel candidate genes for autism, and it is extensively used in other large-scale studies of human diseases. This protocol gives an overview of the algorithm and describes how to use Scalpel to perform highly accurate indel calling from whole-genome and whole-exome sequencing data. We provide detailed instructions for an exemplary family-based *de novo* study, but we also characterize the other two supported modes of operation: single-sample and somatic analysis. Indel normalization, visualization and annotation

Correspondence should be addressed to G.N. (gnarzisi@nygenome.org).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

AUTHORS CONTRIBUTIONS G.N. is the lead developer of Scalpel. M.C.S. contributed to the development of Scalpel and wrote the microsatellite detector scripts. H.F. contributed to enhance Scalpel, compiled the Scalpel resource bundle and generated the figures in this article. E.D. and V.J. performed the Sanger validation. G.N., M.C.S. and M.W. conceived the Scalpel software project. G.N., M.C.S. and H.F. wrote the initial draft of the manuscript. M.C.S. is the principal investigator. All authors contributed to the development and approval of the final manuscript.

COMPETING FINANCIAL INTERESTS The authors declare competing financial interests: details are available in the online version of the paper.

of the mutations are also illustrated. Using a standard server, indel discovery and characterization in the exonic regions of the example sequencing data can be completed in ~5 h after read mapping.

INTRODUCTION

Reductions in the cost of whole-genome sequencing (WGS) and whole-exome sequencing (WES) are opening the door to affordable sequencing of patients and the development of precision medicine[1]. Historically, genomic studies have focused on single-nucleotide polymorphisms (SNPs) because of their high prevalence and the relative simplicity of detecting them[2]. However, recent advancements in sequencing technologies and computational methods have broadened the focus to include the role of insertion and deletion (indel) mutations. Indel mutations are defined by the addition or loss of one or more nucleotides of a DNA sequence. Frameshift mutations are a highly disruptive class of indel mutations that alter the reading frame of protein coding sequences[3] and have been strongly implicated in neurodevelopmental disorders, cardiovascular diseases, cancer and many other human diseases[4–7]. In evolutionary analysis, the role of indels has been established and emphasized in both eukaryotic and prokaryotic genomes[8,9]. In particular, studies have shown widespread occurrences of loss-of-function variants, especially indels, in protein-coding genes of humans, plants and other species[10–12].

Recent studies have shown that indels are ubiquitous in human genomes, causing a similar level of variation as SNPs in terms of the total number of base pair (bp) changes, but with great diversity in size[13]. As the second-largest group of variants in the human genome, there are typically more than 1 million small indels (in the size range from 1 to 100 bp) per diploid genome compared with the human reference, with the majority of them being <10 bp (refs. 14,15). The sizes of the indels in a human exome, relative to the reference, approximately follow a log-normal distribution, with similar numbers of insertions and deletions[16]. However, indels are still very challenging to detect for multiple reasons: (i) long indels, especially long insertions, are hard to detect with Illumina short reads, as there will be few bases of the read mapping to the reference; (ii) small-scale repeats, short tandem repeats (STRs) and near-identical repeats increase the degree of ambiguity for mapping and assembly[17] and confound the signature of the detected variant (Box 1); (iii) nonuniform coverage distribution, irregularity in capture efficiency in exome sequencing and targeted resequencing can easily increase the number of false-negative and false-positive calls, depending on the type of study (e.g., *de novo* versus single sample); and (iv) sequencing and PCR error, with PCR being especially error-prone around homopolymer A or T runs in the sequencing data[18], leading to the mapping/assembly problems described in (ii).

Box 1

Representing and annotating indels

Unlike SNPs, which are always represented with a unique genomic coordinate and base substitution, an indel can have an ambiguous representation. For example, if there is a 1-bp deletion in a long homopolymer (…AAAAAA…), deleting any A will give rise to the

same haplotype, but it will have a different position. A more complex example, which gives rise to two logically equivalent 3-bp deletions, is shown in the following figure:

where ALT1 represents a left-normalized indel (used by Scalpel), ALT2 represents a right-normalized indel and REF represents the reference sequence.

Note that two different 3-bp sequences can be deleted (GGA or AGG) at two different locations, generating the same alternative sequence. The solution to this ambiguity is to consistently left- or right-normalize the signature of the mutations. This operation consists of shifting the start position of the mutation to the left (or right) as long as the resulting sequence (after the deletion or insertion of the specific number of bases) is still the same as the one generated by the original mutation. Note that at the end of this process the new signature for the indel can have a new coordinate, as well as a new (deleted or inserted) sequence, but the size must remain the same as that of the original. For example, in the case of the previous 3-bp indel, the deletion is shifted to the left by two positions and a new 3-bp sequence (GGA) is deleted.

As different methods might report different signatures for the same indel, it is essential to consistently normalize the signature (typically left-normalization) when comparing indels called by different tools or when querying different databases (dbSNP, 1000G, OMIM and so on). Scalpel always returns a list of variants that are left-normalized. However, if it is unclear what representation has been used for a set of variants made by other callers, there are different tools available that can normalize a list of input variants, including ‘vt normalize’[48], ‘bcftools norm’[21] and ‘GATK LeftAlignIndels’[49]. Indel normalization is now becoming standard practice and widely used variant annotation software, such as Annovar[45], is now enforcing left-normalization as the default representation for indels. Updated variants databases (e.g., 1000 Genome Project, dbSNP and ExAC) were made available, and Annovar users are highly encouraged to left-normalize (using any of the previously listed tools) the variants before annotation. However, note that some of the databases use right-normalization or normalize relative to the sense of the transcript, so users are encouraged to refer to the documentation for each tool separately.

Annotation of variants is a common and crucial step that is necessary to identify potentially disease-relevant DNA mutations. Among the many different tools available for this task, three of the most widely used annotators are Annovar[45], SnpEff[50] and Variant Effect Predictor[51]. It has been shown that the choice of both the annotation software and transcript set can have a large impact on the classification of variants[52], suggesting the use of multiple annotation software when analyzing variants. The choice of the annotation program to use is also based on the ability to support different sets of transcripts, different cancer variants, multiple variant databases, integration with other tools and so on. Different annotators often produce different results, although here we opted to use Annovar because it is easy to use, the software is well documented, it is

continuously updated with the most recent databases and, as previously discussed, it supports left-normalization as the default indel representation.

A common approach for variant calling (of SNPs, indels or other variants) is to align reads one at a time to a reference genome, and to recognize when the reads disagree from the reference[19,20]. Although this approach works well for SNPs, it is less reliable for indel detection. For example, reads containing a long insertion will contain few bases matching the reference and will fail to map correctly. Although reads supporting a deletion consist of bases from the reference, it may be hard to unambiguously map both sides of the deletion. In both cases the aligner may ignore parts of the reads ('soft-clip') in order to place them on the reference or may fail to map them at all.

Earlier methods for indel detection relied on paired-end and split-read information as a computational signature for the presence of an indel. Some tools, such as GATK UnifiedGenotyper[19], SAMtools[21] and Dindel[20], use paired-end information to screen for indels where one read of a pair aligns well but the other read does not. After identifying such regions, the algorithms use a local realignment of the reads to detect indels, although the sensitivity declines quickly for mutations longer than 5 bp (ref. 18). By using split-read information, in which the alignment for an individual read is split into two segments spanning structural variation (SV) breakpoints, methods such as Pindel[22] and Splitread[23] are able to detect indels, especially deletions. Theoretically, this approach should be effective for deletions of any size, but the sensitivity is reduced because of the short read length of current sequencing technologies. Cortex, one of the first approaches for variant detection using whole-genome *de novo* assembly with de Bruijn graphs, was reported to overcome such issues caused by short reads and alignment artifacts[24]. However, in practice this method is less sensitive than expected, and accurate indel detection instead requires a fine-grained and localized analysis. Thus, in recent years, there has been much interest in developing specialized local assembly and microassembly methods[17].

One of the most sensitive and accurate approaches for indel detection from short read data is a microassembly algorithm, Scalpel. It was previously demonstrated to have substantially improved accuracy over eight algorithms, including GATK-HaplotypeCaller[25] (v3.0) and SOAP-indel[26] (v2.01), whereas other methods report a large number of false-negative calls[16]. In fact, Scalpel achieves very high accuracy (positive predictive value = 90%) of indel detection even on 30× WGS data (Fig. 1). In this protocol, we describe the use of Scalpel for indel detection from whole-genome and whole-exome capture sequencing experiments. We introduce three different modes of indel detection—*de novo*, somatic and single-sample—for different study designs. First, the *de novo* mode is useful for calling germline *de novo* variants in nuclear families up to four people. Second, the somatic mode is useful for identifying somatic changes within matched samples, especially tumor/normal pairs in cancer studies. Finally, the single-sample mode is useful for studies of a single proband.

Overview of Scalpel microassembly strategy

Scalpel is a computational tool specifically designed to detect indels in next-generation sequencing (NGS) data. Figure 2 outlines the main steps for the analysis of a sequencing data set using Scalpel. To highlight the main focus of this protocol, the left panel of Figure 2 depicts the specific scenario of detecting *de novo* indels in a quartet family composed of two parents and two children. We highly recommend reviewing the original Scalpel publication for a more extensive description of the method[16]. Here we describe the main ideas of the microassembly strategy used by Scalpel, as well as the strategies and filters that can be applied for optimizing the accuracy with different experimental designs or sequencing conditions, and describe the new developments since the original publication of the software (v0.1.1 beta).

Before running Scalpel, the sequencing reads (whole genome, whole exome or custom capture) must be aligned to a reference genome using a short-read-mapping algorithm such as BWA-MEM (http://bio-bwa.sourceforge.net/), similar to the steps used for SNP calling or other analyses. It is worth noting that computationally expensive procedures such as indel realignment and base quality recalibration are not necessary with Scalpel. Unlike in those analyses, the alignments are not directly used to find indels but instead are used to localize the analysis into computationally tractable regions. After alignment, Scalpel examines all the genomic regions provided in the input by the user in Browser Extensible Data (BED) format (right panel, Fig. 2). For each region, reads that align in the region or whose mates align in the region are extracted from the alignment and assembled independently of the reference using a de Bruijn assembly paradigm. If the size of a region is larger than the user-defined window-size parameter, a sliding-window approach will be applied to this target region based on the window-size and step-size parameters. In order to reduce the number of errors in locally highly repetitive regions, Scalpel automatically performs a local repeat analysis coupled with a self-tuning *k*-mer strategy that iteratively increases the *k*-mer size until a 'repeat-free' local assembly graph is built. In this context, a repeat-free graph is a graph without exact repeats, which would introduce cycles into the de Bruijn graph, as well as near-identical repeats (up to three mismatches by default). The advantage of this strategy is that each genomic window will be analyzed using an optimal *k*-mer specifically tuned according to its sequence composition. The graph is then exhaustively explored to identify end-to-end paths spanning the selected region. These paths, representing *de novo*-assembled sequences of the short reads, are then aligned to the reference window to detect candidate mutations using a sensitive gapped sequence aligner based on the Smith–Waterman algorithm.

Scalpel supports three modes of operation: single, *de novo* and somatic. In the single mode, Scalpel detects indels in one single data set (e.g., one individual exome or genome). In the *de novo* mode, Scalpel detects *de novo* indels in a quad family (father, mother, affected child and unaffected sibling). In the somatic mode, Scalpel detects somatic indels from the sequencing data coming from matched tumor and normal samples. In this protocol, we illustrate the use of Scalpel by focusing on the *de novo* mode; however, we describe the alternative operation modes in Box 2, including a discussion of the computational requirements for running Scalpel and how those differ between whole-genome and whole-

exome studies. Box 3 provides guidelines on how to export and filter the mutations based on coverage and quality scores and how those operations could affect sensitivity and specificity. Finally, Box 4 presents additional advanced operations for when using Scalpel for deep-sequencing projects or detection of longer indels (>100 bp).

Box 2

Alternative operation modes and computational requirements

Scalpel is designed for UNIX-type operating systems, and it provides a command-line interface. Users are expected to have a basic familiarity with operating in a UNIX environment. The discovery pipeline of Scalpel is executed from the command line via a master (perl) script called 'scalpel-discovery', and it requires a minimum number of parameters describing the input alignment files (BAM format), the reference genome (FastA format) and the target region (BED format) to analyze.

Scalpel supports three operation modes: single, somatic and de novo. The de novo mode is described and used in the procedure of this protocol. Here we report the basic usage and command-line parameters for the other two operation modes. To call variants on one single sample (e.g., a single-exome or single whole-genome data set), use the following command:

```
$ scalpel-discovery --single --bam file.bam --bed regions.bed --ref
genome.fa
```

where 'file.bam' is the bwa aligned BAM file of the reads (after sorting, indexing and PCR duplicates marking), 'regions.bed' contains the set of target regions in BED format (typically the list of exonic coding regions) and 'genome.fa' is the reference sequence in FastA format. It is important to provide to Scalpel the same reference file that was used to align the reads in the BAM file.

If calling variants on a tumor/matched normal pair, execute Scalpel as follows:

```
$ scalpel-discovery --somatic --normal normal.bam --tumor tumor.bam --bed
regions.bed --ref genome.fa --two-pass
```

where 'regions.bed' and 'genome.fa' are the same files as described before; 'normal.bam' and 'tumor.bam' are the BWA-aligned BAM files of the reads for the normal tissue and the tumor tissue sample, respectively. Also note the use of the '--two-pass' option, which enables Scalpel to perform a second round of indel verification on the candidate list of somatic mutations to reduce the number of false-positive calls. For example, in the case of a tumor/normal pair, a more sensitive analysis is performed on the normal sample to identify any signature of the candidate mutation in the tumor that was missed during the first pass of the analysis. We highly recommend using the '--two-pass' option for *de novo* and somatic studies. Exceptions to this rule are studies with extremely high coverage

(e.g., 1,000× or more) that can be obtained, for example, in panel studies of cancer samples, for which the use of the two-pass option is not required.

It is best to run Scalpel on a multicore computer with at least 64 GB of RAM. The relative computational requirements depend on the type of data being analyzed. For example, in the case of whole-exome analysis, ten CPUs and a minimum of 10 GB of RAM will be enough to perform the analysis in a few hours. In the case of a whole-genome study, in order to reduce the memory requirements, it is recommended that Scalpel be run on each chromosome separately and then that the lists of detected call sets be merged. Given the more uniform coverage distribution of whole-genome data and the increasing read length of Illumina technology, we also recommend increasing the window size (default 400 bp) to 600 bp or larger. For example, the following command can be used to call variants on chromosome 22 using ten CPUs:

```
$ scalpel-discovery --single --bam file.bam --ref genome.fa --bed
22:1-51304566 --window 600 --numprocs 10
```

Box 3

Exporting variants and filtering considerations

By default, Scalpel exports the list of detected indels in a VCF file within the selected output directory according to the default parameters. However, it is recommended that different filtering criteria be explored using the export tool ('scalpel-export'). The raw list of mutations detected by Scalpel is always available in a database within the output directory, which can be queried with the export tool using the following command:

```
$ scalpel-export [single|somatic|denovo] --db database.db --bed
regions.bed -ref genome.fa [options] > variants.vcf
```

All detected mutations are exported, but following the standard practices of the VCF format, high-quality mutations are flagged as 'PASS' in the 'FILTER' column. For non-PASS mutations, the 'FILTER' field contains the list of filters that were applied to the variant, explaining why the variant was considered to be of low quality. For example, in the VCF snippet below, the first indel is of high quality and is labeled as 'PASS', whereas the second indel does not satisfy the minimum Phred-scaled Fisher's exact test score requirement and it is flagged as 'LowFisherScore':

CHROM	POS	ID	REF	ALT	QUAL	FILTER	INFO	FORMAT normal	tumor
1	65099715	.	TC	T	10.16	PASS	.	GT:AD:DP 0/1:41,0:41	0/1:60,5:65
1	36884530	.	GA	G	8.77	LowFisherScore	.	GT:AD:DP 0/0:33,0:33	0/1:79,6:85

These filters can be further controlled using some of the command-line parameters available in the 'scalpel-export' tool. Like the scalpel-discovery command, the export tool requires the mode of operation to be specified according to type of study (single, somatic and *de novo*). Different parameters and filters are available for each operation mode; these are summarized in the following table:

Filter class	Single	*De novo*	Somatic
Indel size		--min-ins-size	
		--max-ins-size	
		--min-del-size	
		--max-del-size	
Allele counts	--min-alt-count	--min-alt-count-affected	--min-alt-count-tumor
	--max-alt-count	--max-alt-count-unaffected	--max-alt-count-normal
Variant allele fraction	--min-vaf	--min-vaf-affected	--min-vaf-tumor
		--max-vaf-unaffected	--max-vaf-normal
Statistics tests		--min-χ^2 score	--min-phred-fisher
		--max-chi2-score	
Coverage		--max-coverage-unaffected	--max-coverage-normal
		--min-coverage-affected	--min-coverage-tumor
		--max-coverage-affected	--max-coverage-tumor
		--min-coverage-unaffected	--min-coverage-normal

Here we discuss some of the most important ones and give recommendations on how to adjust them according to the type of study.

VAF and support

A high number of supporting reads and high VAF are typical signals of strong evidence for a variant. All three modes of operation provide parameters to control the thresholds used for the minimum number of supporting k-mers ('--min-alt-count', '--min-alt-count-tumor', '--max-alt-count-normal') and minimum allele fraction for the alternative allele ('--min-vaf', '--min-vaf-tumor', '--min-vaf-affected'); these are used to filter low-quality variants. Although default values are provided, optimal cutoffs for these numbers depend on the coverage available for the sample and the type of study (single, somatic and *de novo*). For example, as somatic calls are typically found at lower VAFs in the data, the user may need to adjust these parameters according to the level of purity, ploidy and clonality (if available) of the data.

Contamination in normal samples

The normal sample matched to the tumor is typically assumed not to be contaminated with the tumor sample. However, in practice, it is possible to have a very low level of contamination of the tumor in the normal sample as well. In this scenario, two parameters, '--min-alt-count-tumor' and '--min-vaf-tumor' (which by default assume no

contamination in the normal sample), can be used to allow mutations in the tumor that are also found in the normal sample at low allele fraction to be called as somatic.

Statistical test and scoring

For germline (inherited and *de novo*) mutations, the relative balance between the alternative and reference counts is estimated using the χ^2-test statistic. The cutoff used can be adjusted via the 'max-chi2-score' parameter. A larger value will increase sensitivity but produce a larger number of false positives. We recommend using χ^2 score 20 to export high-confidence indels. Differently from germline mutations, which are expected to be relatively balanced in their reference and alternative counts, somatic variants are usually out of balance due to several known characteristics of cancer data (e.g., ploidy, clonality and purity). The Fisher's exact test is generally used to determine whether there are nonrandom associations between the allele balances in the tumor and those in the normal samples. Scalpel internally scores the mutations by computing a Phred-scaled *P* value Fisher's exact test score, and the filtering cutoff can be adjusted via the '--min-phred-fisher' parameter. By default, this parameter is set to 10, but lower values will increase sensitivity at the cost of specificity.

Box 4

Advanced Scalpel operations

Variant calling is a computational step that is widely applied to a multitude of different projects and data sets, and, as expected, the default parameters of the tool cannot handle all situations equally well. Here we describe how to adjust the optional parameters in a few common scenarios.

Deep sequencing

Some projects require deep sequencing to allow in cancer, for example, the detection of low-allele-fraction mutations. With higher coverage, there is also enrichment of the errors in the data. In the most extreme scenarios of very deep sequencing experiments with 1,000-fold coverage or greater (e.g., cancer gene panels), these errors contribute to an increased complexity of the assembly graph to the point at which the associated region to assemble would be discarded. Thus, it becomes necessary to increase the minimum *k*-mer coverage used to remove low-coverage nodes (parameters '--lowcov' and '--covratio'). This can typically solve the problem by reducing the complexity to a level at which the graph can be efficiently analyzed. In addition, by default, regions that have >10,000-fold coverage are not processed. If higher coverage is expected, the maximum average coverage allowed per region must be adjusted accordingly ('--maxregcov' parameter).

Detecting very long (>100 bp) indels

There are cases in which researchers might have some evidence or prior knowledge about the presence of larger (>100 bp and up to 1 kb) indels in their data set. In this scenario, Scalpel can be used to genotype these loci for the presence of the mutations. Two parameters can be adjusted to handle and improve the sensitivity in such cases:

- '--coords': using this parameter the user can specify a list of selected coordinates to examine. The expected format is a tab-delimited list of chromosome names and positions.
- '--window': by default, the list of positions is analyzed using a window size of 400 bp. For indels approaching the window size or larger, it is necessary to increase this parameter to allow for enough unique sequence on both sides of the mutation. For example, in the case of a 400-bp insertion, we recommend using a window size of at least 600 bp so that 100 bp of unique sequence can be used to anchor the mutation to the reference on both sides.

Inspecting the assembly

In some special cases, the user may be interested in examining the final assembly generated by Scalpel. This information is stored internally by the program in the log files. By default, the log files are not saved in the output directory, as they can be extremely large in size, especially for whole-genome analysis. It is possible, however, to change this behavior by using the '--logs' option, but we recommend doing so only if a relatively short list of small regions is being analyzed. The file contains detailed information describing the different stages of the assembly. It is out of the scope of this article to describe the complete format of the log files; instead, we will focus on the section containing the final assembly and alignment of the region of interest. A typical alignment of the assembled sequence to the reference will look like the following one:

```
r': AGAGAGATTTTATTTATTTATTTATG----TATGTATTTATTTATTTATTTATTTACCTTGAGACAGAGT
p':
AGAGAGATTTTATTTATTTATTTATGTATTTATTTATTTATTTATTTATTTATTTACCTTGAGACAGAGT
43.3 [6.5 - 57.7]
d': ^^^^ x
>p_1:65100032-65100097_2 cycle: 0 match: 65 snp: 1 ins: 4 del: 0
65100058:----|TATT|
6.0|5|G|G 65100061:G|T|5.0|5|T|T
```

where r' is the reference sequence, `p'` is the sequence assembled by Scalpel, followed by information about the minimum and maximum coverage across the assembly, and `d'` is the alignment string showing the differences between the reference and the assembled sequence. In this case, the assembly contains a complex mutation composed of an insertion of four bases (`TATT`) together with single-base substitution (`G>T`). The last line reports the genomic coordinates of the region, followed by (i) the number of cycles detected (`cycle: 0`), (ii) the total number of matches to the reference (`match: 65`), (iii) the number of SNPs, insertions and deletions (`snp: 1 ins: 4 del: 0`) and (iv) a list of signatures describing each mutation. The signature starts with the position of the mutation, followed by ':' and a list of fields separated by the symbol '|' (e.g., `65100058:----|TATT|6.0|5|G|G`). In order, each field contains the following:

- Position

- Reference sequence
- Alternative sequence
- Average coverage supporting the mutation
- Minimum coverage supporting the reference
- Base pair preceding the mutation in the reference sequence
- Base pair preceding the mutation in the alternative sequence

In the first version of Scalpel (v0.1.1), all possible paths in the final graph were exhaustively examined using a breadth-first-search traversal approach. This strategy worked well for the majority of the human genome, with limited numbers of mutations, leading to the generation of one or two paths. However, this step is computationally expensive for a small number of regions with high levels of heterozygosity or higher sequencing error rates that generate exponentially many alternative paths, because the variants are not linked by the same *k*-mer. Since the release of a new version (v0.4.1), Scalpel instead enumerates only the minimum number of source-to-sink paths that cover every edge of the graph using a network flow approach. This strategy still detects all the mutations in the graph but substantially reduces the computational requirements by aligning to the reference a much smaller set of paths. Another important addition in the new version of Scalpel is the ability to better handle regions characterized by sudden drops in coverage. After removal of low-coverage nodes, the de Bruijn graphs associated with these regions can be disconnected into multiple connected components, which are now analyzed independently. Finally, the somatic mode of Scalpel is entirely new since the previous publications.

Comparison with other methods

Several hundred software packages are now available for analyzing WGS and WES data[27], including dozens of methods each for quality assessment, read alignment, variant identification, annotation and other applications. Most of the variant analysis packages are specialized for detecting one or a few types of mutations, because each type requires a different computational and statistical framework. For example, SNPs are generally found directly from read alignments, copy-number variations and SVs from read coverage and/or split-read approaches, whereas the leading methods for detecting indels rely on alignment or localized sequence assemblies.

A few other indel-finding software packages implement a localized sequence assembly strategy similar to the one used by Scalpel. These include GATK HaplotypeCaller[25], SOAPindel[26], Platypus[28], ABRA[29], TIGRA[30], DISCOVAR[31], Bubbleparse[32], Manta[33] and ScanIndel[34]. Although they all use a local read assembly step, these tools differ in how they explore the graphs and in their relative ability to handle repeat structures. Scalpel is unique because of on-the-fly repeat analysis that it uses to automatically optimize the parameters used for different regions of the genome, and the extensive set of filters that can be applied to correct for different sequencing conditions, among several other enhancements. In combination, these features enable Scalpel to accurately identify indel variants in diverse sequencing conditions and sequence contexts. Small-scale repeats are especially challenging

for most other indel-finding algorithms, although they are carefully detected and properly analyzed by Scalpel. We encourage the users to read the review on the challenge of small-scale repeats for indel discovery for a more in-depth discussion of the differences[17].

Most indel-finding tools, including Scalpel, have been designed to be general variant callers for detecting mutations across every region of the reference genome. However, some classes of indel, specifically the ones located within STRs, are known to be inherently more difficult to detect because of the high level of replication slippage events (e.g., homopolymers) associated with Illumina technology. Very few tools have been designed to specifically deal with the complexity of calling within STR regions. Users who specifically require the ability to call variants within STRs are strongly advised to use the following two tools: RepeatSeq[35] and lobSTR[36]. More recently, more complex classes of indels have been also discovered and analyzed in which a simultaneous deletion and insertion of DNA fragments of different sizes can co-occur at the same genomic location. A new tool, Pindel-C, has been specifically designed to handle these complex indels[37], and we encourage the user to use such tools for detecting complex indels in cancer-associated genes.

Limitations of the protocol and software

Scalpel provides several advantages over standard mapping approaches but, similar to any bioinformatics algorithm, it does not attempt to address all possible types or sizes of mutations at once. In our experiments, Scalpel was able to reliably detect deletions of up to 400 bp (including deletions of *Alu* mobile elements) and insertions shorter than 200 bp, but its sensitivity is reduced for longer indels, given the available read lengths (data not shown). Even within this size range, Scalpel—and all pipelines—has lower sensitivity for indels in low-coverage regions that are supported by very few reads. In the worst-case scenario, a combination of low coverage within a complex repeat region may require a *k*-mer size that is too large for assembling across the mutation, leading to false negatives. Phasing of the discovered mutations is not supported and, given the locality of the assembly, it would be possible to phase only mutations within the same window (400 bp, by default). Thanks to the new advances in long-molecule sequencing technologies (e.g., PacBio, 10× Genomics), in the near future it will be possible to combine such technologies for phasing mutations that are hundreds of kilobases to megabases apart.

For variant-calling purposes, it is ideal to have a high-quality reference genome available. This is also true for indel calling with Scalpel because assembly errors might incorrectly increase the number of variants, and the read localization will not be effective unless a complete representation of the genome is available. Users working with data from a genome without a reference should first generate a high-quality assembly using one of the several whole-genome assemblers[38,39]. This procedure can be easily adapted to work with a draft assembly, but no testing has been performed and the results could be unpredictable. Tumor/normal samples and samples from multiple family members can be analyzed together, but joint calling across a large number of samples is not supported by Scalpel, although population frequencies can be used to identify systematic sequencing errors. This protocol also assumes that sequencing was performed using the Illumina sequencing platform, including MiSeq, HiSeq 2000 and HiSeq X sequencers. Other sequencing technologies (e.g,

Ion Torrent, Sanger and SOLiD) can also be used for studies such as the one reported here, but the software pipeline used in this protocol does not support them. Finally, no graphical user interface is available for the steps performed in this protocol; all the operations are performed through the UNIX shell. Some of the tools used here, such as BWA and Picard Tools, are now available through cloud-based web interface systems such as Galaxy (https://usegal-axy.org/). We look forward to seeing Scalpel integrated into such systems in the near future.

Overview of the protocol

In the PROCEDURE, we present a step-by-step protocol for identifying *de novo* variants in a HapMap family from PCR-free Illumina HiSeq2000 data. Here, we provide an overview of using Scalpel to discover *de novo* and inherited indel mutations within a quad family of two parents and two children, one affected and one unaffected with a certain phenotype. It should be noted that internally within the algorithm, the two children are treated identically, which can support additional use cases. The input to the algorithm can be data from WGS, WES or targeted sequencing experiments. A two-pass search mode is used by Scalpel when calling *de novo* or somatic mutations. In the first pass, Scalpel identifies indels in each of the samples using parameters designed to balance between sensitivity and specificity. In the second pass, Scalpel performs a more sensitive search in the parents for the indels identified in the children to reduce false-positive *de novo* calls in regions of low coverage in the parents. We also show how to extract indel calls that fall into target regions and filter out false-positive calls with respect to their sequence composition and variant quality (Fig. 3). Finally, we present one of the available methods for annotating the mutations, to prioritize particularly any potential disease-related mutations. Although we use Scalpel for generating the indel calls, the protocol provides general guidelines for standard operations required to analyze and evaluate indel calls. We also illustrate several sources of indel calling errors, which could be introduced by library construction, sequencing or alignment. Whenever possible, visualization of data/results is performed using Integrative Genomics Viewer (IGV) alignments, and auxiliary scripts are provided for plotting size, allele fraction distribution and so on.

This protocol is based on the use of v0.5.3 of the Scalpel software. Users should keep in mind that the software is continuously under development, and some of the parameters, file names and output formats could change in the new releases of the software. The most recent version of the code and documentation is always available at http://scalpel.sourceforge.net. This protocol follows very closely a typical usage of the software; however, we recommend that the users perform the full procedure described herein before running the pipeline on their own data.

Experimental design

In this protocol, we use publicly available WGS data to detect and analyze indels within a family. However, when designing a new study, researchers are typically faced with the problem of choosing suitable sequencing and bioinformatics strategies to answer the relevant scientific questions. There are many factors that have a role in study design, including depth of coverage, read length, parameter tuning, WGS versus WES protocols, the use of PCR

amplification and cost per base pair. In this section, our goal is to provide some guidelines on the impact of such different experimental design choices on the sensitivity and accuracy of indel detection.

WGS versus WES—Although WES is a cost-effective approach to identifying genetic mutations within the coding region, it suffers from several major limitations due to a combination of coverage biases, low capture efficiency and errors introduced by PCR amplification. For example, an indel located near the end of a target region may not be well covered by sequencing reads, which limits detection ability. In addition, exome capture kits are typically designed to pull down a region of ~400 bp around an exon, which can limit detection of large indels within coding regions or near splice sites. On the other hand, albeit with higher cost, WGS comes with several benefits, including more uniform coverage, freedom from capture efficiency biases and the inclusion of the noncoding genome. In the context of detecting indels, it has been shown that the accuracy of indel detection with WGS data is much greater than that with WES data, even within the targeted regions[18]. Table 1 shows that the validation rate of WGS-specific indels is much higher as compared with that of WES-specific indels (84% versus 57%). Specifically, WGS has a unique advantage over WES in identifying many more indels that are longer than 5 bp; these were successfully confirmed by experimental validation (25 versus 1). When using WGS, it was estimated that 60× depth of coverage from the HiSeq platform would be needed to recover 95% of the indels detected by Scalpel. In particular, detection of heterozygous indels naturally requires deeper sequencing coverage relative to detection of homozygous indels (Fig. 4). WGS at 30× using the HiSeq platform is not sufficient for sensitive indel discovery, resulting in at least 25% false-negative rates for heterozygous indels. However, these requirements can rapidly change with the longer reads and lower error rates provided by newer instruments.

PCR-free protocols—PCR is a widely used and useful technique for amplifying DNA fragments of interest and for attaching various linkers or barcodes for sequencing. However, small amounts of contaminating material can also be amplified without discrimination. In addition, PCR amplification introduces errors during the library construction step, especially in regions near STRs such as homopolymer A or T runs. These types of errors are due to replication slippage events and result in high variability in the number of repeat elements (Fig. 5). It then becomes very difficult to distinguish true events at these loci from stutter errors. Moreover, as described in Box 1, candidate mutations within STRs can have an ambiguous signature. Therefore, for indel analysis, we recommend using PCR-free protocols, which can substantially reduce the number of errors around those loci. Moreover, as reported in this protocol, filtering based on the combination of alternative allele coverage (aac) and *k*-mer χ^2 score is an effective strategy for filtering out additional false positives without compromising much on sensitivity.

Population studies—Large-scale sequencing studies, involving hundreds or thousands of samples, are now becoming more and more widespread. Here we aim to introduce some of the advantages of having access to a collection of sequenced individuals. Even though Scalpel does not directly provide an application program interface for joint calling on more than four samples, we provide examples and a recommendation for how to take advantage of

such information if available. The basic idea is to aggregate all the genetic variants detected in the samples into a database framework with associated genotypes and genomic annotation. There are existing flexible systems, such as GEMINI[40], for exploring genetic variation for disease and population genetics. Analyzing the genetic code of a large cohort of individuals has the potential to shed light on the mechanisms underpinning complex diseases such as autism and schizophrenia. These studies are generally focused on the detection and analysis of rare variants that can explain the phenotype of the affected individuals.

The population frequency of such rare mutations is usually so low that it is obscured by the noise in the sequencing data, making any real biological signal undetectable. In these circumstances, the population can be used to devise effective filtering strategies. For example, in a large-scale autism study in which Scalpel was used[7], the population database was used to identify rare variants by filtering highly polymorphic loci with many more mutations than expected in the general population, as well as common variants using minor allele frequency cutoffs. Typically, variants for which the minor allele is present in >1% of a population are considered common. By removing these locations from the analysis, the biological signal started to emerge: an enrichment of frameshift *de novo* mutations in the affected child as compared with the unaffected sibling. The highly polymorphic regions were later found to enrich for homopolymers and other STRs, which are known to be more susceptible to sequencing errors. In the case of *de novo* studies, it is extremely unlikely that the same mutation is present as *de novo* in multiple individuals; in such a case, the population information can be used again to filter out such candidates as artifacts in the sequencing.

Cancer studies—Detection of somatic variation in tumor- normal matched samples is complicated by different factors such as ploidy, clonality and purity of the input material. Moreover, the sensitivity and specificity of any somatic mutation calling approach varies along the genome because of differences in sequencing read depths, error rates, variant allele fractions (VAFs) of mutations and so on. Accounting for all these variables poses a very complex and challenging problem. However, the proper filtering parameters can eliminate the majority of Scalpel's false-positive calls. For example, Figure 6 show the effects of different Phred-scaled Fisher's exact score cutoffs used for filtering of a pair of highly concordant primary and metastatic samples from Branon *et al.*[41]. Figure 6 demonstrates that indels with a Phred-scaled Fisher's exact score below 10 tend to have low variant allele fraction (VAF) values and are much more likely to be sequencing errors. In fact, the allele fraction of mutations exclusive to either the primary tumor or the metastasis is substantially lower with higher (more stringent) cutoffs. Similarly, the VAF distribution of the indels found only in the primary tumor shifts toward the expected distribution for these samples (with a peak at ~20%) as more conservative Fisher's exact test cutoffs are used. Not all errors are eliminated, however, especially in regions where very low support for a mutation in the normal sample or the tumor precludes the assembly of the reads. We are actively researching enhanced algorithms for such regions, including using a joint assembly within the same de Bruijn graph of the reads from both the tumor and the normal samples.

MATERIALS

EQUIPMENT

▲ **CRITICAL** Make sure that the listed software tools are available within your UNIX PATH setting. For example, if you have your tools installed in a '/path/to/your/tools' directory, you can update your PATH setting to include this path by using the following command:

```
% export PATH=/path/to/your/tools:$PATH
```

You can also add the command to your UNIX setting file ~/.bashrc.

- Data ▲ **CRITICAL** This protocol and bioinformatics software is generally applicable and optimized for Illumina NGS data, including WGS and exome-captured sequencing data. We use the publicly available HiSeq WGS data from the Illumina Platinum Genomes[42] (http://www.ebi.ac.uk/ena/data/view/ERP001960) for the family of NA12878 as an illustration in this protocol. However, some specific parameters, such as the filtering criterion, may need to be adjusted accordingly for a different data set.
- BWA (ref. 43) v0.7.12 (http://bio-bwa.sourceforge.net/)
- SAMtools[21] v1.3 (http://samtools.sourceforge.net/)
- bcftools v1.2 (http://samtools.github.io/bcftools/)
- Picard v1.130 (http://broadinstitute.github.io/picard/)
- Scalpel v0.5.3 (http://scalpel.sourceforge.net)
- bedtools[44] v2.23.0 (https://github.com/arq5x/bedtools2)
- PyVCF v0.6.7 (https://github.com/jamescasbon/PyVCF)
- Annovar[45] v2015-03-22 (http://annovar.openbioinformatics.org/)
- R v2.15 (http://www.r-project.org)
- gnuplot v4.4 (http://www.gnuplot.info/)
- IGV(ref. 46) v2.3 (https://www.broadinstitute.org/igv/)

EQUIPMENT SETUP

Hardware setup—The software used in this protocol is intended for operation on a 64-bit machine, running a 64-bit version of the Linux operating system. We recommend using a machine with at least 1.2 TB of disk storage available for whole-genome analysis and a minimum of 64 GB of RAM. The software will scale to the number of cores available; we recommend the use of at least ten cores, if possible, especially for whole-genome analysis.

Software setup—Download Scalpel's resource bundle containing the scripts for visualization and quality control of the indels (available as part of the Scalpel distribution). Download relevant files hosted on the Scalpel website, including the resource bundle:

```
 % wget --no-check
http://sourceforge.net/projects/scalpel/files/scalpel-0.5.3.tar.gz
; tar zxvf scalpel-0.5.3.tar.gz; cd scalpel-0.5.3; make; export PATH=../
scalpel-0.5.3:$PATH
 % tar zxvf protocol_bundle-0.5.3.tar.gz; cd protocol_bundle-0.5.3
```

Download the tools that are required for the indel analysis, including BWA, SAMtools, bcftools, Picard, Scalpel, vt, bedtools and Annovar (needs registration (http://www.openbioinformatics.org/annovar/anno-var_download_form.php)):

```
 % wget --no-check
http://sourceforge.net/projects/bio-bwa/files/bwa-0.7.12.tar.bz2
; tar jxf bwa-0.7.12.tar.bz2; cd bwa-0.7.12; make; cd ../; export PATH=./
bwa-0.7.12:$PATH
 % wget --no-check
http://sourceforge.net/projects/samtools/files/samtools/1.3/
samtools-1.3.tar.bz2
; tar jxf samtools-1.3.tar.bz2; cd samtools-1.3; make; cd ..; export PATH=./
samtools-1.3:$PATH
 % wget -no-check
https://github.com/samtools/bcftools/releases/download/1.2/
bcftools-1.2.tar.bz2
; tar bcftools-1.2.tar.bz2; cd bcftools-1.2; make; cd ..; export PATH=./
bcftools-1.2:$PATH
 % wget --no-check
https://github.com/broadinstitute/picard/releases/download/1.130/picard-
tools-1.130.zip
; unzip picard-tools-1.130.zip; export PATH=./picard-tools-1.130:$PATH
 % wget --no-check
```

```
https://github.com/arq5x/bed-tools2/releases/download/v2.23.0/
bedtools-2.23.0.tar.gz

; tar zxvf bedtools-2.23.0.tar.gz;cd bedtools2; make; cd ..; export PATH=./
bedtools-2.23.0/bin:$PATH
 % wget --no-check

http://www.openbioinformatics.org/annovar/download/register-for-download/
annovar.latest.tar.gz

; tar zxvf annovar.latest.tar.gz; export PATH=./annovar:$PATH
 % git clone

https://github.com/jamescasbon/PyVCF.git

 cd PyVCF; python setup.py install; cd ..
```

PROCEDURE

▲ **CRITICAL** This protocol includes 26 steps encompassing the whole procedure from downloading the input data sets to identification of frameshift variants. The protocol bundle, available within the Scalpel software package, contains a master script called 'run_protocol_0.53.sh' with the complete list of commands (excluding software setup) required to replicate the results presented in this procedure. This script can also be modified to automate the processing of user samples.

Downloading of the example sequencing data and reference ● TIMING ~6 h

1| Download the example sequencing reads of the Hapmap quad family from the Illumina Platinum Genome project ('*_1*fastq.gz' and '*_2*fastq.gz' denote paired-end reads):

```
% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194146/
ERR194146_1.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194146/
ERR194146_2.fastq.gz

% wget --no-check
```

```
ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194147/
ERR194147_1.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194147/
ERR194147_2.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194151/
ERR194151_1.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR194/ERR194151/
ERR194151_2.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR324/ERR324432/
ERR324432_1.fastq.gz

% wget --no-check

ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR324/ERR324432/
ERR324432_2.fastq.gz
```

2| Download the human reference genome hg19:

```
% wget --no-check

http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/hg19.2bit

% wget --no-check

http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/twoBitToFa
```

3| Convert the *.2bit genome to *.fa format and index it with BWA (note that you can also download the .fasta file directly, although this may take much longer):

```
% chmod +x twoBitToFa; ./twoBitToFa hg19.2bit hg19.fa
% bwa index hg19.fa
```

Alignment of the NGS reads to the genome ● TIMING ~40 h

4| Align reads to reference for each sample separately with bwa mem:

```
% bwa mem -t 10 -R '@RG\tID:NA12877\tSM:NA12877' hg19.fa
ERR194146_1.fastq.gz ERR194146_2.fastq.gz | samtools view -h -S -b
> NA12877.bam
% bwa mem -t 10 -R' @RG\tID:NA12878\tSM:NA12878' hg19.fa
ERR194147_1.fastq.gz ERR194147_2.fastq.gz | samtools view -h -S -b
> NA12878.bam
% bwa mem -t 10 -R' @RG\tID:NA12881\tSM:NA12881' hg19.fa
ERR324432_1.fastq.gz ERR324432_2.fastq.gz | samtools view -h -S -b
> NA12881.bam
% bwa mem -t 10 -R' @RG\tID:NA12882\tSM:NA12882' hg19.fa
ERR194151_1.fastq.gz ERR194151_2.fastq.gz | samtools view -h -S -b
> NA12882.bam
```

? TROUBLESHOOTING

5| Sort the bam files by chromosome coordinates with SAMtools, and then delete the unsorted versions:

```
% samtools sort -m 4G -o NA12877.sort.bam NA12877.bam
% samtools sort -m 4G -o NA12878.sort.bam NA12878.bam
% samtools sort -m 4G -o NA12881.sort.bam NA12881.bam
% samtools sort -m 4G -o NA12882.sort.bam NA12882.bam
% rm -f NA12877.bam NA12878.bam NA12881.bam NA12882.bam
```

6| Mark duplicated reads within the alignment with Picard Tools:

```
% java -jar -Xmx10g picard.jar MarkDuplicates
INPUT=NA12877.sort.bam OUTPUT=NA12877.sort.markdup.bam
METRICS_FILE=NA12877.sort.metric
% java -jar -Xmx10g picard.jar MarkDuplicates
INPUT=NA12878.sort.bam OUTPUT=NA12878.sort.markdup.bam
METRICS_FILE=NA12878.sort.metric
% java -jar -Xmx10g picard.jar MarkDuplicates
INPUT=NA12881.sort.bam OUTPUT=NA12881.sort.markdup.bam
METRICS_FILE=NA12881.sort.metric
% java -jar -Xmx10g picard.jar MarkDuplicates
```

```
INPUT=NA12882.sort.bam OUTPUT=NA12882.sort.markdup.bam
METRICS_FILE=NA12882.sort.metric
% rm -f NA12877.sort.bam NA12878.sort.bam NA12881.sort.bam
NA12882.sort.bam
```

7| Perform a basic quality control of the alignment files with SAMtools:

```
% samtools flagstat NA12877.sort.markdup.bam >
NA12877.sort.markdup.bam.simplestats
% samtools flagstat NA12878.sort.markdup.bam >
NA12878.sort.markdup.bam.simplestats
% samtools flagstat NA12881.sort.markdup.bam >
NA12881.sort.markdup.bam.simplestats
% samtools flagstat NA12882.sort.markdup.bam >
NA12882.sort.markdup.bam.simplestats
```

▲ **CRITICAL STEP** To generate reliable indel calls, accurate alignment of the NGS short reads is of great importance. If the DNA is derived from a blood sample, the mapping rate of Illumina Hiseq reads is typically higher than 90%. Lower mapping rates indicate either contaminations of DNA from other species (e.g., bacterial DNA from saliva samples) or poor quality of the sequencing experiments. In addition, excessive numbers of duplicated reads are usually due to issues with library construction and PCR amplification. Table 2 lists the number of reads generated for each sample and the reads mapped to the human genome hg19.

? TROUBLESHOOTING

Exonic indel variant calling and downstream filtering ● TIMING ~8 h

8| Run Scalpel in the ‘de novo’ mode to perform multisample calling for a quad family. In this example, we use the NA12882 genome to represent the affected individual. The NA12881 genome represents the unaffected individual accordingly:

```
% scalpel-discovery --denovo --dad NA12877.sort.markdup.bam --mom
NA12878.sort.markdup.bam --aff NA12882.sort.markdup.bam --sib
NA12881.sort.markdup.bam --bed
SeqCap_EZ_Exome_v3_primary.scalpel.bed --ref hg19.fa --numprocs 10
--two-pass
```

▲ **CRITICAL STEP** In both ‘de novo’ and ‘somatic’ mode, Scalpel is optimized to achieve high sensitivity, but it may include some false positives. To control for this, we recommend using the ‘--two-pass’ option in Scalpel, which undergoes a second round of indel verification to reduce the likely false calls.

9| Export the inherited and *de novo* mutations from the Scalpel database (in target only):

```
% scalpel-export --denovo --db outdir/main/inherited.db --bed
SeqCap_EZ_Exome_v3_primary.scalpel.bed --ref hg19.fa --intarget --
min-alt-count-affected 10 --max-chi2-score 10.8 >
inherited.onepass.vcf
% scalpel-export --denovo --db outdir/twopass/denovos.db --bed
SeqCap_EZ_Exome_v3_primary.scalpel.bed --ref hg19.fa --intarget --
min-alt-count-affected 10 --max-chi2-score 10.8 --min-coverage-
unaffected 20 > denovo.twopass.vcf
```

? TROUBLESHOOTING

10| Identify and mark indels within STR regions using the microsatellite annotation software (msdetector) distributed with the protocol bundle:

```
% sh ./msdetector/msdetector.sh -r 50 -d 2 -g hg19.fa -i
inherited.onepass.vcf > inherited.onepass.vcf.ms
% sh ./msdetector/msdetector.sh -r 50 -d 2 -g hg19.fa -i
denovo.twopass.vcf > denovo.twopass.vcf.ms
```

11| Save indels within and outside STR regions into different variant calling format (.vcf) files (note: the number of fields to keep with the UNIX cut command depends on the number of samples in the .vcf file):

```
% awk -F"\t"' {if($0 ~ /^#/){print $0} else{if($16=="yes")
print}}' inherited.onepass.vcf.ms | cut -f1-13 >
inherited.onepass.vcf.ms.in
% awk -F"\t"' {if($0 ~ /^#/){print $0} else{if($16=="no") print}}'
inherited.onepass.vcf.ms | cut -f1-13 >
inherited.onepass.vcf.ms.out
% awk -F"\t"' {if($0 ~ /^#/){print $0} else{if($16=="yes")
print}}' denovo.twopass.vcf.ms | cut -f1-13 >
denovo.twopass.vcf.ms.in
% awk -F"\t"' {if($0 ~ /^#/){print $0} else{if($16=="no") print}}'
denovo.twopass.vcf.ms | cut -f1-13 > denovo.twopass.vcf.ms.out
```

▲ **CRITICAL STEP** Low-quality indel calls (potential false positives) are usually found within low-coverage regions or have an unbalanced number of reads supporting the alternative allele.

? TROUBLESHOOTING

12| Filter out false-positive calls by adjusting coverage and/or χ^2 score thresholds for your data:

```
% awk -F"\t" '{if($0 ~ /^#/){print $0} else {if(! ($7~ /
LowAltCntAff/ && $7 ~ /HighChi2score/) ) print}}'
inherited.onepass.vcf.ms.out > inherited.onepass.vcf.ms.out.hq
% awk -F"\t" '{if($0 ~ /^#/){print $0} else {if(! ($7~ /
LowAltCntAff/ || $7 ~ /High-Chi2score/ || $7 ~ /LowCovUnaff/))
print}}' denovo.twopass.vcf.ms.out > denovo.twopass.vcf.ms.out.hq
```

13| (Optional) Perform additional filtering of the *de novo* calls using the Python script provided in the Scalpel resource bundle. This script supports filtering indels by aac, χ^2 scores and parental coverage:

```
% python denovo-multi-filter.py -i denovo.twopass.vcf.ms.out -f
NA12877 -m NA12878 -a NA12882 -u NA12881 -aac 10 -chi 10.8 -pc 20 -
o denovo.twopass.vcf.ms.out.filter
```

14| (Optional) Extract a subset of indels based on other annotations using bedtools. Here we show how to extract the variants that overlap any of the mutations in the ClinVar main database.

```
% bedtools intersect -wa -u -a inherited.onepass.vcf.ms.out.hq -b
clinvar_main.bed > inherited.onepass.vcf.ms.out.hq.clinvar
```

? TROUBLESHOOTING

15| Summarize indel calls with a histogram of mutations by size:

```
% grep -v"#" inherited.onepass.vcf.ms.out.hq
denovo.twopass.vcf.ms.out.hq | awk' {print length($5)-length($4)}'
> all.indel.size.txt
% gnuplot44 -e"outfile='indel_size_dist.pdf';
infile=' all.indel.size.txt'" size_dist.gnu
```

16| Characterize low-quality homopolymer indel calls with a histogram of mutations by VAF:

```
% cat denovo.twopass.vcf.ms inherited.onepass.vcf.ms | grep -v'#' |
grep 'yes' | awk -F"\t"' {if(($7~ /LowAltCntAff/ && $7 ~ /
HighChi2score/) || $7 ~ /LowCovUnaff/) print}'> combine.ms.txt
% for i in A C G T; do awk -v j=$i' $0! ~ /^#/ {if($15==j)
{split($12,a,":"); if(a[1]== "0/1"|| a[1]== "1/1") split(a[2],b,
```

```
","); print b[1] "\t" b[2]} 'combine.ms.txt > poly${i}.VAF.txt;
done
% gnuplot44 -e "outfile= 'homo.vaf.pdf'; infileA= 'polyA.VAF.txt';
infileC= 'polyC.VAF.txt'; infileG= 'polyG.VAF.txt'; infileT=
'polyT.VAF.txt'" hp.vafdist.gnu
```

▲ **CRITICAL STEP** There are usually much higher sequencing biases in GC-extreme regions. Indels within STRs, especially homopolymer A or T runs, are major sources of false-positive variant calls.

17| Summarize inherited indels with VAF %:

```
% awk -F'\t' '$0! ~ /^#/ {split($12,a,":"); if(a[1]== "0/1" ||
a[1]== "1/1") split(a[2],b, ","); print b[1] "\t" b[2]}'
inherited.onepass.vcf.ms.out > inherited.onepass.vcf.ms.out.vaf
% awk -F'\t' '$0! ~ /^#/ {split($12,a,":"); if(a[1]== "0/1" ||
a[1]== "1/1") split(a[2],b, ","); print b[1] "\t" b[2]}'
inherited.onepass.vcf.ms.out.hq >
inherited.onepass.vcf.ms.out.hq.vaf
% gnuplot44 -e "outfile='inherited.VAFdist.pdf'; infileAll=
'inherited.onepass.vcf.ms.out.vaf'; infileHq=
'inherited.onepass.vcf.ms.out.hq.vaf'
" vafdistplot.inherited.qual.gnu
```

▲ **CRITICAL STEP** The filtering cascade should not reduce the sensitivity of inherited indels by a lot. One should expect a relatively balanced number of reads supporting each inherited indel, indicating high confidence for these calls.

18| Determine the number of indels remaining after each step of the filtering:

```
% for i in *.vcf.* ; do echo $i; grep -v "#" $i | wc -l;done >
indel.count.txt
```

19| Split the multisample VCF file to create an individual file for NA12882:

```
% for file in *.hq; do bgzip -c $file > $file.gz; tabix -p vcf
$file.gz; done
% for file in *.hq.gz; do bcftools view -c1 -Ov -s NA12882 -o $
{file/.gz*/.NA12882.vcf} ${file}; done
```

20| Filter the single VCF files based on χ^2 score and allele coverage:

```
% python single-vcf-filter.py -i
inherited.onepass.vcf.ms.out.hq.NA12882.vcf -mc 10 -chi 10.8 -o
```

```
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf
% python single-vcf-filter.py -i
denovo.twopass.vcf.ms.out.hq.NA12882.vcf -mc 10 -chi 10.8 -o
denovo.twopass.vcf.ms.out.hq.NA12882.filter.vcf
```

Annotation and visualization of the indel calls ● TIMING <5 min

21| Prepare and create the input format required by Annovar:

```
% annovar=/path-to-annovar/
% $annovar/convert2annovar.pl -format vcf4
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf >
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf.avinput
% $annovar/convert2annovar.pl -format vcf4
denovo.twopass.vcf.ms.out.hq.NA12882.filter.vcf >
denovo.twopass.vcf.ms.out.hq.NA12882.filter.vcf.avinput
```

22| Annotate and intersect indels with gene regions using Annovar:

```
% $annovar/annotate_variation.pl -buildver hg19
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf.avinput
$annovar/humandb
% $annovar/annotate_variation.pl -buildver hg19
denovo.twopass.vcf.ms.out.hq.NA12882.filter.vcf.avinput $annovar/
humandb
```

23| Summarize coding region indels by size in R:

```
% cat
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf.avinput.exonic_v
ariant_function | egrep -v'unknown|stopgain' | cut -f 2,7,8 | cut -
d" " -f 2 | awk '{if($2=="-") print $1 "\t" length($3);else if
($3== "-") print $1 "\t" length($2)' >type_and_size.txt
% R
> indel=read.table("type_and_size.txt", header=FALSE)
> colnames(indel)= c("type", "size")
> indel_30=indel[indel[,2]<=30,]
> indel.table <-
table(indel_30$type,factor(indel_30$size,lev=1:30))
> pdf('indelsize_by_type.pdf', width=16, height=7)
> mar.default <- c(5,4,4,2) + 0.1
> par(mar = mar.default + c(0, 4, 0, 0))
> barplot(indel.table, main= "indel distribution within coding
sequence (CDS)", xlab= "indel size", ylab= "number of indels",
```

```
col=c("green", "red"), cex.axis=2, cex.names=2, cex.lab = 2,
cex.main=2, cex.sub=2)
> legend('topright',rownames(indel.table), fil=c('green', 'red'),
bty= 'n', cex=2)
> dev.off()
```

24| Filter the indels based on population allele frequencies:

```
% $annovar/annotate_variation.pl -filter -out
inherited.onepass.vcf.ms.out.hq.NA12882.filter -dbtype
popfreq_max_20150413 -build hg19
inherited.onepass.vcf.ms.out.hq.NA12882.filter.vcf.avinput
$annovar/humandb/
% $annovar/annotate_variation.pl -filter -out
denovo.twopass.vcf.ms.out.hq.NA12882.filter -dbtype
popfreq_max_20150413 -build hg19
denovo.twopass.vcf.ms.out.hq.NA12882.filter.vcf.avinput $annovar/
humandb/
```

25| Annotate novel indels that were not reported by a population database before (1000G, ESP6500, ExAC and CG46):

```
% $annovar/annotate_variation.pl -buildver hg19
inherited.onepass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201
50413_filtered $annovar/humandb
% $annovar/annotate_variation.pl -buildver hg19
denovo.twopass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201504
13_filtered $annovar/humandb
```

26| Retrieve frameshift mutations, which are potentially loss-of-function:

```
% awk '{if($2=="frameshift") print}'
inherited.onepass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201
50413_filtered.exonic_variant_function >
inherited.onepass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201
50413_filtered.exonic_variant_function.fs.txt
% awk '{if($2=="frameshift") print}'
denovo.twopass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201504
13_filtered.exonic_variant_function >
denovo.twopass.vcf.ms.out.hq.NA12882.filter.hg19_popfreq_max_201504
13_filtered.exonic_variant_function.fs.txt
```

? TROUBLESHOOTING

Troubleshooting advice can be found in Table 3.

● TIMING

Following this protocol, it will take ~48 h to complete the analysis of the exonic indels in the example WGS data on a machine with ten processing cores and at least 53 GB of RAM. However, the time could be variable depending on the user's actual computational power. Most of the run time is spent on read alignment. Notably, we can expect that, in the near future, there will be studies with larger sample sizes and deeper sequencing. Therefore, the run time may be longer, assuming the computer power remains the same.

Steps 1–3, downloading of the example sequencing data and reference: ~6 h

Steps 4–7, alignment of the NGS reads to the genome: ~40 h

Steps 8–20, exonic indel variant calling and downstream filtering: ~8 h

Steps 21–26, annotation and visualization of the indel calls: <5 min

ANTICIPATED RESULTS

Expected distribution of indels and signatures of low-quality calls

After filtering for acc, χ^2 scores and STR regions, the size of the high-quality inherited indels should follow a log-normal distribution (Fig. 7). Similar observations of such a size distribution were also reported in the 1000 Genomes Project[47] and an analysis of 179 human genomes[13]. We also observed a much higher abundance of homopolymer A or T indels, relative to homopolymer C or G indels in the low-quality call set (Fig. 8). Homopolymer A or T indels usually have low VAF, because homopolymer A or T molecules are enriched for PCR stutter/slippage artifacts. Conversely, the VAF of high-quality inherited indels follows an approximately normal distribution, with a mean of around or slightly <50% (Fig. 9). This indicates that we observed equal read evidence for the two alleles in the genome.

Expected number of indels that remains during the filtering cascade

As calling *de novo* indels requires a more sensitive analysis of the family members, we recommend using the '--two-pass' search option when discovering *de novo* events. Many more inherited indels will persist through the filtering cascade, as compared to the number of *de novo* events. This is because *de novo* events are extremely rare in comparison with inherited indels. *De novo* mutations are also particularly vulnerable to batch effects and random errors, as a correct analysis requires both high sensitivity and specificity in the entire family. In fact, among the in-target indels, ~51% of the inherited ones are of high quality, whereas only 5% of the *de novo* ones survived the filtering cascade (Fig. 10).

Indel distribution within coding sequence

Because frameshift mutations can cause loss of function of a gene, these mutations are expected to be less frequent than frame-preserving mutations in the coding region. As shown

in Figure 11 (produced at Step 23), indels whose size is a multiple of three are much more abundant than others with similar sizes (+1 or −1).

A list of novel inherited frameshift mutations in the family

Although this family has been investigated in many studies, many frameshift indels were not discovered in any public database, including 1000G, ExAC and ESP (Table 4). We observe a total of six novel frameshift mutations. Many of these indels are of a size larger than 5 bp. On the basis of Sanger validation of these loci, all 20 genotypes in four family members were successfully validated/confirmed (Supplementary Results; Supplementary Methods). With the improvement of the indel-calling protocol introduced in this article, we are able to identify these previously undiscovered loss-of-function mutations. We also inspected the VCF file generated by the Illumina Platinum Genome project (release 8.0.1) for the presence of the six discovered frameshifts. Although the VCF file was generated using five different variant callers (Freebayes, Platypus, GATKv3, Cortex and Issac2), it contained only two of the six indels in Table 4. This indeed further demonstrates the power of Scalpel over other methods, especially for detecting large indels.

The *de novo* indel in the child and the alignment IGV screenshot

High-quality *de novo* indels usually share the following characteristics: (i) the number of reads in the region is close to the genome-wide mean coverage; (ii) there are balanced numbers of reads supporting both the reference and the alternative allele; (iii) these indels are not located within or near STR regions; and (iv) in the parents' genome, there are no reads supporting the same indel presented in the child's genome. Table 5 reports the *de novo* deletion found in the affected child. This is a 1-bp heterozygous frameshift deletion located in exon 4 of the gene *HFM1*. This *HFM1 de novo* deletion was also successfully validated in Sanger experiments (Supplementary Results; Supplementary Methods). The genomic coordinate is chr1: 91859889, relative to the reference genome hg19. This variant has not been reported before in any of the widely used variant databases, such as 1000G, ESP6500, ExAC and CG46. Figure 12 shows the screenshot of the IGV alignment of all four genomes. We can see a distinct signature of the deletion presented only in the affected child, but not in anyone else in the family.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The project was supported in part by grants from the US National Institutes of Health (R01-HG006677 and U01-CA168409) and the US National Science Foundation (DBI-1350041) to M.C.S. and by grants from the Cold Spring Harbor Laboratory (CSHL) Cancer Center Support (5P30CA045508), the Stanley Institute for Cognitive Genomics and the Simons Foundation (SF51 and SF235988) to M.W.

References

1. Collins FS, Varmus H. A new initiative on precision medicine. N Engl J Med. 2015; 372:793–795. [PubMed: 25635347]

2. Highnam G, et al. An analytical framework for optimizing variant discovery from personal genomes. Nat Commun. 2015; 6:6275. [PubMed: 25711446]
3. Watson, JD., Baker, TA., Gann, A., Levine, M., Losick, R. Molecular Biology of the Gene. 7. Cold Spring Harbor Laboratory Press; 2013.
4. Nik-Zainal S, et al. Mutational processes molding the genomes of 21 breast cancers. Cell. 2012; 149:979–993. [PubMed: 22608084]
5. Zaidi S, et al. De novo mutations in histone-modifying genes in congenital heart disease. Nature. 2013; 498:220–223. [PubMed: 23665959]
6. Iossifov I, et al. De novo gene disruptions in children on the autistic spectrum. Neuron. 2012; 74:285–299. [PubMed: 22542183]
7. Iossifov I, et al. The contribution of *de novo* coding mutations to autism spectrum disorder. Nature. 2014; 515:216–221. [PubMed: 25363768]
8. Gupta RS. Protein phylogenies and signature sequences: a reappraisal of evolutionary relationships among archaebacteria, eubacteria, and eukaryotes. Microbiol Mol Biol Rev. 1998; 62:1435–1491. [PubMed: 9841678]
9. Tian D, et al. Single-nucleotide mutation rate increases close to insertions/deletions in eukaryotes. Nature. 2008; 455:105–108. [PubMed: 18641631]
10. MacArthur DG, et al. A systematic survey of loss-of-function variants in human protein-coding genes. Science. 2012; 335:823–828. [PubMed: 22344438]
11. Fukuoka S, et al. Loss of function of a proline-containing protein confers durable disease resistance in rice. Science. 2009; 325:998–1001. [PubMed: 19696351]
12. Denver DR, et al. High mutation rate and predominance of insertions in the *Caenorhabditis elegans* nuclear genome. Nature. 2004; 430:679–682. [PubMed: 15295601]
13. Montgomery SB, et al. The origin, evolution, and functional impact of short insertion-deletion variants identified in 179 human genomes. Genome Res. 2013; 23:749–761. [PubMed: 23478400]
14. Mullaney JM, et al. Small insertions and deletions (INDELs) in human genomes. Hum Mol Genet. 2010; 19:R131–R136. [PubMed: 20858594]
15. Jiang Y, Turinsky AL, Brudno M. The missing indels: an estimate of indel variation in a human genome and analysis of factors that impede detection. Nucleic Acids Res. 2015; 43:7217–7228. [PubMed: 26130710]
16. Narzisi G, et al. Accurate *de novo* and transmitted indel detection in exome-capture data using microassembly. Nat Methods. 2014; 11:1033–1036. [PubMed: 25128977]
17. Narzisi G, Schatz MC. The challenge of small-scale repeats for indel discovery. Front Bioeng Biotechnol. 2015; 3:8. [PubMed: 25674564]
18. Fang H, et al. Reducing INDEL calling errors in whole genome and exome sequencing data. Genome Med. 2014; 6:89. [PubMed: 25426171]
19. DePristo MA, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 2011; 43:491–498. [PubMed: 21478889]
20. Albers CA, et al. Dindel: accurate indel calls from short-read data. Genome Res. 2011; 21:961–973. [PubMed: 20980555]
21. Li H, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009; 25:2078–2079. [PubMed: 19505943]
22. Ye K, et al. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics. 2009; 25:2865–2871. [PubMed: 19561018]
23. Karakoc E, et al. Detection of structural variants and indels within exome data. Nat Methods. 2012; 9:176–178.
24. Iqbal Z, et al. De novo assembly and genotyping of variants using colored de Bruijn graphs. Nat Genet. 2012; 44:226–232. [PubMed: 22231483]
25. Van der Auwera GA, et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics. 2013; 11:11 10 1–11 10 33.
26. Li S, et al. SOAPindel: efficient identification of indels from short paired reads. Genome Res. 2013; 23:195–200. [PubMed: 22972939]

27. Pabinger S, et al. A survey of tools for variant analysis of next-generation genome sequencing data. Brief Bioinform. 2014; 15:256–278. [PubMed: 23341494]
28. Rimmer A, et al. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. Nat Genet. 2014; 46:912–918. [PubMed: 25017105]
29. Mose LE, et al. ABRA: improved coding indel detection via assembly-based realignment. Bioinformatics. 2014; 30:2813–2815. [PubMed: 24907369]
30. Chen K, et al. TIGRA: a targeted iterative graph routing assembler for breakpoint assembly. Genome Res. 2014:24310–24317.
31. Weisenfeld NI, et al. Comprehensive variation discovery in single human genomes. Nat Genet. 2014; 46:1350–1355. [PubMed: 25326702]
32. Leggett RM, et al. Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de Bruijn graphs. PLoS One. 2013; 8:e60058. [PubMed: 23536903]
33. Chen X, et al. Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics. 2016; 32:1220–1222. [PubMed: 26647377]
34. Yang R, et al. ScanIndel: a hybrid framework for indel detection via gapped alignment, split reads and *de novo* assembly. Genome Med. 2015; 7:127. [PubMed: 26643039]
35. Highnam G, et al. Accurate human microsatellite genotypes from high-throughput resequencing data using informed error profiles. Nucleic Acids Res. 2013; 41:e32. [PubMed: 23090981]
36. Gymrek M, et al. lobSTR: A short tandem repeat profiler for personal genomes. Genome Res. 2012; 22:1154–1162. [PubMed: 22522390]
37. Ye K, et al. Systematic discovery of complex insertions and deletions in human cancers. Nat Med. 2016; 22:97–104. [PubMed: 26657142]
38. Gnerre S, et al. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proc Natl Acad Sci USA. 2011; 108:1513–1518. [PubMed: 21187386]
39. Berlin K, et al. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol. 2015; 33:623–630. [PubMed: 26006009]
40. Paila U, et al. GEMINI: integrative exploration of genetic variation and genome annotations. PLoS Comput Biol. 2013; 9:e1003153. [PubMed: 23874191]
41. Brannon AR, et al. Comparative sequencing analysis reveals high genomic concordance between matched primary and metastatic colorectal cancer lesions. Genome Biol. 2014; 15:454. [PubMed: 25164765]
42. Eberle, MA., et al. A reference dataset of 54 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree. 2016. bioRxiv http://dx.doi.org/10.1101/055541
43. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. ArXiv e-prints. 2013; 1303:3997.
44. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010; 26:841–842. [PubMed: 20110278]
45. Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res. 2010; 38:e164. [PubMed: 20601685]
46. Robinson JT, et al. Integrative genomics viewer. Nat Biotechnol. 2011; 29:24–26. [PubMed: 21221095]
47. Genomes Project, C. et al. A map of human genome variation from population-scale sequencing. Nature. 2010; 467:1061–1073. [PubMed: 20981092]
48. Tan A, Abecasis GR, Kang HM. Unified representation of genetic variants. Bioinformatics. 2015; 31:2202–2204. [PubMed: 25701572]
49. Van der Auwera GA, et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics. 2013; 43:11 10 1–11 10 33. [PubMed: 25431634]
50. Cingolani P, et al. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. Fly (Austin). 2012; 6:80–92. [PubMed: 22728672]

51. McLaren W, et al. Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. Bioinformatics. 2010; 26:2069–2070. [PubMed: 20562413]

52. McCarthy DJ, et al. Choice of transcripts and software has a large effect on variant annotation. Genome Med. 2014; 6:26. [PubMed: 24944579]

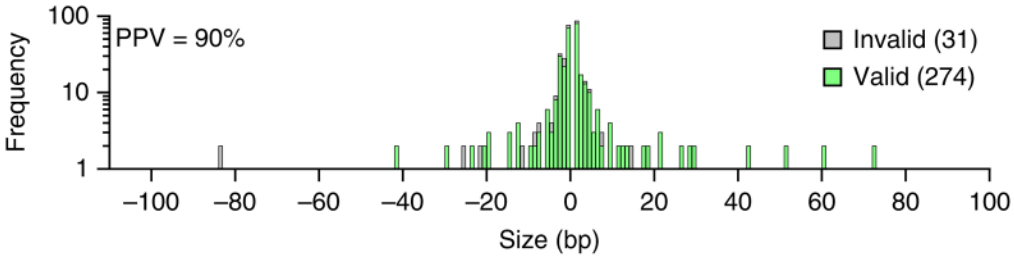

Figure 1.

High accuracy of indel detection using Scalpel on WGS data. Scalpel was run in single mode on 30× Illumina Hiseq 2000 2 × 100 bp WGS data described in Narzisi *et al.*[16] and later analyzed in Fang *et al.*[18]. This figure shows the size distribution of valid (green) and invalid (gray) indels that were randomly selected for validation (using targeted resequencing) from the two previous studies. This validation set includes 160 and 145 candidate variants that were WGS–WES intersected and WGS-specific, respectively. Among a total of 305 candidates, 90% of them (274) were successfully validated. Positive predictive value (PPV) is computed by PPV = no. TP/(no. TP + no. FP), where no. TP is the number of true-positive calls and no. FP is the number of false-positive calls.

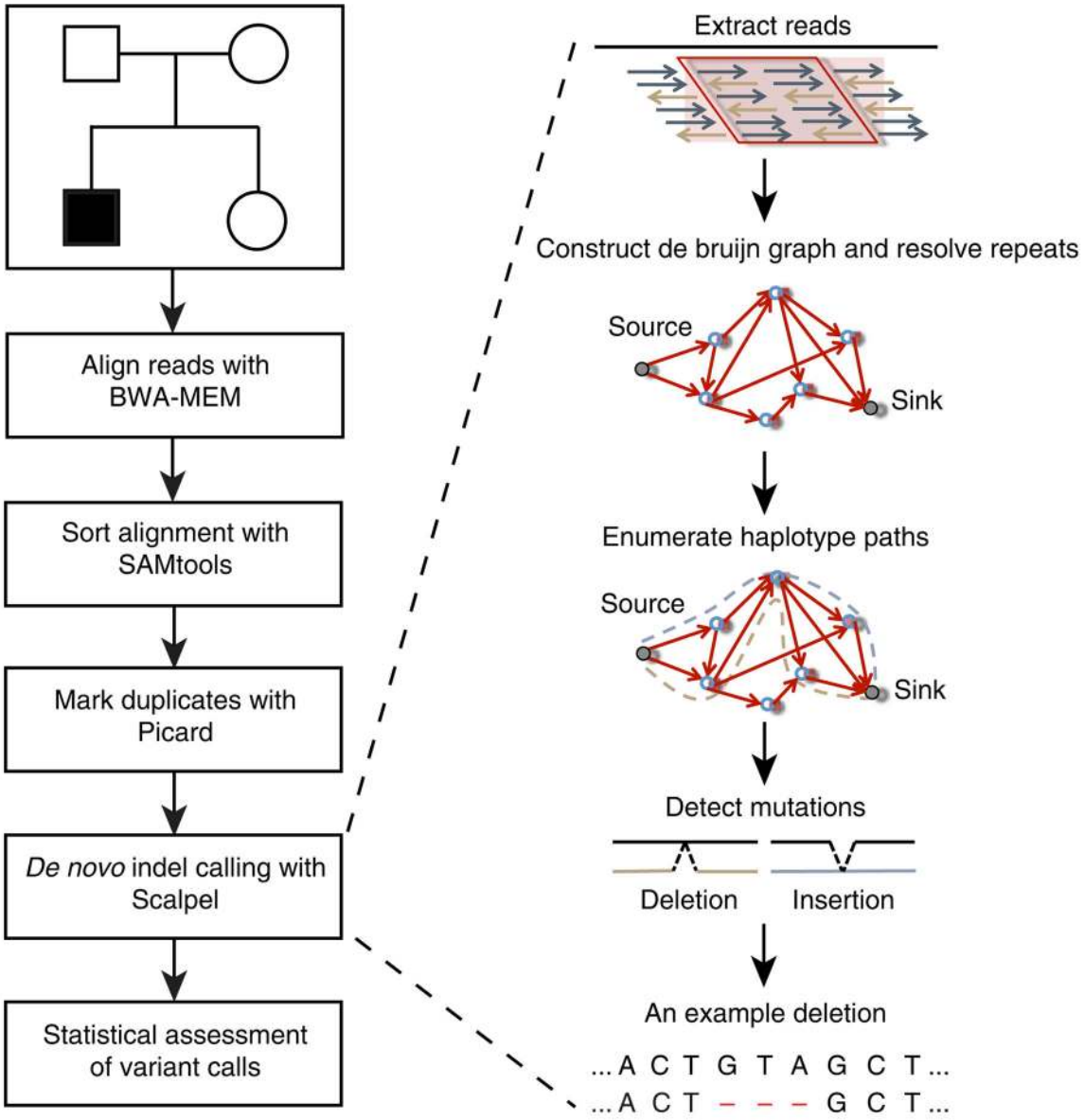

Figure 2.

Main steps in the Scalpel protocol. Starting from raw sequencing data, reads are first aligned to the human genome using the BWA[43] software package (Step 4 in PROCEDURE). After the standard practices in the field, the alignments are sorted (using SAMtools[21], Step 5 in PROCEDURE) and duplicates are marked (using Picard Tools—http://broadinstitute.github.io/picard/, Step 6 in PROCEDURE). Finally, indels can be called with Scalpel (Steps 8 and 9 in PROCEDURE), and statistical assessment of the variant calls can provide diagnostics of the data (Steps 10–20 in PROCEDURE). Note that as Scalpel locally reassembles the reads, this procedure is free of computationally expensive techniques such as indel realignment and base quality recalibrations. The BAM files obtained after the earlier steps are the input for the Scalpel microassembly procedure. Scalpel then localizes the reads

within a window, constructs a de Bruijn graph, resolves repeat structure and enumerates haplotype paths. Image adapted with permission from ref. 16, Nature Publishing Group.

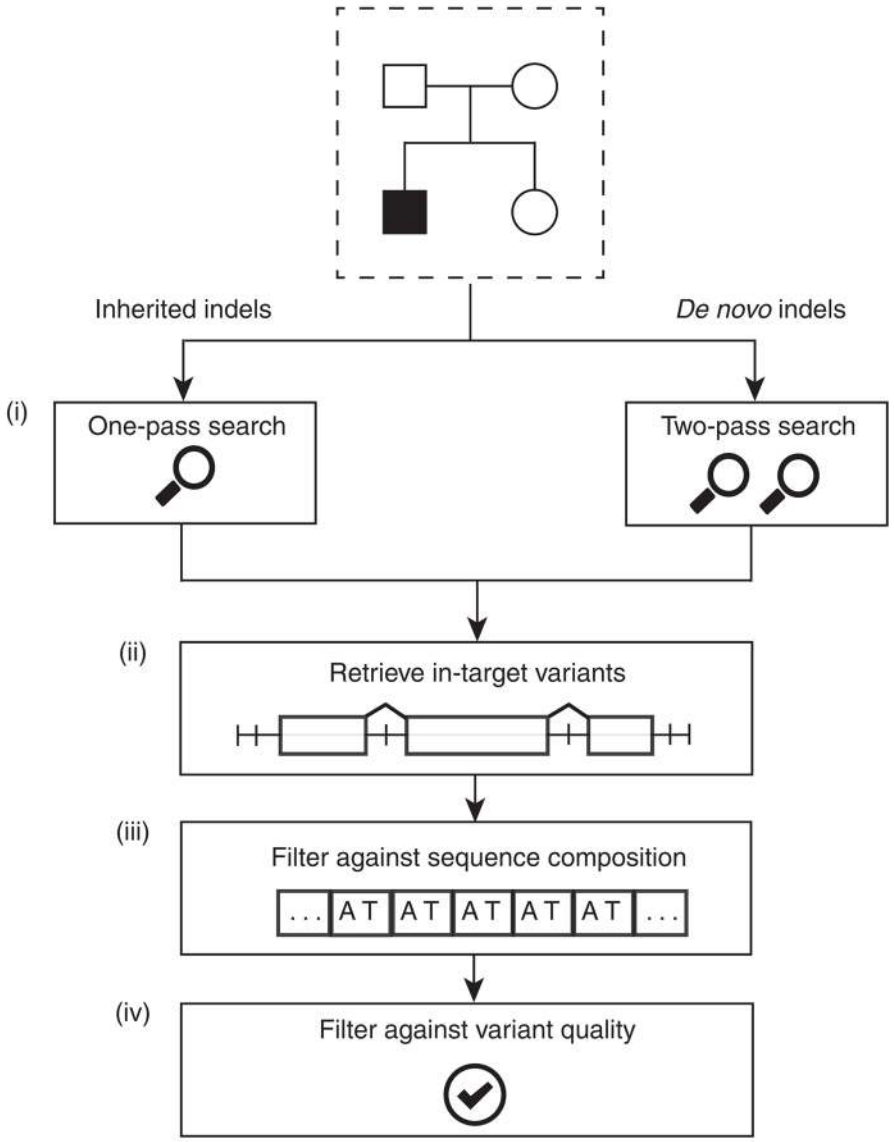

Figure 3.

Overview of the indel variant filtering cascade. This figure is a conceptual representation of the filtering procedure (Steps 9–12 in the PROCEDURE). It is used to report high-quality *de novo* and inherited indels within the target region, coding regions in this case. (i) Inherited and *de novo* indels are analyzed separately; (ii) only variants within the target regions are exported; (iii) low-quality indels are identified and removed based on sequence composition (e.g., STRs); and (iv) additional filters based on supporting coverage and allele balance are used to reduce the number of false positives.

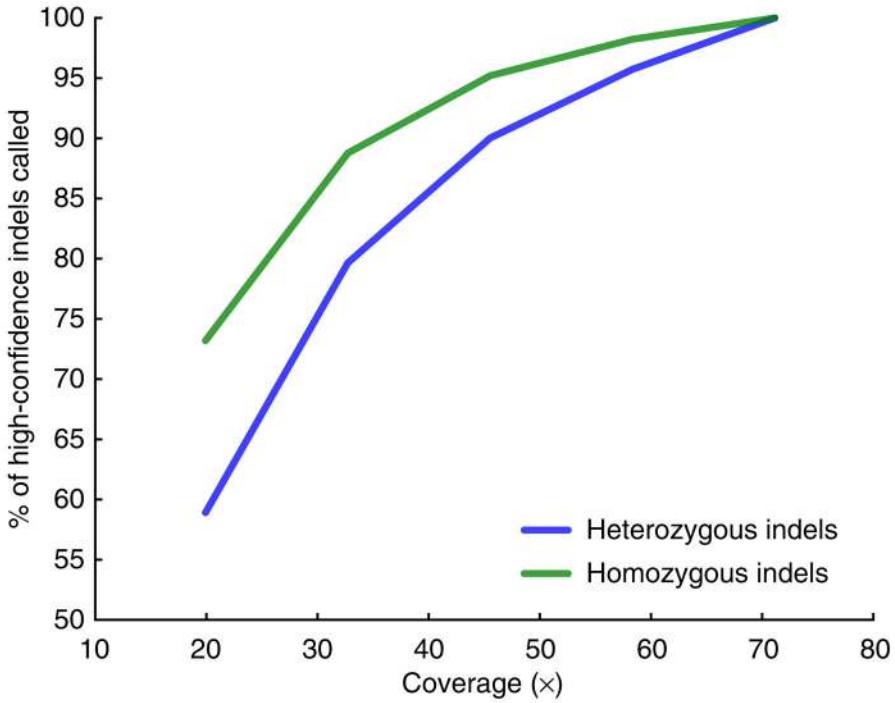

Figure 4.

Higher coverage can improve Scalpel's sensitivity performance for indel detection with WGS data. The sensitivity performance is assessed using the high-confidence call set shared by WGS and WES data (both Illumina HiSeq2000 platform) from eight samples using all available coverage (70× mean coverage). We down-sampled the reads to a fraction of the original coverage and performed indel calling again. Compared with the original set at 70× mean coverage, we report the percentage of variants that could still be called at a reduced coverage. The *y* axis represents the percentage of the high confidence indels revealed at a down-sampled data set. The *x* axis represents the mean coverage of the eight down-sampled genomes. Among the entire call set, ~61% of the indels are heterozygous and the remaining 39% are homozygous. Performance for heterozygous (blue) and homozygous (green) indel detection is shown by separate curves. Reduced coverage indeed affected the detection of heterozygous indels more than that of homozygous ones.

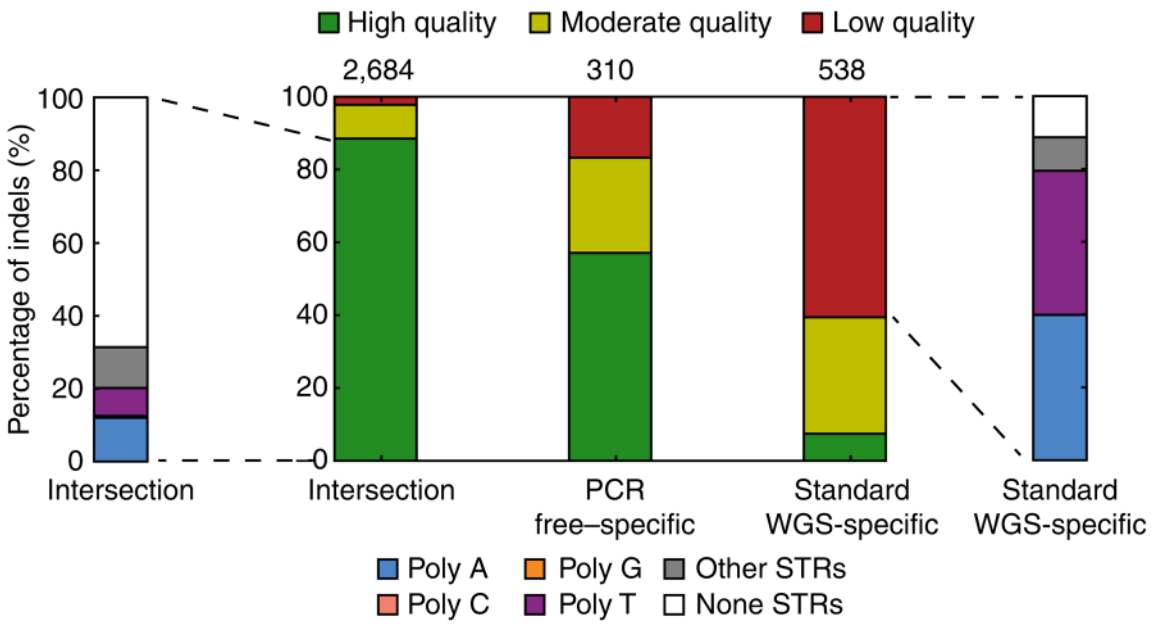

Figure 5.

Comparison of standard WGS and PCR-free data based on indel quality. Indel quality was defined with respect to alternative allele coverage and χ^2 score, which is used in the PROCEDURE and described in Fang *et al.*[18]. 'Intersection' represents the shared indels from both the PCR-free and standard WGS indels. The number reported above a call set represent the total number of indels in that subset; the two data shared 2,684 variants, whereas 310 and 538 were specific to standard WGS and PCR-free data, respectively. Indel calls are further categorized (side-bars) based on their sequence composition: Poly A, Poly C, Poly G, Poly T, other-STR and non-STR. Note: although Poly C and Poly G indels exist in the call-set, their fractions are too minimal to be visualized in the plot. In fact, Poly A, Poly T and non-homopolymer STRs dominate the STR indels. Poly, homopolymer.

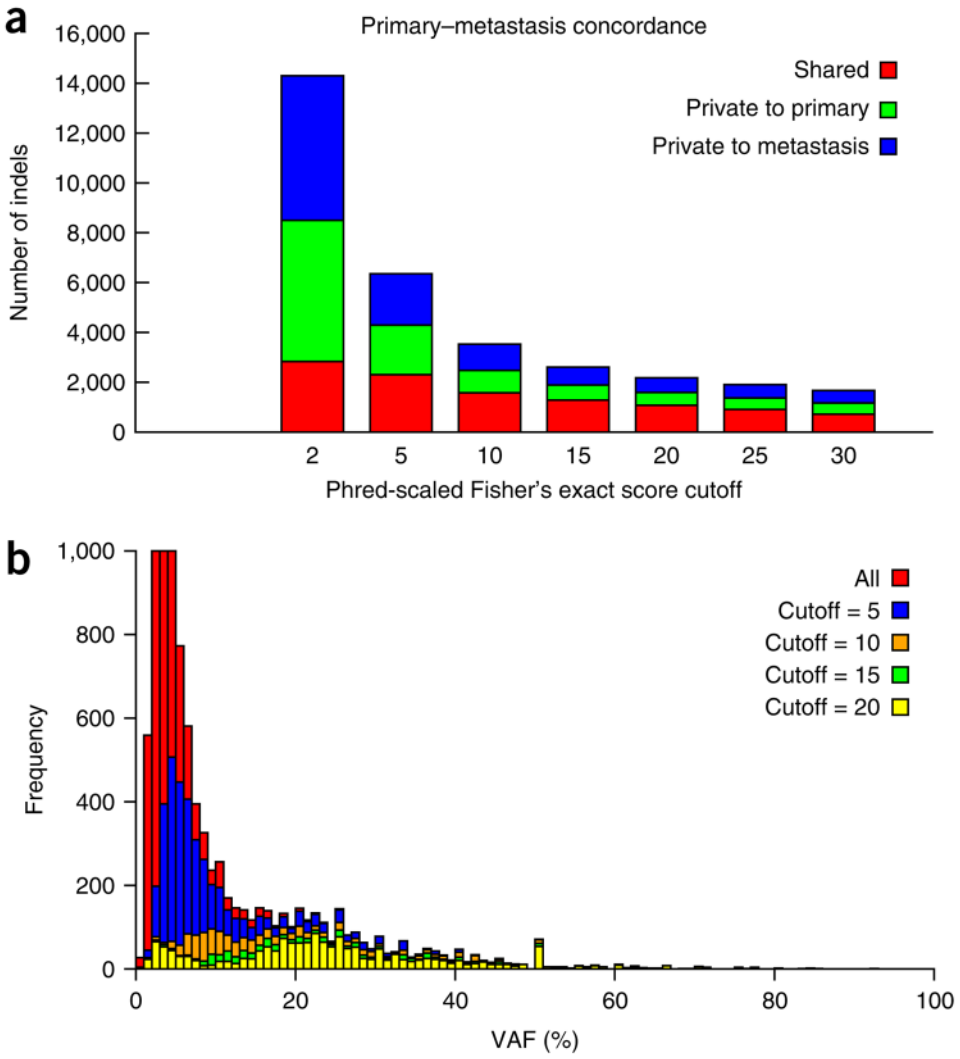

Figure 6.

Whole-genome mutational concordance. (**a**) Concordance and discordant indel mutations as a function of the Phred-scaled Fisher's exact score cutoff between primary and metastasis for a pair of highly concordant colorectal cancer samples from Branon *et al.*[41]. Increasing the Fisher's exact score cutoff substantially reduces the number of private indels while maintaining a similar number of shared ones. This demonstrates the Fisher's exact score's ability to discriminate true mutations from the false-positive ones. (**b**) Distribution of variant allele fraction (VAF) as a function of different Phred-scaled Fisher's exact score cutoffs for the somatic indels detected in the primary tumor. Increasing the cutoff shifts the distribution to the expected 20% VAF for these samples.

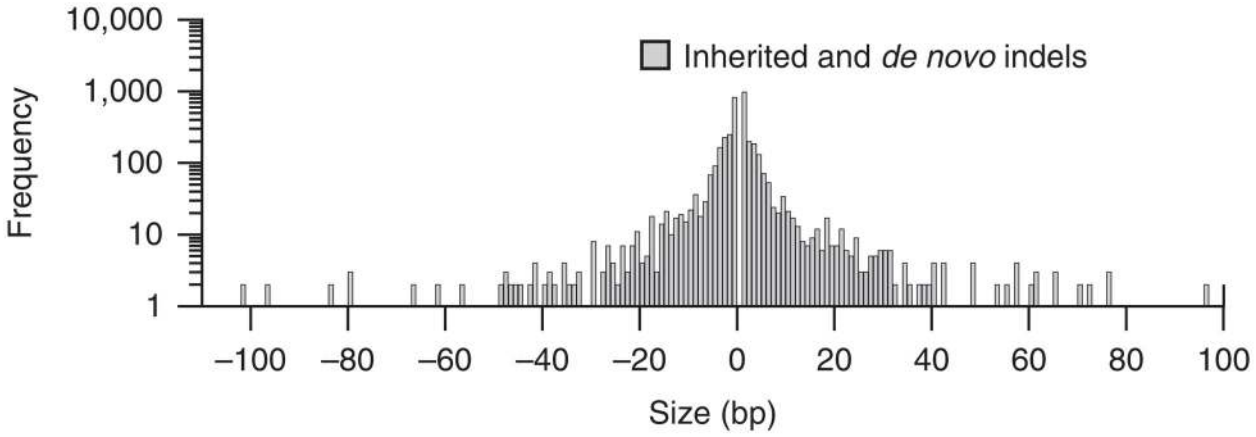

Figure 7.

Size distribution of inherited and *de novo* indels. The *y* axis represents the number of indels, whereas the *x* axis represents the size of indels in base pairs. We should expect a log-normal distribution of indels, with a majority of them being short—i.e., <5 bp in the human exonic regions[16]. This figure was generated using the data from Step 15.

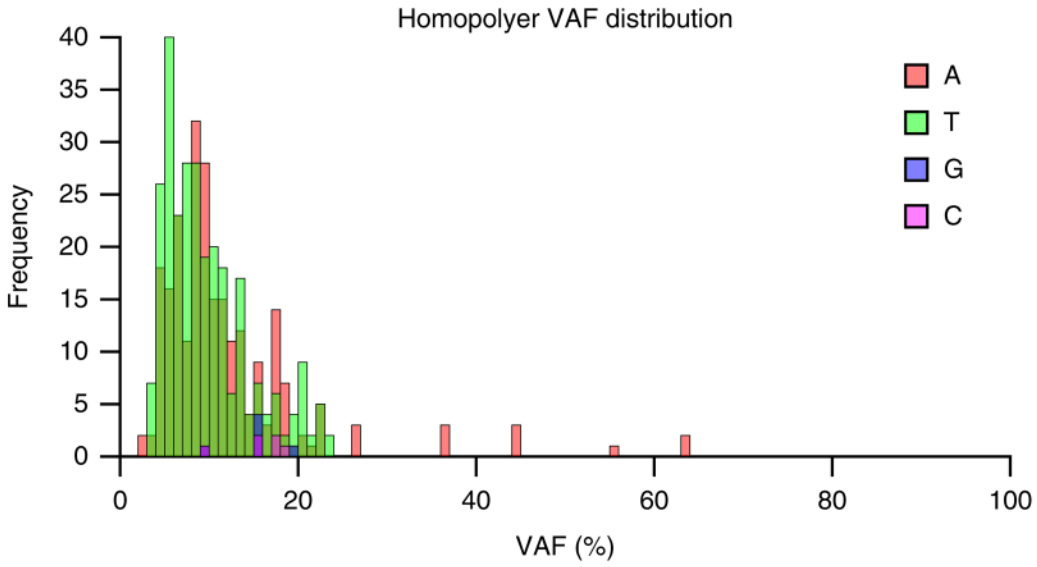

Figure 8.

Histograms of low-quality homopolymer indels by category. The *y* axis represents the number of indels, whereas the *x* axis represents the variant allele fraction (VAF). Homopolymer A or T indels should be more abundant than C or G indels in the call set, especially indels with very low VAF. Due to the limitations of PCR amplification, homopolymer A or T runs are more likely to result in inaccurate molecules[18]. This figure was generated using the data from Step 16.

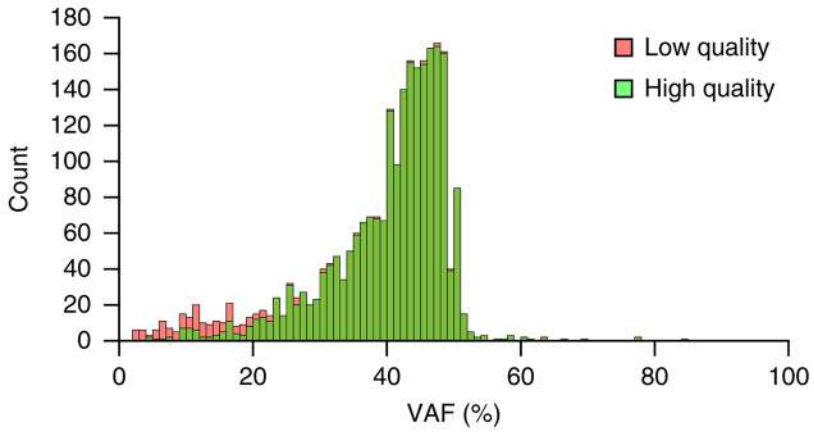

Figure 9.

Variant allele fractions (VAF %) of the inherited indels. Low/high-quality indels here were defined with respect to the coverage and χ^2 scores described in Steps 11 and 12. The VAF of high-quality inherited indels should follow an approximately normal distribution, with a mean of ~50%. In practice, because of sequencing and alignment biases, the mean of the normal distribution is usually slightly less than 50%. Low-quality indels usually have low VAF values, generally tending to be lower than 20%. This figure was generated using the data from Step 17.

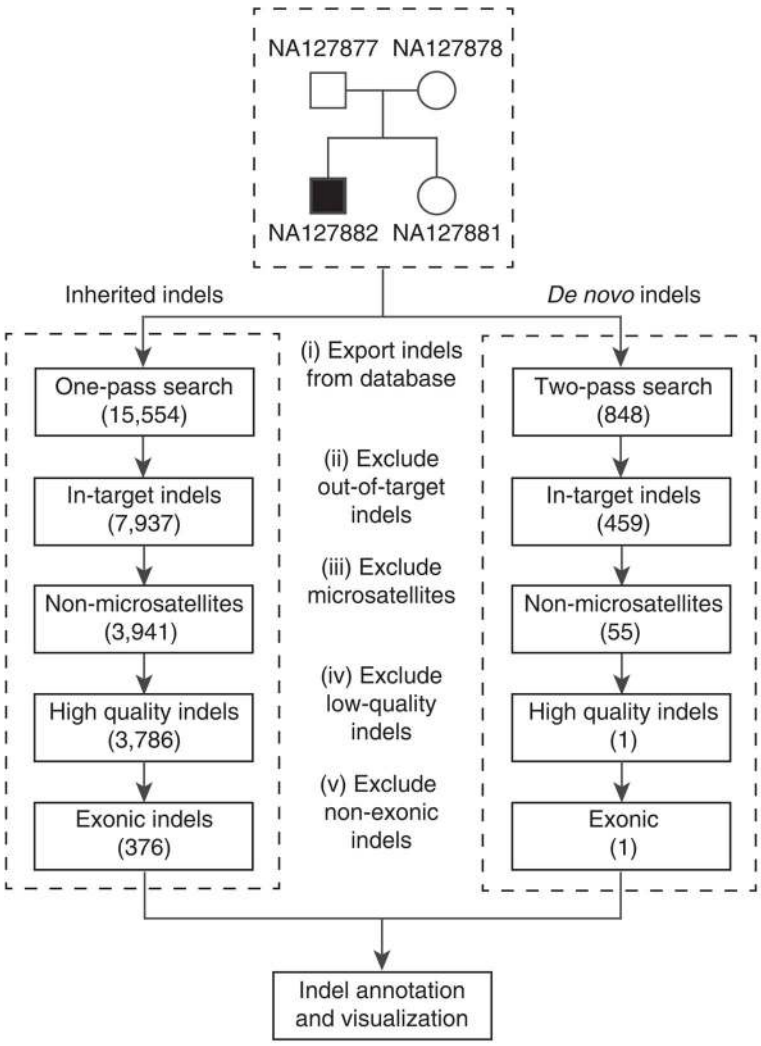

Figure 10.

Filtering cascade of inherited and *de novo* indel calls. The numbers in each box denote the expected numbers of indel calls remaining after filtering. The *de novo* indels underwent a two-pass search to reduce the number of false positives. The numbers in this figure were obtained from Steps 9–12 and 22. It is important to use a two-pass search in *de novo* indel calling, as false-positive calls can be reduced by using a more sensitive parameter setting for the parents' data.

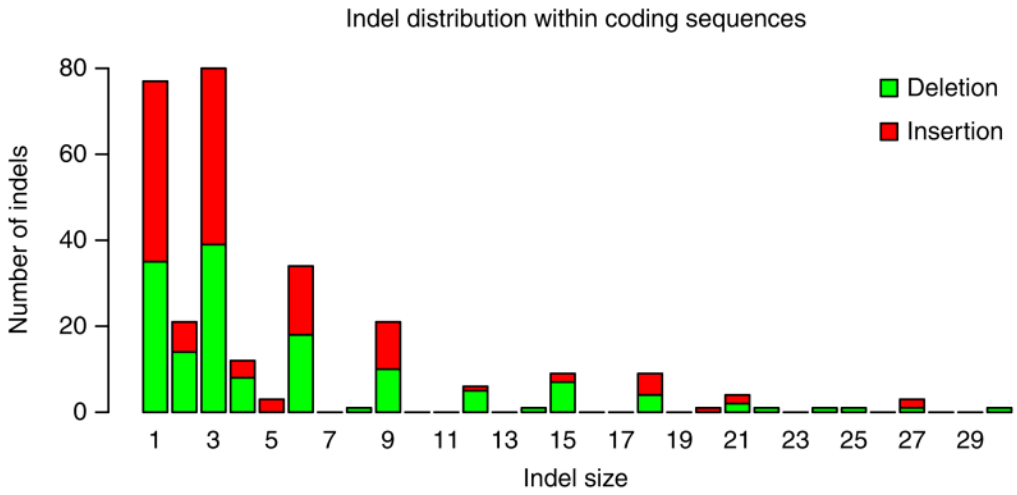

Figure 11.

Frame-preserving indels are more abundant within coding sequences. This figure was generated using data generated by Step 23, which was the set of inherited indels from the proband, NA12882. The *y* axis represents the number of indels, whereas the *x* axis shows the indel size. Stacked bar plots of insertions (red) and deletions (green) are shown in this figure. Indels with a size that is a multiple of three (frame-preserving) are more abundant than the frame-disrupting ones.

Figure 12.

Screenshot of the alignment of the *de novo* deletion in the IGV browser. From the top to the bottom, the alignment is as follows: NA12877 (father), NA12878 (mother), NA12881 (sibling) and NA12882 (proband). The black lines in the alignment of NA12882 show the T deletion in the genome. It is clear from the alignment of the reads that this deletion is present only in the proband and not in any other family members.

TABLE 1

Comparisons and validation rates of indel detection with WGS and WES.

	Indels	Valid	PPV (%)	Indels (>5 bp)	Valid (>5 bp)	PPV (>5 bp; %)
WGS–WES intersection	160	152	95.0	18	18	100
WGS specific	145	122	84.1	33	25	75.8
WES specific	161	91	56.5	1	1	100

The validation rate, positive predictive value (PPV), is computed by the following: PPV = no. TP/(no. TP + no. FP), where no. TP is the number of true-positive calls and no. FP is the number of false-positive calls. Both WGS (mean coverage = ~70×) and WES (mean coverage = ~330×) were performed on Illumina Hiseq 2000 sequencers under 2× 100 bp mode (described in Fang *et al.*18). The construction of WGS libraries here involved a procedure of PCR amplification. The exome capture kit used for WES was NimbleGen SeqCap EZ Exome v2.0, which was designed to pull down 36 Mb of the human genome.

Nat Protoc. Author manuscript; available in PMC 2017 July 12.

TABLE 2

Expected QC-passed read and mapping statistics.

Sample	QC-passed reads	Duplicated reads	Duplicated rate (%)	Mapped reads	Mapping rate (%)
NA12877	1,629,579,046	39,439,277	2.42	1,618,796,107	99.34
NA12878	1,578,485,183	36,266,744	2.30	1,568,334,656	99.36
NA12881	1,559,137,724	39,169,529	2.51	1,547,550,351	99.26
NA12882	1,617,281,311	38,592,443	2.39	1,607,709,559	99.41
Average	1,596,120,816	38,366,998	2.40	1,585,597,668	99.34

QC, quality control.

Nat Protoc. Author manuscript; available in PMC 2017 July 12.

TABLE 3

Troubleshooting table.

Step	Problem	Possible reason	Solution
4	'Bash' cannot find the command for a tool	An incorrect directory for the tool was exported into $PATH	Make sure that the root directory containing the executable files is exported into $PATH; otherwise, just use the absolute path of the tool
7	Low mapping rate (<90%)	Untrimmed barcodes on reads, poor-quality reads, contamination or invalid mapper settings	Evaluate the read quality using FastQC and trim if necessary. Reads that fail to map can be realigned with more sensitive settings. Reads that still fail to map are probably contamination and can be safely ignored
9	There are very few inherited indels in the outputs and/or there is an excessive number of *de novo* indels Excessive numbers of *de novo* indels	An incorrect database might have been used Poor-quality reads require most stringent filtering	Use the 'main' folder for exporting inherited indels and use the 'two-pass' folder for exporting *de novo* indels Adjust the minimum coverage and χ^2 parameters to reduce the number of false-positive calls
11	The number of variants reduces after running 'ms-detector'	'ms-detector' might have introduced blank space into some columns	Make sure that you follow the protocol commands and use '-F '\t'' in the awk script
14	'gnuplot' is not producing the figure	Some earlier versions of 'gnuplot' do not have the necessary functions	Use 'gnuplot' v0.44 or later

TABLE 4

A list of inherited frameshift mutations that are not reported in 1000G ExAC or ESP databases.

Type	Gene	Chr	Start	Ref	Alt	Zygosity	Validation status
Del	OR2T2	Chr1	248617057	TGATCAGGAAGGGCTAGCAGGGACTCCCACAGCATCAGAGTGGTGACTGTGATCGGGAAGGATTAGCGGGGACTCCCAGAGCATCAGGGGTGGTGAC	—	Het	Confirmed[a]
Del	POLR1B	Chr2	113300002	TCCGGCGTGTACCGAGAGACTGGCG	—	Het	Validated
Ins	ZNF806	Chr2	133075904	—	A	Het	Confirmed[a]
Del	ZNF806	Chr2	133076118	A	—	Het	Validated
Ins	AGAP3	Chr7	150783918	—	G	Hom	NA[b]
Del	MYO7A	Chr11	76895771	GGAGGCGGGGACACCAGGGCCT	—	Het	Validated

The build of annotation database for Annovar used here is v20150413. We performed Sanger validation of these loci (6 × 4 = 24) in all family members. With the exception of the locus that did not yield PCR amplicons for Sanger validation, the remaining 20 mutations were successfully validated/confirmed. Chr, chromosome; Del, deletion; Het, heterozygous; Hom, homozygous; Ins, insertion; NA, not applicable.

[a]A status of 'validated' means that the Sanger experiments returned clear signals in all four people and validated the variant calls. A status of 'confirmed' means that the experiments yielded low Sanger signals in the carriers, whereas the experiments (correctly) yielded no Sanger signal in the noncarriers. The low Sanger signal is because two confirmed mutations are located in one of the class LINE repeats of the human genome (Supplementary Results; Supplementary Methods).

[b]Because of the extremely high GC content (86% in a 100-bp window) near chr7:150783918, the PCR did not yield amplicons and thus Sanger validation for these loci was not possible.

TABLE 5

Overview of the *de novo* deletion in the affected child.

Type	Gene	Chr	Start	Ref	Alt	Zygosity	Sanger validation
Del	HFM1	Chr1	91859889	T	—	Het	Validated

A 1-bp heterozygous frameshift deletion was detected in exon 4 of gene *HFM1*. Sanger validation confirmed that the *de novo* deletion in *HFM1* is present only in the proband, and is not seen in any other family member (see Supplementary Results and Supplementary Methods for more information). Chr, chromosome; Del, deletion; Het, heterozygous.

Nat Protoc. Author manuscript; available in PMC 2017 July 12.