doi: 10.1093/narcan/zcaa034

Unique *k*-mer sequences for validating cancer-related substitution, insertion and deletion mutations

HoJoon Lee[1,†], Ahmed Shuaibi[1,†], John M. Bell[2], Dmitri S. Pavlichin[1] and Hanlee P. Ji[1,2,*]

[1]Division of Oncology, Department of Medicine, Stanford University School of Medicine, Stanford, CA 94305, USA and [2]Stanford Genome Technology Center, Stanford University, Palo Alto, CA 94304, USA

Received June 23, 2020; Revised October 23, 2020; Editorial Decision November 9, 2020; Accepted November 12, 2020

ABSTRACT

Cancer genome sequencing has led to important discoveries such as the identification of cancer genes. However, challenges remain in the analysis of cancer genome sequencing. One significant issue is that mutations identified by multiple variant callers are frequently discordant even when using the same genome sequencing data. For insertion and deletion mutations, oftentimes there is no agreement among different callers. Identifying somatic mutations involves read mapping and variant calling, a complicated process that uses many parameters and model tuning. To validate the identification of true mutations, we developed a method using *k*-mer sequences. First, we characterized the landscape of unique versus non-unique *k*-mers in the human genome. Second, we developed a software package, KmerVC, to validate the given somatic mutations from sequencing data. Our program validates the occurrence of a mutation based on statistically significant difference in frequency of *k*-mers with and without a mutation from matched normal and tumor sequences. Third, we tested our method on both simulated and cancer genome sequencing data. Counting *k*-mer involving mutations effectively validated true positive mutations including insertions and deletions across different individual samples in a reproducible manner. Thus, we demonstrated a straightforward approach for rapidly validating mutations from cancer genome sequencing data.

INTRODUCTION

Next-generation sequencing analysis has been widely adopted in cancer research for identifying mutations and other genetic aberrations (1). For example, The Cancer Genome Atlas (TCGA) Project has relied on exome sequencing to identify numerous driver mutations in over 30 cancers. These catalogues of cancer genetic alterations provide insight into the underlying mechanisms of cancer and are used clinically for predicting response to certain therapies and have prognostic implications (2–4). However, the analysis of cancer genome sequencing data relies on human genome assemblies for reference alignment. Since mapping sequence reads enables the identification of mutations, their positions and their allelic fractions, accurate variant analysis depends on alignment reference mapping accuracy.

Although cancer genome sequencing has become routine for biomedical research studies and diagnostic genetic testing of tumors, there are significant challenges in accurately identifying cancer mutations. The complexity of this task is evident in the discordant results produced by different mutation callers (5). The relatively sparse overlap among various mutations callers is a major dilemma and directly stems from the use of the human reference genome for sequence alignment (6). The human reference build is a static representation of assembled sequences derived from the genomes of 13 individuals, encompasses only a small proportion of human genome diversity and lacks feature indicating structural complexity. The broad spectrum of novel and complex somatic alterations present in cancer genomes is often missed or misclassified due to these limitations of the reference genome. For example, short sequence reads unique to a specific tumor genome and containing a novel mutation may prove to be unmappable and thus are excluded from analysis. Unknown homologous or paralogous genes are similarly problematic due to uncertain mapping locations of the genomic reads.

A major challenge in alignment-based variant calling is the identification of insertions and deletions (indels); this challenge is intrinsically related to alignment scoring metrics. Single-nucleotide variants such as substitutions have a lower alignment penalty score than indels. As a result, reads with substitutions have higher alignment scores compared to reads with indels. Lowering the penalty for indels does not resolve this issue (7). For example, we observed that modifying the penalty threshold resulted in the interpretation of true substitutions as one-base indels whenever adja-

[*]To whom correspondence should be addressed. Tel: +1 650 721 1503; Fax: +1 650 725 1420; Email: genomics_ji@stanford.edu
[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

cent bases matched the alternate allele. Altering these settings leads to the calling of spurious indels for ∼25% of the substitutions. Furthermore, variant calling programs such as GATK (8), Mutect2 (9) and VarScan2 (10) require statistical models to identify mutation events from mapped reads. Despite these programs' ability to identify true positive mutations, there are issues with reproducibility when making comparisons among different callers or even when using the same caller repeatedly (11,12). For example, repeating the analysis for discovering mutations can lead to a different set of variant calls despite starting from the same dataset of mapped reads.

For this study, we examined the properties of *k*-mers, short segments of DNA sequence, that include somatic mutations identified in cancer genome sequencing data. Our goal was to determine the *k*-mer properties of somatic mutations and to assess the properties of unique *k*-mers for validating true versus artifactual mutation calls. Previous studies have utilized *k*-mers to analyze sequences from organisms lacking a complete reference genome (13). *K*-mers are used in sequencing alignment programs such as BLAST, BLAT, BWA (14–16) and RNA-seq analysis (17,18). Some studies have used *k*-mers to genotype known variants (19), capture reads for efficient target mutation validation from targeted sequencing (20) or construct local assemblies for identifying somatic variants in tumor samples (21). However, none of these previously published studies has conducted a thorough, systematic evaluation of the property of *k*-mers related to cancer mutations found in exome or whole genome sequence data.

For this analysis, we developed a computational tool, KmerVC, that determines the properties of *k*-mers such as their uniqueness in the human genome reference. Moreover, we determined which specific *k*-mer properties defined mutations as somatic. After obtaining *k*-mer counts, we used a binomial statistical test to validate cancer mutations and given mutation calls. Our results suggest that using *k*-mers without a conventional static reference has the potential for first-pass mutation calling.

MATERIALS AND METHODS

We implemented KmerVC as a Python 3.6/2.7 software package consisting of a command-line program, kmervc.py, and a reusable library, kmervclib. The GitHub repository is open access at https://github.com/compbio/kmerVC. The KmerVC program requires three inputs: (i) a list of mutation calls of interest in VCF or BED file format; (ii) the reference genome sequence; and (iii) the primary sequencing data in FASTQ or FASTA format.

The overall analysis pipeline of KmerVC

The overall structure of KmerVC is outlined in Figure 1. Our tool has five steps: (i) pre-processing to assess the uniqueness of a *k*-mer in the human genome; (ii) counting *k*-mers in sequencing data; (iii) retrieving expected *k*-mers from regions surrounding mutation calls of interest; (iv) compiling *k*-mer counts related to called mutations from a variant caller; and (v) validating true positive mutations.

Pre-processing. With the default settings, we used JELLYFISH, a *k*-mer frequency counting software to obtain *k*-mer counts (22) using the following command-line call:

```
jellyfish count -m 31 -s 100M -t 24 –C –o grch38_31mer.jf grch38.fa
```

Counting k-mers. Accounting for multiple FASTQ input files, we obtain *k*-mer counts using the following command-line calls:

```
jellyfish count -m 31 -s 100M -t 24 -C -o normal.jf -F 2 normal_R1.fastq normal_R2.fastq
jellyfish count -m 31 -s 100M -t 24 -C -o tumor.jf -F 2 tumor_R1.fastq tumor_R2.fastq
```

Retrieving expected k-mers. We extracted the sequence regions and generated a set of *k*-mers encompassing each mutation. This process occurs with a list of mutation coordinates used to generate a BED file. For a given mutation coordinate and a specific *k*-mer length denoted by *k*, each segment is defined by the mutation coordinate minus (k – 1) and plus *k*. With this BED file, we extract the corresponding sequences from the reference genome using the genomic analysis toolkit, BedTools. Then, we generated a FASTA file of the surrounding sequence regions with the following command:

```
bedtools getfasta -fi reference.fa -fo regions.fa -bed regions.bed -name
```

The BedTools getfasta functions through indexing the reference genome FASTA file and quickly identifying the specific sequence segment. We selected the use of Bedtools over another genome analysis toolkit Samtools, due to its faster runtime efficiency. Using the segment of sequence surrounding each variant, we determined whether the *k*-mers were wild type, representing the reference versus being one derived from mutation. Wild-type *k*-mers were obtained by generating all *k*-mers that cover a segment containing the variant position of interest. Mutation-containing *k*-mers were generated: substituting the specified mutation at the given variant position and similarly generating all *k*-mers that include the mutation such as a substitution. Subsequently, we constructed a list of wild-type and mutant *k*-mers particular to each variant provided as input. This yielded a set of *k*-mers for each variant that we utilize to assess its validation.

Further, where multiple somatic mutations exist in one *k*-mer region, KmerVC validates them independently. However, -m (multiple_mutations) arguments examine two potential scenarios that would result due to the diploid chromosome: (i) both mutations exist on the same chromosome or (ii) each of the mutations exists on different chromosomes. The mutation *k*-mers differ in these two cases and they are validated separately. In addition, we considered a scenario in which there are multiple mutations in a *k*-mer region. In this case, we deal with each pair of consecutive mutations as we would in the case of two mutations in a region as just described. Cases in which more than two mutations are contained in a single *k*-mer region were discarded from further analysis and reported as such.

Compiling. We evaluated the total counts of unique *k*-mers in the reference genome per variant. To identify a vari-

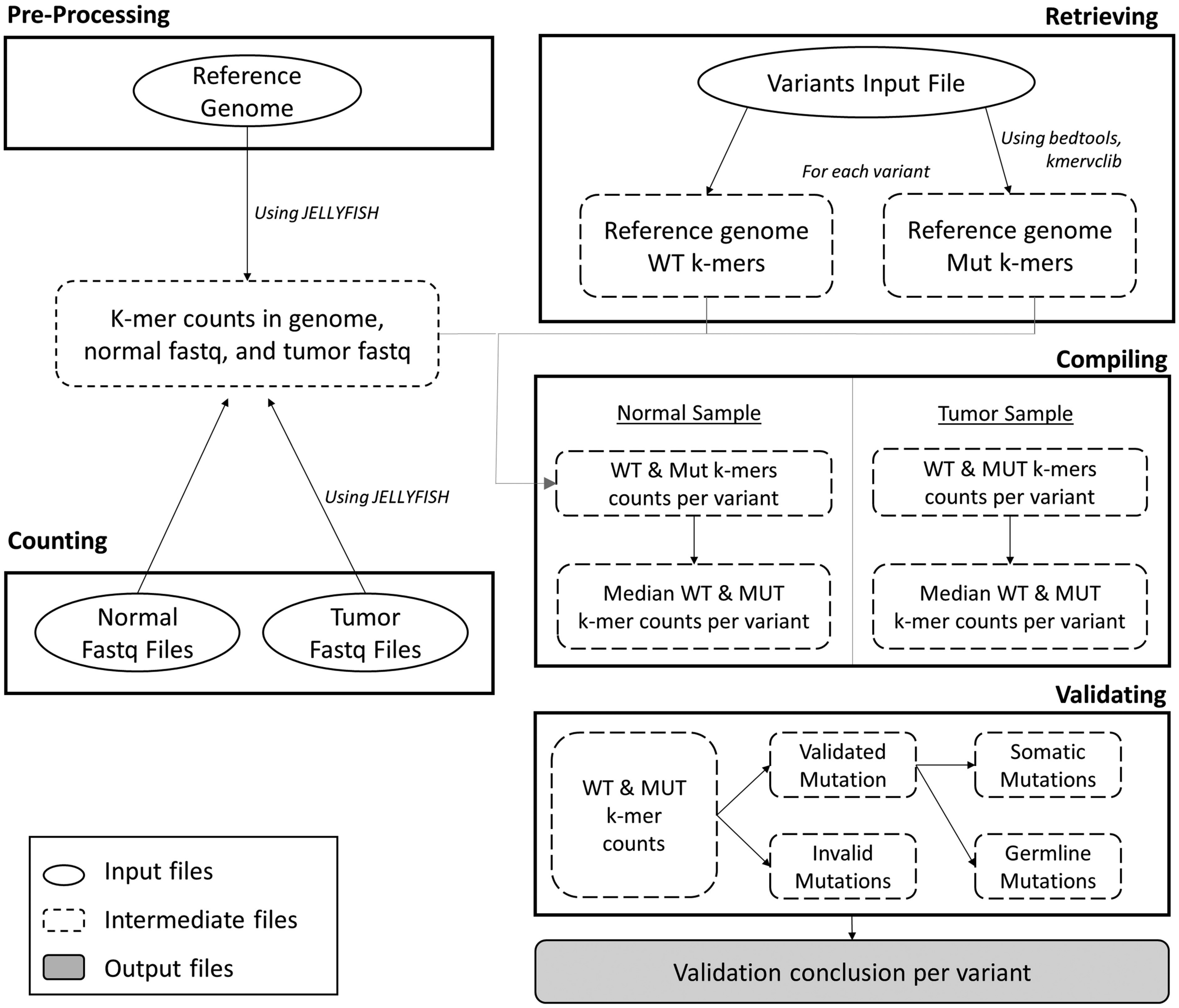


Figure 1. Overview of pipeline. (**A**) Preprocessing. (i) Determining the frequency of every distinct *k*-mer in the reference genome to ascertain its uniqueness. (ii) Determining the frequency of *k*-mers in normal and tumor FASTQ input files using JELLYFISH: a fast *k*-mer counting software. (**B**) Extraction. For all variants, we obtained the surrounding sequence region and generated a set of respective *k*-mers that include the target. Overlapping regions have the variants considered separately and consecutively accounted for in decomposition. Finally, non-unique normal and nonzero mutant *k*-mers are filtered from the sets. (**C**) Compilation. For all variants, we obtained the frequency of the corresponding *k*-mer set from the pre-processed count dictionaries. (**D**) Validation. We assess whether the variants are germline, somatic or otherwise using a binomial test. For the binomial test, we utilize a sequencing error rate of 0.01 and an alpha value of 0.01. We determine whether the median counts of wild-type and mutant *k*-mers are nonequivalent in the normal sample in the first test and whether the median counts of wild-type and mutant *k*-mers are nonequivalent in the tumor sample in the latter.

ant given its surrounding *k*-mers, it is important that a portion of the *k*-mers is unique. We only proceeded with the statistical analysis of variants that had five or more unique *k*-mers in its surrounding region. We calculated the median count for each variant's generated set of wild-type and mutant *k*-mers. We used the median value since it was more robust to sequencing errors and yielded more reliable results.

Validating. We performed statistical analysis using a binomial or Poisson test provided by the scipy.stats Python package. First, we tested whether a candidate variant is germline by determining the difference between wild-type and mutation *k*-mer counts in the normal sample. Thereafter, we determined whether the variant is somatic using the difference between counts of wild-type and mutation *k*-mer counts in the tumor sample. We assumed a sequencing error rate of 1% and utilized an alpha value of 0.01. For the insertion/deletion, we fixed the sequencing error rate as 0.01 regardless of their length, which is more stringent threshold compared to adjusted lower sequencing error rate for longer indels. These hyperparameters were tuned using simulated data although users can choose their own value. In addition, we applied a Bonferroni correction to the alpha value based on the number of tested variants. We validated a variant when the tests (i) failed to reject null hypothesis (i.e. that it is wild type) in the normal sample and (ii) rejected the null hypothesis in the tumor sample, thereby qualifying it as a somatic mutation. In the case that both null hypotheses were rejected, the variant was determined to be germline. In all other cases, the variant was considered an artifact. We required that the count of unique wild-type *k*-mers and the count of unique mutant *k*-mers, which should not be present in the genome, were a threshold of 5 or above. This threshold value ensured that the identified variants could be mapped back to the genome with accuracy and robustness. Calls passing this criterion were determined to be true positive somatic variants and thus validated.

We provided a validity assessment for each input variant regarding its validity as described among one of several categories (Figure 2). A variant marked as 'insufficient' had not met the proper conditions regarding the count of surrounding unique wild-type and unique mutant *k*-mers. As a result, assessment of this variant and its validation were unreliable. A variant marked *SNP_affected* possesses a count of wild-type *k*-mers near 0; thus, the *k*-mer properties were insufficient to proceed with the validation.

We consolidated the software's analysis and results into one final summary (Figure 2). In the final table, we organized the *k*-mer counts by mutations and determine whether the counts of mutant *k*-mers sufficiently verified the mutations using binomial tests that assume a sequencing error rate of 1%. The summary table has reporting headers and 17 columns (Supplementary Table S1) with information regarding the variants and their surrounding *k*-mer regions.

Reference and sequence data

Reference genome and annotations. We downloaded GRCh38 from the National Center for Biotechnology Information (www.ncbi.nlm.nih.gov). Only the canonical chromosomes and chromosome M were used in the analysis. The coordinates of gaps (N) and repeats were obtained from UCSC Genome Browser: gap.txt.gz. For the definition of coding sequence, we downloaded consensus coding sequence (CCDS) from NCBI. RepeatMasker by Institute for Systems Biology provides the list of the repetitive DNA families. We downloaded hg38.fa.out.gz (RepeatMasker open-4.0.6) from repeatmasker.org/species/hg.html.

Simulated data. We simulated sequence dataset containing substitutions, insertions, deletions and indels from 10 randomly selected regions by BedTools after excluding sex chromosomes and regions with >10% of Ns (Supplementary Table S2). For instance, a segment of human GRCh38 chromosome 1 from positions 629640 to 5629640 (5 Mb) was used as the reference sequence for these *in silico* data. Then, we introduced mutations at 120 random positions into this segment. The indels had six different base pair lengths: 1, 3, 5, 10, 15 and 20. Two different datasets were generated with one representing 'normal' and the other containing the mutations. We used the read simulator wgsim (http://github.com/lh3/wgsim) to generate simulated short paired-end reads with 100 bp length at a depth of 25 (normal) and 50× coverage (mutation-containing). This simulation incorporated a 1% sequencing error model. The following command was used:

wgsim -e0.01 -N625000 -1100 -2100 -r0.0 -R0.0 -X0.0 -S4 chrT.fa seq_R1.fq seq_R2.fq

All simulated sequencing data are publicly available at our website (https://dna-discovery.stanford.edu/publicmaterial/software/kmervc/simulation/).

Exome sequencing data from TCGA. We downloaded exome sequencing data (BAM files) of 50 colorectal cancers from the Genomic Data Commons (GDC) data portal (portal.gdc.cancer.gov). FASTQ files were derived from BAM files using Picard SamToFastq (version 2.9.0). We also downloaded four MAF (Mutation Annotation Format) files generated by Mutect2, SomaticSniper, MuSE, and VarScan2 Variant Aggregation and Masking from the GDC data portal: (i) TCGA.COAD.mutect.03652df4-6090-4f5a-a2ff-ee28a37f9301.DR-10.0.somatic.maf.gz; (ii) TCGA.COAD.somaticsniper.70835251-ddd5-4c0d-968e-1791bf6379f6.DR-10.0.somatic.maf.gz; (iii) TCGA.COAD.muse.70cb1255-ec99-4c08-b482-415f8375be3f.DR-10.0.somatic.maf.gz; and (iv) TCGA.COAD.varscan.8177ce4f-02d8-4d75-a0d6-1c5450ee08b0.DR-10.0.somatic.maf.gz. We only included variants with 'PASS' in the 'FILTER'.

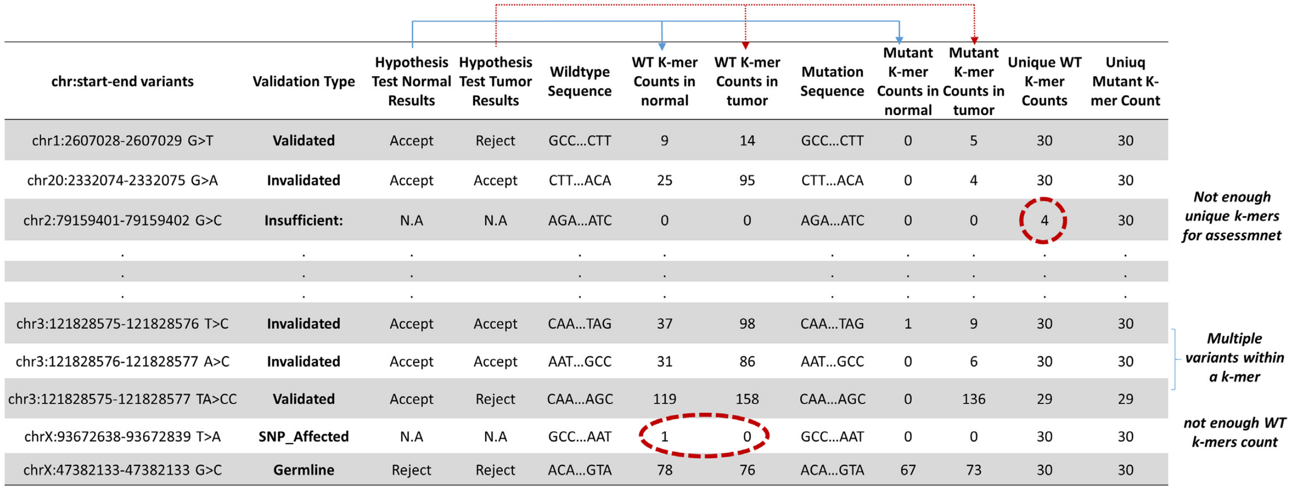
In silico sequence data for indels. To test the performance of our approach for validating indels, we examined a category of sequence that is one of the most challenging cases. Specifically, microsatellites (MSs) are composed of simple tandem repeats (STRs) that are present throughout the human genome. STRs have different classes of repeat motifs that include mono-, di-, tri- and tetranucleotide sequences. MSs are prone to accumulating indel mutations at a high frequency and are extremely difficult to identify reliably given their repetitive structure.

We generated virtual VCF files that could be used as inputs. First, we located MSs by searching for particular mononucleotide repeat motifs. This process involved matching of single-nucleotide repeats against the reference genome. This method gave us a total of 5751 MS regions. Next, we identified insertions or deletions of up to three bases occurring in these MS regions. Therefore, six variant calls were generated: three deletions and three insertions with each of the MS's repeated nucleotide with lengths from 1 to 3 bp. Finally, we created VCF files recoding the simulated indels at these regions and utilized these files as input to the KmerVC to evaluate their significance as plausible mutations. This file, MS_DNA_indels_grch38.vcf, is available at https://github.com/compbio/kmerVC.

RESULTS

Overview of *k*-mer evaluation

Our analysis of *k*-mer counts relied on the analysis of matched samples to confirm candidate mutations (Figure 3). The use of matched normal versus tumor samples enables us to eliminate germline polymorphisms, rare variant and hereditary mutations. We determined the properties of a somatic mutation and how it generates a series of novel (neo) *k*-mers. These sequences are different from *k*-mers present in the reference genome or the matched germline comparison. These neo *k*-mers identify the somatic mutations in the tumor. For instance, a substitution such as a G to T will generate nine neo *k*-mers (length of nine bases) as seen in Figure 3A. We utilized a sliding window to obtain the *k*-mers of the designated lengths spanning the mutation region of interest; this provided both mutation-containing and wild-type *k*-mers. A normal genome yields no mutation *k*-mers, while a tumor sample with mutations yields a significant number of mutant *k*-mers. Similarly, the number of



chr:start-end variants	Validation Type	Hypothesis Test Normal Results	Hypothesis Test Tumor Results	Wildtype Sequence	WT K-mer Counts in normal	WT K-mer Counts in tumor	Mutation Sequence	Mutant K-mer Counts in normal	Mutant K-mer Counts in tumor	Unique WT K-mer Counts	Uniuq Mutant K-mer Count
chr1:2607028-2607029 G>T	Validated	Accept	Reject	GCC…CTT	9	14	GCC…CTT	0	5	30	30
chr20:2332074-2332075 G>A	Invalidated	Accept	Accept	CTT…ACA	25	95	CTT…ACA	0	4	30	30
chr2:79159401-79159402 G>C	Insufficient:	N.A	N.A	AGA…ATC	0	0	AGA…ATC	0	0	4	30
.	.	.		.	.		.	.	.	.	.
.	.	.		.	.		.	.	.	.	.
.	.	.		.	.		.	.	.	.	.
chr3:121828575-121828576 T>C	Invalidated	Accept	Accept	CAA…TAG	37	98	CAA…TAG	1	9	30	30
chr3:121828576-121828577 A>C	Invalidated	Accept	Accept	AAT…GCC	31	86	AAT…GCC	0	6	30	30
chr3:121828575-121828577 TA>CC	Validated	Accept	Reject	CAA…AGC	119	158	CAA…AGC	0	136	29	29
chrX:93672638-93672839 T>A	SNP_Affected	N.A	N.A	GCC…AAT	1	0	GCC…AAT	0	0	30	30
chrX:47382133-47382133 G>C	Germline	Reject	Reject	ACA…GTA	78	76	ACA…GTA	67	73	30	30

Not enough unique k-mers for assessmnet

Multiple variants within a k-mer

not enough WT k-mers count

Figure 2. An example of final summary table. The final summary table with the validation status of variants by KmerVC.

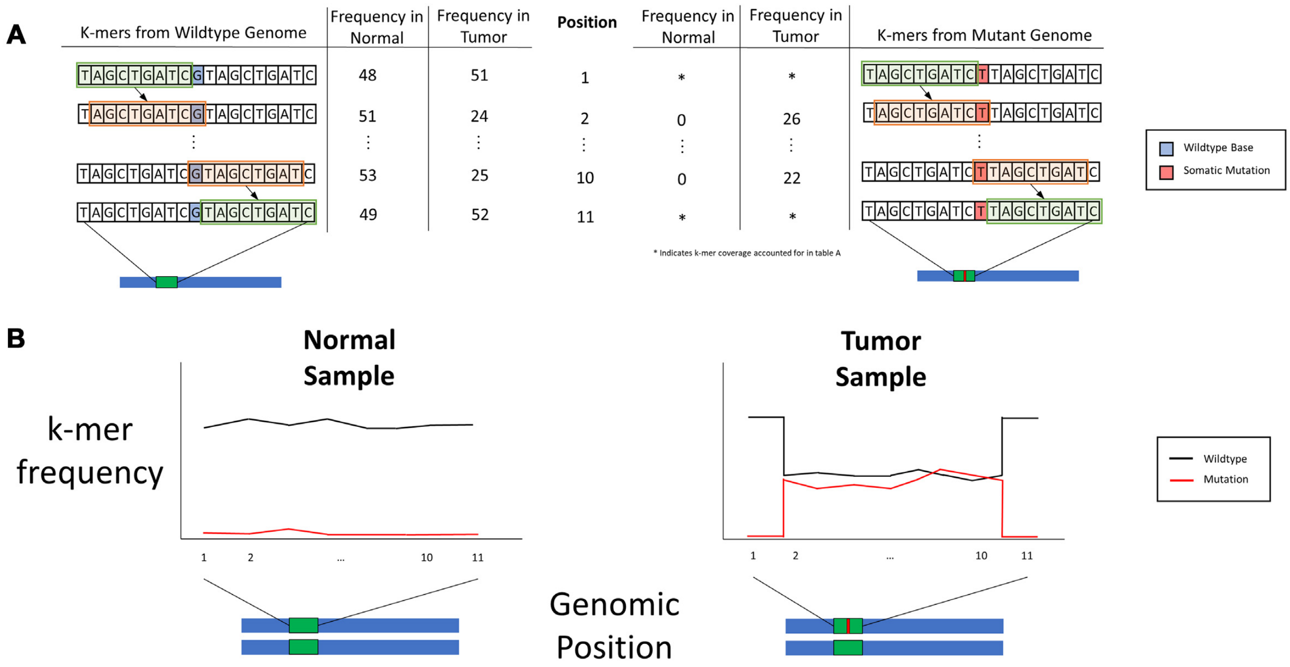


Figure 3. Principle of *k*-mer counts for variant detection. (**A**) The majority of mutations generate neo *k*-mers. (**B**) The counts of *k*-mers are affected by mutations. The mutation is noted by a red bar occurring with an exon that is denoted by a green box.

corresponding wild-type *k*-mers in the tumor will be lower if the mutation is heterozygous and diminished if the mutation is homozygous. For example, with a coverage of 50× the count of mutant *k*-mers is expected to be 50 for homozygous and 25 for heterozygous mutations in the tumor sample, assuming a 100% tumor purity (Figure 3B illustrates a heterozygous example). A difference in the counts of wild-type and mutant *k*-mers between the matched normal and tumor sequencing data allows for the statistical assessment of the validity of a somatic mutation variant.

Evaluation of genome-wide base positions by unique *k*-mers

For this study, one of the most important properties of a *k*-mer is the uniqueness of its sequence compared to the total number of *k*-mers present in the human genome reference (GRCh38). Specifically, this means whether a given *k*-mer appears only once in either the forward or reverse direction of the genome reference. We postulated that mutations appearing in these unique 'neo' *k*-mers have specific properties that make them readily distinguishable (Supplementary Figure S1).

We determined the number of unique *k*-mers in the reference. All *k*-mers in the reference genome were counted using 1 bp increments over its length. When the length of a *k*-mer is ≥19 bases, >90% of distinctive *k*-mers are uniquely represented in the human genome reference (Supplementary Figure S2). When *k*-mers are 31 bases in length, 96.96% of these sequences are unique. Based on our analysis, *k*-mer lengths of >31 bases lead to only minimal increases (<2% with 100-mers) in the proportion of unique *k*-mers among distinctive *k*-mers (Supplementary Figure S2). This result has been confirmed elsewhere (23).

It is important to note that the length of *k*-mers directly determines the total fraction of unique *k*-mers. However, we found that increasing the size of *k*-mers also leads to greater probability of sequencing error artifact. Specifically, we observed that there was a trade-off in that the distribution of longer *k*-mers and their properties could be artifactually skewed by sequencing errors. Shorter length *k*-mers were less prone to artifact from these errors. Another study confirmed this issue (18). Thus, our software allows users to select different *k*-mer lengths for their studies.

Having identified 31-mers as a length with suitable properties, we identified the total number of detectable bases within the human genome reference. We used the following definition of unique *k*-mer as related to individual base positions in the reference. If one examines any given position in the human genome reference, there are a total of 31 *k*-mers (length = 31) that overlap this base. Stating it differently, a sliding window provides a series of *k*-mers that cover a given base position B with the first having B as its last position and the last having B as its first position. So long as one of these 31-mers is unique, this base is considered mappable. For our analysis, we defined several terms. We defined a 'detectable base' as one that overlaps five or more unique 31-mers. This metric provides robust and unique mapping from multiple 31-mers even when they contain regions with sequencing errors.

Similarly, we conducted an extensive analysis of genome positions that are not mapped with unique *k*-mers. We used the term 'dark base' to refer to a base position that is covered by less than five unique 31-mers (Figure 4A). We used the term 'dark region' to refer to a segment of sequence containing two or more adjacent dark bases. These are segments of genome sequence that may have issues aligning correctly to the reference. We provide the coordinates of dark regions as BED files (dark_bed.zip), which can be downloaded from https://dna-discovery.stanford.edu/publicmaterial/software/kmervc/.

Based on these definitions, we determined that 86% of the reference genome bases can be mapped using unique *k*-mers. As an added filter, we eliminated the unknown bases (typically annotated with the character N), which comprise 5.3% of the reference genome and make up a total of 164 Mb. As a result, we found that 87.78% of the known bases in genome reference are detectable by at least five or more unique 31-mers. This high level of overall coverage based on unique *k*-mers is remarkable given that more than two-thirds of the human genome is composed of repeat elements and low-complexity regions (24). As expected, most (96%) of dark regions were located within repetitive DNA families. Mostly, short interspersed nuclear elements, long interspersed nuclear elements and satellite DNAs contribute 38.3%, 30.6% and 20.1% of dark regions, respectively.

If one considers only exon regions, which total ∼33 Mb according to CCDS (25), 95.6% of the exon bases are detectable by at least five or more unique 31-mers. The vast majority (79.5%) of genes do not contain any dark regions and 89.3% of genes have <10% as dark regions. The exons of established cancer genes such as *TP53*, *APC* and *KRAS* have a very high portion (96.2%) of detectable bases on average (100% by median). In general, the genes with dark regions belong to families (Supplementary Table S3).

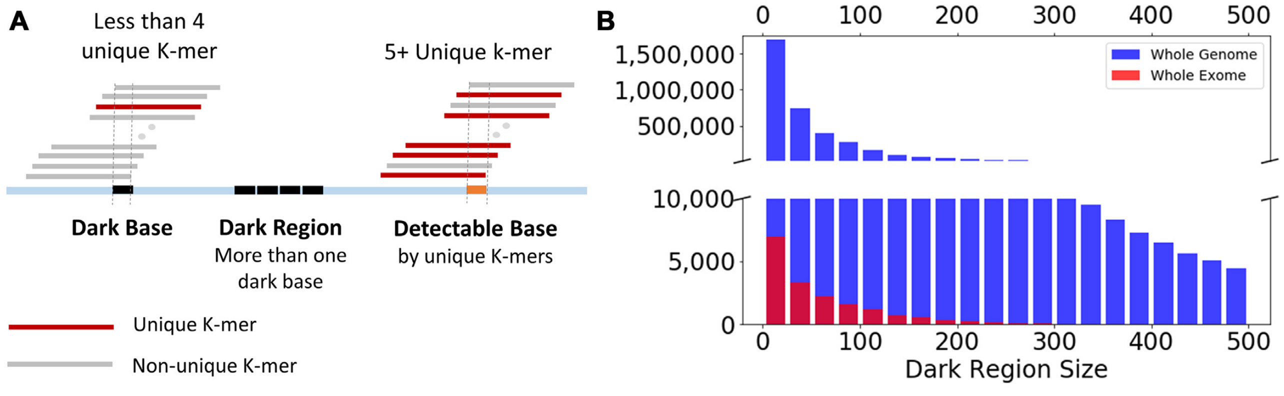
We determined that these dark regions (i.e. gaps in detectable bases) generally were <150 bases in length (Figure 4B); this length is shorter than the typical sequence read from an Illumina system. We identified adjacent unique 31-mers as either upstream or downstream anchors of the dark region within the length of a single sequence read (i.e. 150–300 bases) or within the insert size distribution that occurs for paired-end reads (i.e. 150–500 bases). With this extended definition of detectable base using pair-end reads, we found that the detectable regions of whole genomes and exomes increased from 87.78% and 95.68% to 93.84% and 98.60%, respectively (Figure 4C). With this observed increase in exon detectable bases, there were no dark regions found in 17 996 (94.1%) of the 19 124 genes annotated per CCDS. If one considers the long sequence reads available from Oxford Nanopore or Pacific Biosciences, these dark regions can be further reduced in size and more bases can be identified using our *k*-mer approach. This prediction can be seen in the theoretical example of sequence read lengths of 1 kb (Figure 4C).

There are other resources that consider the 'mappability' of *k*-mers. For example, the Umap resource available from the UCSC Genome Browser provides mapping information for sequences of varying lengths, ranging from 24 to 100 bases (26). This feature is used for determining which sequences can be accurately aligned to the reference genome using *k*-mers. Our method is different in several ways. First, we define a single base as 'detectable' if it overlaps with five or more unique *k*-mers. Using Umap's formula, our 'detectable' base has the minimum 'mappability' of 0.167 (=5/30). We use this definition of a 'detectable' base to validate somatic mutations, which has not been described for Umap. Our method also allows us to introduce and characterize the 'dark regions' of the genome where conventional mapping could be problematic.

Validation of simulated variants from *in silico* sequence data

We evaluated the feasibility of *k*-mer counts to validate the identification of a ground truth set of mutations generated *in silico*. We applied the KmerVC program to a simulated sequencing dataset of a wild-type genome with three simulated sequencing data files generated from the mutated genome that contain (i) 120 substitutions, (ii) 120 insertions and (iii) 120 deletions, respectively (see the 'Materials and Methods' section). We measured the performance by TP, FP and FN, where TP is true positive, FP is false positive and FN is false negative. TP indicates the number of correctly validated simulated mutations. FP indicates the number of validated mutations that were not one of the simulated ones. FN indicates the number of simulated mutations that were not validated.

We generated a ground truth set of mutations embedded within *in silico* sequence data. Categories of different mutations included (i) 120 simulated substitutions + 380 random substitutions + 1352 raw variants from GATK 3.8, (ii) 120 simulated insertions + 380 random insertions + 1346 raw



C

Filtering	Whole genome coverage %		Whole exome coverage %	
	Mappable base with one+ unique 31-mer	Detectable base with 5+ unique 31-mers	Mappable base with one+ unique 31-mer	Detectable base with 5+ unique 31-mers
Excluding unknown bases (N's)	89.42	87.78	97.52	95.68
Excluding dark regions < 200 bp	94.34	93.84	99.52	98.60
Excluding dark regions < 1000 bp	96.92	96.80	100.0	99.65

Figure 4. The characteristics of detectable and dark. (**A**) Definition of detectable and dark base/region. (**B**) The distribution of size of dark regions. (**C**) Coverage of unique *k*-mers in GRCh38 after excluding dark regions.

variants from GATK 3.8 and (iii) 120 simulated deletions + 380 random deletions + 1362 putative variants from GATK 3.8. Substitution mutations were provided to the program as a BED file. We provided the actual sequences with the indel mutations seeing that microhomology motifs complicate the use of coordinates for reporting the location of either insertions or deletions.

Figure 5 shows the number of TPs, FNs and FPs of KmerVC for simulations based on an alpha value of 0.01 with multiple testing correction. We tested different lengths of *k*-mers and found that 31-mers performed better than 21-mers, but there was no significant improvement with 41- or 51-mers (Supplementary Figure S3). However, we observed the trade-off between shorter and longer *k*-mers. For instance, we observed that some of variants were validated by 31-mers, but not by 51-mers because the number of mutant 51-mer counts was reduced as longer *k*-mers are likely to incur a higher frequency of sequencing errors with miscalled bases (Supplementary Table S4). To see whether genomic composition affects the performance, we conducted analysis on other simulation dataset derived from nine randomly selected regions. We observed similar outcomes (Supplementary Table S2).

Furthermore, we examined the performance by different thresholds of minimum unique *k*-mers, which we set 5 as default (Supplementary Table S5). There was no significant difference in TPs, FNs and FPs between the different thresholds of minimum unique *k*-mers. This is expected given that there is only 1.64% of the entire genome that is mappable (by one or more unique *k*-mers) but not detectable (by five or more unique *k*-mers). In fact, while the difference is minimal, the threshold of 5 generates the best performance in substitution simulation (see Supplementary Table S5) compared to 1 or 3, because it is more desirable to have smaller number of FPs than FNs. In the case of insertion simulation, we saw that the better performance in the lesser thresholds was due to our longest insertions (20 bp), which indicates that the insertions that can be reliably validated by 31-mers are <20 bp. As such, we conclude that the maximum size of insertion that can be reliably validated is half of the selected *k*-mer size. Lastly, the simulations of deletions performed equally well regardless of the threshold. It is expected that longer insertions suffer more from stringent threshold than deletions and substitutions because an insertion generates [k − (size of insertion) − 1] mutant *k*-mers, while a deletion and a substitution generate (k − 3) and k mutant *k*-mers, respectively. As a result, we believe that the threshold of five or more unique *k*-mers provides optimal performance. Users also have the option to change this parameter within the framework of KmerVC if necessary.

From the ground truth sequence data, we used Mutect2 to call mutations. This program was run in both its GATK 3.8 and GATK 4 versions. Because the upgrades of GATK 4 involved improvements in specificity of variant calling, we observed a significantly lower performance in detecting indels in the simulated data. This calling decrement was apparent even after extensive filtering. Therefore, we used variant calls from GATK 3.8. The performance of the Mutect2 pipeline is shown in Figure 5. When we analyzed the GATK

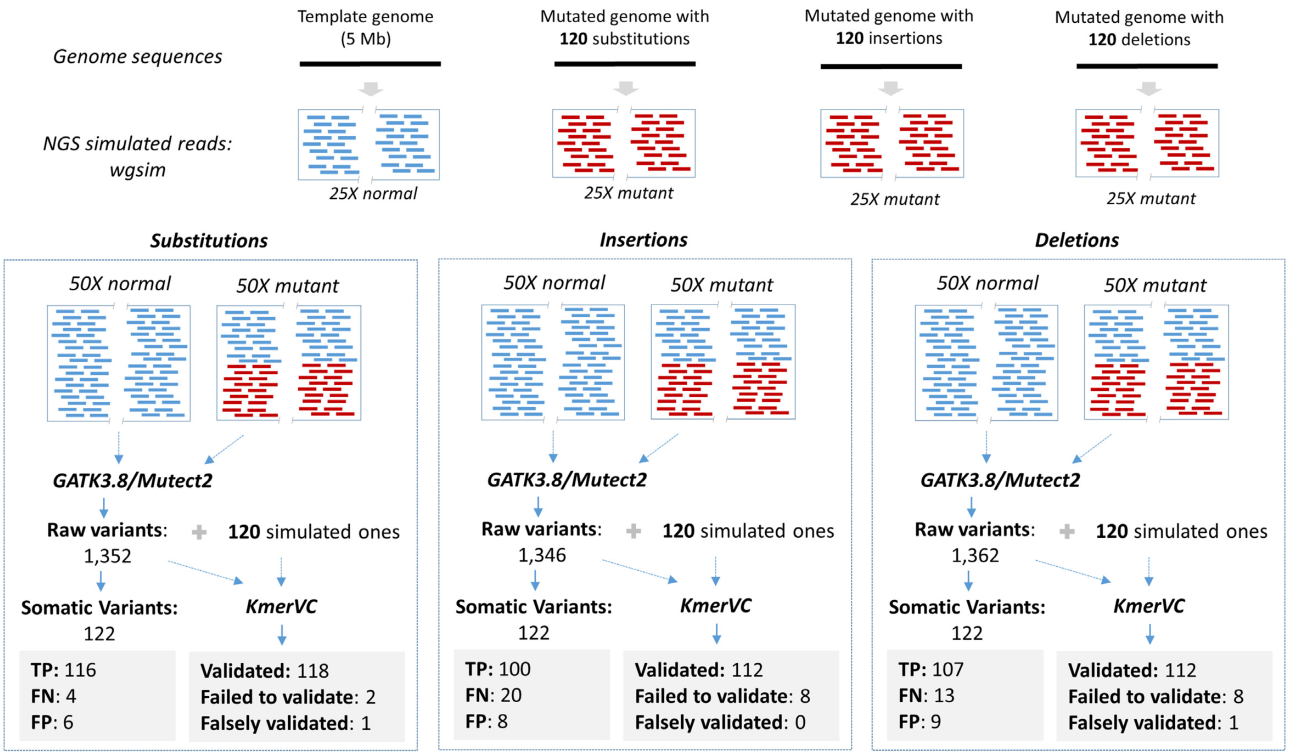


Figure 5. Validation of simulated substitutions, insertions and deletions in sequencing data.

substitution variants, Mutect2 called 116 TPs, 4 FNs and 6 FPs. Of the four FNs, one was not called at all while three did not pass the clustered_events filter (i.e. proximal variant calls failed on the assumption that the clustering is a proxy for mapping artifacts). Against the insertion data, Mutect2 found 100 TPs, 20 FNs and 8 FPs. Two of the FNs were not called at all; 7 failed at least in part because of the clustered_events filter, while 10 failed because they failed to pass the triallelic_sites filter. This situation occurs when at least two candidate alternate alleles are found to be equally likely; since this is unlikely to occur in a tumor, it is assumed to be evidence of an artifact. Against the deletion data, Mutect2 found 107 TPs, 13 FNs and 9 FPs. Here, five TPs were not called, four were filtered by the clustered_events filter, and two were filtered because they involved deletions of tandem repeats. As we noted previously, Mutect2 had had reduced performance for calling insertions with a lower recall rate relative to deletions.

Next, we ran KmerVC to validate the simulated variants along with all the raw Mutect2 calls. Our program confirmed nearly all of the ground truth mutations successfully and invalidated nearly all other tested mutations regardless of variant types (Figure 5). Remarkably, all FPs but two, which were called by Mutect2, were not validated by KmerVC and thus invalidated them properly. Furthermore, KmerVC successfully validated 2 out of 4 FNs in substitutions, 17 out of 20 FNs in insertions and 5 out of 13 FNs in deletions; our program was able to increase the number of TPs from FNs. Overall, these results indicated that our *k*-mer analysis validated TPs while avoiding false calls made by a well-established variant caller.

KmerVC analysis of TCGA exome data

Next, we determined our application's performance for validating mutations from actual exome sequence data representing 50 normal–tumor pairs. We downloaded the matched normal/tumor FASTQ files and VCF files derived from four pipelines (Mutect2, VarScan, MuSE and SomaticSniper), all available from the TCGA data portal. The VCF files for each mutation caller are available separately for each caller pipeline for the 50 cancer samples and separated into two groups: (i) substitutions and (ii) indels. For example, Mutect2 called a total of 21 673 substitutions and 3409 indels across the 50 tumor samples.

Substitutions. We used KmerVC to process the FASTQ and VCF files of each sample. For our first analysis, we examined the Mutect2 calls. Our analysis determined which of 21 673 Mutect2 substitutions could be validated for each sample. For any given normal–tumor pair, KmerVC successfully validated on average 92.4% of Mutect2 variants for an alpha value of 0.01 and 85.6% after multiple testing correction (Supplementary Table S6). The difference in the percentages that were validated is an indicator of the smaller alpha value due to Bonferroni correction; a lower alpha value imposes a more stringent threshold that excludes lower quality variants.

To determine what features distinguished the validated versus non-validated mutations, we conducted a manual inspection of the calls from Mutect2 that were not validated by KmerVC in the tumor sample of TCGA-AA-3350. KmerVC failed to validate 4 out of 111 mutations derived from Mutect2. Three of the four missed mutations were due to proximal single-nucleotide polymorphisms (SNPs), while one was due to the mutation being located at the end of most of the reads.

With an alpha value of 0.01 with multiple testing correction, KmerVC validated 89.6%, 91.2% and 93.4% of variants derived from VarScan, MuSE and SomaticSniper, respectively, on average (Supplementary Table S6). Interestingly, the number of Mutect2 variant calls validated by KmerVC was larger than the number of Mutect2 variants also identified by any of the three other variant callers (Supplementary Table S7).

Insertions and deletions. We analyzed indels among these 50 tumors. Our validation analysis required there to be a flanking base on both sides of the mutation to ensure the length of insertion and the indel's identity. Therefore, the number of affected *k*-mers was extrapolated using the size of the indels. For instance, a two-base insertion in turn overlaps with 28 of the surrounding 31-mers.

We examined all 3409 indels identified in 50 samples. On average, KmerVC validated the greater majority of indels (78.0%) derived from Mutect2 with a corrected alpha value of 0.01. The percentage of validated indels went down to 73.5% if one decreased alpha value by Bonferroni correction (Supplementary Table S8). In conclusion, KmerVC validated most substitution calls as well as indels identified by Mutect2 with a high performance. The KmerVC outcomes (tcga.zip) for all TCGA samples are available at https://dna-discovery.stanford.edu/publicmaterial/software/kmervc/.

Validating MS mutation

KmerVC has the flexibility to validate different classes of mutations. One of the most challenging somatic mutations to identify includes indels present within MS sequences. To determine whether our approach had applicability to verifying MS indels, we generated a VCF file that contains potential indels at 5751 MS DNA in the coding regions (see the 'Materials and Methods' section). KmerVC successfully validated 65 (median) indels at MS DNA per sample ranging from 7 to 658 MS DNA with a corrected alpha value of 0.01 (Supplementary Table S9). We examined sequence alignment plots by Golden Helix GenomeBrowse 3.0.0 (www.goldenhelix.com), and visually confirmed the presence of a set of these indels (Supplementary Figure S4).

DISCUSSION

Our study examined the properties of *k*-mers from the human genome reference. With this information, we developed an application that validates somatic mutation calls for TPs using the characteristics of counts and uniqueness extrapolated from primary sequencing data. Our approach provides a way of rapidly validating indels. One can imagine strategies where the indel threshold can be lowered to improve sensitivity with follow-up KmerVC analysis and validation to improve specificity.

We evaluated the performance of KmerVC using simulated sequencing data and exome data from TCGA. KmerVC achieved high performance by validating >93% of true mutations with nearly zero false validation based on the simulation data for all variant types: substitutions, insertions and deletions. Furthermore, KmerVC successfully validated most of the variants derived from Mutect2 in TCGA exome sequencing data. Remarkably, we observed that KmerVC rarely validates any variants falsely, which is a strength in identifying mutations with implications for disease and diagnosis.

Several studies have utilized the *k*-mer count for identifying previously known germline mutations or indels. However, there are few if any studies examining the application of *k*-mer counts to evaluating somatic mutations in cancer sequencing data. As demonstrated in this study, KmerVC validates the mutation of interest with a straightforward binomial test using only four numbers: (i) wild-type counts in normal; (ii) mutation counts in tumor; (iii) wild-type counts in tumor; and (iv) mutation counts in tumor. The simplicity of this metric enables us to validate variants easily without the need for hidden heuristics or complex optimizations. This feature is extremely useful for identifying cancer mutations of interest when sequencing data are available from multiple different projects. The counts of expected *k*-mers for these mutations can be quickly compared between samples. Any validated variant calls by KmerVC can be compared among different samples easily, providing an added level of quality control.

Our method has some limitations. First, insert sizes larger than the *k*-mer length cannot be detected. For a future study, we plan to extend our approach to detect structural variations, which should generate novel combinations of *k*-mers. These potential features may allow us to identify insertions larger than the *k*-mer size. Second, some regions do not overlap with any unique *k*-mers despite the high number of unique *k*-mers across the human genome reference. Our characterization of the dark regions of the genome highlights our use of unique *k*-mers. Moreover, different lengths of sequence reads that cover these mutations provide a number of opportunities to improve validation analysis. Third, KmerVC is strictly a validation program of variant calls and does not account for FNs. These unaccounted variants may arise from mutation-related *k*-mers that are not unique or expected *k*-mer counts for a given coordinate position that occur as the result of SNPs. The use of matched normal samples provides a way of eliminating SNPs. However, we anticipate that a substantial number of somatic mutations lead to neo *k*-mers that are unique and this property may enable us to identify FNs and provide a way to transition our tool into a full-fledged mutation caller.

In conclusion, for this study, we comprehensively examined the landscape of mappable regions by unique *k*-mers in the human genome. Using the characteristics of *k*-mers, including their uniqueness in the reference and the ease of counting them, provides an excellent way to determine whether called mutations are TPs, especially for indels.

DATA AVAILABILITY

The KmerVC program and all simulated sequencing data are available in the GitHub repository (https://github.com/compbio/kmerVC). All whole exome sequencing data are available at TCGA data portal (https://portal.gdc.cancer.gov/).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Cancer Online.

ACKNOWLEDGEMENTS

We wish to thank Stephanie Greer, Billy Lau and Sue Grimes for discussions and comments on the manuscript.

Author contributions: H.J.L. and H.P.J. conceived and designed the study. A.S., H.J.L. and D.S.P. implemented the computational method. H.J.L. and D.S.P. developed the statistics. A.S. wrote the Python script. J.M.B analyzed simulation data using the different versions of GATK. H.J.L. and H.P.J. analyzed the data. All authors contributed to the manuscript writing. H.J.L. and H.P.J. supervised the project.

FUNDING

National Institutes of Health [2R01HG006137-04 to H.P.J., P01HG00205ESH to J.M.B and H.P.J., U01HG010963 to H.J.L., D.S.P. and H.P.J.]; American Cancer Society [RSG-13-297-01-TBG to H.P.J.]; National Science Foundation [1807371 to H.P.J.]; Clayville Foundation [to H.P.J.].

Conflict of Interest statement. None declared.

REFERENCES

1. Shendure,J. and Ji,H. (2008) Next-generation DNA sequencing. *Nat. Biotechnol.*, **26**, 1135–1145.
2. Biankin,A.V., Piantadosi,S. and Hollingsworth,S.J. (2015) Patient-centric trials for therapeutic development in precision oncology. *Nature*, **526**, 361–370.
3. Chapman,P.B., Hauschild,A., Robert,C., Haanen,J.B., Ascierto,P., Larkin,J., Dummer,R., Garbe,C., Testori,A., Maio,M. *et al.* (2011) Improved survival with vemurafenib in melanoma with BRAF V600E mutation. *N. Engl. J. Med.*, **364**, 2507–2516.
4. Swanton,C., Soria,J.C., Bardelli,A., Biankin,A., Caldas,C., Chandarlapaty,S., de Koning,L., Dive,C., Feunteun,J., Leung,S.Y. *et al.* (2016) Consensus on precision medicine for metastatic cancers: a report from the MAP conference. *Ann. Oncol.*, **27**, 1443–1448.
5. O'Rawe,J., Jiang,T., Sun,G., Wu,Y., Wang,W., Hu,J., Bodily,P., Tian,L., Hakonarson,H., Johnson,W.E. *et al.* (2013) Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing. *Genome Med.*, **5**, 28.
6. Rosenfeld,J.A., Mason,C.E. and Smith,T.M. (2012) Limitations of the human reference genome for personalized genomics. *PLoS One*, **7**, e40294.
7. Mount,D.W. (2008) Using gaps and gap penalties to optimize pairwise sequence alignments. *Cold Spring Harb. Protoc.*, **2008**, pdb.top40.
8. DePristo,M.A., Banks,E., Poplin,R., Garimella,K.V., Maguire,J.R., Hartl,C., Philippakis,A.A., del Angel,G., Rivas,M.A., Hanna,M. *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.*, **43**, 491–498.
9. Cibulskis,K., Lawrence,M.S., Carter,S.L., Sivachenko,A., Jaffe,D., Sougnez,C., Gabriel,S., Meyerson,M., Lander,E.S. and Getz,G. (2013) Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat. Biotechnol.*, **31**, 213–219.
10. Koboldt,D.C., Zhang,Q., Larson,D.E., Shen,D., McLellan,M.D., Lin,L., Miller,C.A., Mardis,E.R., Ding,L. and Wilson,R.K. (2012) VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res.*, **22**, 568–576.
11. Bianchi,V., Ceol,A., Ogier,A.G., de Pretis,S., Galeota,E., Kishore,K., Bora,P., Croci,O., Campaner,S., Amati,B. *et al.* (2016) Integrated systems for NGS data management and analysis: open issues and available solutions. *Front Genet*, **7**, 75.
12. Kanwal,S., Khan,F.Z., Lonie,A. and Sinnott,R.O. (2017) Investigating reproducibility and tracking provenance: a genomic workflow case study. *BMC Bioinformatics*, **18**, 337.
13. Nordstrom,K.J., Albani,M.C., James,G.V., Gutjahr,C., Hartwig,B., Turck,F., Paszkowski,U., Coupland,G. and Schneeberger,K. (2013) Mutation identification by direct comparison of whole-genome sequencing data from mutant and wild-type individuals using *k*-mers. *Nat. Biotechnol.*, **31**, 325–330.
14. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
15. Kent,W.J. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.
16. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
17. Bray,N.L., Pimentel,H., Melsted,P. and Pachter,L. (2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**, 525–527.
18. Patro,R., Mount,S.M. and Kingsford,C. (2014) Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.*, **32**, 462–464.
19. Pajuste,F.D., Kaplinski,L., Mols,M., Puurand,T., Lepamets,M. and Remm,M. (2017) FastGT: an alignment-free method for calling common SNVs directly from raw sequencing reads. *Sci. Rep.*, **7**, 2537.
20. Chen,S., Huang,T., Wen,T., Li,H., Xu,M. and Gu,J. (2018) MutScan: fast detection and visualization of target mutations by scanning FASTQ data. *BMC Bioinformatics*, **19**, 16.
21. Narzisi,G., Corvelo,A., Arora,K., Bergmann,E.A., Shah,M., Musunuri,R., Emde,A.K., Robine,N., Vacic,V. and Zody,M.C. (2018) Genome-wide somatic variant calling using localized colored de Bruijn graphs. *Commun. Biol.*, **1**, 20.
22. Marcais,G. and Kingsford,C. (2011) A fast, lock-free approach for efficient parallel counting of occurrences of *k*-mers. *Bioinformatics*, **27**, 764–770.
23. Li,W., Freudenberg,J. and Miramontes,P. (2014) Diminishing return for increased mappability with longer sequencing reads: implications of the *k*-mer distributions in the human genome. *BMC Bioinformatics*, **15**, 2.
24. de Koning,A.P., Gu,W., Castoe,T.A., Batzer,M.A. and Pollock,D.D. (2011) Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet.*, **7**, e1002384.
25. Pruitt,K.D., Harrow,J., Harte,R.A., Wallin,C., Diekhans,M., Maglott,D.R., Searle,S., Farrell,C.M., Loveland,J.E., Ruef,B.J. *et al.* (2009) The consensus coding sequence (CCDS) project: identifying a common protein-coding gene set for the human and mouse genomes. *Genome Res.*, **19**, 1316–1323.
26. Karimzadeh,M., Ernst,C., Kundaje,A. and Hoffman,M.M. (2018) Umap and Bismap: quantifying genome and methylome mappability. *Nucleic Acids Res.*, **46**, e120.