SCIENTIFIC REPORTS
nature research

OPEN

Denoising of Aligned Genomic Data

Irena Fischer-Hwang[1*], Idoia Ochoa[2], Tsachy Weissman[1] & Mikel Hernaez[3*]

Noise in genomic sequencing data is known to have effects on various stages of genomic data analysis pipelines. Variant identification is an important step of many of these pipelines, and is increasingly being used in clinical settings to aid medical practices. We propose a denoising method, dubbed SAMDUDE, which operates on aligned genomic data in order to improve variant calling performance. Denoising human data with SAMDUDE resulted in improved variant identification in both individual chromosome as well as whole genome sequencing (WGS) data sets. In the WGS data set, denoising led to identification of almost 2,000 additional true variants, and elimination of over 1,500 erroneously identified variants. In contrast, we found that denoising with other state-of-the-art denoisers significantly worsens variant calling performance. SAMDUDE is written in Python and is freely available at https://github.com/ihwang/SAMDUDE.

The ability to sequence genetic material has expanded our understanding of genes and their roles in biological processes, opened up new areas of biological inquiry, and are guiding the trajectory of modern biomedical research[1]. Raw sequencing data are typically stored in the FASTQ file format and converted to the SAM file format following alignment to a reference genome. Both file types comprise sequences of nucleotide bases called "reads," which are accompanied by sequences of quality scores that indicate the sequencing machine's confidence in the base calls making up the reads. However, the genomic sequencing process is imperfect and can result in reads containing various types of noise including base substitutions, insertions, and deletions (INDELs).

Although noise characteristics vary across sequencing technologies, they are well characterized for some sequencing platforms. For example, Illumina sequencing technologies produce "short" reads on the order of hundreds of bases, with an average substitution error rate of less than 1%, and INDEL rates orders of magnitude lower[2]. Furthermore, these errors were found to be correlated with position within the read, resulting in position-dependent noise characteristics. These errors can affect downstream applications, with an important application being variant calling, or the identification of genetic polymorphisms unique to individuals. Variant identification from WGS is increasingly being used for diagnosis and treatment design in the clinical setting, especially in the field of rare genetic disease research[3]. Thus, accuracy of variant identification is paramount. Note that there is a clear distinction between the natural variations of DNA sequences (i.e., genetic polymorphisms)—which are the target of variant calling—and noise due to sequencing errors introduced by the sequencing platform, which can be represented as base call mismatches (single base-substitutions) and INDELs.

Algorithms for removing noise, or denoisers, have been proposed for genomic sequencing data[4], as well as for other biological methods relying on genomic sequencing, like ChiP-seq[5]. These denoisers attempt to rectify sequencing errors by only changing individual bases in reads, while retaining the original quality scores. They are typically tested on simulated and real data sets in FASTQ format, and have been shown to perform well on some of the early stages of genomic sequencing pipelines, such as correcting base calling errors in the simulated data sets, increasing both breadth and depth of reads coverage during alignment[6], or improving de novo assembly of real data sets[7]. However, these analyses often do not extend to later steps in genomic sequence analysis pipelines, and those that do focus on non-human data sets[8]. To our knowledge, none of these works examines the effect that denoising might have on variant calling. Current variant calling procedures are complex and rely on alignment information, including quality scores which are a direct function of the analog signals used to determine the called base. In fact, a survey of lossy quality score compressors has already shown that changing quality scores alone can sometimes have a beneficial effect on variant calling[9]. This result demonstrates that in a sense, lossy quality score compression removes noise from, or denoises the genomic data, resulting in more accurate variant calling. Taken together, the existing body of work on denoising and variant calling suggests that reads denoising could be improved by incorporating alignment and quality score information during error correction.

[1]Stanford University, Department of Electrical Engineering, Stanford, 94305, USA. [2]University of Illinois Urbana-Champaign, Department of Electrical and Computer Engineering, Urbana, 61801, USA. [3]University of Illinois Urbana-Champaign, Carl R. Woese Institute for Genomic Biology, Urbana, 61801, USA. *email: ihwang@stanford.edu; mhernaez@illinois.edu

In this work, we propose a novel denoising method, SAMDUDE, which takes advantage of alignment information contained in the SAM file in order to both denoise reads and update quality scores. We evaluate the effect of denoising on variant calling by comparing variants identified in files before and after denoising by SAMDUDE. We also evaluate files that have been denoised using other state-of-the-art denoisers that operate solely on reads in FASTQ files. This variant calling comparison methodology has already been used to analyze the effect of lossy compression on quality scores beyond the early steps in a genomic sequencing pipeline[9]. To our knowledge, this is the first application of such a comparison methodology on denoised genomic sequencing data, and provides a unique framework for directly evaluating the effect of denoising on sequencing data. To highlight the potential utility of simultaneous base denoising and quality score updating in a clinical setting, we perform denoising and variant calling comparisons on human data sets. We show that the simultaneous reads denoising and quality score updating procedure either maintains or improves variant calling with respect to the original SAM file, while denoising schemes that change only the reads result in degraded variant calling performance.

Survey of Denoisers for Genomic Data

Current state-of-the-art denoisers perform denoising based on a variety of techniques including *k*-mer counting and statistical error models, and target either substitution errors, insertion and deletion errors, or a combination of both[4]. We chose Musket[10] and RACER[11] to serve as benchmarks for SAMDUDE denoising performance, since both were touted for their ability to handle human WGS data sets[6]. In addition to Musket and RACER, BFCounter[12] and Lighter[13] are also preferred in the field for their memory efficiency and speed, respectively, and are thus the most likely to be used in practice. We omitted comparisons with BLESS 2[14], another denoising tool especially popular for its speed, due to installation difficulties. Here, we briefly describe the techniques underlying these four denoisers.

Musket[10] uses a *k*-mer spectrum approach in which reads that are suspected to be erroneous are changed until their *k*-mers appear frequently in the entire data set. The *k*-mer spectrum is constructed using a parallelized master-slave model, resulting in Musket's highly competitive execution time and excellent parallel scalability. Denoising is performed using a multistage workflow which begins with multiple iterations of two-sided conservative base correction. Two-sided conservative base correction is followed by multiple iterations of one-sided aggressive correction and voting-based refinement. Musket is able to denoise paired-end reads data sets simultaneously.

RACER[11] uses a *k*-mer counting approach to denoise FASTA and FASTQ data. Its *k*-mer counting method retains *k*-mers with counts above a given threshold, while correcting all other ones. RACER utilizes a unique and efficient hash table-based data structure which makes it extremely space efficient. While RACER does not denoise both files in a paired-end reads data set at the same time, each of the FASTQ files can be denoised independently and recombined in subsequent analysis steps. RACER requires approximate genome size as a parameter.

BFCounter[12] also uses a *k*-mer counting approach coupled with a Bloom filter in a two-pass denoising process. The use of a Bloom filter results in reduced memory requirements of nearly 50% memory savings, as compared to popular *k*-mer counting software. However, the two-pass implementation requires a significant amount of time for completing denoising, especially on human WGS data. Like RACER, BFCounter requires approximate genome size as a parameter.

Unlike the previously mentioned denoisers, Lighter[13] avoids *k*-mer counting and instead relies entirely on Bloom filters to perform denoising. Compared with most denoising methods, Lighter is extremely fast and memory-efficient, but like RACER and BFCounter it also requires an estimate of genome size as a parameter.

Results

To formulate the proposed denoising method, we assume a setting in which a genetic sample undergoes high-throughput shotgun sequencing, producing a large number of short, overlapping reads of length on the order of hundreds of base pairs. The errors introduced during the sequencing process are assumed to be primarily substitution errors, while INDELs are assumed to be negligible. We also assume that a reference genome is available, and that the reads can be aligned to the reference.

Our proposed denoising method, SAMDUDE, is based on the Discrete Universal Denoiser (DUDE) algorithm proposed in[15]. DUDE is a sliding-window discrete denoising scheme which is universally optimal in the limit of input sequence length when applied to an unknown source with finite alphabet size corrupted by a known discrete memoryless channel. The universal optimality of the DUDE guarantees that in the asymptotic limit of input sequence length it does as well as the best scheme of its type, regardless of the characteristics of the underlying noise-free sequence. In brief: DUDE uses a two-pass procedure to first infer statistics of the source sequence based on the noisy sequence, and then denoise the noisy sequence using the inferred statistics and noise channel characteristics. See Fig. 1 for a schematic of the SAMDUDE setting and algorithm.

In order to apply the DUDE-like denoising framework to the genomic sequencing setting, we make a number of algorithm design choices based on assumptions about the problem setting. While the universal denoising setting assumes a single noise-free input sequence and a single noise-corrupted output sequence of equal length to the input sequence, in the high-throughput sequencing setting the channel input is a single, noise-free sequence and the output are numerous overlapping, short, noisy sequences. The reads may not necessarily all be of the same length, but are all assumed to be much shorter than the noise-free sequence length. Despite the difference between these settings, the sequencing reads can be thought of as samples of a single, noise-corrupted sequence that can be inferred from the reads using alignment information. Under this assumption, we aggregate information from each read into statistics about the inferred noisy sequence. The universal denoising setting also assumes that the noise channel is memoryless and known, and corrupts sequences only with substitution errors. In the genomic sequencing setting, substitution errors are the primary form of noise in sequencing-by-synthesis methods, such as Illumina technologies. Today, sequencing-by-synthesis methods are among the most commonly

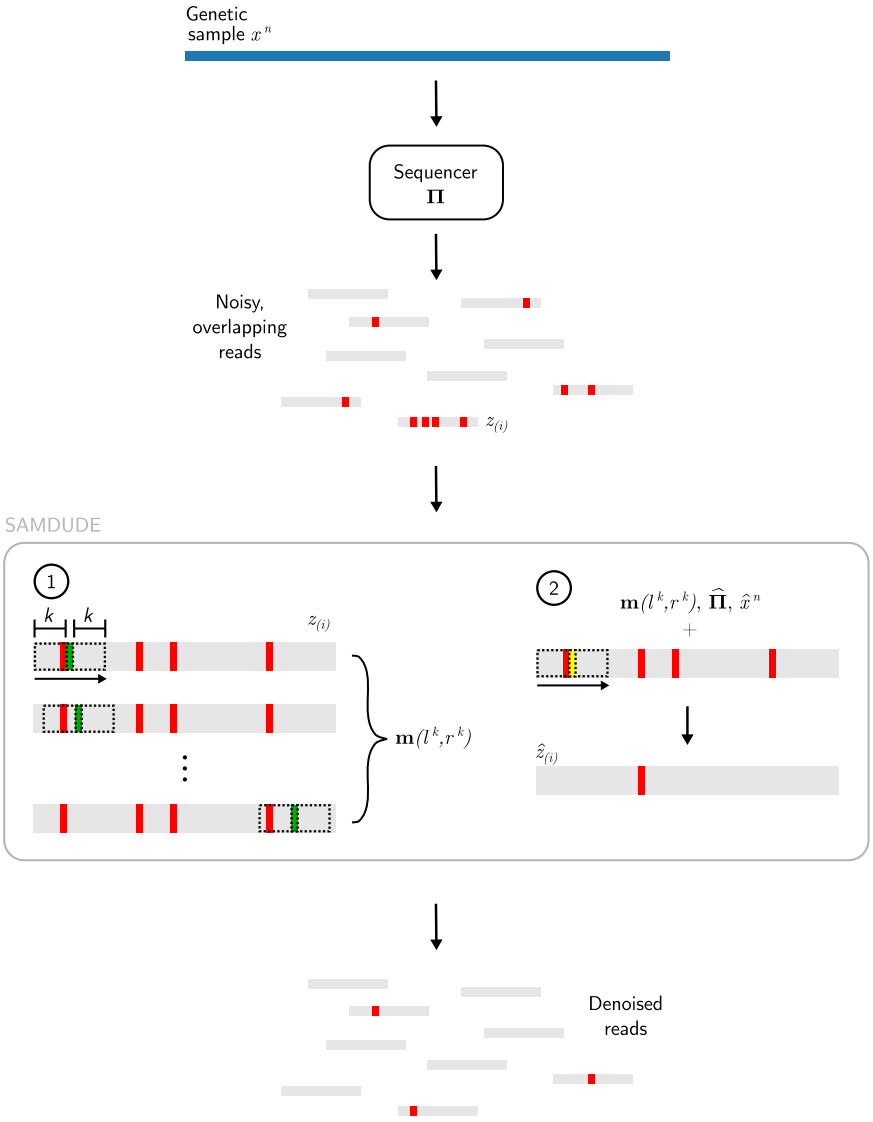

Figure 1. Schematic of the SAMDUDE setting and algorithm. A genetic sample, x^n (blue bar), is corrupted by a noise channel $\mathbf{\Pi}$, producing noisy reads $z_{(i)}$ with errors (red lines). In the first pass of SAMDUDE, a sliding window (dashed rectangle) sequentially records the central symbol (green line) flanked by a context of length $2k$, $l^k r^k$ for each location in each read. These central symbols and contexts are combined across all reads into a vector of counts $\mathbf{m}(l^k, r^k)$. During the first pass, a sequence estimate $\hat{x}^n$ and estimate of the noise channel $\hat{\mathbf{\Pi}}$ are also obtained. In the second pass, each read is again traversed sequentially for denoising. For each context in each read $\mathbf{m}$, $\hat{\mathbf{\Pi}}$ and $\hat{x}^n$ are all used to denoise the central symbol (yellow line) for each context in each read. The output of SAMDUDE is a collection of denoised reads.

used in practice[16]. SAMDUDE is a natural fit for such methods, and thus we assume that errors introduced during the sequencing process are primarily substitution errors, while INDELS are assumed to be negligible[2,17]. We also assume that a reference genome is available, and that raw sequencing reads stored in the FASTQ file format can be aligned to the reference in order to produce a SAM file of aligned reads. We consider only the primary mapping of the aligned reads, and disregard any other mappings, i.e., if more than one alignment location is reported for a read, only the primary one is considered (all secondary alignments are omitted). While DUDE is designed to operate in the setting where the corrupting noise channel is known and memoryless, in the genomic sequencing setting it is often difficult to obtain precise information about a particular machine's noise characteristics. As a result, we use alignment information in the SAM file to generate an estimate of the particular sequencer's noise-injecting characteristics. Operationally, this strategy has the added benefit of accounting for individual variations in performance from one sequencing machine to another. Finally, while paired-end reads are acceptable inputs to the denoiser, the pairing information is not used in the denoising process. The SAMDUDE denoising scheme is depicted in Fig. 1, and described in detail in the Methods section.

While all sequencing technologies inject all three types of errors, the noise channel model used in SAMDUDE is a particularly accurate reflection of Illumina sequencing technologies. Furthermore, due to the importance

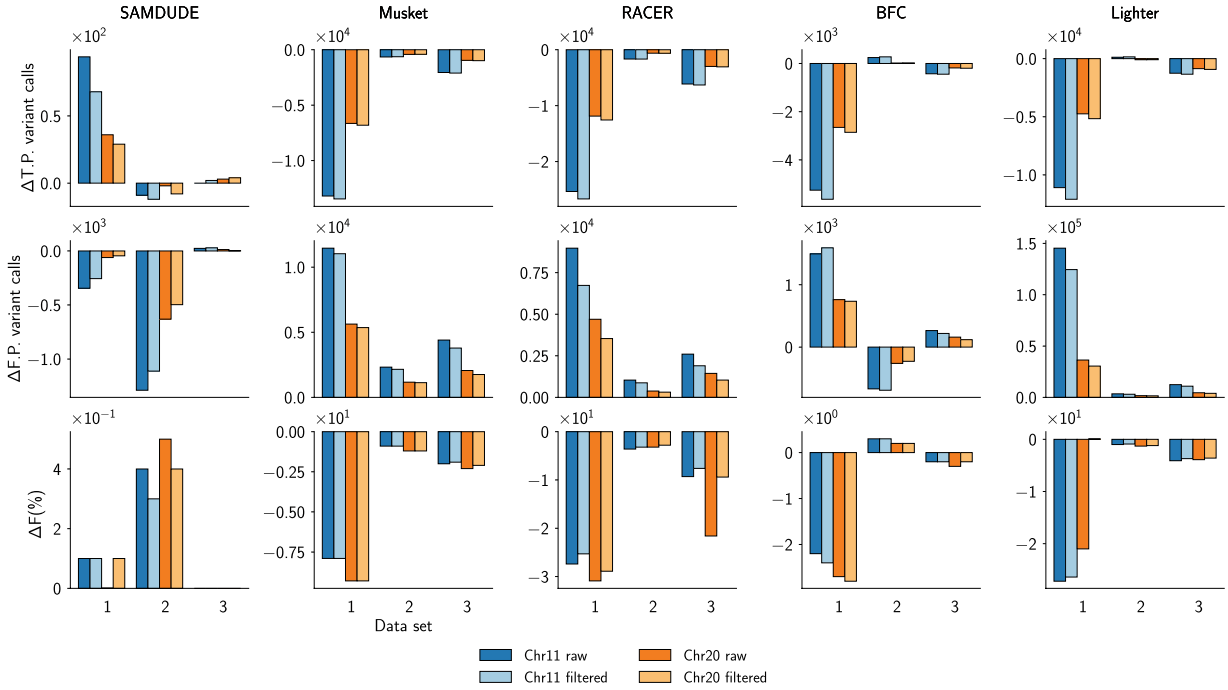

Figure 2. Denoising results for SAMDUDE (left column), Musket (second from left), RACER (center), BFCounter (second from right), and Lighter (right) for chromosomes 11 (blue) and 20 (orange). ΔT.P. indicates the change in True Positives, ΔF.P. indicates the change in False Positives, and ΔF indicates the change in F-score. In particular, positive ΔT.P. and ΔF indicate increases in true positive variants called and improvement of F-score, respectively. A negative ΔF.P. indicates that fewer variants were erroneously called, i.e., there were fewer false positives. Raw (dark colors) and filtered (light colors) variant call values are shown.

of variant identification in the clinical setting, we evaluated the effect of denoising on variant calling in human data sets. We tested SAMDUDE on three different paired-end WGS data sets of the *H. Sapiens* individual NA12878. The data sets are: ERR262997 corresponding to 30×-coverage, CEUTrio.HiSeq.WGS corresponding to 100×-coverage, and NA12878_V2.5_Robot_2 corresponding to 40×-coverage. For convenience, we refer to these data sets as 1, 2 and 3, respectively. Variant calls were compared against the version 37 gold standard call set for individual NA12878 released by the National Institute of Standards and Technology's (NIST) Genome in a Bottle consortium (GIAB)[18]. While the GIAB gold standard call set is a fairly conservative estimate of the individual NA12878 true variant call set, it is widely regarded by the field as the standard benchmark for evaluating sequence analysis algorithms. Furthermore, there exist "gold standard" (consensus of polymorphisms) variant call sets for certain human individuals, which can be used for an intuitive and direct method of assessing variant calling performance.

Denoising performance was evaluated with respect to variant calling of single nucleotide polymorphisms (SNPs). We first analyzed the effect of denoising on variant calling performance for individual human chromosomes using SAMDUDE and other state-of-the-art denoisers. We also compared the effect of denoising reads to the effect of lossy quality score compression. Lossy quality score compressors were developed to decrease the size of SAM files while still maintaining variant calling performance. Previous work showed that while quality score compressors effectively reduced SAM file size, they also sometimes had the unintended effect of improving variant calling performance[9]. For this reason, the lossy quality score compressors serve as a counterpoint to the SAMDUDE algorithm's procedure of changing both reads and quality scores in tandem with the explicit goal of improving variant calling performance. Finally, we analyzed the effect of SAMDUDE denoising on human WGS data.

Human chromosome denoising with SAMDUDE. For individual chromosome denoising experiments, we chose to use chromosomes 11 and 20. Chromosome 11 was chosen as representative of the median chromosome length in the human genome, and chromosome 20 was chosen since it is frequently used in genomic data tool assessment as representative of a small human chromosome[9]. The leftmost column of Fig. 2 shows that SAMDUDE can have varied effects across different data sets. For data set 1, denoising with SAMDUDE resulted in an increase in T.P. variants called concomitant with a decrease in F.P. variants, resulting in a modest gain in F-score. Although denoising resulted in a slight decrease of T.P. variants called in data set 2 relative to the original file, it also resulted in a very large decrease in number of F.P. variants called, resulting in a large gain in F-score. In contrast, a handful of additional T.P. and F.P. variants were called in data set 3, resulting in no change in F-score. Despite variation in results across data sets, the effect of SAMDUDE denoising is consistent across chromosomes within a given data set.

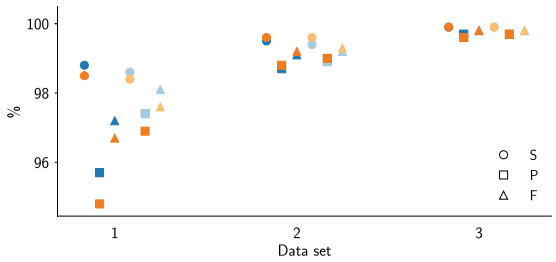

Figure 3. Statistics for variant call sets generated from the original data sets (blue, chromosome 11 and orange, chromosome 20). The statistics are: sensitivity (circles), precision (squares) and F-score (triangles). Statistics are grouped by variant call type: raw (dark colors) and filtered (light colors).

While differences in performance across data sets seem to indicate inconsistency in the SAMDUDE denoising algorithm, these results make sense in the context of coverage and initial data quality, which are summarized in Fig. 3. The original performance metrics for data set 3 are by far the highest. In contrast, while data set 2 also has a high initial sensitivity, its relatively low precision leaves room for improvement in F-score. Data set 1 has the most room for improvement, with low initial sensitivity, precision and F-score.

Because of SAMDUDE's conservative behavior, it might be suspected that the gains from updating quality scores are small compared to the effect of denoising bases in the reads. To test this hypothesis, we created partially-denoised SAM files where the reads were those obtained from SAMDUDE denoising, but were paired with the original quality score strings. These partially-denoised files were then analyzed using the same variant calling pipeline, and the results of this experiment are summarized in Supplementary Table 1 (labeled Partial Denoising). For all data sets, partial denoising resulted in either no change or an increase in F-score, even after GATK filtering. However, for data sets 1 and 2 the gains were not as large as those attained using SAMDUDE after GATK filtering of variant calls.

It might also be suspected that the same amount of random base changes would additionally result in improvements in variant calling. To check this hypothesis, we ran the variant calling pipeline on the SAM files for chromosome 20 with bases changed at random based on a uniform distribution over the possible nucleotide bases (see Supplementary Table 2). The results of this experiment are summarized in Supplementary Table 1 (labeled Random noise). For all data sets, addition of random noise resulted in either no change or a decrease in sensitivity, concurrent with a uniform increase in precision. Overall, the effect of random noise is an increase in F-score for all raw variant calls, but a decrease in F-score for GATK-filtered variant calls in data sets 1 and 2.

Together, these results support our claim that SAMDUDE is a conservative denoising method which will not adversely affect data sets that do not need denoising, while improving those that can benefit from denoising. Furthermore, SAMDUDE's performance is robust and consistent for both raw and filtered variant calls, unlike that of partial denoising and random noise addition.

Comparison of SAMDUDE denoising with state-of-the-art genomic denoisers and quality score compressors. The effects of denoising by the state-of-the-art genomic denoisers Musket, RACER, BFCounter and Lighter are shown in the center and right columns of Fig. 2. In most cases, the denoisers resulted in significant decreases in the number of T.P. variants called, with significant increases in the number of F.P. variants called, leading overall to significant decreases in F-score for both the raw and GATK-filtered variants. The only exception is the denoising of data set 2 using BFCounter, which resulted in a slight increase in T.P. variants called, slight decrease in F.P. variant called, and a corresponding slight increase in F-score. However, this improvement in variant calling is not consistent, as denoising data sets 1 and 3 using BFCounter resulted in worse F-scores. These changes demonstrate that although current state-of-the-art denoisers have been shown to improve early steps of the genome analysis pipeline, their denoising choices tend to have adverse effects on variant calling.

The trends are consistent even when we consider different variant call filtering levels. Supplementary Figures 1–6 show the variant calling precision as a function of sensitivity for different filtering criteria. We focus on the variant call filtering results for data set 2 (Supplementary Figs. 3 and 4) since for this data set SAMDUDE denoising resulted in the largest number of total changes to the variant call set, and also in the largest change in F-score. We also focus on comparing sensitivity filtering for the results of Musket and RACER due to their consistent denoising trends. To construct these curves, we set percentile thresholds starting at the 10^{th} and ending at the 90^{th} percentile at increments of 10% for quality of depth (QD) of the raw variant calls. All variants with QD below the threshold were filtered out, and the remaining variant calls were evaluated against the gold standard call set. The top row of Supplementary Fig. 4 shows that even under these filtering criteria, the curves corresponding to the variant call sets of Musket- and RACER-denoised SAM files lie far below the others. In other words, for a given sensitivity level, the variants called under Musket- and RACER-denoised SAM files have significantly worse precision than SAMDUDE.

In addition to variant call filtering results for call sets from SAMDUDE, Musket-, and RACER-denoised SAM files, the bottom rows of Supplementary Figs 1–6 also show the filtered variant call set curves for the original call set, the call set resulting from the addition of random noise to the reads, and two call sets resulting from lossy compression of the original file using P-Block and R-Block with compression parameters 3 and 40, respectively. P-Block and R-Block are two state-of-the-art lossy quality score compressors[19] which have also been shown to improve variant calling performance. Another state-of-the-art lossy compressor, QVZ[20], has been omitted from

	ΔT.P. raw	ΔT.P. filtered	ΔF.P. raw	ΔF.P. filtered
SAMDUDE	1,954	1,642	−1,509	−1,923
Musket	−208,144	−215,524	324,957	305,362

Table 1. Results for raw and filtered variant calling on denoised data set 1 using SAMDUDE and Musket.

the analysis since it is unable to compress SAM files with variable length reads. Again we focus our discussion on data set 2 and observe that in Supplementary Figs. 3 and 4 the effect of lossy compression on the variant call set is almost indistinguishable from that of SAMDUDE. However, the rightmost points in each of the filtered variant call set curves show that SAMDUDE outperforms all other schemes when sensitivity and precision are both high. SAMDUDE's dominance, especially at high sensitivity levels for chromosome 11, makes sense since SAMDUDE's denoising performance improves as read coverage increases.

Human whole-genome denoising with SAMDUDE. Next, we evaluated the effect of denoising on variant calling for an entire WGS human data set. For this experiment, we used all reads of data set 1, and the results are summarized in Table 1. After SAMDUDE denoising the raw variant call set included in total 1,954 additional true positive variants. Furthermore, 1,509 false positive variants were eliminated relative to the original set of variant calls. After the GATK pipeline, 1,642 of the additional true positive variants were validated, and the number of additional false positive variant calls eliminated increased to 1,923. In contrast, raw variant calls based on the Musket-denoised file resulted in 208,144 fewer true positive calls and 324,957 additional false positive calls relative to the original set of variant calls. After the GATK pipeline, the number of true positive variant calls missed increased by 215,524 while the number of false positive variant calls increased by 305,362. Variant calls from the RACER-denoised data set were unable to be validated against the gold standard call set due to pipeline errors.

Discussion

We have presented SAMDUDE, a denoising method that uses alignment information in SAM files and a statistical model of the genomic data in order to improve variant calling. Because of the assumptions used by the denoising model, SAMDUDE's intended use is for improving the quality of short sequencing reads obtained from healthy tissue. The potential problem of applying SAMDUDE to samples from unhealthy, e.g. cancerous, tissues is that SAMDUDE may incorporate k-mers from the unhealthy tissue into analysis, and incorrectly denoise bases that were indeed correct. This could happen when the variant allele frequency (VAF) of a given somatic mutation is below or similar to the substitution error rate from the sequencing technology being used[21]. In such cases, differentiating between genomic sequencing errors and somatic mutation in unhealthy tissue is more challenging.

Taken together with the initial data quality statistics in Fig. 3, the range of improvements observed in Fig. 2 imply that SAMDUDE is a "conservative" denoising algorithm that makes few or no changes to the reads and quality scores when the original data set is already of very high quality, but makes sound denoising choices resulting in variant calling improvements when the original data set is of lesser quality. The quality of denoising performance also correlates with data coverage: data set 2 has the highest coverage and SAMDUDE denoising performance is the best on this set. Since SAMDUDE relies on empirical estimates of k–mer distributions and the noise channel, the higher the coverage and the more accurate the empirical estimates, the better the denoising performance. It is also notable that the trends in performance hold for both raw variant and filtered variant calls, adding to our confidence in SAMDUDE as a conservative denoiser that integrates well with existing recommended genomic data analysis pipelines.

The results of denoising aligned whole genome data are consistent with those observed for individual chromosomes. While the relative number of true positive variants identified might seem relatively small, the extra information provided by each extra variant could be invaluable. Single point mutations are responsible for numerous human diseases, and other diseases once assumed to be caused by a single variant with large effect are now being understood to be the result of multiple monogenic mutations, or of collections of rare variants in previously identified genes[22–24]. Perhaps more importantly, the elimination of false positive variants is crucial to accurate diagnosis and appropriate treatment design[25–27]. Thus, in the clinical context the implications of every additional true positive variant identified and each false positive variant eliminated are far larger than the objective tally.

Our results emphasize that the quality score updating step of SAMDUDE is crucial to improving variant calling outcome, and that denoising reads alone is insufficient for higher quality of variant calls. SAMDUDE was able to both identify thousands of additional variants and eliminate a similar number of false positive variants from a single human whole-genome data set. In contrast, state-of-the-art denoisers, which were designed to improve earlier steps in the sequencing pipeline and are limited to changing information only in the reads and not the quality scores, led to degraded variant calling performance. Our results also highlight the importance of evaluating denoisers on the variant calling step of the genomic sequencing pipeline using real data sets with gold standards.

As a proof-of-concept denoiser, SAMDUDE shows great promise in improving the accuracy of variant calling based on individual sequencing data sets. Furthermore, these encouraging results motivate further experimentation of the parameters and elements of the denoising procedure, including context length k, majority and confidence thresholds, quality score updating rule, and additional refinement of the implementation in order to reduce computational memory and time requirements. We anticipate that the SAMDUDE denoising method will inspire an efficient and powerful denoising software that will be a valuable tool for researchers and clinicians alike.

Methods

Problem setting. We have the following problem setting: x^n is the true genomic sequence of length n, and the sequencing procedure involves a noise channel, $\hat{\mathbf{\Pi}}$, which generates a set of m noisy reads $\{z_{(1)}, z_{(2)}, \ldots, z_{(m)}\}$ with components taking values in the set of all possible nucleotide bases, alphabet $\mathcal{A} = \{\text{A, C, G, T}\}$. The reads set is accompanied by a set of quality score strings $\{q_{(1)}, q_{(2)}, \ldots, q_{(m)}\}$ with components taking values in the set of ASCII characters quantifying basecalling quality on a quality score scale. Our goal is to both denoise the bases in the reads and to update the corresponding quality scores in order to improve the accuracy of variant identification, while still preserving polymorphisms that are unique to the individual. The denoiser has access to the reads and the quality score strings, as well as to the alignment information of each of the reads to the reference sequence. Note that in this setting, the true genome sequence x^n is unknown.

We assume that for a particular location i in the reference genome, the majority of reads covering that position will have base calls that agree, and that the minority of base calls which do not match with the majority are likely to be errors. Under this assumption, the sequence estimate is obtained by recording the majority base, for some majority threshold, for all reference genome positions covered by the reads. For noise channel estimation, creating vectors of counts and denoising, we couple base calls and quality scores by combining bases with their corresponding quality scores. In other words, we assume that the input to the channel is a noise-free sequence taking values in alphabet $\mathcal{A}$, while the channel output is a tuple of the called based and the quality score associated with that base. Hence the output alphabet is given by $\mathcal{B} = \mathcal{A} \times \mathcal{Q}$, where $\mathcal{Q}$ denotes the alphabet of the quality scores. The typical size of $\mathcal{Q}$ is 42, which carries a significant computational burden due to the resulting size of $\mathcal{A} \times \mathcal{Q}$. To avoid this burden, we adopt the quality score binning method recommended by Illumina for reducing quality score resolution[28]. This method reduces the original alphabet of quality scores from 42 to only 8 bins; hence, $\mathcal{Q} \in \{\text{bin}_1, \text{bin}_2, \ldots, \text{bin}_8\}$, with bin limits corresponding to those recommended by Illumina (see Supplementary Table 3). This set of output tuples also incorporates the fact that certain sequencing technologies, like Illumina technologies, are known to produce reads with position-varying noise characteristics. Typically, different noise characteristics would be characterized by different noise channels—in the position-varying case, up to one channel per position. However, by combining each nucleotide base with its quality score, we can broadly account for various possible position-dependent trends in noise without setting hard boundaries and limiting ourselves to particular assumptions about the noise characteristics.

With this formalization of the problem setting, the noise channel estimate $\hat{\mathbf{\Pi}}$ is of size $|\mathcal{A}| \times |\mathcal{B}|$, reflecting the assumption that the noise channel corrupts a sequence composed of symbols belonging to input alphabet $\mathcal{A}$, and produces noisy sequences composed of symbols belonging to output alphabet $\mathcal{B}$. The vector of counts $\mathbf{m}(l^k, r^k)$, of size $|\mathcal{B}|$, records the number of times the subsequence $l^k b r^k$, comprising left and right contexts l^k and r^k, is observed in the collection of reads, with $l^k, r^k \in \mathcal{A}^k$, and central tuple $b \in \mathcal{B}$. We limit the alphabet of the context to $\mathcal{A}$ to ensure that each possible context is observed a significant number of times.

Once the noise channel estimate $\hat{\mathbf{\Pi}}$ and vectors of counts $\mathbf{m}$ are acquired, the SAMDUDE algorithm proceeds follows. For ease of exposition and with some abuse of notation, we denote an arbitrary read as z with i^{th} component z_i accompanied by quality score q_i. Subsequences of z are denoted as $z_a^b = (z_a, \ldots, z_b)$.

1. For each base z_i and its associated quality score q_i in read z, identify the length $2k$ context string $z_{i-k}^{i-1} z_{i+1}^{i+k}$ surrounding position i, and bin bin_i to which q_i belongs.
2. Calculate the estimated probability of observing the left and right contexts z_{i-k}^{i-1} and z_{i+1}^{i+k} in x^n with different central symbols belonging in $\mathcal{A}$. The probability is distributed over all symbols in $\mathcal{A}$ and given by

$$\hat{\mathbf{q}}(z_{i-k}^{i-1}, (z_i, \text{bin}_i)), z_{i+1}^{i+k}) = \pi_{(z_i, \text{bin}_i)} \odot [(\hat{\mathbf{\Pi}}\hat{\mathbf{\Pi}}^T)^{-1}\hat{\mathbf{\Pi}}\mathbf{m}(z_{i-k}^{i-1}, z_{i+1}^{i+k})], \tag{1}$$

   where $\hat{\mathbf{\Pi}}$ is the channel estimate matrix comprising column vectors $\{\pi_1, \pi_2, \ldots, \pi_{|\mathcal{B}|}\}$, and $\odot$ represents element-wise multiplication for vectors (see[15], Eq. (25)).
3. Replace z_i with the base corresponding to the argument of the maximum of the distribution estimate, and update q_i using the maximum of the distribution estimate (see subsec. Quality score updating for details).

We devote the following subsections to detailed descriptions of the components of SAMDUDE: estimates of the channel and true sequence, noise statistics, and vectors of counts. We also describe the quality score updating procedure, specify implementation details in terms of reads padding and describe the overall evaluation workflow.

Channel and sequence estimation. To compute the channel estimate $\hat{\mathbf{\Pi}}$, we use alignment information from the SAM file to perform a sequence pileup at every reference genome position by cataloging all reads at that position. At each position we assume that the majority base, for some majority threshold $t_m \geq 0.5$, is the true base. That is, the base in the pileup with normalized counts greater than or equal to t_m is declared to be the majority base. If there is no clear majority base, then we do not use information from that position for channel estimation. This rule allows us to use the overwhelming majority of genomic positions in order to estimate the channel characteristics. Additionally, the threshold ensures that information is not used from positions in the genome that display heterozygosity, e.g., due to polyploidy of the organism being sequenced. This, in turn, prevents the conflation of differences in reads overlapping heterozygous positions with noise, and precludes erroneous denoising of reads at those positions. For each base in $\mathcal{A}$ we record the number of bases in an 4×32 conditional counts matrix

$$\mathbf{N} = \begin{bmatrix} \mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\mathrm{A}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\mathrm{A}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\mathrm{A}} \\ \mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\mathrm{C}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\mathrm{C}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\mathrm{C}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\mathrm{T}} & \cdots & \mathrm{n}_{(\mathrm{T},\mathrm{bin}_1)|\mathrm{T}} & \cdots\cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\mathrm{T}} \end{bmatrix},$$

where $\mathrm{n}_{(i_1,i_2)|j}$ is the number of positions in all reads for which the read contains base i_1 with accompanying quality score in bin i_2 at a position whose majority base is j. $\mathbf{N}$ is row-normalized to obtain $\hat{\mathbf{\Pi}}$

$$\hat{\mathbf{\Pi}} = \begin{bmatrix} \frac{\mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\mathrm{A}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{A}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\mathrm{A}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{A}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\mathrm{A}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{A}}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\mathrm{T}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{T}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{T},\mathrm{bin}_1)|\mathrm{T}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{T}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{T},\mathrm{bin}_8)|\mathrm{T}}}{\sum_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\mathrm{T}}} \end{bmatrix}. \quad (2)$$

The majority bases are recorded as the sequence estimate.

Counts vector acquisition. The vectors of counts $\mathbf{m}(l^k, r^k)$ record all appearances of context $l^k r^k$ surrounding a central symbol b, with $l^k, r^k \in \mathcal{A}^k$ and $b \in \mathcal{B}$. There are up to $|\mathcal{A}|^{2k}$ possible vectors $\mathbf{m}$ for each possible context $(l^k r^k)$, and each $\mathbf{m}$ is of length $|\mathcal{B}|$. For each b appearing between the left and right context components l^k and r^k, respectively, we record the number of times the sequence $l^k b r^k$ appears in the collection of reads as the b^{th} component of $\mathbf{m}(l^k, r^k)$:

$$\begin{aligned} \mathbf{m}(l^k, r^k)[b] &= |\{z, i\colon k+1 \le i \le \mathrm{length}(z) - k \\ \forall\, z_{i-k}^{i+k} &= l^k b r^k, z \in \{z_{(1)}, z_{(2)}, \ldots, z_{(m)}\}|. \end{aligned} \quad (3)$$

For positions $i < k+1$ or $i > \mathrm{length}(z) - k$, we employ information from the sequence estimate to acquire context counts. This process is detailed in subsec. Reads padding.

Position-based denoising rules. Denoising is performed at read positions where z_i as well as the surrounding context string contain only bases in $\mathcal{A}$, excluding symbols indicating ambiguity. We use this simple genomic alphabet in order to avoid basing denoising decisions on non-uniquely identifiable context strings. For efficiency, we set a quality score threshold above which denoising is not attempted, t_p. At read positions with quality scores corresponding to a probability above t_p, SAMDUDE does not attempt denoising for expediency.

Quality score updating. The maximum of the conditional distribution estimate $\hat{\mathbf{q}}$ is used to update the quality score accompanying the denoised base. The updating procedure depends on whether the base the denoiser selected matches the original base z_i. If the maximum of $\hat{\mathbf{q}}$ corresponds to the same base as the original one, the quality score is adjusted as follows: convert the original quality score q_i into a confidence probability

$$p_i = 1 - 10^{-q_i/10}, \quad (4)$$

take the arithmetic mean of p_i and the maximum of the estimated conditional probability $\hat{\mathbf{q}}$, denoted by $p_{\max}$, and back-convert the averaged probability into the updated quality score. In other words, the updated quality score is

$$\tilde{q} = -10\log\left(1 - \frac{p_i + p_{\max}}{2}\right),$$

when the denoiser does not recommend a different base. The updated quality score is re-inserted into the quality score string after conversion to an ASCII character as per the sequencing machine's quality score encoding method. For example, if the sequencing machine encodes on a Phred + 33 scale, the quality score string's i^{th} component is replaced with the ASCII character for $\tilde{q} + 33$. On the other hand, if the denoiser recommends a base change, q_i is simply replaced with $\tilde{q} = -10\log(1 - p_{\max})$. Again, the quality score string's i^{th} component is replaced with the appropriately encoded $\tilde{q}$.

This procedure was chosen in order to balance the denoiser's conditional probability estimates with the original quality scores, which reflect the sequencing machine's confidence in the base calls. Since the original quality score is a function of the original base call, if the denoiser agrees with the base call, the denoiser's probability estimate should be combined with the sequencer's quality score. However, if the denoiser decides on a different base, then the original quality score is unrelated to the denoiser's chosen base and we can disregard the original quality score in favor of the quality score converted from the denoiser's probability estimate.

Reads padding. The reads reported from a sequencing machine cannot always be mapped directly to the reference genome in their entirety, since they may contain bases that are insertions relative to the reference genome, lack bases that correspond to deletions from the reference genome, or contain stretches of bases at the beginning and end of the read that simply do not match the reference genome. These inconsistencies relative to the reference genome are summarized by the sequence aligner in a CIGAR string accompanying the read (https://samtools.

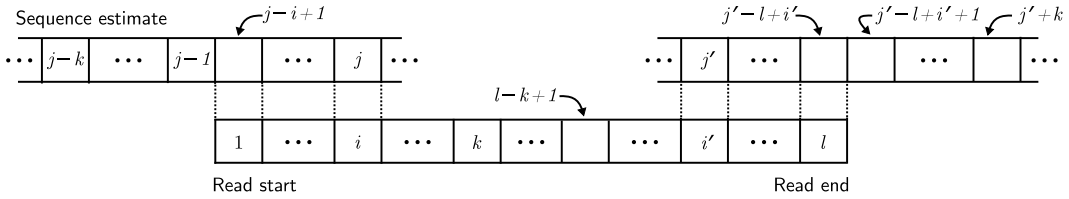

Figure 4. During context acquisition and denoising processes, reads are padded with bases from the sequence estimate. At left: position i in the read corresponds to position j in the sequence estimate, and $1 \leq i \leq k$. The left half of the context for that base is padded with sequence estimate bases from positions $j - k$ up to $j - i + 1$, where k is half of the context length. At right: here, position i' in the read corresponds to position j' in the sequence estimate. $i' \geq l - k + 1$, where l is the length of the processed read. The base at position i' can now use positions $j' - l + i' + 1$ up to $j' + k$ of the sequence estimate as a right-hand context.

github.io/hts-specs/SAMv1.eps). Additionally, large portions of the read may be assigned very low quality scores, indicating entire regions of the read for which the sequencer has low confidence. One strategy for dealing with these inconsistencies is to simply eliminate non-matching or low-quality bases, but this can lead to loss of potentially valuable information. Instead, we retain this information and tailor our use of it to process the reads during channel estimation, counts vector acquisition, and denoising.

The channel estimation procedure relies on the creation of pileups at reference genome positions. As a result, this step considers only bases that are mapped to the reference genome and relies on the CIGAR string information to accurately map bases to reference genome positions. Bases that are designated in the CIGAR string as low-confidence and non-matching (i.e., "soft-clipped", or simply "clipped") or inserted relative to the reference genome are not considered during channel estimation. When deletions are indicated in the CIGAR string, reference mapping positions are adjusted accordingly.

The counts vectors are simply histograms of the central base appearing with context strings of length $2k$. These context strings are unique to the individual and should include bases that are inconsistent relative to the reference genome, since those bases may be true polymorphisms. Thus, during counts vector acquisition the reads retain bases that are marked by the aligner as insertions since those insertions may be inherent to the true sequence. However, as in the channel estimation process, bases that are designated in the CIGAR string as clipped are omitted to avoid large sections of low-confidence base calls. In order to maximize the number of context strings obtained from a processed read, we additionally pad the read with a header and footer of length up to k if the read begins or ends, respectively, with bases that are mapped to the reference genome, i.e., not insertions. The padding process, illustrated in Fig. 4, allows the denoiser to obtain context information from up to $2k$ additional locations in each read.

Evaluation. *Evaluation procedures and metrics.* Denoising performance was evaluated with respect to variant calling of single nucleotide polymorphisms (SNPs). To quantify denoiser performance, we used the common performance metrics of true positives (T.P.), false positives (F.P.), and false negatives (F.N.). T.P. variants are the calls present in the gold standard call set, F.P. variants are the calls not present in the gold standard call set, and F.N. variants are those present in the gold standard set but not called. To summarize the changes in T.P., F.P. and F.N. variants identified, we used the following performance metrics: sensitivity (S), which measures the proportion of all the variants that are correctly called $(\mathrm{T.P.}/(\mathrm{T.P.} + \mathrm{F.N.}))$, precision (P), which measures the proportion of called variants that are true $(\mathrm{T.P.}/(\mathrm{T.P.} + \mathrm{F.P.}))$, and F-score (F), which is the harmonic mean of the sensitivity and precision $(2(S \times P)/(S + P))$.

SAMDUDE parameters. For all denoising experiments, we used a single-sided context length of $k = 7$ (14 bases total in the double-sided context). This context length was chosen for computational feasibility, but also maximizes the number of counts in each context histogram without skewing the histograms towards a uniform distribution, which occurs when k is either too small or too large (see Supplementary Tables 4–6). For sequence and channel estimation we used a majority threshold of $t_m = 0.9$ for high confidence in our estimate of the "true" genomic sequence, and also to definitively eliminate potentially confounding effects at heterozygous genomic positions which might not have a clear majority base. Finally, based on experiments with different quality value thresholds (see Supplementary Tables 5 and 6), we attempted denoising only at bases for which the sequencer's confidence probability p, Eq. (4)) is less than a chosen confidence threshold $t_p = 0.9$.

Denoising and variant calling pipeline. Individual chromosomes were extracted in binary SAM (BAM) file format from the aligned data sets and sorted using the SAMtools utility[29]. For denoising using Musket, RACER, BFCounter and Lighter, copies of the sorted BAM file were converted from the BAM to FASTQ format via the biobambam2 BAM file processing toolkit[30]. BAM files for each chromosome were also converted to the SAM format. The extracted SAM files were denoised using SAMDUDE.

The denoised FASTQ files were then aligned to a reference file using BWA-mem[31], generating denoised SAM files. All denoised SAM files then underwent SNP calling using the SNP calling pipeline recommended by the Broad Institute[32–34], and compared to the gold standard call set. We report results for both raw variants and variants filtered under the GATK Best Practices-recommended variant filtering process. For more details regarding the variant calling, filtering and evaluation pipelines, we refer the reader to the Variant calling pipeline section in the Supplementary data.

Computational requirements and machine specifications. We ran most experiments on a workstation computer with 12 Intel Xeon cores at 3.4 GHz and 32 GB of RAM, running Linux Ubuntu 14.04.4. SAMDUDE denoising for the chromosome 11 file of data set 3 was run on a different workstation with 80 Intel Deon cores at 2.2 GHz and 504 GB RAM, running CentOS 7.4.1708. Time and peak computational memory requirements for denoising data sets 1, 2 and 3 using SAMDUDE, Musket, RACER, BFCounter and Lighter are summarized in the Supplementary Table 7. In its current manifestation, SAMDUDE generally uses about an order of magnitude more memory than Musket and RACER. This is due to the large number of context histogram vectors that SAMDUDE acquires. SAMDUDE also generally requires about an order of magnitude more runtime than Musket and RACER. This result is not surprising, given that SAMDUDE is currently implemented in Python with no parallelization.

Received: 7 January 2019; Accepted: 25 September 2019;
Published online: 21 October 2019

References

1. Costa, F. F. Big data in biomedicine. *Drug discovery today* **19**, 433–440 (2014).
2. Minoche, A. E., Dohm, J. C. & Himmelbauer, H. Evaluation of genomic high-throughput sequencing data generated on illumina hiseq and genome analyzer systems. *Genome biology* **12**, R112 (2011).
3. Boycott, K. M., Vanstone, M. R., Bulman, D. E. & MacKenzie, A. E. Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics* **14**, 681–691 (2013).
4. Laehnemann, D., Borkhardt, A. & McHardy, A. C. Denoising dna deep sequencing data—high-throughput sequencing errors and their correction. *Briefings in bioinformatics* **17**, 154–179 (2016).
5. Koh, P. W., Pierson, E. & Kundaje, A. Denoising genome-wide histone chip-seq with convolutional neural networks. *Bioinformatics* **33**, i225–i233, https://doi.org/10.1093/bioinformatics/btx243, /oup/backfile/content_public/journal/bioinformatics/33/14/10.1093_bioinformatics_btx243/2/btx243.pdf (2017).
6. Molnar, M. & Ilie, L. Correcting illumina data. *Briefings in bioinformatics* **16**, 588–599 (2014).
7. Heydari, M., Miclotte, G., Demeester, P., Van de Peer, Y. & Fostier, J. Evaluation of the impact of illumina error correction tools on de novo genome assembly. *BMC bioinformatics* **18**, 374 (2017).
8. Lee, B., Moon, T., Yoon, S. & Weissman, T. Dude-seq: Fast, flexible, and robust denoising for targeted amplicon sequencing. *PloS one* **12**, e0181463 (2017).
9. Ochoa, I., Hernaez, M., Goldfeder, R., Weissman, T. & Ashley, E. Effect of lossy compression of quality scores on variant calling. *Briefings in bioinformatics* **18**, 183–194 (2016).
10. Liu, Y., Schröder, J. & Schmidt, B. Musket: a multistage k-mer spectrum-based error corrector for illumina sequence data. *Bioinformatics* **29**, 308–315 (2013).
11. Ilie, L. & Molnar, M. Racer: rapid and accurate correction of errors in reads. *Bioinformatics* **29**, 2490–2493 (2013).
12. Melsted, P. & Pritchard, J. K. Efficient counting of k-mers in dna sequences using a bloom filter. *BMC Bioinformatics* **12**, 333, https://doi.org/10.1186/1471-2105-12-333 (2011).
13. Song, L., Florea, L. & Langmead, B. Lighter: fast and memory-efficient sequencing error correction without counting. *Genome Biology* **15**, 509, https://doi.org/10.1186/s13059-014-0509-9 (2014).
14. Heo, Y., Ramachandran, A., Hwu, W.-M., Ma, J. & Chen, D. Bless 2: accurate, memory-efficient and fast error correction method. *Bioinformatics* **32**, 2369–2371, https://doi.org/10.1093/bioinformatics/btw146 (2016).
15. Weissman, T., Ordentlich, E., Seroussi, G., Verdú, S. & Weinberger, M. J. Universal discrete denoising: Known channel. *IEEE Transactions on Information Theory* **51**, 5–28 (2005).
16. Pfeiffer, F. *et al*. Systematic evaluation of error rates and causes in short samples in next-generation sequencing. *Scientific reports* **8**, 10950 (2018).
17. Fox, E. J., Reid-Bayliss, K. S., Emond, M. J. & Loeb, L. A. Accuracy of next generation sequencing platforms. *Next generation, sequencing & applications* **1** (2014).
18. Zook, J. M. *et al*. Integrating human sequence data sets provides a resource of benchmark snp and indel genotype calls. *Nature biotechnology* **32**, 246–251 (2014).
19. Cánovas, R., Moffat, A. & Turpin, A. Lossy compression of quality scores in genomic data. *Bioinformatics* **30**, 2130–2136, https://doi.org/10.1093/bioinformatics/btu183, /oup/backfile/content_public/journal/bioinformatics/30/15/10.1093_bioinformatics_btu183/2/btu183.pdf (2014).
20. Malysa, G. *et al*. Qvz: lossy compression of quality values. *Bioinformatics* **31**, 3122–3129, https://doi.org/10.1093/bioinformatics/btv330, /oup/backfile/content_public/journal/bioinformatics/31/19/10.1093_bioinformatics_btv330/3/btv330.pdf (2015).
21. Kandoth, C. *et al*. Mutational landscape and significance across 12 major cancer types. *Nature* **502**, 333 (2013).
22. Gilissen, C., Hoischen, A., Brunner, H. G. & Veltman, J. A. Disease gene identification strategies for exome sequencing. *European Journal of Human Genetics* **20**, 490 (2012).
23. Rabbani, B., Mahdieh, N., Hosomichi, K., Nakaoka, H. & Inoue, I. Next-generation sequencing: impact of exome sequencing in characterizing mendelian disorders. *Journal of human genetics* **57**, 621 (2012).
24. Bastarache, L. *et al*. Phenotype risk scores identify patients with unrecognized mendelian disease patterns. *Science* **359**, 1233–1239 (2018).
25. Goldfeder, R. L. *et al*. Medical implications of technical accuracy in genome sequencing. *Genome medicine* **8**, 24 (2016).
26. Dewey, F. E. *et al*. Clinical interpretation and implications of whole-genome sequencing. *Jama* **311**, 1035–1045 (2014).
27. Altman, R. B. *et al*. A research roadmap for next-generation sequencing informatics. *Science translational medicine* **8**, 335ps10–335ps10 (2016).
28. Illumina. Reducing whole-genome data storage footprint (white paper, available at https://www.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf, 2014).
29. Li, H. *et al*. The sequence alignment/map format and samtools. *Bioinformatics* **25**, 2078–2079 (2009).
30. Tischler, G. & Leonard, S. biobambam: tools for read pair collation based algorithms on bam files. *Source Code for Biology and Medicine* **9**, 13 (2014).
31. Li, H. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997* (2013).
32. McKenna, A. *et al*. The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data. *Genome research* **20**, 1297–1303 (2010).
33. DePristo, M. A. *et al*. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics* **43**, 491–498 (2011).
34. Van der Auwera, G. A. *et al*. From fastq data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. *Current protocols in bioinformatics* 11–10 (2013).

Acknowledgements
The authors would like to thank Jiantao Jiao and Dmitri Pavlichin for helpful discussions and suggestions. This work has been supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370, and by NIH Grant 5 U01 CA198943-03.

Author contributions
I.O., M.H. and T.W. conceived the experiments, I.F.-H. conducted the experiments and analyzed the results. All authors reviewed the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41598-019-51418-z.

Correspondence and requests for materials should be addressed to I.F.-H. or M.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.