Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Universal Denoising of Aligned Genomic Sequences

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

[1] Department of Electrical Engineering, Stanford University, Stanford, 94305, USA and
[2] Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign, Urbana, 61801, USA

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on January 26. 2017; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Advances in sequencing technology have spurred the production of sequencing data at ever increasing speed and volume. Despite the proliferation of new technologies and improvements in sequencing techniques, noise in sequencing data remains a confounding factor in data analysis and interpretation.
Results: We propose a denoising scheme that denoises nucleotide bases in the reads while also updating quality score values when necessary. We show that this scheme results in more accurate variant calling in high-coverage datasets.
Availability and implementation: SAMdude is written in Python and can be downloaded from https://github.com/ihwang/SAMdude
Contact: ihwang@stanford.edu or mhernaez@stanford.edu or idoia@illinois.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Raw sequencing data is stored in the FASTQ file format and is converted to the SAM file format following alignment of the data to a reference genome. Both file types are primarily made up of the nucleotide sequences, or "reads," as well as accompanying per-base quality score sequences that indicate the sequencer's confidence in the base calls. Although the error rate in reads varies across sequencing technologies, for some sequencing platforms the noise characteristics are well characterized. For example, Illumina sequencing technology produces "short" reads on the order of hundreds of bases, with an average substitution error rate of less than 1% and insertion and deletion rates orders of magnitude lower (Minoche *et al.*, 2011). These errors can affect downstream applications, with an important application being variant calling, or the detection of single nucleotide polymorphisms (SNPs) including substitutions, and insertions or deletions (INDELs). Variant identification from whole genome sequencing data is increasingly being used to diagnose, gain biological insight to, and design treatments in the clinical setting, especially in the field of rare genetic disease research(Boycott *et al.*, 2013). Thus, precision in variant identification is paramount.

Many efforts to denoise reads have been proposed, relying on a variety of techniques including k-mer counting and statistical error models, and also targeting either substitution errors, insertion and deletion errors, or a combination of both (Laehnemann *et al.*, 2016). While many denoisers perform well in terms of correcting basecalling errors (Lee *et al.*, 2015) and increasing both breadth and depth of reads coverage (Molnar and Ilie, 2015), they do not address the effect of changing base pairs independently of adjusting the corresponding quality scores. Because the quality scores are a direct function of the analog signals used to determine the called base, the quality score and called base are intimately related and one cannot be changed without considering a potential effect on the other. Furthermore, only a select few denoisers such as Musket (Liu *et al.*, 2013), Racer (Ilie and Molnar, 2013) and SGA (Simpson and Durbin, 2010) can accomodate human whole genome sequencing (WGS) data.

We propose a denoising scheme, SAMdude, based on the discrete universal denoiser (DUDE) detailed in Weissman *et al.*, 2005. Unlike the aforementioned denoisers which denoise reads stored in the FASTQ file format, SAMdude takes advantage of alignment information contained in the SAM file in order to both denoise reads as well as update quality scores, toward the end goal of improving the quality of variant calls. In contrast, the previously mentioned denoisers operate on pre-aligned FASTQ files and focus on increasing the coverage of pre-aligned reads, as well as improving the mappability of unaligned reads. As a result, it is difficult to directly compare the proposed denoiser's performance to that of other denoisers. We instead benchmark SAMdude's performance using variant calling on a dataset for which a consensus sequence exists. We assess our denoiser's performance on real datasets by evaluating the accuracy of variant calls when reads are denoised and quality scores are updated using SAMdude. We use the pipeline described in Krusche, 2016, which benchmarks single SNP and INDEL variant calls against gold standard truth datasets.

2 Methods

In this section we outline the problem of denoising in the genomic sequencing setting, and describe the proposed denoiser.

2.1 Problem setting

High-throughput sequencing methods use a shotgun approach and generate a large number of short, overlapping reads of length on the order of hundreds of base pairs. When the sample source is known and a reference genome is available, the reads are aligned against a reference genome in order to generate a sequence estimate. However, errors introduced during the sequencing process result in reads with noise (Figure 1). Our goal is to denoise individual bases in reads and update the corresponding quality scores in order to improve the accuracy of the downstream procedure of variant identification, while still preserving polymorphisms that are unique to the individual.

In the genomic setting, the denoising problem is as follows: we assume a clean sequence $\mathbf{x}^n$ corresponding to the true genome sequence of length n, and consider a set of noisy sequences $\{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$ corresponding to the reads generated by the sequencing machine with components taking values in alphabet $\mathcal{A}$, accompanied by a corresponding set of quality score strings $\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_m\}$ with components taking values in the set of ASCII characters quantifying basecalling quality on some quality score scale (Group, 2016). Typically, $\mathcal{A}$ is the set of all possible nucleic bases $\{\mathrm{A}, \mathrm{T}, \mathrm{G}, \mathrm{C}\}$ as well as potentially other symbols indicating ambiguity. Since the reads are considered with their quality score strings, the alphabet of the input to the denoiser is the set of tuples of nucleotide bases and quality scores. We assume that a reference genome is available, the reads are aligned and that alignment information is available. Our goal is to denoise the nucleotide bases and update their corresponding quality scores in order to improve variant calling.

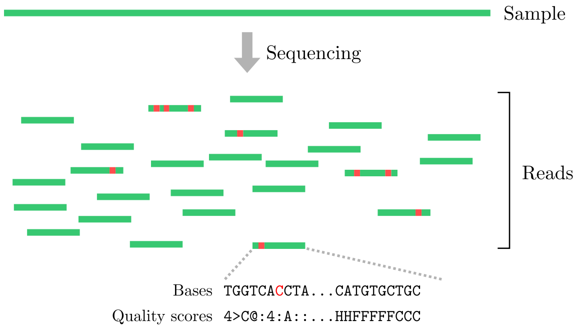
Sample
Sequencing
Reads
Bases TGGTCACCTA...CATGTGCTGC
Quality scores 4>C@:4:A::...HHFFFFFCCC

Fig. 1. Cartoon illustrating the noisy sequencing process. The sample is represented by the green line, with reads depicted as shorter green lines. Red vertical bars in the reads indicate positions with sequencing errors.

2.2 Denoising method

Because our denoising method is inspired by the DUDE algorithm proposed in Weissman *et al.*, 2005, we first briefly describe the DUDE denoiser and then describe in detail our proposed denoiser, SAMdude.

2.3 Denoising background: DUDE

The DUDE Weissman *et al.*, 2005 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. Intuitively, the DUDE denoising procedure can be viewed as an estimation of a conditional distribution on the noisy sequence conditioned on the noise-free sequence.

Let us consider the setting in which the components of the noise-free sequence x^n take values in an A-letter alphabet $\mathcal{A}$, the components of the noisy sequence z^n take values in an B-letter alphabet $\mathcal{B}$ and the noise channel is specified by $A \times B$-dimensional transition matrix $\mathbf{\Pi}$. As a sliding-window denoiser, DUDE considers each string u_{-k}^k of length $2k+1$ in a corrupted sequence z^n and estimates the probability of observing the center-flanking string $u_{-k}^{-1}u_1^k$ of length $2k$ in the noise-free sequence x^n with central symbol $u_0 = a$ for all $a \in \mathcal{A}$.

For every double-sided context string (u_{-k}^{-1}, u_1^k) of length $2k$ observed in the corrupted sequence z^n with central symbol u_0, the conditional probability distribution is estimated using only $\mathbf{\Pi}$ and a B-dimensional vector of counts whose b^{th} component ($b \in \mathcal{B}$) is equal to

$$\mathbf{m}(z^n, u_{-k}^{-1}, u_1^k)[b] = |\{i : k+1 \leq i \leq n-k, z_{i-k}^{i+k} = u_{-k}^{-1} b u_1^k\}|, \tag{1}$$

or in words, the number of appearances of the string $u_{-k}^{-1} b u_1^k$ in z^n.

The conditional distribution estimate takes the form

$$\hat{\mathbf{q}}(z^n, x^n, u_{-k}^k) = \pi_{u_0} \odot [\mathbf{\Pi}^{-T}\mathbf{m}(z^n, u_{-k}^{-1}, u_1^k)], \tag{2}$$

where $\hat{\mathbf{q}}$ is a vector of length A and is close to the desired conditional distribution. For nonsquare channel matrices, the inverse transpose matrix $\mathbf{\Pi}^{-T}$ is replaced with pseudoinverse $(\mathbf{\Pi}\mathbf{\Pi}^T)^{-1}\mathbf{\Pi}$. We refer the reader to Weissman *et al.*, 2005 for more details.

2.4 Denoising aligned genomic sequences: SAMdude

In order to apply a DUDE-like denoising framework to the genomic sequencing setting, we make several assumptions about the problem setting and make certain algorithm design choices which we outline below.

While the universal denoising setting assumes a single noise-free input sequence and a single noise-corrupted output sequence, both of equal length, in the high-throughput sequencing setting we consider the reads to be overlapping, individual noisy output sequences output from a noisy channel whose input is a single noise-free input sequence. 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. The universal denoising setting also assumes that the noise channel is memoryless and known, and corrupts sequences only with substitution errors. In the sequencing setting, the error characteristics of sequencing technologies are generally known, but in order to account for indivdual variations in performance from machine to machine and to avoid the potentially confounding effects of difficult-to-sequence regions in the genome, we use read alignment information to generate our own estimate of the channel's noise-injecting characteristics. Furthermore, while all sequencing technologies inject all three types of errors, the preponderance of substitution errors in Illumina's sequencing technology allows us to consider Illumina sequencing data to be a good candidate for DUDE-based universal denoising.

For these reasons we propose SAMdude which performs denoising on SAM files containing read alignment information and quality scores. We devote the following subsections to detailed descriptions of the denoising procedure as well as the denoising subtasks of channel estimation and reads processing. In these subsections, for ease of exposition and with some abuse of notation, we denote an arbitrary read as $\mathbf{z}$ with i^{th} component z_i accompanied by quality score q_i which is converted from the i^{th} component of the ASCII quality score string accompanying the read, using the appropriate scale (e.g., Phred+33).

2.4.1 Denoising procedure

Before detailing each of the steps involved in SAMdude, we first give a brief overview of the proposed denoising procedure.

The denoising procedure begins with channel estimation and counts vector acquisition. Once the channel estimate and vectors of counts $\mathbf{m}$ for all context strings of length $2k$ are obtained, SAMdude produces a conditional distribution estimates $\hat{\mathbf{q}}$ via Equation (2) for each context string using the channel estimate $\hat{\mathbf{\Pi}}$ and the counts vector $\mathbf{m}$. The denoising procedure is as follows: for base z_i and corresponding quality score q_i in read $\mathbf{z}$, identify the length $2k$ context string $u_{i-k}^{i-1}u_{i+1}^{i+k}$ surrounding position i. Calculate distribution estimate $\hat{\mathbf{q}}(\mathbf{z}, \mathbf{x}^n, u_{i-k}^{i-1}u_{i+1}^{i+k})$, use the argmax of the distribution estimate as the denoised base, and update the base's corresponding quality score using the max of the distribution estimate.

We perform denoising at read positions where z_i as well as the surrounding context string contain only bases in $\mathcal{A} = \{\mathrm{A}, \mathrm{T}, \mathrm{G}, \mathrm{C}\}$ comprising the nucleotide bases and none of the symbols indicating ambiguity. We limit ourselves to this simple alphabet in order to avoid basing denoising decisions on non-uniquely identifiable context strings. We also set a quality score threshold above which we do not attempt denoising, since bases that the sequencer assigns a high quality score are most likely not noisy and may not benefit from denoising.

2.4.2 Channel estimation

The channel input is assumed to be simply the noise-free sequence alphabet $\mathcal{A}$. The channel output is a tuple of the called base and the quality score associated with that base. Typically, the alphabet size of quality scores is around 40. The large size of the quality score alphabet makes computation unfeasible, and so we bin the quality scores into eight bins $\{\mathrm{bin}_1, \mathrm{bin}_2, ..., \mathrm{bin}_8\}$ with bin limits corresponding to those recommended by Illumina for reducing quality score resolution (Illumina, 2012). As a result, our output alphabet is of size 32 and is the set of tuples of all possible combinations of the nucleotide bases and the 8 quality score bins, i.e., $\mathcal{B} = \{(\mathrm{A}, \mathrm{bin}_1), (\mathrm{T}, \mathrm{bin}_1), (\mathrm{G}, \mathrm{bin}_1), (\mathrm{C}, \mathrm{bin}_1), \ldots, (\mathrm{A}, \mathrm{bin}_8), (\mathrm{T}, \mathrm{bin}_8), (\mathrm{G}, \mathrm{bin}_8), (\mathrm{C}, \mathrm{bin}_8)\}$.

To estimate $\mathbf{\Pi}$, we first perform a sequence pileup at every reference genome position by cataloging all reads at that position, as illustrated in Figure 2. At each position we assume that the majority base, for some majority threshold t_m, is the true base at that position. If there is no clear majority base, then we do not use bases at that position for channel estimation. This measure allows us to utilize an overwhelming majority of genomic positions in order to estimate the channel characteristics, while preventing confusion from positions in the genome that are heterozygous due to the polyploidy of the organism being sequenced. For each base in $\mathcal{A}$ we record the number of bases in an 8×32 conditional counts matrix

$$\mathbf{N} = \begin{bmatrix} \mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\,\mathrm{A}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\,\mathrm{A}} & \cdots & \mathrm{n}_{(\mathrm{C},c_8)|\,\mathrm{A}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\,\mathrm{C}} & \cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\,\mathrm{C}} & \cdots\cdots & \mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\,\mathrm{C}} \end{bmatrix},$$

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 an estimate of $\mathbf{\Pi}$, which we denote by $\hat{\mathbf{\Pi}}$

$$\hat{\mathbf{\Pi}} = \begin{bmatrix} \frac{\mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\,\mathrm{A}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{A}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\,\mathrm{A}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{A}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\,\mathrm{A}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{A}}} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \frac{\mathrm{n}_{(\mathrm{A},\mathrm{bin}_1)|\,\mathrm{C}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{C}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_1)|\,\mathrm{C}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{C}}} & \cdots & \frac{\mathrm{n}_{(\mathrm{C},\mathrm{bin}_8)|\,\mathrm{C}}}{\sum\limits_{\forall b \in \mathcal{B}} \mathrm{n}_{b|\,\mathrm{C}}} \end{bmatrix}. \qquad (3)$$

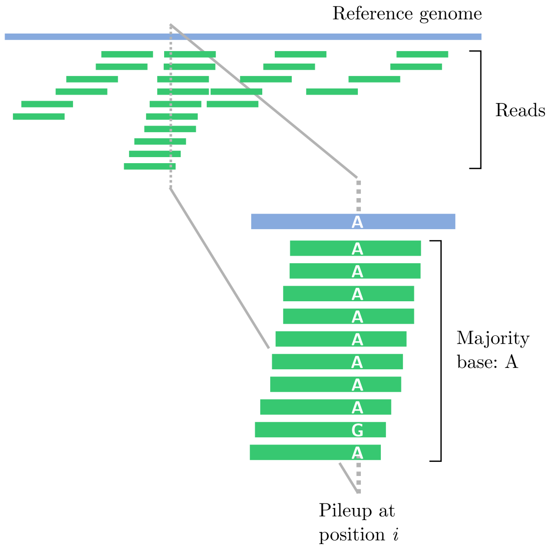

Fig. 2. Cartoon of the pileup procedure. The reference genome against which read alignment was obtained is shown in blue, and reads are shown in green. Bases in the reads at given position i are shown in white.

Note that the channel matrix is nonsquare, and thus when calculating equation 2, we use the pseudoinverse $(\hat{\mathbf{\Pi}}\hat{\mathbf{\Pi}}^T)^{-1}\hat{\mathbf{\Pi}}$.

Because the matrix output alphabet comprises the set of tuples of nucleotide bases and quality score bins, when calculating the distribution estimate $\hat{\mathbf{q}}$ SAMdude employs channel matrix column $\pi_{(z_i,\mathrm{bin}_i)}$ where bin_i is the bin containing quality score q_i.

2.4.3 Reads processing

The raw reads reported from a sequencing machine frequently cannot be mapped directly to a 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 are summarized by the sequence aligner in a CIGAR string accompanying the read (Group, 2016). Additionally, large portions of the read may be assigned very low quality scores, indicating entire regions of the read for which the denoiser 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.

Channel estimation relies on the creation of pileups at reference genome positions. As a result, in this step only bases that map to the reference genome are considered and bases that are designated in the CIGAR string as low-confidence and non-matching ("soft-clipped", or simply "clipped") or inserted relative to the reference genome are disregarded.

When acquiring vectors of counts $\mathbf{m}$ in Equation (1), our goal is to obtain histograms of the appearance of all unique context strings of length $2k$ flanking a central base. These context strings are unique to the individual and should include the individual's 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. In order to maximize the number of context strings obtained from a processed read, we additionally pad the read with a headers and footers of length up to k if the read begins or ends, respectively, with bases that are able to be mapped to the reference genome, i.e. not insertions. The padding process is illustrated in Figure 3.

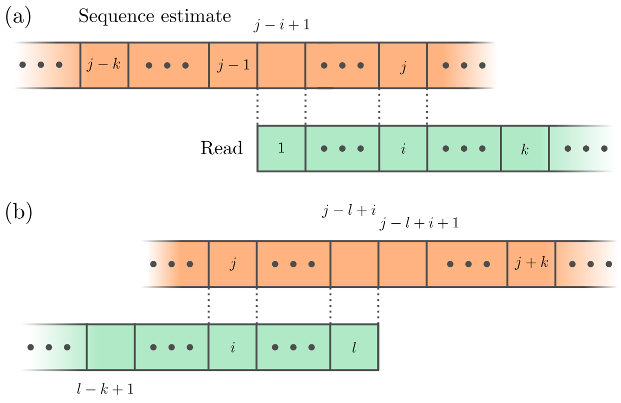

Fig. 3. During the context acquisition or denoising processes, reads (green) are padded with bases taken from the sequence estimate. a) The central base for which a context string is being acquired is located at position i in the read corresponding to position j in the sequence estimate, and $1 \leq i \leq k$. The left half of the context for that base is padded with sequence estimate bases from positions $j-k$ up to $j-i+1$. b) Here, $i \geq l-k+1$, where l is the length of the processed read. The right half of the context for the base at position i is padded with sequence estimate bases from position $j-l+i+1$ up to $j+k$.

When denoising, we again require the context string of length $2k$ surrounding a given base to be denoised. Now, we consider all bases regardless of their categorization are utilized since bases in the reads that were designated as insertions or deletions relative to the reference genome may very well be true polymorphisms, and clipped regions may benefit from denoising. When the read begins or ends in bases that can be mapped to the reference genome, it is padded with bases from the sequence estimate using the same procedure described above.

2.4.4 Quality score updating

The max of the conditional distribution estimate $\hat{\mathbf{q}}$ is used to update the quality score accompanying the denoised base. The updating procedure varies depending on whether or not the base the denoiser selected matches the original base z_i. If the denoiser does not recommend a change of base, the quality score is adjusted as follows: the original quality score q_i is converted into a confidence probability $p_i = 1 - 10^{-q_i/10}$ (i.e., $p_i = 1 - p_{i,e}$, where $p_{i,e}$ is the sequencer's base-calling error probability for base z_i), average p_i with the maximum estimated conditional probability $p_{\max} = \max \hat{\mathbf{q}}(\mathbf{z}, \mathbf{x}^n, u_{-k}^k)$, and back-convert the averaged probability into a quality score, i.e. the updated quality score is

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

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 incorporate the denoiser's conditional probability estimates with the sequencer's confidence. Since the original quality score is a function of the original base call, if the denoiser agrees with the basecall, we consider both the denoiser's probability estimate and the sequencer's quality score equally. However, if the denoiser decides on a different base then the original quality score is unrelated to the denoiser's chosen base and we disregard the original quality score in favor of a quality score converted from the denoiser's probability estimate.

3 Results

In order to gauge the performance of SAMdude, we performed tests on real human datasets of the *H. Sapiens* individual NA12878 for which extensive sequencing has resulted in the availability of various "ground truth" variant calling datasets. In this study, we use the ground truth released by the Genome in a Bottle consortium (GIAB) released by the National Institute of Standards and Technology (NIST) .

3.1 Evaluation criteria

We evaluate our denoiser's performance by analyzing the effect on variant calling when bases are denoised and quality scores are updated. The datasets used in this study have a consensus sequence or ground truth, so we can calculate the number of variant calls for the denoised and original files that are true positives (T.P.), false positives (F.P.), and false negatives (F.N.). In brief: T.P. variants are variants that were identified using the denoised reads known to be present in the consensus sequence, and F.N. variants are variants that were not identified using the denoised reads, but were indeed present in the consensus sequence. We consider the following performance metrics: sensitivity, which measures the proportion of all the variants that are correctly called $\left(\frac{\text{T.P.}}{\text{T.P.+F.N.}}\right)$, precision, which measures the proportion of called variants that are true $\left(\frac{\text{T.P.}}{\text{T.P.+F.P.}}\right)$, and F-score, which is the harmonic mean of the sensitivity and precision. We use the pipeline in Krusche, 2016 to compare the variant calls of the denoised files to those of the original file, and compared the effect of denoising on both raw variant calls as well as those filtered by the GATK variant quality score recalibration (VQSR) procedure DePristo *et al.*, 2011. For more details on the evaluation pipeline, we refer the reader to the Supplementary Data.

3.2 Human chromosome denoising

For all denoising experiments, we used a single-sided context length of $k = 7$ (14 bases total in the double-sided context), both for computational feasibility and also in order to maximize the number of counts in each context histogram without skewing the histograms towards a uniform distribution. 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 eliminate potentially confounding effects at heterozygous genomic positions which might not have a clear majority base. We used a confidence probability threshold of $t_p = 0.9$, focusing on denoising only the bases for which the sequencer has low basecalling confidence. For more details on parameter choice, we refer the reader to the Supplementary Data.

We tested the proposed denoiser on extracted SAM files for chromosomes 20 and 11 from NA12878 paired-end WGS dataset ERR262997, corresponding to 30×-coverage. Chromosome 20 was chosen for its ubiquity in testing, and chromosome 11 was chosen as representative of the longer chromosomes. The results of variant-calling on the SAMdude-denoised datasets are shown in Table 1. In all cases, SAMdude denoising and quality score updating resulted in improvements in both sensitivity and precision. For chromosome 11, denoising resulted in fewer variants being called overall, while for chromosome 20, denoising resulted in more variants being called overall. Despite this difference in total number of variants called, denoising of these two chromosomes resulted in the identification of 140 additional true positive variants and 406 fewer false positives over the original files.

The gains in improvement are slightly smaller after VQSR filtering, indicating that some of the variants identified after denoising are low-confidence variants. However, it is notable that the change in both sensitivity and precision are still positive even after the variant calls are VQSR filtered. This indicates that the denoising process does not interfere with current recommended analysis pipelines, and may indeed enhance their performance.

	Raw variant calls				VQSR filtered variant calls			
Chr	ΔC	ΔS [%]	ΔP [%]	ΔF	ΔC	ΔS [%]	ΔP [%]	ΔF
11	-259	0.06	0.21	0.001	-215	0.04	0.15	0.001
20	86	0.05	0.09	0.001	72	0.04	0.07	–

Table 1. Denoising results, with changes (Δ), T.P. and F.P. calculated relative to the original file. C is the number of variants called for each condition, sensitivity $\mathrm{S} = \left(\frac{\mathrm{T.P.}}{\mathrm{T.P.+F.N.}}\right)$, precision $\mathrm{P} = \left(\frac{\mathrm{T.P.}}{\mathrm{T.P.+F.P.}}\right)$, and F-score $\mathrm{F} = \frac{1}{2}(\mathrm{S} + \mathrm{P})$. For S, P and F, positive Δ indicates improvement with respect to the original data, and horizontal lines indicate no change.

Table 2 compares the number of bases SAMdude chose to change and the total number of bases in the original SAM files. The number of bases changed is on the conservative end of the estimates of Illumina's noise-injecting characteristics, but within the expected range. Similarly, Table 3 displays the number of true and false positives in the original SAM file, and the number of additional true positives and false positives (a negative change in false positives indicates a decrease in the number of false positives) in the denoised SAM file, respectively.

Chr	N_0	N_Δ	%Δ
11	4,675,126,885	15,910,155	0.34
20	2,063,339,605	7,239,425	0.35

Table 2. Table summarizing the number of bases changed by SAMdude. Number of bases in the original SAM file (N_0) is shown in the second column, number of bases changed by SAMdude is shown in the third column (N_Δ), and the percent of bases changed by SAMdude is shown in the fourth column (%Δ).

	Raw variant calls				VQSR filtered variant calls			
Chr	T_0	F_0	ΔT	ΔF	T_0	F_0	ΔT	ΔF
11	153,840	7,075	36	-61	153,555	2,090	29	-45
20	64,664	3,575	94	-345	64,539	4,091	68	-256

Table 3. Table summarizing the change in number of true and false positive bases due to denoising using SAMdude with $k = 7$. Number of T.P. and F.P. in the original SAM file (T_0 and F_0, respectively) are shown in the second and third columns, respectively, and number of changes in T.P. and F.P. (ΔT and ΔF) are shown in the third column and fourth columns, respectively.

3.3 Comparison to random noise

As a sanity check, we compared the performance of SAMdude and quality score updating scheme to that of a file in which bases were changed at random, and quality scores were not updated. For this comparison test, we employed the chromosome 20 data a smaller value of k for SAMdude which still resulted in improvements in variant calling. Addition of random noise (base changes only) to a high-coverage dataset at a rate equal to the base changes prescribed by SAMdude with $k = 6$ resulted in many fewer variant calls, resulting in an increase in the precision of calls but at the cost of simultaneously decreasing the sensitivity (Table 4). While this sometimes resulted in a slight increase in F-score for unfiltered variant calls, the ratio of decrease in sensitivity to increase in precision was much larger for VQSR filtered variant calls, and resulted in a relatively large decrease in F-score. In contrast, SAMdude for $k = 6$ increased the number of true positive calls while simultaneously decreasing the number of false positive and false negative calls (see Supplementary Data). The robustness of our denoiser is seen in the maintenance of the F-score and the consistent increase in both precision and sensitivity for both raw and VQSR filtered variant calls.

	Raw variant calls				VQSR filtered variant calls			
Noising Run	ΔC	ΔS [%]	ΔP [%]	ΔF	ΔC	ΔS [%]	ΔP [%]	ΔF
1	-2428	-0.94	1.14	0.001	-2452	0.43	-0.65	-0.006
2	-2495	-0.98	1.33	0.002	-2464	0.64	-0.48	-0.005

Table 4. Variant calling results for random noise added to chromosome 20 at a rate of SAMdude base changes for $k = 6$. 5,595,920 bases were changed.

4 Discussion

We have presented SAMdude, an algorithm that performs both genomic sequence denoising and quality score updating. Unlike other denoisers, the algorithm utilizes alignment information in order to improve variant calling. We have shown that denoising and quality score updating via the SAMdude procedure results in improvements in both sensitivity and precision of variant calling, and occasionally even boosts F-score. Importantly, application of the algorithm in no way hinders post-variant calling filtering methods.

Future work includes comparison of denoising performance against existing state-of-the-art denoisers. Such a comparison will necessitate denoising of the entire WGS dataset, since SAMdude operates on a different file format than other denoisers. Additionally, changes to the implementation to improve computational efficiency and feasibility will be valuable for evaluating larger datasets.

Acknowledgements

The authors would like to thank Dmitri Pavlichin for helpful discussions and suggestions.

Funding

This work has been supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

Boycott, K. M., Vanstone, M. R., Bulman, D. E., and MacKenzie, A. E. (2013). Rare-disease genetics in the era of next-generation sequencing: discovery to translation. *Nature Reviews Genetics*, **14**(10), 681–691.

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. *Nature genetics*, **43**(5), 491–498.

Group, T. S. F. S. W. (2016). Sequence alignment/map format specification.

Ilie, L. and Molnar, M. (2013). Racer: Rapid and accurate correction of errors in reads. *Bioinformatics*, page btt407.

Illumina, I. (2012). Illumina White Paper: Informatics sam/bam and related specifications.

Krusche, P. (2016). Haplotype comparison tools.

Laehnemann, D., Borkhardt, A., and McHardy, A. C. (2016). Denoising dna deep sequencing dataâŁ"high-throughput sequencing errors and their correction. *Briefings in Bioinformatics*, **17**(1), 154–179.

Lee, B., Moon, T., Yoon, S., and Weissman, T. (2015). DUDE-Seq: Fast, flexible, and robust denoising of nucleotide sequences. *ArXiv e-prints*.

Liu, Y., Schröder, J., and Schmidt, B. (2013). Musket: a multistage k-mer spectrum-based error corrector for illumina sequence data. *Bioinformatics*, **29**(3), 308–315.

Minoche, A. E., Dohm, J. C., and Himmelbauer, H. (2011). Evaluation of genomic high-throughput sequencing data generated on illumina hiseq and genome analyzer systems. *Genome biology*, **12**(11), 1.

Molnar, M. and Ilie, L. (2015). Correcting illumina data. *Briefings in Bioinformatics*, **16**(4), 588–599.

Simpson, J. T. and Durbin, R. (2010). Efficient construction of an assembly string graph using the fm-index. *Bioinformatics*, **26**(12), i367–i373.

Weissman, T., Ordentlich, E., Seroussi, G., Verdu, S., and Weinberger, M. J. (2005). Universal discrete denoising: known channel. *IEEE Transactions on Information Theory*, **51**(1), 5–28.