Sequence analysis

DUDE-Seq: Fast, flexible, and robust denoising of nucleotide sequences

Byunghan Lee, Taesup Moon, Sungroh Yoon, and Tsachy Weissman

Bioinformatics Institute & Electrical and Computer Eng., Seoul National University, Seoul 08826, Korea, Department of Information and Communication Engineering, DGIIST, Daegu 42988, Korea, and Department of Electrical Engineering, Stanford University, Stanford, CA 94305, USA

Abstract

Motivation: We consider the correction of errors from nucleotide sequences produced by next-generation sequencing. The error rate in reads has been increasing with the shift of focus of mainstream sequencers from accuracy to throughput. Denoising in high-throughput sequencing is thus becoming a crucial component for boosting the reliability of downstream analyses.

Results: Our methodology, named DUDE-Seq, is derived from a general setting of reconstructing finite-valued source data corrupted by a discrete memoryless channel and provides an effective means for correcting substitution and homopolymer indel errors, the two major types of sequencing errors in most high-throughput sequencing platforms. Our experimental studies with real and simulated data sets suggest that the proposed DUDE-Seq not only outperforms existing alternatives in terms of error-correction capabilities and time efficiency, but also boosts the reliability of downstream analyses. Further, the flexibility of DUDE-Seq enables us to robustly apply it to different sequencing platforms and analysis pipelines by a simple update of the noise model.

Availability: http://data.snu.ac.kr/pub/dude-seq

Contact: tsmoon@dgist.ac.kr, sryoon@snu.ac.kr

1 Introduction

A new generation of high-throughput, low-cost sequencing technologies, referred to as next-generation sequencing (NGS) (Metzker 2010), is reshaping biomedical research including large-scale comparative and evolutionary studies (Astbury 1961; Bateson 1894; Riesenfeld et al. 2004). Compared with the automated Sanger sequencing, NGS machines produce significantly shorter reads in a large quantity, posing various new computational challenges (Pop and Salzberg 2008).

To detect the sequences of fluorescent labels at the molecular level, NGS technologies normally rely on imaging systems that require amplified templates using emulsion polymerase chain reaction (PCR) or solid-phase amplification (Metzker 2010). These amplification and imaging processes can cause erroneous reads, the origin of which can be traced into incorrect determination of homopolymer lengths, erroneous insertion/deletion/substitution of nucleotide bases, and PCR chimera (Shendure and Ji 2008). Substitution errors dominate in many platforms including Illumina, while homopolymer errors manifested as insertions and deletions (indels) are also abundant in Roche’s 454 and Ion Torrent.

Erroneous reads must be properly handled since they complicate downstream analysis (e.g., variant calling and genome assembly), often lowering the quality of the whole analysis pipeline. Soft clipping, which trims 3’-ends of a read based on the quality scores of individual bases, may be the simplest approach, but it results in loss of information (Yang et al. 2013). More sophisticated methods aim at detecting and correcting errors in sequence data (Ilie et al. 2011; Kao et al. 2011; Kelley et al. 2010; Qi et al. 2009; Salmela 2010; Salmela and Schröder 2011; Schröder et al. 2009; Wijaya et al. 2009; Yang et al. 2011, 2010). Given the widespread use of Illumina sequencing platforms, most error-correction algorithms have targeted substitution errors (Yang et al. 2013).

As summarized in the recent survey papers (Yang et al. 2013; Laehnemann et al. 2015), the current error-correction methods for NGS can be categorized as follows: k-mer (i.e., oligonucleotide of length k) frequency/spectrum based, multiple sequence alignment (MSA) based,
and statistical error-model based methods. The idea of \( k \)-mer based methods [Kelley et al., 2010; Yang et al., 2010; Medvedev et al., 2011; Nikolenko et al., 2013; Greenfield et al., 2014; Lim et al., 2014] is that \( k \)-mers within a small edit distance from each other belong to the same location at a reference genome. Under the assumption that the errors are rare and random, and that the coverage is uniform, for sufficiently large \( k \), it is reasonable to expect that most of the substitution errors alter \( k \)-mers to inexact ones in a genome. Thus, along with high-coverage genome sequences from NGS data, we may identify suspicious \( k \)-mers and correct them to a consensus. MSA based methods [Salmen and Schröder, 2011; Kao et al., 2011; Bragg et al., 2012] work by aligning three or more evolutionarily related sequences and then creating a putatively error-free consensus sequence. The statistical error-model based methods [Meacham et al., 2011; Yin et al., 2013; Schulze et al., 2014] create an empirical confusion model from datasets, exploiting the information obtained from the Phred quality score [Ewing et al., 1995] (i.e., a measure of the quality of the identification of the nucleobases generated by automated DNA sequencing) or alignment results. Many of the existing tools combine \( k \)-mer based, MSA based, and statistical error-model based techniques in a complementary way for performance boosts.

While the above methods often deliver state-of-the-art performance, they commonly suffer from several limitations. First, although some algorithms make certain stochastic modeling assumptions on the underlying DNA sequences, little attention is given to the validity and accuracy of such modeling assumptions, let alone to theoretical analysis on whether near optimum or sound error-correction performance is attained. Second, the performance of an algorithm is sensitive to the sequencing platforms; for example, the \( k \)-mer based schemes tend to perform well when a sequencer has uniform coverage over the sequences, whereas the MSA based methods tend to perform well when the coverage is expected to vary over the sequences. Third, existing algorithms require meticulous fine-tuning to each sequencing platform in order to obtain reasonable results. This requires time-consuming manual tweaks, and it is consequently difficult to rapidly and flexibly apply these algorithms to various (existing and emerging) sequencing platforms. Finally, the number of sequences is often not preserved after denoising, given that most existing algorithms return only representative (consensus) denoised sequences created by merging input sequences. In some applications this may result in inconsistencies in the downstream analyses.

To alleviate such limitations, we adapt an algorithm called Discrete Universal DENoiser (DUDE) [Weissman et al., 2005] to the DNA sequence error correction problem. DUDE was developed for a general setting of reconstructing sequences with finite-valued components (source symbols) corrupted by a discrete memoryless channel (DMC), a noise mechanism that corrupts each source symbol independently and statistically identically. The original paper [Weissman et al., 2005] showed rigorous performance guarantees of DUDE for the semi-stochastic setting; namely, that where no stochastic modeling assumptions are made on the underlying source data, while the corruption mechanism is assumed to be governed by a known DMC. DUDE is shown to universally attain the optimum denoising performance (in a sense appropriate for the semi-stochastic setting) for any source data as the data size grows.

The semi-stochastic modeling approach from the DUDE framework naturally fits the setting of DNA sequence denoising problems and alleviates the first limitation mentioned above. Namely, it is difficult to come up with accurate stochastic models for DNA sequence, but it is simple and fairly realistic to assume certain memoryless noise models (i.e., DMC models) for the sequencing devices. Furthermore, the algorithmic property of DUDE, which will be elaborated shortly, enables overcoming the other limitations as well; applying the algorithm to different sequencing devices requires a simple change of the DMC model for each device in flexible manner, and the number of reads is preserved since DUDE performs denoising one sequence read at a time. Furthermore, our experimental results show that the performance of the DUDE-based method is robust to the change of sequencing platforms and consistently outperforms the representative state-of-the-art schemes with fast running time.

We apply two versions of DUDE separately for substitution and homopolymer errors, the two major types of sequencing error. For substitution error, our approach utilizes the original DUDE which was developed in a framework that does not cover errors of the homopolymer type. To correct homopolymer errors, we thus adopt a variant of DUDE for general-output channels [Dembo and Weissman, 2005]. Our homopolymer-error correction is applicable to the cases in which base-called sequences and the underlying flowgram intensities are available (e.g., pyrosequencing). For brevity, we refer to both of our DUDE-based approaches as DUDE-Seq in the remainder of the paper. The reader will be able to disambiguate from the context.

2 Discrete Universal DENoiser (DUDE)

Fig. 1 shows the concrete setting of the discrete denoising problem. We denote the underlying source data as \( \{X_i\} \) and assume each component takes values in some finite set \( \mathcal{X} \). The resulting noisy version of the source corrupted by a DMC is denoted as \( \{Z_i\} \), and its components take values in, again, some finite set \( \mathcal{Z} \). The DMC is completely characterized by the channel transition matrix \( \Pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{Z}|} \), of which the \( (x, z) \)-th element, \( \Pi(x, z) \), stands for \( \Pr(Z_i = z | X_i = x) \), i.e., the conditional probability of the noisy symbol taking value \( z \) given the original source symbol was \( x \). Furthermore, throughout this paper, we generally denote a sequence \( (n\text{-tuple}) \) as, e.g., \( a^n = (a_1, \ldots, a_n) \), and \( a_i' \) refers to the subsequence \( (a_1, \ldots, a_j) \).

As shown in Fig. 1, a discrete denoiser observes the entire noisy data \( Z^n \) and reconstructs the original data with \( \hat{X}^n = (\hat{X}_1(Z^n), \ldots, \hat{X}_n(Z^n)) \). The goodness of the reconstruction by a discrete denoiser \( \hat{X}^n \) is measured by the average loss,

\[
L_{\mathcal{X}}(X^n, \hat{X}^n) = \frac{1}{n} \sum_{i=1}^{n} \Lambda(X_i, \hat{X}_i(Z^n)),
\]

where \( \Lambda(x_i, \hat{x}_i) \) is a single-letter loss function that measures the loss incurred by estimating \( x_i \) with \( \hat{x}_i \) at location \( i \). The loss function can be also represented with a loss matrix \( \mathbf{A} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|} \).

DUDE [Weissman et al., 2005] is a two-pass algorithm that has linear complexity in the data size \( n \). During the first pass, the algorithm collects the statistics vector

\[
\mathbf{m}(z^n, t^k, r^k)[a] = \{i : k + 1 \leq i \leq n - k, z_{i+k} = t^k r^k \}, \quad a \in \mathcal{Z},
\]

which is the count of the occurrence of the symbol \( a \in \mathcal{Z} \) along the noisy sequence \( z^n \) in the double-sided context \( (t^k, r^k) \in \mathcal{Z}^{2k} \). Once the \( \mathbf{m} \) vector is collected, for the second pass, DUDE then applies the rule

\[
\hat{X}_i(z^n) = \arg \min_{\hat{x} \in \mathcal{X}} \mathbf{m}^T(z^n, z_{i-k}^{-1} z_{i+k}^{1+k} \Pi^{-1} \lambda_{\hat{x}} \otimes \pi_{z_i})
\]

for each \( k + 1 \leq i \leq n - k \), where \( \pi_{z_i} \) is the \( z_i \)-th column of the channel matrix \( \Pi \), and \( \lambda_{\hat{x}} \) is the \( \hat{x} \)-th column of the loss matrix \( \mathbf{A} \). Note \( (\hat{x} \otimes \mathbf{A}) \) assumes \( \mathcal{Z} = \mathcal{Z} \) and \( \mathbf{A} \) is invertible for simplicity, but [Weissman et al., 2005] deals with more general cases as well. The form [1] shows that DUDE is a sliding window denoiser with window size \( 2k + 1 \), i.e.,

1 In some communities, \( \Pi \) is referred to as a confusion matrix.
DUDE returns the same denoised symbol at all locations with the same value of \( z_{i-k}^t \). DUDE is guaranteed of attaining the optimum performance attainable by the sliding window denoisers with the same window size as the observation length \( n \) increases. For more details, e.g., the intuition of the formula (3) and the theoretical analyses, we refer to the original paper (Weissman et al. 2005).

The original DUDE dealt exclusively with the case of \([X] \) and \([Z] \) finite. Dembo and Weissman (2005) generalized DUDE to the case of discrete input and general output channels; namely, the noisy outputs need not have their values in some finite set, but can have continuous values as well. As in Weissman et al. (2005), the memoryless noisy channel model, which in this case is characterized by the set of densities \( \{ f_x \}_{x \in X} \), was assumed known. As shown in Dembo and Weissman (2005, Fig. 1), the crux of the arguments is to apply a scalar quantizer \( Q \) to each continuous-valued noisy output \( \{ Y_i \} \) and derive a virtual DMC, \( \Gamma \in R^{[X] \times [Z]} \), between the discrete input \( \{ X_i \} \) and the quantized (hence, discrete) output \( \{ Z_i \} \). Such \( \Gamma \) can be readily obtained by the knowledge of \( \{ f_x \}_{x \in X} \) and evaluating the following integral for each \( (x, z) \); 

\[
\Gamma(x, z) = \int f_z(y|z) f_x(y) dy
\]

Once the virtual DMC is obtained, the rest of the algorithm in Dembo and Weissman (2005) proceeds similarly as the original DUDE; that is, to obtain the statistics vector \( \mathbf{m} \) for the quantized noisy outputs \( \{ Z_i \} \) during the first pass, then apply a sliding window denoising rule similar to (3), which depends on the statistics vector \( \mathbf{m} \), the virtual DMC \( \Gamma \), \( \{ f_x \}_{x \in X} \), and the noisy sequence \( Y^n \), during the second pass. A concrete denoising rule can be found in Dembo and Weissman (2005) Eqs. (16), (19), and (20).

In Dembo and Weissman (2005), a formal analysis of the generalized DUDE shows that it attains the optimum denoising performance among sliding window denoisers with the same window size, that base their denoising decisions on the original continuous-valued outputs \( Y^n \). We refer readers to the paper for more details. In the next section, we show how we adopt the idea of this generalized DUDE in our DUDE-Seq to correct the homopolymer errors in DNA sequencing.

### 3 DUDE-Seq: DUDE for DNA Sequence Denoising

#### 3.1 Substitution errors

It is straightforward to apply the original DUDE of Weissman et al. (2005) for correcting substitution errors, since the occurrence of substitution errors can be naturally modeled via a DMC, \( \Pi \). For this setting, we set \( X = Z = \{ A, C, G, T \} \), and the loss function as the Hamming loss, i.e., \( \Lambda(x, \hat{x}) = 1 \) if \( x \neq \hat{x} \), and \( \Lambda(x, \hat{x}) = 0 \), otherwise. Furthermore, using predefined reference sequences, we obtain an estimated DMC model, \( \Pi \) (more details in the next section). Fig. 3 shows the sliding-window procedure of the two passes of DUDE, namely, collecting the statistics vector \( \mathbf{m} \) and the actual denoising, with a toy example.

A more formal summary of the DUDE-Seq for the substitution errors is given in Algorithm 1. Note that the pseudocode in Algorithm 1 skips those bases whose Phred quality score is higher than a user-specified threshold and invokes DUDE-Seq only for the bases with low quality scores (lines 10–14). This is in accord with the common practice in sequence preprocessing and has nothing to do with the DUDE-Seq algorithm itself. Furthermore, for simplicity, we denoted \( z^t Z^t \) as the entire noisy DNA sequence, although the sequencing devices typically generate multiple short reads of lengths 100–200 in practice. Hence, in our experiments, we combined all the statistics vectors \( \mathbf{m} \) obtained from multiple short reads before applying the denoising rule (3).

#### 3.2 Homopolymer errors

The homopolymer errors in pyrosequencing occur while handling the observed flowgram, and a careful understanding of the error injection procedure is necessary to correct them. As described in Quince et al. (2012), in pyrosequencing, the light intensities, i.e., flowgram, that correspond to a fixed order of four DNA bases \( \{ T, A, C, G \} \) are sequentially observed. The intensity value increases when the number of consecutive nucleotides (i.e., homopolymers) for each DNA base
Algorithm 1: The DUDE-Seq for substitution errors

Require: Observation \( z^n \), Channel matrix \( \Pi \in \mathbb{R}^{k \times 4} \), Hamming loss \( \mathbf{A} \in \mathbb{R}^{10 \times 10} \), Context size \( k \). Pired quality score \( Q_n \)

Ensure: The denoised sequence \( \hat{X}^n \)

1: Define \( \mathbf{m}(z^n, t^k, r^k) \in \mathbb{R}^k \) for all \((t^k, r^k) \in \{(a, c, g, t)\}^{2k}\)
2: Initialize \( \mathbf{m}(z^n, t^k, r^k) = 0 \) for all \((t^k, r^k) \in \{(a, c, g, t)\}^{2k}\) and for all \( \alpha \in \{a, c, g, t\} \)
3: for \( i = k + 1, \ldots, n - k \) do
   \> First pass
4: \( \mathbf{m}(z^n, z_{i-k}^{-1}, z_{i+k}^1) = \mathbf{m}(z^n, z_{i-k}^{-1}, z_{i+k}^1) + 1 \) \> Update the count statistics vector
5: end for
6: for \( i = 1, \ldots, n \) do
   \> Second pass
7: if \( i \leq k \) or \( i \geq n - k + 1 \) then
8:   \( \hat{X}_i = z_i \)
9:  else
10: \( \hat{Q}_i > \text{threshold} \) then \> Quality score
11: \( \hat{X}_i = z_i \)
12: else
13: \( \hat{X}_i(z^n) = \arg \min_{z \in \{a, c, g, t\}} \mathbf{m}^T(f^n, z_{i-k}^{-1}, z_{i+k}^1) \mathbf{Pi}^{-1}(k) \cap \pi_{z_i} \) \> Apply the denoising rule
14: end if
15: end if
16: end for

Algorithm 2: The DUDE-Seq for homopolymer errors

Require: Flowgram data \( f^n \), Flowgram densities \( \{P(f|N)\}_{N=0}^\infty \), Hamming loss \( \mathbf{A} \in \mathbb{R}^{10 \times 10} \), Context size \( k \)

Ensure: The denoised sequence \( \hat{D} \)

1: Let \( Q_R(f) \) be the rounding quantizer in \[4\]
2: Let base\( (i) \in \{T, A, C, G\} \) be the DNA base corresponding to \( f_i \)
3: Define \( \mathbf{m}(f^n, t^k, r^k) \in \mathbb{R}^{10} \) for all \((t^k, r^k) \in \{(0, 1, \ldots, 9)\}^{2k}\)
4: Initialize \( \mathbf{m}(f^n, t^k, r^k) = 0 \) for all \((t^k, r^k) \in \{(0, 1, \ldots, 9)\}^{2k}\) and for all \( \alpha \in \{0, \ldots, 9\} \)
5: Let \( D = \phi, I = 0 \)
6: for \( i = 0, \ldots, 9 \) do
7: for \( j = 0, \ldots, 9 \) do
8: Compute \( \Gamma_i(j) \) following \[5\] \> Computing the virtual DMC \( \Gamma \)
9: end for
10: end for
11: for \( i = 1, \ldots, n \) do
12: Obtain \( z_i = Q_R(f_i) \) \> Note \( z_i \in \{0, \ldots, 9\} \)
13: for \( i \leq k \) or \( i \geq n - k + 1 \) then \> First pass
14: \( \mathbf{m}(f^n, z_{i-k}^{-1}, z_{i+k}^1) = \mathbf{m}(f^n, z_{i-k}^{-1}, z_{i+k}^1) + 1 \)
15: end for
16: for \( i = 1, \ldots, n \) do \> Second pass
17: if \( i \leq k \) or \( i \geq n - k + 1 \) then \( \hat{X}_i(f^n) = z_i \)
18: else
19: \( \hat{X}_i(f^n) = \arg \min_{z \in \{0, \ldots, 9\}} \mathbf{m}^T(f^n, z_{i-k}^{-1}, z_{i+k}^1) \mathbf{Pi}^{-1}(k) \cap \pi_{z_i} \) \> Note \( \hat{X}_i(z^n) \in \{0, \ldots, 9\} \)
20: end if
21: if \( \hat{X}_i(f^n) \geq 1 \) then \> Reconstructing the DNA sequence
22: \( \hat{X}_i(f^n) \) do \( \hat{D}_{I+j} = \text{base}(i) \)
23: end for
24: end if
25: \( I \leftarrow I + \hat{X}_i(f^n) \)
26: end for

4 Experimental Results

In this section, we present experimental results that show the robustness and competitiveness of DUDE-Seq across all the used platforms and data types, demonstrating its performance and flexibility. We used both real and simulated NGS datasets and compared the performance of DUDE-Seq with several state-of-the-art error correction methods. Where the flowgram intensities of base-calling were available, we corrected both homopolymer and substitution errors, otherwise, the substitution errors only. The specification of the machine we used is as follows: Ubuntu 12.04.3 LTS, 2x Intel Xeon X5650 CPUs, 64 GB main memory, and 2 TB HDD.

DUDE-Seq has a single hyperparameter \( k \), the context size, that needs to be determined. In similar to the popular \( k \)-mer based schemes, there is no analytic way to choose the best \( k \) for finite data size \( n \) (except for the asymptotic order result of \( k|X|^k = o(n/\log n) \) in \[9\], but a heuristic rule of thumb is to try with values between 2 and 8. In Section 4.1, we show the extensive comparison of the performances of DUDE-Seq with several \( k \) values for the pyrosequencing data. Furthermore, as shown in Eq. \[8\], the two adjustable matrices, \( \Pi \) and
A, \(k\), are required for DUDE-Seq. The DMC matrix \(\mathbf{D}\) for substitution errors is empirically determined by aligning each sampled read to its reference sequence as in Quince et al. (2011). Fig. 4(a) shows the non-negligible variations of the empirically obtained \(\mathbf{D}\) across the sequencing platforms. The loss matrix \(\mathbf{L}\) can be any loss matrix (i.e., Hamming or BLOSUM), and we used the Hamming loss throughout our experiments.

4.1 Experiments on real data

4.1.1 454 pyrosequencing data

Pyrosequenced 16S rRNA genes are commonly used to characterize microbial communities, due to its relatively longer reads than those of other NGS technologies (Reeder and Knight, 2010). In metagenome analysis (Schloss and Handelsman, 2005), grouping reads and assigning them to operational taxonomic units (OTUs) (i.e., binning) are essential processes. However, due to the erroneous reads, inexistnet OTUs may occur, which results in the common problem of overestimating the ground truth OTUs. Such overestimation bottlenecks the overall microbiome analysis, hence, removing the errors of reads as much as possible before assigning them to OTUs becomes a critical issue (Quince et al., 2011). With this motivation, in some of our experiments below, we used the difference of the number of assigned OTUs and the ground truth number of OTUs as a proxy of denoising performance.

We tested the performance of DUDE-Seq with the eight datasets used in Quince et al. (2011), which are mixtures of an environmental clone library. Dataset P1 has 90 clones that are mixed in two orders of magnitude difference while P2 has 23 clones that are mixed in equal proportions. In P3, P4, P5 and P6, P7, P8, there are 87 mock communities mixed in even and uneven proportions, respectively. In all datasets, both homopolymer and substitution errors exist, and the flowgram intensity values as well as uneven proportions, respectively. In all datasets, both homopolymer and substitution errors exist, and the flowgram intensity values as well as differences from golden OTUs.

In Fig. 4(b), we compare the error correction performance of three schemes, AmpliconNoise, Coral, and DUDE-Seq, in terms of the measure of concordance (MoC) (Pfitzner et al., 2009). MoC is a popular similarity measure for pairs of clusterings, and MoC of one (zero) represents perfect (no) concordance between the two clusterings. Here, the two clusterings compared are the golden OTU clusterings and the clusterings returned by denoisers. We observe that for all eight datasets, the number of OTUs generated by DUDE-Seq is consistently closer to the ground truth, giving higher MoC values than the other two schemes.

Furthermore, Fig. 5(c) compares the running time of the three schemes on the eight data sets. We can clearly see that the running time of DUDE-Seq is substantially faster than the other two. Particularly, we stress that the
running time of DUDE-Seq, even when implemented and executed with a single CPU, is two orders of magnitude faster than that of parallelized AmpliconNoise, which was run on four powerful GPU processors in this experiment. We believe such a significant boost over the state-of-the-art scheme in running time could be a compelling reason for adoption of DUDE-Seq in microbial community analysis.

4.1.2 Illumina data
At the time of writing, Illumina platforms, such as GAIIx, MiSeq, and HiSeq, are ubiquitous platforms in genome analysis. These platforms intrinsically generate paired-end reads (forward and reverse reads), due to the relatively short reads compared to the automated Sanger sequencing [Bartram et al. 2011]. Merging the forward and reverse reads from paired-end sequencing gives us elongated reads (e.g., \(2 \times 300\) bp for MiSeq) that improve the performance in downstream pipelines [Magoc and Salzberg 2011].

The Illumina platforms primarily inject substitution errors. A realistic error model is not the DMC, though, as the error rates of the Illumina tend to increase from the beginning to the end of reads; thus, the assumptions under which the DUDE was originally developed do not exactly apply to the error model of Illumina. In our experiments with DUDE-Seq, however, we still used the empirically obtained DMC model \(\Pi\) in Fig. 4, which was computed by averaging all the error rates throughout different Illumina platforms.

Fig. 6 shows per-base error correction performance on the eight real Illumina datasets reported by Schirmer et al. [2015], in which four organisms (Anaerocellum thermophilum Z-1320 DSM 6725, Bacteroides thetaotaomicron VPI-5482, Bacteroides vulgatus ATCC 8482, and Caldicellulosiruptor saccharolyticus DSM 8903) were sequenced using two different configurations (see the caption for Fig. 6 for details).

In our experiments, to see how the number of reads in a dataset affects the denoising performance, we derived 10 subsets from the original datasets by randomly subsampling 10,000 to 100,000 reads in the unit of 10,000 reads. In addition to Coral, we compared the performance of DUDE-Seq with Trowel [Lim et al. 2014], which is a recent extension of the \(k\)-mer based method. Trowel does not use a coverage threshold for its \(k\)-mer spectrum and iteratively boosts the quality values of bases after making corrections only with \(k\)-mers that have high quality values.

From the figure, we observe that DUDE-Seq robustly outperforms the two competing schemes throughout the datasets and the subsampling rates. We specifically emphasize that DUDE-Seq shows a strong performance notwithstanding the DMC assumption does not hold for the sequencer. Coral also gave a stable results but showed consistently worse results than DUDE-Seq. On the contrary, Trowel’s performance was sensitive to the \(k\)-mer size (in the figure, Trowel\((k)\) denotes a Trowel run with its \(k\)-mer size used) and mostly worse than DUDE-Seq and Coral. We believe that this result of DUDE-Seq outperforming the state-of-the-art algorithms (one based on MSA and the other based on \(k\)-mer spectrum) on the real Illumina datasets strongly demonstrates the competitiveness of DUDE-Seq as a general DNA sequence denoiser.

4.2 Experiments on simulated data
In this section, we carry out more detailed experiments using Illumina simulators in order to further highlight the strong denoising performance of DUDE-Seq, including the effect on the downstream analyses.

Fig. 7(a) shows the results we obtained using the Grinder simulator [Angly et al. 2012] and the comparison with Coral. We
**DUDE-Seq: Fast, flexible, and robust denoising of nucleotide sequences**

Fig. 6. Comparison of reads correction performance on eight real Illumina datasets (labeled Q19–Q26 [Scharmer et al. 2015]) (parameters: \(k = 5\) for DUDE-Seq; \(k = 5, 10, 15, 20\) for Trowel; \((k, m_r, m_m, g) = (21, 1, 1, 1000)\) for Coral) (Organisms: *Anaerocellum thermophilum* Z-1320 DSM 6725 (Q19 and Q23), *Bacteroides thetaiotaomicron* VPI-5482 (Q20 and Q24), *Bacteroides vulgatus* ATCC 8482 (Q21 and Q25), *Caldicellulosiruptor saccharolyticus* DSM 8903 (Q22 and Q26) [Q19–Q22: Miseq (Library: nested single index, Taq: Q5 neb, Primer: 515 & 805RA)] [Q23–Q26: Miseq (Library: NexteraXT, Taq: Q5 neb, Primer: 341f & 806rcb)]

Fig. 7. Reads correction performance on simulated dataset (parameters: \(k = 5\) for DUDE-Seq; \(k = 10\) for Trowel; \((k, m_r, m_m, g) = (21, 1, 1, 1000)\) for Coral; optimal values set by tool seq-analy for Reptile): (a) varying error rates using the Grinder simulator [Angly et al. 2012] (b) varying reads composition using the GemSiM simulator [McElroy et al. 2012] (values on top of each bar represent the error rates).
generated nine synthetic datasets of forward reads that had the error rates at the end of the sequence varying from 0.2% to 1.0%, as denoted in the horizontal axis. For all cases, the error rate at the beginning of the sequence was 0.1%. We again used the average DMC model for the entire sequence for DUDE-Seq. Note that the error rates for the raw data, i.e., the red bars, match the average of the error rates at the beginning and the end of the sequence. From the figure, as was with the real datasets in Section 8.1.2, we clearly see that DUDE-Seq significantly outperforms Coral for all levels of tested error rates.

In order to evaluate the performance of DUDE-Seq for the paired-end reads, we generated datasets in Table 1 with the GemSIM sequencing data simulator (McElroy et al. 2012). As shown in the table, we used 23 public reference sequences (Quince et al. 2011) for generating the dataset A5, and a single reference sequence for S5. We used the error model v5 that has the error rate of 0.28% for forward reads and 0.34% for reverse reads. In Fig. 7(b), in addition to DUDE-Seq, Coral, and Trowel, we included the result of Reptile (Yang et al. 2010), another k-mer spectrum based method that outputs the read-by-read denoising results. We again observe from the figure that DUDE-Seq outperforms Coral, Reptile, and Trowel with significant margins.

In Table 2, we show that the error-corrected reads produced by DUDE-Seq can also improve the downstream pipeline, such as the paired-end merging. We applied four different paired-end merging schemes, CASPER (Kwon et al. 2014), COPE (Liu et al. 2012), FLASH (Magoc and Salzberg 2011), and PANDAseq (Masella et al. 2012), for the two datasets A5 and S5 in Table 1. The used metrics are defined as usual: a true positive (TP) is defined as a merge with correct mismatching resolution in the overlap region, and a false positive (FP) is defined as a merge with incorrect mismatching resolution in the overlap region. Furthermore, a false negative (FN) is a merge that escapes the detection, and a true negative (TN) is defined as correct predictions of the reads that do not truly overlap.

The accuracy and F1 score are computed based on the above metrics (Witten and Frank 2005). For each dataset, we compared the results for three cases: performing paired-end merging without doing any denoising, after correcting errors with Coral, and after correcting errors with DUDE-Seq. Reptile and Trowel were not included in this experiment since Coral generally outperformed them as shown in Fig. 7(b). The accuracy and F1 score results show that correcting errors with DUDE-Seq consistently yields better paired-end merging performance, not only compared to the no denoising case, but also to the case of correcting errors with Coral. This result highlights the potential in applying DUDE-Seq for boosting the performance in downstream analyses of DNA sequences.

### Table 1. Details of the public data (Kwon et al. 2014) used for our experiments on simulated data

| dataset ID | total reads | references | fragment length | read overlap length | simulator (error model) |
|------------|-------------|------------|-----------------|--------------------|-------------------------|
| A5         | 1,000,000   | 23         | 160–190         | 10–40              | GemSIM (v5)             |
| S5         | 1,000,000   | 1          | 160             | 10–40              | GemSIM (v5)             |

### Table 2. Paired-end reads merging performance statistics [parameters: k = 5 for DUDE-Seq; (k, mr, mm, g) = (21, 1, 1, 1000) for Coral]

| tool      | dataset ID | # merges | TP   | FP   | FN   | accuracy | F1   |
|-----------|------------|----------|------|------|------|----------|------|
| CASPER    | A5         | 999,973  | 997,302 | 2,771 | 27.0 | 0.997    | 0.999|
| COPE      | A5         | 924,634  | 915,951 | 8,653 | 75,366 | 0.916    | 0.856|
| FLASH     | w/ Coral   | 999,578  | 977,355 | 22,223 | 422  | 0.977    | 0.989|
| PANDAseq  | A5         | 999,122  | 978,720 | 20,402 | 878  | 0.979    | 0.989|
| CASPER    | S5         | 999,974  | 995,899 | 4,075  | 26   | 0.986    | 0.998|
| COPE      | S5         | 927,757  | 918,733 | 9,024  | 72,243 | 0.919    | 0.958|
| FLASH     | w/ DUDE-Seq| 999,742  | 978,814 | 20,928 | 258  | 0.979    | 0.989|
| PANDAseq  | S5         | 999,351  | 979,899 | 19,452 | 649  | 0.980    | 0.990|

5 Discussion

Our experimental results show that DUDE-Seq can robustly outperform k-mer based, MSA based, and statistical error model based schemes on both real-world datasets, such as 454 pyrosequencing and Illumina, and simulated datasets. This performance advantage in denoising further allowed us to obtain improved results in downstream analysis tasks, such as OTU binning and paired-end merging. Furthermore, the time demand of DUDE-Seq based OTU binning is order(s) of magnitude lower than that of the current state-of-the-art. We also demonstrated the robustness and flexibility of DUDE-Seq by showing that a simple change of Π matrix is enough to apply the exact same DUDE-Seq to data from different sequencing platforms. In particular, we experimentally showed that even when the memoryless channel assumption does not hold, as in Illumina, DUDE-Seq still solidly outperforms the state-of-the-art schemes.

The sliding window nature of DUDE-Seq may seem to resemble the popular k-mer based schemes in the literature. However, while all the existing k-mer based schemes rely on heuristically choosing the thresholds for determining the errors in the reads regardless of the error model of the sequencing platform, DUDE-Seq applies an analytic denoising rule that explicitly takes the error model Π into account. Therefore, even for the exact same noisy reads z*m, DUDE-Seq may result in different denoised sequences depending on the Π’s of different sequencing platforms, whereas the k-mer based scheme will always result in the same denoised sequence. Furthermore, DUDE-Seq can deal with the non-uniform coverage case, for which the k-mer based schemes are known to be ill-suited, by separately dealing with the noisy symbols for each double-sided context ((k, r) ∈ 2^k). The performance gains reported in this paper compared to the state-of-the-art baselines, including the k-mer based schemes, substantiate the competiveness of our method for DNA sequence denoising.

Another advantage of DUDE-Seq is its read-by-read error-correction capability. This feature is important for a number of bioinformatics tasks including de novo sequencing, metagenomics, resequencing, targeted resequencing, and transcriptome sequencing, which typically require extracting subtle information from small variants in each read. In addition to the types of tasks presented in this paper (per-based error correction, OTU binning, and paired-end merging), we plan to apply DUDE-Seq to additional tasks as mentioned above.

Additional venues for further investigation include the procedure for estimating the noise mechanism represented by Π, which is currently empirically determined by aligning each read to the reference sequence and is thus sensitive to read mapping and alignment. For more
robust estimation, we may employ an expectation-maximization-based algorithm, as was recently proposed for estimating substitution emissions for the data from the nanopore technology [Lian et al. 2015]. Considering uncertainties in Σ may also be helpful; hence, it may be useful to investigate the relevance of the framework in [Gemelos et al. 2006]. Additionally, it will likely be fruitful to utilize the information delivered by the Phred quality scores for making decisions about noisy bases and for fine-tuning the objective loss function in our approach. Using a lossy compressed version of the quality scores may be one possible direction to boost the inferential performance of some downstream applications as was shown in [Qu, et al. 2016]. Furthermore, particularly for the homopolymer error correction, there are several hyperparameters whose choices can be experimented with in the future for potentially substantial performance boosts; for example, the choice of alphabet size (in lieu of the current value of 10), the choice of the loss function that may be proportional to the difference between the true and estimated value of N (in lieu of the current Hamming loss), and the choice of quantization (in lieu of [3]). Moreover, we may apply the full generalized DUDE in [Dembo and Weissman 2005] for homopolymer error correction to see if better performance can be achieved at the cost of increased complexity. Finally, we plan to test DUDE-Seq on several other sequencing platforms such as PacBio and Oxford Nanopore, which tend to result in longer and more noisy sequences, to further substantiate the robustness and effectiveness of our algorithm.

References

Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P., and Tyson, G. W. (2012). Grinder: a versatile ampiclon and shotgun sequence simulator. *Nucleic acids research*, 40(12), e94–e94.

Astbury, W. T. (1961). Molecular biology or ultrastructural biology? *Nature*, 190(4789), 547.

Astbury, W. T. (1961). Molecular biology or ultrastructural biology? *Nature*, 190(4789), 547.

Bateson, W. (1894). *Materials for the Study of Variation, Treated with Especial Regard to Discontinuity in the Origin of Species*. Macmillan.

Bragg, L., Stone, G., Imelfort, M., Hugenholtz, P., and Tyson, G. W. (2012). Fast, accurate error-correction of ampiclon pyrosequences using acacia. *Nature Methods*, 9(5), 425–426.

Dembo, A. and Weissman, T. (2005). Universal denoising for the finite-input general-channel. *Information Theory, IEEE Transactions on*, 51(4), 1507–1517.

Ewing, B., Hillier, L., Wendl, M. C., and Green, P. (1998). Base-calling of automated sequencer traces using phred. i. accuracy assessment. * Genome research*, 8(3), 175–185.

Gemelos, G. M., Sigurjonsso, S., and Weissman, T. (2006). Algorithms for discrete denoising under channel uncertainty. *Signal Processing, IEEE Transactions on*, 54(6), 2263–2276.

Greenfield, P., Duensing, K., Papanicolaou, A., and Bauer, D. C. (2014). Blue: correcting sequencing errors using consensus and context. *Bioinformatics*, 30(19), 2723–2732.

Huang, H., Zhang, L., Li, Y., Chen, X., Sun, J., Li, Y., Liu, H., and Li, Y. (2013). Probabilistic error correction for short reads using unsupervised dimensionality reduction. *Bioinformatics*, 29(6), 701–708.

Ilie, L., Farazdel, F., and Iie, S. (2011). Hitec: accurate error correction in high-throughput sequencing data. *Bioinformatics*, 27(3), 295–302.

Kao, W.-C., Chan, A. H., and Song, Y. S. (2011). Echo: a reference-free short-read error correction algorithm. *Genome research*, 21(7), 1181–1192.

Kelley, D. R., Schatz, M. C., Salzberg, S. L., et al. (2010). grinder: a versatile amplicon and shotgun sequence simulator. *Bioinformatics*, 26(17), 2157–2163.

Liu, B., Yuan, J., Yiu, S.-M., Li, Z., Xie, Y., Chen, Y., Shi, Y., Zhang, H., Li, Y., Lam, T.-W., et al. (2012). Cope: an accurate k-mer-based pair-end reads connection tool to facilitate genome assembly. *Bioinformatics*, 28(22), 2870–2874.

Magoz, T. and Salzberg, S. L. (2011). Flash: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, 27(21), 2957–2963.

Medvedev, P., Scott, E., Kakaradov, B., and Pevzner, P. (2011). Error correction of high-throughput sequencing datasets with non-uniform coverage. *Bioinformatics*, 27(13), i37–i41.

McElroy, K. E., Luciani, F., and Thomas, T. (2012). Gemssim: general, error-model based simulator of next-generation sequencing data. *BMC bioinformatics*, 13(1), 31.

Morton, N. and Pevzner, P. (2005). Algorithms for discrete denoising under channel uncertainty. *IEEE Information Theory, IEEE Transactions on*, 51(1), 25–40.

Qu, W., Hashimoto, S.-i., and Morishita, S. (2009). Efficient frequency-based de novo error correction algorithm. *Genome research*, 19(3), 361–394.

Riemenschneider, S. and German, J. (2011). Improved data analysis for the minion nanopore sequencer. *Bioinformatics*, 27(13), 513–519.

Salmela, L. (2010). Correction of sequencing errors in a mixed set of reads. *BMC bioinformatics*, 11, 1284–1290.

Salmela, L. and Schröder, J. (2011). Cope: an accurate k-mer-based pair-end reads connection tool. *Bioinformatics*, 27(11), 1309–1315.

Shendure, J. and Ji, H. (2008). Next-generation dna sequencing. *Nature biotechnology*, 26(10), 1135–1145.

Yang, X., Dorman, K. S., and Aluru, S. (2010). Reptile: representative tiling for deep sequencing data. *IEEE Biotechnology*, 20, 525–552.

Zhao, X., Hara, K., and Wang, D. (2010). Quality-aware error correction for illumina sequencing reads. In *Proceedings of the 8th Annual Biomedical Engineering Society Conference*, pages 21–24.

Zheng, J., Li, Y., and Feng, Z. (2009). Fast and accurate error correction for illumina sequencing reads. *Bioinformatics*, 25(16), 2157–2163.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.

Zhu, J., Li, P., Liu, J., and Xu, H. (2013). A survey of error-correction methods for next-generation sequencing. *Briefings in bioinformatics*, 14(1), 56–66.