博士論文

Simulation for long read sequencers
（ロングリードシーケンサーのシミュレーション）

小野幸輝

ABSTRACT

Recent advances of long read sequencers including Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (Nanopore) sequencers have been accelerating studies on genome, epigenome, transcriptome, and others. It is known that reads generated by long read sequencers include more errors than those generated by short read sequencers (e.g. Illumina HiSeq), and many tools and algorithms that target specifically for the long read sequencers have been developed. In the development of the tools/algorithms for long read sequencers, however, it is generally difficult to evaluate the tools/algorithms by using real data. Therefore, simulators that generate reads with error information such as alignments between reads and the reference sequences are useful for the evaluation of new tools/algorithms. Those simulators are also useful for experimental design such as estimating the depth coverage required for genome assembly and variant detection.

My analysis of 13 PacBio datasets showed characteristic features of PacBio reads. I have developed a read simulator, PBSIM, that captures these features using either a model-based or sampling-based method. Using PBSIM, I conducted several hybrid error correction and genome assembly tests for PacBio reads, suggesting that a continuous long reads coverage depth of at least 15 in combination with a circular consensus sequencing coverage depth of at least 30 achieved extensive genome assembly results.

PacBio sequencers have less systematic (or context-specific) errors than short read sequencers. On the other hand, it has been reported that PacBio reads have regional bias of error distribution within the reads, and very low quality regions are sometimes observed. The low quality regions are caused by chimeras and undetected adapter sequences, and also by non-uniformity of errors. To capture characteristics of errors in reads for long read sequencers more precisely, especially to simulate the non-uniformity of quality scores, I developed a generative model for quality scores, based on a hidden Markov Model in combination with latest model selection criteria. My computational experiments show that PBSIM2, the new version of PBSIM, simulates quality scores that are more consistent with real reads of PacBio and Nanopore than other existing simulators. Also, I improved the correlation between read length and accuracy, and the relationship between error rate and quality scores, both of which PBSIM was unable to simulate properly.

Acknowledgements

I am very grateful to Professor Kiyoshi Asai and Michiaki Hamada for giving me the opportunity and direction of research . This thesis could never be written without their help and support. I would like to extend our gratitude to Professor Martin Frith for his valuable comments on the manuscript. Finally, I thank the member of Asai laboratory for fruitful discussions. Computations in this study were partially performed on the supercomputer systems at the ROIS National Institute of Genetics.

Contents

Chapter 1

General Introduction

Next-generation sequencing (NGS) techniques are the current standard for the generation of genomic data, producing amounts of information rapidly and at a low cost [1]. NGS facilitates the acquisition of large amounts of massive genomic, transcriptome, DNA-protein interactions, and epigenetics data, which is rapidly changing our view of biological process, but the downstream processing of these data is still a serious bottleneck. NGS presents many bioinformatics challenges, and to overcome them, improved computational methods and more efficient software are constantly being developed to provide faster processing and more accurate inferences [2, 3, 4].

Sanger method differs from NGS, especially it works with relatively large fragments which simplifies assembling [5]. Despite it is laborious, and therefore time consuming and expensive, the Sanger method is still respected as the most reliable technique and therefore serves as the 'gold standard'. Compared with the conventional Sanger method, NGS allows massively parallel acquisition of nucleotide sequences, resulting in higher data throughput, shorter sequencing times, and reduced cost. These feature allows second-generation sequencers (also called high-throughput sequencing), such as Illumina, Roche454, SOLiD, and Ion Torrent, to perform unprecedented analysis [6, 7]. However, their read length are incomparably shorter than that of the Sanger method. Their read length range from 75 nt for SOLiD to 400 nt for Ion Torrent, which are shorter than many repetitive sequences, so it is difficult to accurately map and locate them on the genome or transcriptome, and analyses of repetitive sequences and structural variants are limited [8]. The error rate of Roche 454 and Ion Torrent is 1-2%, which is very high compared to the Sanger sequence, but it is reduced to about 0.01% for Illumina and SOLiD. However, platform-specific artefacts are also reported, which makes it difficult to analyze [9].

In general, The tools/algorithms for analyzing data must be designed to properly capture the characteristics of the data and overcome any drawbacks in the data. Since the advent of NGS, many tools/algorithms have been developed for these reads. Most of them are used for each of the downstream processes after sequencing, such as read alignment, genome assembly, transcriptome analysis, and epigenetic analysis. Since the analysis result often changes depending on the tool, it is essential that these tools/algorithms are properly evaluated in order for the user to properly choose the one that suits their research from among them [2, 3, 4]. There are many tools with similar functionality, each with its own strengths and weaknesses. In addition, some tools, especially most of the *de facto* standard tools, are upgraded frequently, and the *de facto* standard tools are often replaced by new tools. Therefore, it is very difficult to choose the best tool. In general, tools/algorithms are evaluated using real and simulation data, and evaluation with real data is more important. However, real data that meets the necessary conditions cannot always be prepared and the true error information of real data is not easy to obtain. Simulators, on the other hand, are very useful because they allow users control data conditions and generate error information, such as alignments between reads and the reference sequences. In addition, In silico data can significantly reduce tools/algorithms development costs and time. The demand for simulation data is increasing not only in the evaluation of tools/algorithms, but also in the validation of biological models, understanding of biological processes, and design of sequence experiments [10].

It is crucial for the simulator to be able to properly simulate the characteristics of real reads, especially the characteristics of errors. The error model determines the probability of substitution, insertion, or deletion at each position in the read. In second-generation sequencers, errors are not uniformly distributed along a read, and errors rates vary widely between reads within the same dataset [11]. Various error statistics are measured for each sequencing pratform, and each error model is created based on the statistics. It has been reported that each platform has specific error biases. For example, Illumina has been observed to frequently replace A>C [9]. Another example is the high frequency of errors in the homopolymer region in Illumina and Ion Torrent (insertion and deletion in Illumina, only deletion in Ion Torrent). In the case of Ion Torrent, this is due to the limited ability of terminator-free chemistry to accurately sequence long homopolymers [12]. Another important bias is known as GC bias, which is the difference between the observed GC content of reads and the expected GC content based on the reference sequence. For example, Illumina and Ion Torrent have a low GC bias, which affects the depth of coverage and is a major cause of gaps in genome assembly. [12, 13]. Simulation of these sequencing biases is expected to be useful in designing sequencing experiments that can avoid the harmful effects of these biases. More complex simulations include simulations of SNPs, structural variants, and heterozygosities in genomic sequences, as well as simulations of each transcriptome expression level, alternative splicing, and intron retention in transcriptome sequences. In assembling these complex simulations, the simulation of the read and the simulation of the genome sequence or transcriptome from which the read is derived will be designed separately.

Third-generation sequencing technologies including the PacBio and Nanopore sequencing are causing a revolution in genomics study as they provide researchers to study genomes at an unprecedented sequencing read length [14]. Third generation sequencing read is also called 'long read'. The characteristics of PacBio reads is quite different from that of the second-generation sequencing reads. PacBio uses Single Molecule Real Time (SMRT) technology, which enables the real-time detection of nucleotide incorporation events during the elongation of the replicated strand from the non-amplified single stranded template . SMRT technology uses nucleotides containing a fluorescent label on the phosphate chain of the nucleotide rather than on the base. Thus, incorporated nucleotides are detected based on the associated fluorophore that is released and dissipated upon cleavage of the phosphate chain [15]. There are two types of reads in PacBio. The first type is Continuous Long Read (CLR), which averages thousands to tens of thousands bases in length and has a maximum length of hundreds of thousands bases. CLR is much longer than the second-generation sequencing reads, but its error rate is very high at 15%. The second type is Circular Consensus Sequencing (CCS, also called HiFi read), which has improved the error rate to 1-2% by error correction with multi-path sequencing. Due to the constraints of the multi-path sequencing, the read length is shorter than the CLR, but recent technological innovations have increased the CCS length to over ten thousand on average and also reduced the error rate to 0.1%. In Nanopore sequencing, a single stranded DNA or RNA fragment pass through a protein embedded in a membrane via a nanometre-sized channel (this protein is the ‘ nanopore ’) [16]. The sequencing process involves a nucleotide fragment passing through the pore leading to variations in a measured ionic current. The current at any given time primarily depends on a nucleotide subsequence of length 5 or 6 inside the pore at that instant. This raw current signal is sampled, and is used by the base-caller to infer the most likely base sequence that could have induced the raw signal. Nanopore reads average tens of thousands bases, with some exceeding 1M at maximum. Its error rate is 15%, which is the same as PacBio. These long reads can be used to more accurately determine the mapping position on the genome by utilizing their length, which is useful for studying complex regions such as repetitive sequences and structural variants [16, 17]. Also, unlike short reads, these are known to have less error bias. According to the announcement by Pacific Biosciences of California, Inc., errors occur almost randomly, and statistical analysis of reads confirms the randomness, although some weak biases are observed [18]. In Nanopore, it has been reported that there are many errors in homopolymers, but apart from that, the errors are highly random [19]. Notably, long reads have less coverage bias such as GC bias observed in short reads, so using long reads is expected to reduce gaps in genome assembly [19].

Long reads have a much higher error rate than short reads, but their weaknesses has been quickly overcome by the development of tools/algorithms. In the genome assembly, hybrid error correction using high–quality short reads achieved a genome assembly with an error rate of 0.1% or less [20]. Immediately after this, the same or better accuracy was achieved by error correction using only long reads [21]. Many tools/algorithms have been developed to analyze long reads, and the speed of innovation in PacBio and Nanopore technologies has further accelerated their development [22, 23, 24].

The most common approach to read simulation is to understand the characteristics of read generation process and to accurately imitate the process. The process of PacBio sequencing is the formation of DNA double strand by polymerase and the accompanying generation of fluorescent signals, which are decoded by the base-caller. In the case of PacBio CCS, the process of multi-path sequencing and consensus sequence generation is added. In the CCS simulation by SimLoRD [25], the number of passes expected from the read length is first calculated, and the error rate of the read is calculated based on the number of passes. In Nanopore, an electrical signal is emitted from the nucleotide sequence that enters the pore, and the base-caller converts it to a base. Nanopore SimulatION [26] and DeepSimulator [27] first simulate an electrical signal and then use a real base-caller, such as Guppy and Albacore [23] to convert that electrical signal to the expected base. In the short read simulation, a simulator such as ART [28] simulates the bias during PCR amplification. Another approach to read simulation is to learn and imitate the characteristics of the read, which is the final product of sequencing. LongISLND [29], which is one of PacBio read simulators, uses a learn-and-simulate approach. In this approach, LongISLND learns the statistical features of the reads from alignments between the reads and the reference sequences, and simulates reads using the model of statistical features. For Nanopore read simulation, simulators such as NanoSim [30] have adopted a similar approach. If it is difficult to understand the characteristics of the read generation process, this method will be the only option. In this thesis, chapter 2 describes the first software PBSIM [31], which uses the former approach (i.e., imitating sequencing process), and chapter 3 describes the second software PBSIM2 [32], which uses the latter approach (i.e., learn-and-simulate approach). PBSIM is a simulator of PacBio reads, and simulations are performed based on the observation that the process of PacBio read generation is almost random. After that, it has been reported that PacBio reads also have various error biases, albeit weakly [12]. Error bias in the homopolymer region and context-specific errors have been reported, but all causes of the error bias have not been clarified [29]. Therefore, in PBSIM2, learning by Factorized Information Criteria Hidden Markov Model (FIC-HMM) [33, 34] was performed and a read generation model was created. PBSIM2 can also simulate Nanopore reads using the same methods used to simulate PacBio reads. PBSIM precisely simulated the randomness that is characteristic of the PacBio sequencer, and used the simulation of genome assembly as an example to show that the simulation is effective in designing sequence experiments. PBSIM2 modeled characteristics of long reads by machine learning and achieved more accurate simulation than that of PBSIM.

Chapter 2

PBSIM: PacBio reads simulator.toward accurate genome assembly

2.1 Introduction

The advent of high-throughput sequencing technologies enables us to determine various genomes rapidly. A number of sequencers have been developed (e.g. Illumina, 454 and SOLiD), and Pacific Biosciences, or ‘ PacBio ’ for short, has provided a unique sequencer, which produces two types of reads: (i) continuous long reads (CLR) (long reads with high error rates), and (ii) circular consensus sequencing (CCS) reads (short reads with low error rates) (see Table 2.1–2.3 for empirical statistics of CLR and CCS reads). These two types of read set could be useful for hybrid *de novo* genome assembly, and, using the PacBio sequencers, Chin and colleagues have determined the genome sequences of two clinical *Vibrio cholerae* strains [35]. There are several simulators for reads produced by high-throughput sequencing technologies, such as pIRS [36], ART [28], Grinder [37], FlowSim [38], MetaSim [39] and dwgsim in SAMtools [40] (see also Table 2.4). However, no read simulator has targeted the specific generation of PacBio libraries so far. I have therefore developed a simulator (called PBSIM) that simulates both CLR and CCS reads of PacBio sequencers. I adopted two simulation approaches: (i) a sampling-based simulation (in which both length and quality scores are sampled from a real read set), and (ii) a model-based simulation. In addition, I conducted hybrid error correction and assembly tests for datasets simulated by PBSIM, suggesting that a CLR coverage depth of at least 15 in combination with a CCS coverage depth of at least 30 achieved extensive assembly results.

Table 2.1. Statistics of real CLR generated by PacBio RS with the C1 chemistry

Type of data	lambda-phage	*E.coli* C227-11	*E.coli* C227-11	*E.coli* 55989	*S.cerevisiae*	Parrot 7.5kb90min	Parrot 7.5kb45min	Parrot 13kb90min
center [a]	NBACC	PacBio	BI	PacBio	CSHL	Assemblathon	Assemblathon	Assemblathon
coverage depth	675	184	28	61	140	-	-	-
#reads	17,318	294,954	63,183	108,954	622,469	278,610	1,073,100	2,134,448
read length								
average	1,889	3,380	2,368	2,901	2,741	2,110	1,327	1,650
SD	947	2,577	1,163	2,226	1,228	1,686	1,074	1,064
min.	200	101	248	101	221	101	101	500
max.	5,860	22,842	8,069	17,404	7,897	14,601	7,618	15,265
read accuracy								
average	77.35%	78.29%	76.91%	78.16%	77.57%	78.03%	81.95%	79.28%
SD	1.62%	2.02%	1.39%	1.93%	1.77%	1.97%	2.63%	2.32%
min.	76.00%	76.00%	76.00%	76.00%	76.00%	76.00%	76.00%	76.00%
max.	84.00%	89.00%	84.00%	88.00%	91.00%	88.00%	91.00%	89.00%

SD: standard deviation.

[a] NBACC (National Biodefense Analysis and Countermeasures Center);
BI (Broad Institute); CSHL (Cold Spring Harbor Laboratory);
Assemblathon (http://assemblathon.org/)

Table 2.2. Statistics of real CLR generated by PacBio RS with the C2 chemistry

Type of data	*E.coli* K12	*V.cholerae* N5	*V.cholerae* H1
center	PacBio	PacBio	PacBio
coverage depth	47	116	–
#reads	31,815	76,129	3,790
read length			
average	2,997	5,690	3,550
SD	2,145	2,738	1,986
min.	101	428	536
max.	13,640	19,557	13,705
read accuracy			
average	83.81%	80.04%	76.19%
SD	3.39%	3.15%	1.01%
min.	76.00%	76.00%	76.00%
max.	93.00%	91.00%	82.00%

Table 2.3. Statistics of CCS generated by PacBio RS with the C1 and C2 chemistry

Type of data	*E.coli* C227-11	*E.coli* K12
center	PacBio	PacBio
coverage depth	41	19
#reads	502,157	91,473
read length		
average	446	963
SD	168	344
min.	116	500
max.	1,864	2,605
read accuracy		
average	98.23%	97.43%
SD	1.94%	2.09%
min.	76.00%	93.00%
max.	100.00%	100.00%

Table 2.4. Existing high throughput sequencing read simulators

Software	Platform	Features	Reference
ART	Illumina, 454, SOLiD	Built-in, technology-specific read error models	Huang et al. (2012)
dwgsim	General	General but no sequencer-specific error model	Li and et al. (2009)
FlowSim	454	Attempt to model the known aspects of the process of 454	Balzer and Malde (2010)
Grinder	Illumina, 454, Sanger	Amplicon, transcriptomic and genomic reads simulator	Angly et al. (2012)
Mason	Illumina, 454, Sanger	Include position specific error rates and base quality values	http://www.seqan.de/projects/mason/
MetaSim	Illumina, 454, Sanger	Simulating reads for Meta genome	Richter et al. (2008)
pIRS	Illumina	Built in empirical base-calling	Hu et al. (2012)
PBSIM	PacBio	Simulate CLR and CSS reads by sampling/model-based simulations	This work
SimSeq	Illumina	An illumina paired-end and mate-pair short read simulator	https://github.com/jstjohn/SimSeq

Software is sorted in alphabetical order. Simulators for only Sanger sequencers are not included in this table.

2.2 Methods

2.2.1 Analyses of real datasets

Models of read length and quality score were derived from features observed in real PacBio reads publicly available. Only PacBio reads filtered by length (>100 bp) and accuracy (>75%) were used in constructing the models because only the filtered PacBio reads were used in *de novo* assemblies [35, 41]. To learn how to simulate differences (errors) introduced to reads, I analyzed real PacBio reads by aligning them to reference sequences. LAST [42, 43] was used for the alignment with parameters: match=1, mismatch=-2, gap existence=-1 and gap extension=-1. The detailed results are shown in Table 2.5 (basic statistics), Figure 2.1 (patterns of substitutions), Figure 2.2, Table 2.6 (patterns of insertion and deletion) and Figure 2.3 (length of insertion and deletion).

Table 2.5. Alignment results for *real* PacBio data

	read type	chemistry	aligned rate (read)	aligned rate (base)	substitution rate	insertion rate	deletion rate	total error rate
λ-phage	CLR	C1	96.06%	86.69%	0.67%	10.00%	1.59%	12.26%
E.coli C227-11 (PacBio)	CLR	C1	98.40%	92.78%	1.40%	8.40%	4.42%	14.22%
E.coli C227-11 (BI)	CLR	C1	90.22%	79.34%	0.89%	10.80%	1.60%	13.29%
E.coli 55989	CLR	C1	97.46%	90.53%	1.48%	8.84%	4.63%	14.95%
S.cerevisiae	CLR	C1	87.02%	70.97%	0.80%	9.26%	1.81%	11.87%
E.coli K12	CLR	C2	95.94%	91.57%	1.48%	10.79%	3.06%	15.33%
V.cholerae N5	CLR	C2	94.01%	67.30%	1.75%	10.24%	3.93%	15.92%
E.coli C227-11	CCS	C1	97.40%	94.98%	0.19%	0.70%	2.34%	3.23%
E.coli K12	CCS	C2	99.99%	99.86%	0.09%	0.86%	1.04%	1.99%

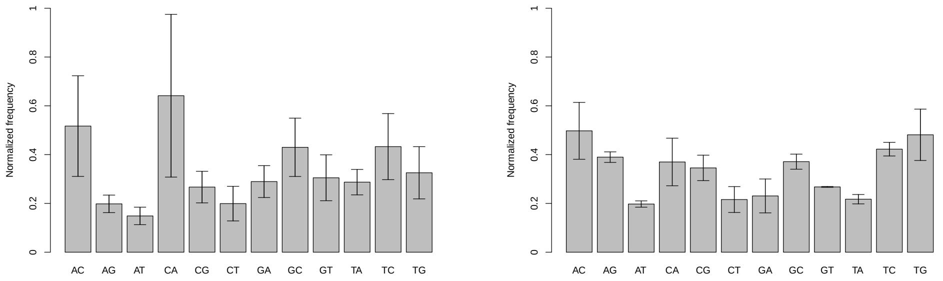
LAST [43] was employed for the alignment with parameters: match=1, mismatch=-2, gap existence=-1, gap extension=-1. LAST was employed for the alignment with parameters: match=1, mismatch=-2, gap existence=-1, gap extension=-1.

Table 2.6. Pattern of INDELs

	read type	chemistry	insertion	deletion
λ-phage	CLR	C1	73.56%	65.35%
E.coli C227-11 (PacBio)	CLR	C1	72.80%	57.92%
E.coli C227-11 (BI)	CLR	C1	73.66%	60.85%
E.coli 55989	CLR	C1	68.77%	57.57%
E.coli K12	CLR	C2	57.72%	57.30%
V.cholerae N5	CLR	C2	55.17%	53.67%
E.coli C227-11	CCS	C1	75.32%	75.99%
E.coli K12	CCS	C2	75.45%	82.82%

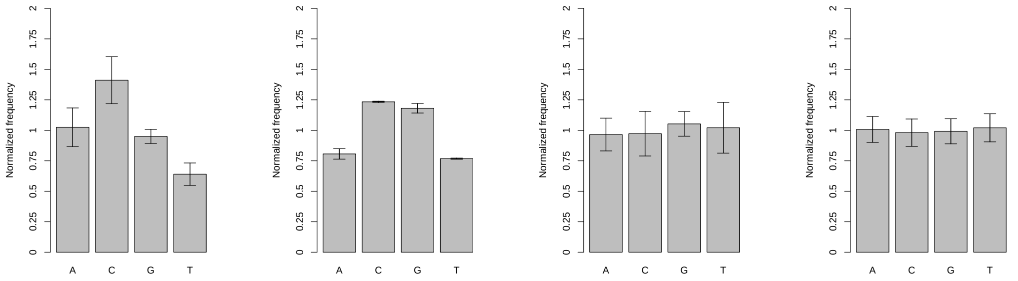
The values show the probability that the nucleotide of an INDEL is the same as either of its neighbors. Notice that the probability is equal to 44% when an INDEL occurs randomly.

Figure 2.1. Pattern of substitutions of *real* PacBio data.


(a) CLR (b) CCS

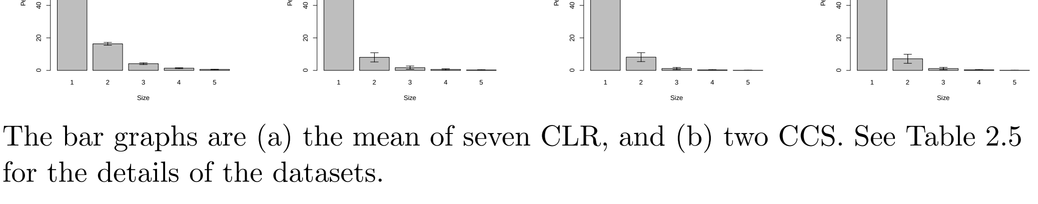
The bar graphs are (a) the mean of seven CLR, and (b) two CCS. See Table 2.5 for the details of the datasets. The frequencies of substitution pattern were normalized by dividing by nucleotide frequencies in the reference sequence. Note that I do not include this pattern in the current version of PBSIM (i.e., substitutions are simulated by using a uniform distribution.).

Figure 2.2. Nucleotide that corresponds to insertion and deletion (INDELs).


(a) CLR Insertion (b) CCS Insertion (c) CLR Deletion (d) CCS Deletion

The bar graphs are (a) the mean of seven CLR, and (b) two CCS. See Table 2.5 for the details of the datasets. The frequencies of INDELs were normalized by dividing by nucleotide frequencies in the reference sequence.

Figure 2.3. Length of INDELs.


(a) CLR Insertion (b) CCS Insertion (c) CLR Deletion (d) CCS Deletion

The bar graphs are (a) the mean of seven CLR, and (b) two CCS. See Table 2.5 for the details of the datasets.

2.2.2 Model-based simulation

According to observed distributions of read length, I used log-normal distributions to model the length of CLR and CCS reads (Figure 2.4–2.6).

For CLR reads, the average accuracy over the length of each read is taken from a normal distribution with parameters (mean, and standard deviation) given by the user. For CCS reads, an exponential function,

$$f(x) = \begin{cases} \exp(0.5(x-75)) & 75 \le x \le 100 \\ 0 & 0 \le x < 75 \end{cases}$$

was utilized for modeling the accuracy of every read (Figures 2.7–2.9).

Errors from single molecule sequencing are considered to be stochastic (random). In fact, no position-specific error profile in CLR and CCS reads was found (cf. Figure 2.10). Quality scores are therefore simulated stochastically, i.e., in the model-based simulation, a quality score at each position of a simulated read is randomly chosen from a frequency table of quality scores. For each accuracy of a read, frequencies of quality scores were precomputed using *E.coli* C227-11/55989 CLR datasets and C227-11 CCS dataset. For accuracies of 0–59% and 86–100% of CLR and 0–84% of CCS, uniform distributions are used because datasets are not sufficiently large. Note that these CLR and CCS datasets were not filtered by the length (>100 bp) and accuracy (>75%).

Simulated read sequences are randomly sampled from a reference sequence, and differences (errors) of the sampled reads are introduced as follows.

The substitutions and insertions are introduced according to the quality scores which are chosen as described above. Their probabilities are computed for each position of a simulated read from the error probability of the position (computed from the quality score of the position) and a ratio of differences (substitution/insertion/deletion) given by a user. The deletion probability is uniform for all positions of each simulated read, which is computed from the mean error probability of the read set and the ratios of differences:

$$P_{del} = \mu_{error} \times \frac{R_{del}}{R_{sub} + R_{ins} + R_{del}}$$

where P_{del} is the deletion probability, μ_{error} is the mean error probability of the read set, R_{sub} is the ratio of substitution, R_{ins} is the ratio of insertion and R_{del} is the ratio of deletion. The substitution and insertion probabilities are computed for each position of a simulated read from the error probability of the position (computed from the quality score of the position) and the ratios of differences:

$$\begin{aligned} P_{error} &= 10^{\frac{-Q}{10}} \\ P_{sub} &= P_{error} \times \frac{R_{sub}}{R_{sub} + R_{ins} + R_{del}} \\ P_{ins} &= P_{error} \times \frac{R_{ins}}{R_{sub} + R_{ins} + R_{del}} \end{aligned}$$

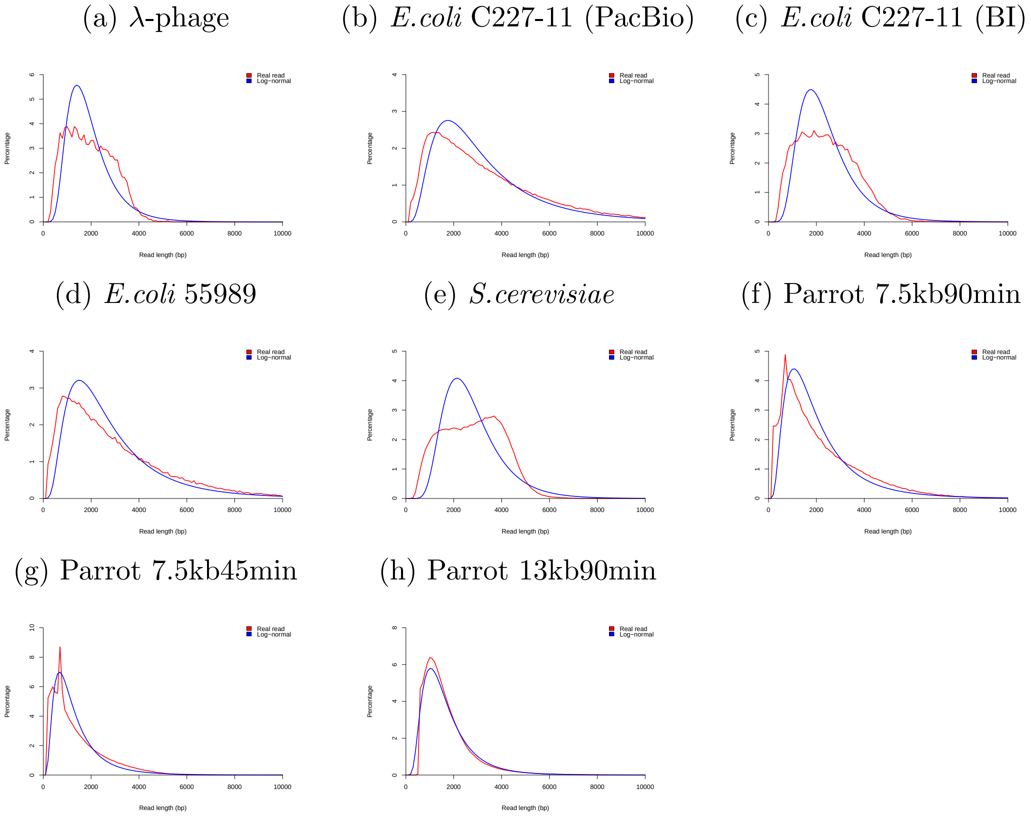
where P_{error} is the error probability of a quality score Q, P_{sub} is the substitution probability and P_{ins} is the insertion probability.

From the observations of the real PacBio reads, I found a weak frequency bias in the substitution pattern (Figure 2.1), but the cause of this bias is not clear; Hence, I do not include this pattern in the current version of PBSIM (i.e., substitutions are simulated by using a uniform distribution.). On the other hand, I found that the probability that inserted nucleotide is the same as either of its neighbors, is significantly higher than that of random choice (Table 2.6), and this bias is considered to be caused by the mechanism known as cognate sampling [18]; Therefore half of inserted nucleotides are chosen to be the same as their following nucleotides and the other half are randomly chosen.

From the observations of the real PacBio reads, I found that the nucleotide frequency of deletion is uniform (Figure 2.2(c),(d)), and that the distribution of deletion length is similar to the geometric distribution (Supplementary Figure 2.3). Therefore the deletion probability is uniform throughout all positions of every simulated read, which is computed from the mean error probability of the read set and the ratio of differences.

It was reported that coverage depth of PacBio reads across a genome and against GC content is nearly uniform [11, 20, 44]. I therefore do not introduce coverage bias and GC bias to simulated sequence reads.

Figure 2.4. Distributions of lengths of CLR generated by PacBio RS with the C1 chemistry.



(a) λ-phage

(b) *E.coli* C227-11 (PacBio)

(c) *E.coli* C227-11 (BI)

(d) *E.coli* 55989

(e) *S.cerevisiae*

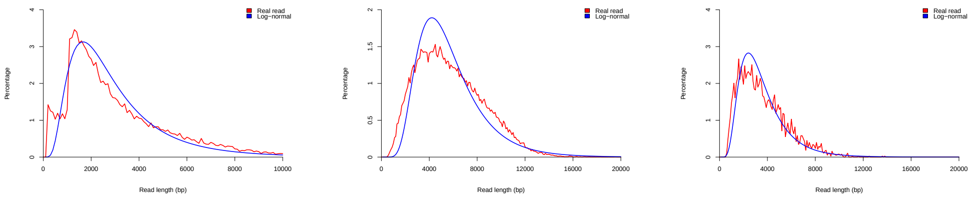
(f) Parrot 7.5kb90min

(g) Parrot 7.5kb45min

(h) Parrot 13kb90min

The red and blue lines indicate the distribution of lengths of *real* reads and a log-normal distribution, respectively. See Table 2.1 for the detailed information of the references (a)–(h).

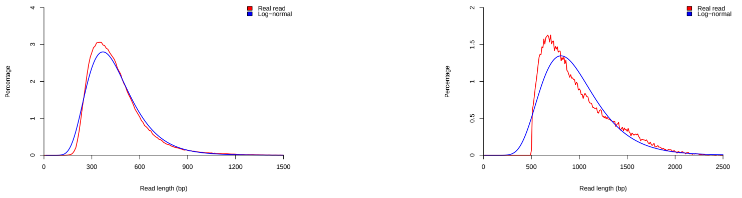
Figure 2.5. Distributions of lengths of CLR generated by PacBio RS with the C2 chemistry.



(a) *E.coli* K12 (b) *V.cholerae* N5 (c) *V.cholerae* H1

The red and blue lines indicate the distribution of lengths of *real* reads and a log-normal distribution, respectively. See Table 2.2 for the detailed information of the references (a)–(c).

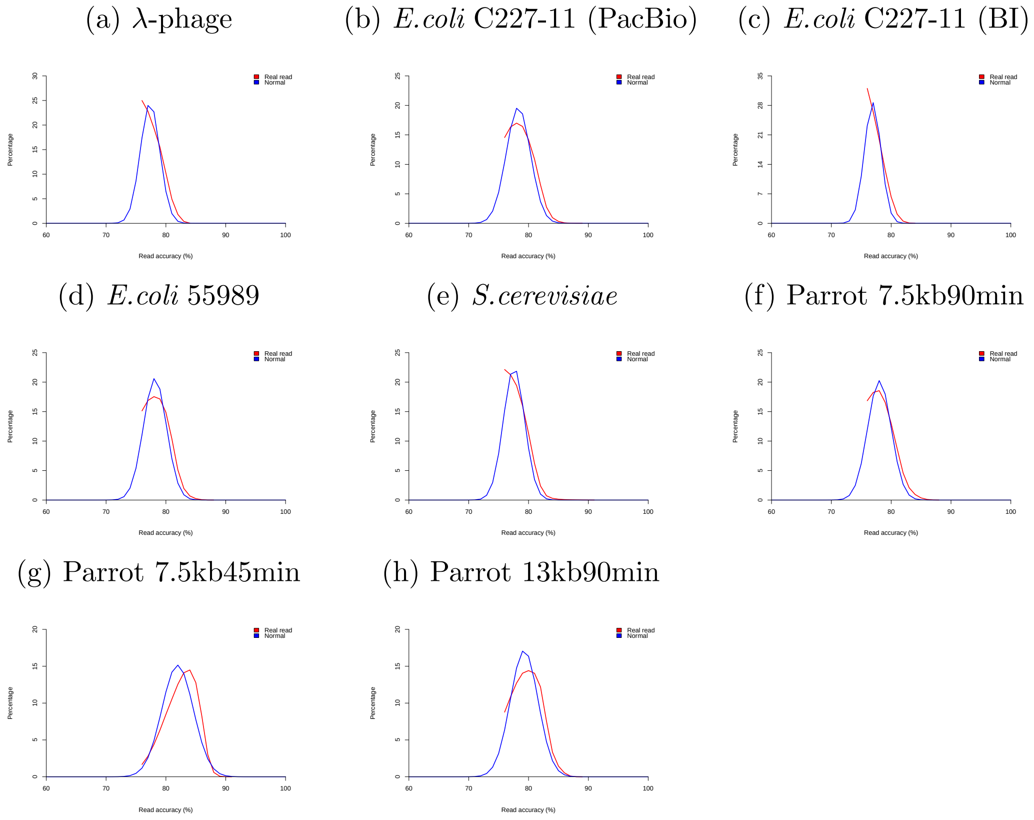
Figure 2.6. Distributions of lengths of CCS generated by PacBio RS with the C1 and C2 chemistry.



(a) *E.coli* C227-11 (C1 chemistry) (b) *E.coli* K12 (C2 chemistry)

The red and blue lines indicate the distribution of lengths of *real* reads and a log-normal distribution, respectively. See Table 2.3 for the detailed information of the references (a) and (b).

Figure 2.7. Distributions of read accuracies of CLR generated by PacBio RS with the C1 chemistry.


(a) λ-phage

(b) *E.coli* C227-11 (PacBio)

(c) *E.coli* C227-11 (BI)

(d) *E.coli* 55989

(e) *S.cerevisiae*

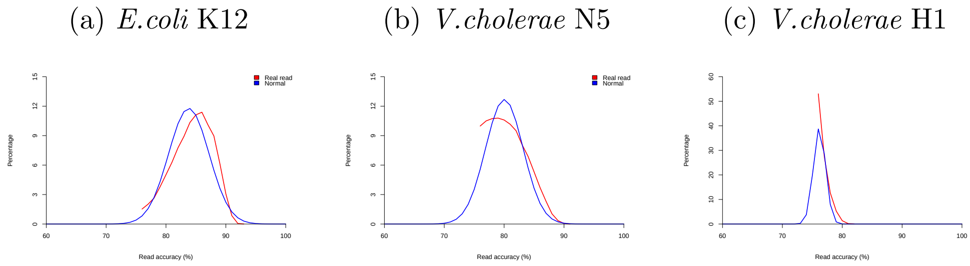
(f) Parrot 7.5kb90min

(g) Parrot 7.5kb45min

(h) Parrot 13kb90min

The red and blue lines indicate the distribution of accuracies of *real* reads and a normal distribution, respectively.

Figure 2.8. Distributions of read accuracies of CLR generated by PacBio RS with the C2 chemistry.

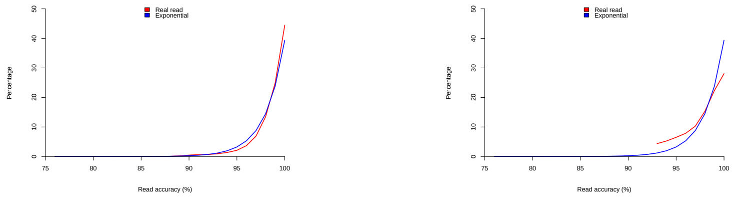

(a) *E.coli* K12

(b) *V.cholerae* N5

(c) *V.cholerae* H1

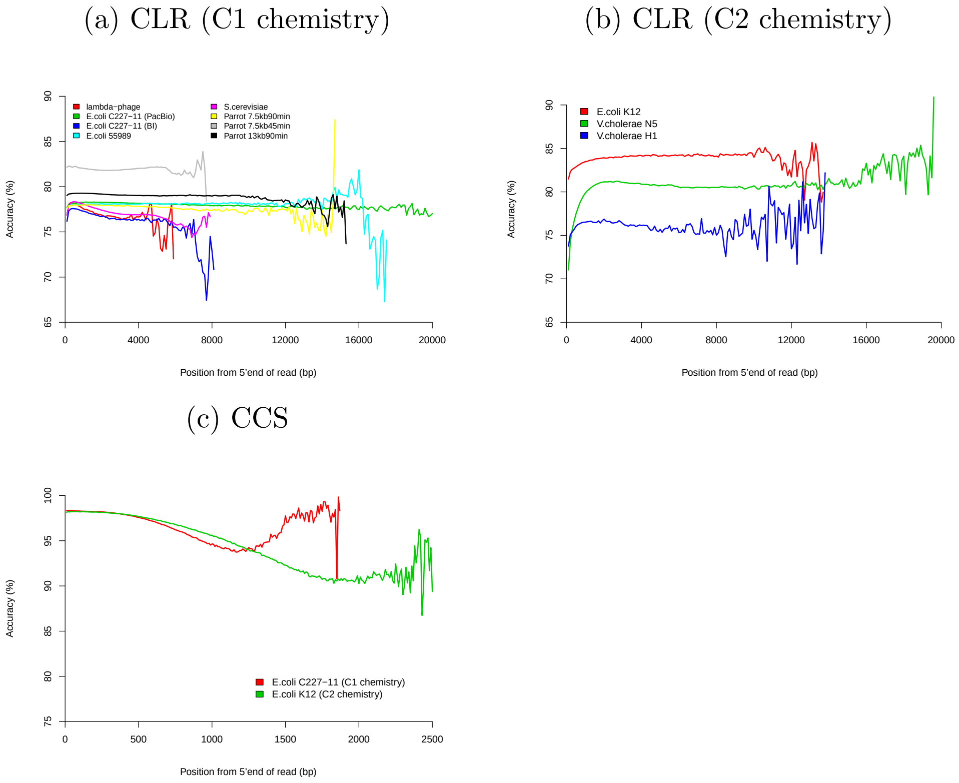
The red and blue lines indicate the distribution of accuracies of *real* reads and a normal distribution, respectively.

Figure 2.9. Distributions of read accuracies of CCS generated by PacBio RS with the C1 and C2 chemistry.


(a) *E.coli* C227-11 (C1 chemistry) (b) *E.coli* K12 (C2 chemistry)

The red and blue lines indicate the distribution of accuracies of *real* reads and a model by exponential function, respectively.

Figure 2.10. Accuracy of each position in CLR and CCS reads.


(a) CLR (C1 chemistry)

(b) CLR (C2 chemistry)

(c) CCS

The origin of the horizontal axis indicates the 5' end of a read. Accuracies are computed by using actual quality scores produced by the PacBio sequencer.

2.2.3 Sampling-based simulation

In the sampling-based simulation, lengths and quality scores of reads are simulated by randomly sampling them in a real library of PacBio reads (provided by the user). Subsequently, their nucleotide sequences are simulated by the same method with the model-based simulation.

2.3 Results and Discussion

PBSIM is implemented using the C language. PBSIM produces a set of simulated reads in the FASTQ format [45] and a list of alignments between a reference sequence and simulated reads in the MAF format (https://cgwb.nci.nih.gov/FAQ/FAQformat.html#format5).

2.3.1 Simulator performance

To test PBSIM's speed, I chose three genomes from Supplementary Table 2.7 as reference sequences, and simulated CLR and CCS reads at 10x, 20x, 50x and 100x coverage to each of the reference sequences. Supplementary Table 2.8 shows the computational time for simulating reads by PBSIM, indicating that PBSIM is sufficiently fast (at most 200 s). On the other hand, the memory requirement of PBSIM depends on the length of the reference sequence.

Because the length and accuracy are selected stochastically, the difference between a set of real reads and a set of simulated reads tends to be larger when the number of simulated reads is smaller. I evaluated this point by using the λ-phage genome (which is the shortest genomes in this study; see Supplementary Table 2.7). In the sampling-based simulation, I used *E.coli* C227-11 real reads as the sample reads. Figures 2.11 and 2.12 show a comparison of real reads and simulated reads. Note that the variance would be much smaller if I used a longer reference sequence. Alignment tests of simulated reads show that simulated reads reproduced CLR and CCS reads well (Table 2.9, compared to Table 2.5).

Table 2.7. Reference sequences used in this study

Reference sequence	Length (bp)	%GC
λ phage genome	48,502	49.85%
E.coli C227-11 genome	5,413,634	50.62%
E.coli 55989 genome	5,154,862	50.66%
E.coli K12 genome	4,639,675	50.79%
V.cholerae N5 genome	3,718,269	47.48%
S.cerevisiae genome	12,157,105	38.15%
D.melanogaster chr2L	23,011,544	41.84%
H.sapiens chr21	48,129,895	40.83%

Table 2.8. Computation time in seconds

			Depth			
Reference	read type	sim. type	10	20	50	100
λ-phage	CLR	sampling-based	19 (9)	21 (9)	16 (8)	20 (8)
genome (48K)	CLR	model-based	1 (1)	1 (1)	1 (1)	1 (1)
	CCS	sampling-based	4(2)	3 (2)	3 (2)	4 (2)
	CCS	model-based	1 (1)	1 (1)	1 (1)	1 (1)
E. coli 55989	CLR	sampling-based	20 (12)	21 (15)	31 (25)	49 (41)
genome (5.1M)	CLR	model-based	5 (4)	10 (9)	25 (23)	50 (47)
	CCS	sampling-based	10 (5)	10 (8)	20 (17)	36 (31)
	CCS	model-based	5 (4)	10 (8)	23 (21)	47 (43)
D. melanogaster	CLR	sampling-based	28 (23)	44 (38)	92 (82)	176 (159)
chr2L (23M)	CLR	model-based	22 (21)	45 (42)	112 (106)	219 (208)
	CCS	sampling-based	19 (16)	37 (29)	78 (70)	151 (138)
	CCS	model-based	22 (20)	44 (40)	109 (100)	216 (200)

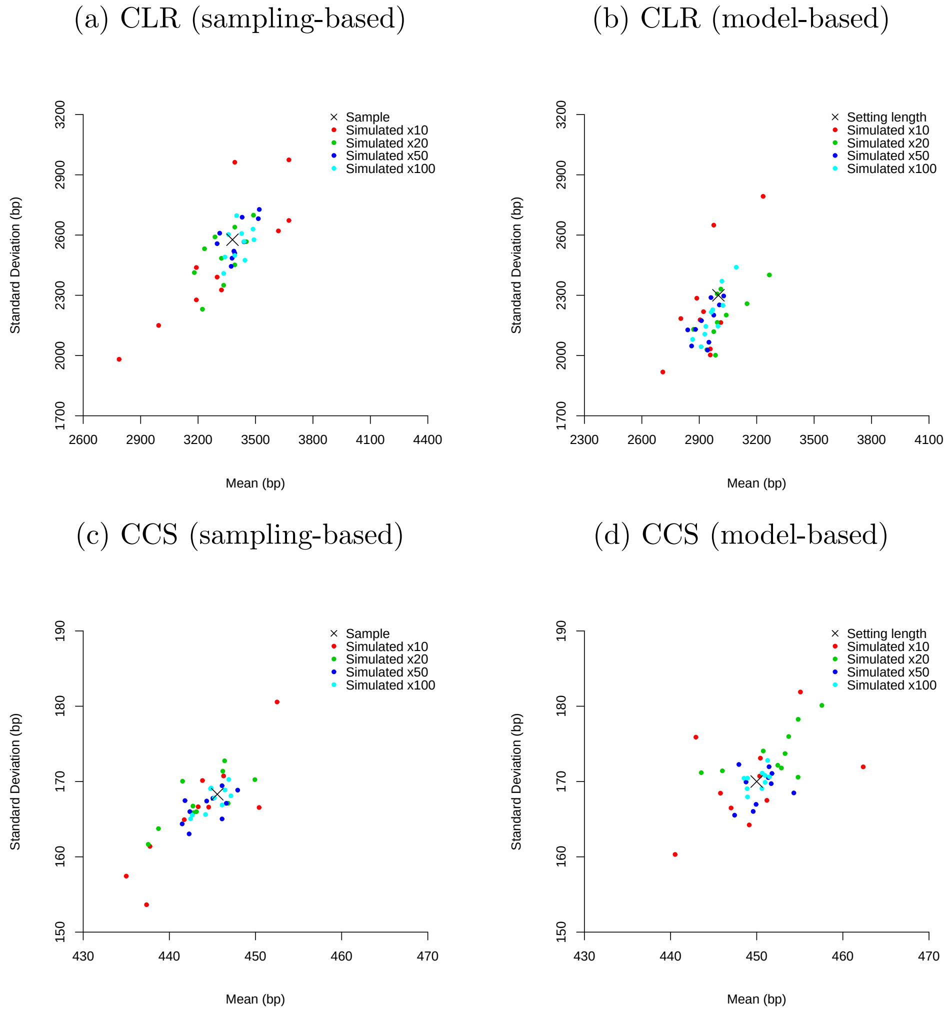
Averaged real time in seconds for 10 simulations is shown. A Linux machine with 2.67 GHz Intel Xeon CPU was used. Times in parentheses indicate CPU times. Memory requirement is the length of the reference genome plus several megabytes.

Table 2.9. Alignment results for simulated data

	read type	sim. type	accuracy	aligned rate(read)	aligned rate(base)	substitution rate	insertion rate	deletion rate	total error rate
E.coli	CLR	sampling	78.30%	99.30%	99.79%	3.23%	10.53%	3.98%	17.74%
55989	CLR	model	77.98%	99.70%	99.81%	3.31%	10.58%	4.00%	17.89%
	CCS	sampling	98.23%	100.00%	99.97%	0.13%	0.39%	1.25%	1.77%
	CCS	model	98.40%	100.00%	99.97%	0.11%	0.32%	1.42%	1.85%

LAST [42] was employed for the alignment with parameters: match=1, mismatch=-2, gap existence=-1, gap extension=-1.

Figure 2.11. Comparison of the simulated lengths and sample (or setting) lengths.



(a) CLR (sampling-based)

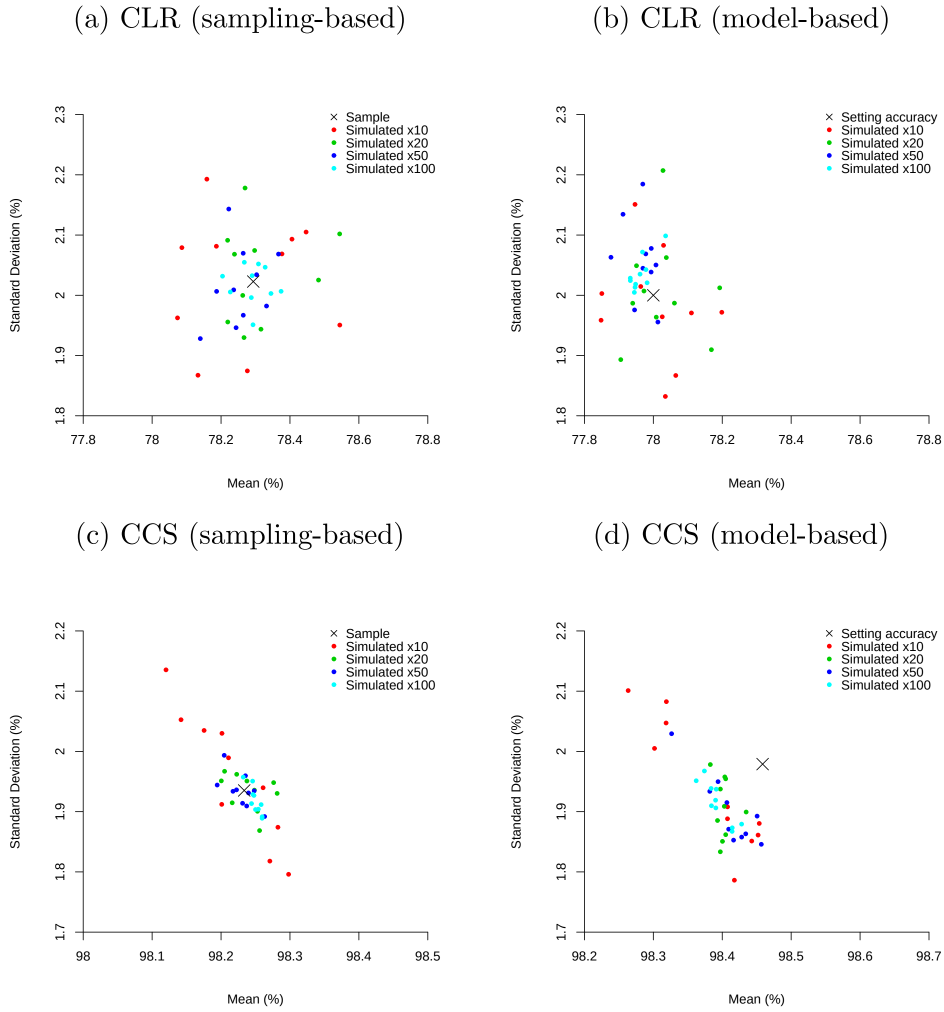
(b) CLR (model-based)

(c) CCS (sampling-based)

(d) CCS (model-based)

I ran PBSIM 10 times for each depth (10, 20, 50 and 100) for sampling- and model-based simulations. The colors indicate the simulated lengths. The cross indicates the sample (or setting) lengths.

Figure 2.12. Comparison of the simulated accuracies and sample (or setting) accuracies.



(a) CLR (sampling-based)

(b) CLR (model-based)

(c) CCS (sampling-based)

(d) CCS (model-based)

I ran PBSIM 10 times for each depth (10, 20, 50 and 100) for sampling- and model-based simulations. The colors indicate the simulated accuracies. The cross indicates the sample (or setting) accuracies.

2.3.2 Assembly test for simulated reads

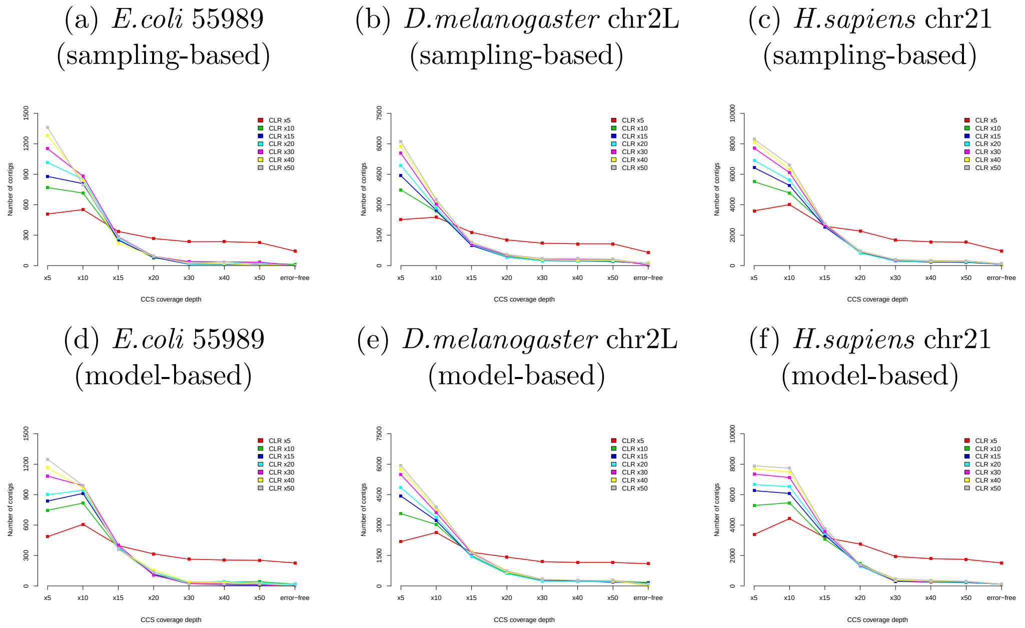
Finally, I conducted hybrid error correction and assembly tests using datasets simulated by PBSIM. I simulated CLR and CCS reads with coverage depth of 5, 10, 15, 20, 30, 40 and 50 (by both model-based and sampling-based simulations), and tested all the combinations of these coverage depth. In the model-based simulation, for CLR reads, the length and accuracy are set to be about 3000 bp and 78%, respectively; For CCS reads, the length and accuracy are set to be about 450bp and 98%, respectively. In the sampling-based simulation, I used *E.coli* C227-11 real reads (from which reads are sampled). Reference sequences tested were *E.coli* 55989, *D.melanogaster* chr2L and *H.sapiens* chr21 (cf. Table 2.7).

For a hybrid assembly of CLR and CCS reads, I employed the PacBioToCA [20], a hybrid error correction method and *de novo* assembly of single-molecule sequencing reads. In the pipeline, error correction of CLR reads was first conducted using CCS reads, and then the corrected (CLR) reads were assembled with the Celera assembler [46]. CLR reads *without* error correction can not be assembled by the Celera assembler due to the high error rate.

The results are shown in Figures 2.13 (the number of contigs), 2.14 (aligned reference bases by PBcR), 2.15 (aligned reference bases by contigs), 2.16 (N50 of contigs) and 2.17 (maximum length of contigs). For every reference sequence, an extensive assembly was obtained with a CLR coverage depth of at least 15 in combination with a CCS coverage depth of at least 30 (Figures 2.16 and 2.17). Additionally I simulated and assembled error-free CLR reads for all the CLR coverage depth tested above. Although the error correction of PacBioToCA improved assembly metrics, assembly of error-free reads was more comprehensive still. Also, higher read coverage did not always translate into larger assembly. These results suggest that there is room for progress in the correction of PacBio errors and read assembly. (see the "error-free" parts in Figures 2.13–2.17).

In this section I have shown that users can employ PBSIM to design sequencing experiments (e.g., to determine the depths of CLR and CCS reads). Note that users can design sequencing experiments of hybrid-assembly of PacBio CLR (simulated by PBSIM) combined with Illumina's short reads (simulated by existing Illumina simulators e.g. pIRS [36]). PBSIM will be also useful for comparisons of hybrid assembly algorithms.

Figure 2.13. Number of contigs in the assembly tests.



(a) *E.coli* 55989 (sampling-based)

(b) *D.melanogaster* chr2L (sampling-based)

(c) *H.sapiens* chr21 (sampling-based)

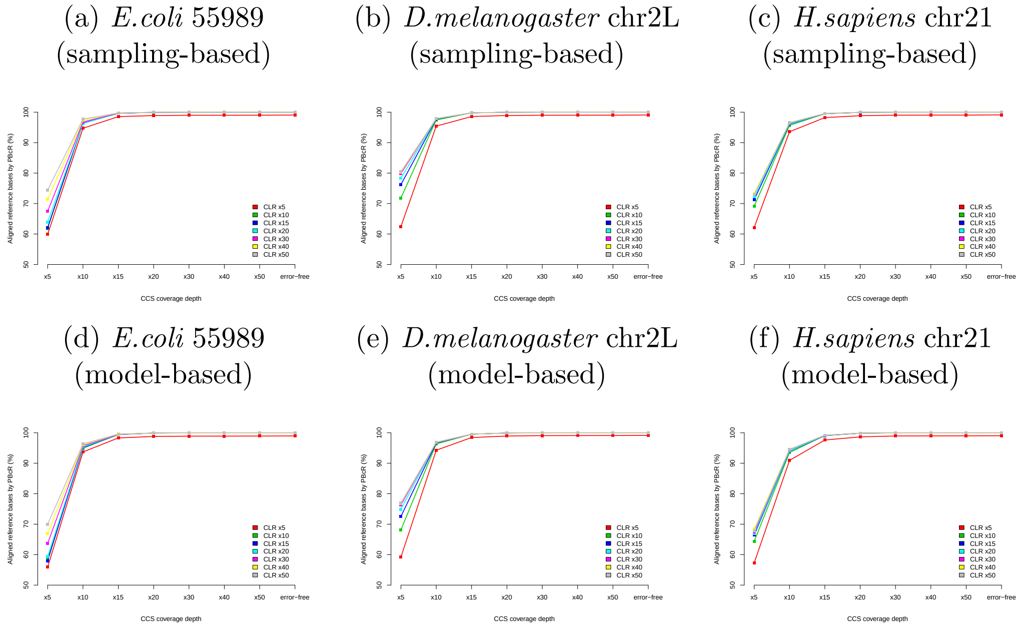
(d) *E.coli* 55989 (model-based)

(e) *D.melanogaster* chr2L (model-based)

(f) *H.sapiens* chr21 (model-based)

In each figure, the horizontal axis (with the exception of the label "error free") indicates circular consensus sequencing (CCS) coverage depth and the vertical axis shows the number of contigs. Both continuous long reads (CLR) and CCS reads were simulated by using a sampling-based simulation in PBSIM for three reference sequences: (a) *E.coli* 55989, (b) *D.melanogaster* chr2L and (c) *H.sapiens* chr21 (cf. Table 2.7). The "error-free" in the horizontal axis shows the case of using only CLR with *no* error (for assembly), where the color indicates the coverage depth of CLR.

Figure 2.14. Aligned reference bases by PBcR in the assembly tests.



(a) *E.coli* 55989 (sampling-based)

(b) *D.melanogaster* chr2L (sampling-based)

(c) *H.sapiens* chr21 (sampling-based)

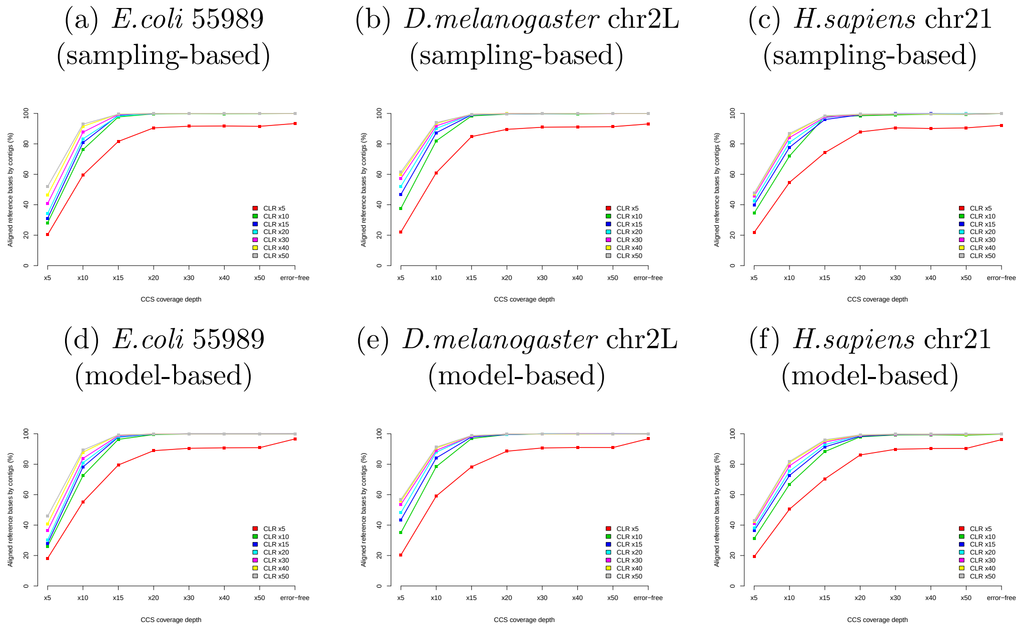
(d) *E.coli* 55989 (model-based)

(e) *D.melanogaster* chr2L (model-based)

(f) *H.sapiens* chr21 (model-based)

Aligned reference bases indicates the percentage of the reference covered by PBcR. MUMmer (http://mummer.sourceforge.net/) was employed to compute this value. PBcR means PacBio corrected reads (i.e. reads after error correction by PacBioToCA). "error-free" at the CCS coverage depth is assembled results of error-free CLR reads.

Figure 2.15. Aligned reference bases by contigs in the assembly tests.



(a) *E.coli* 55989 (sampling-based)

(b) *D.melanogaster* chr2L (sampling-based)

(c) *H.sapiens* chr21 (sampling-based)

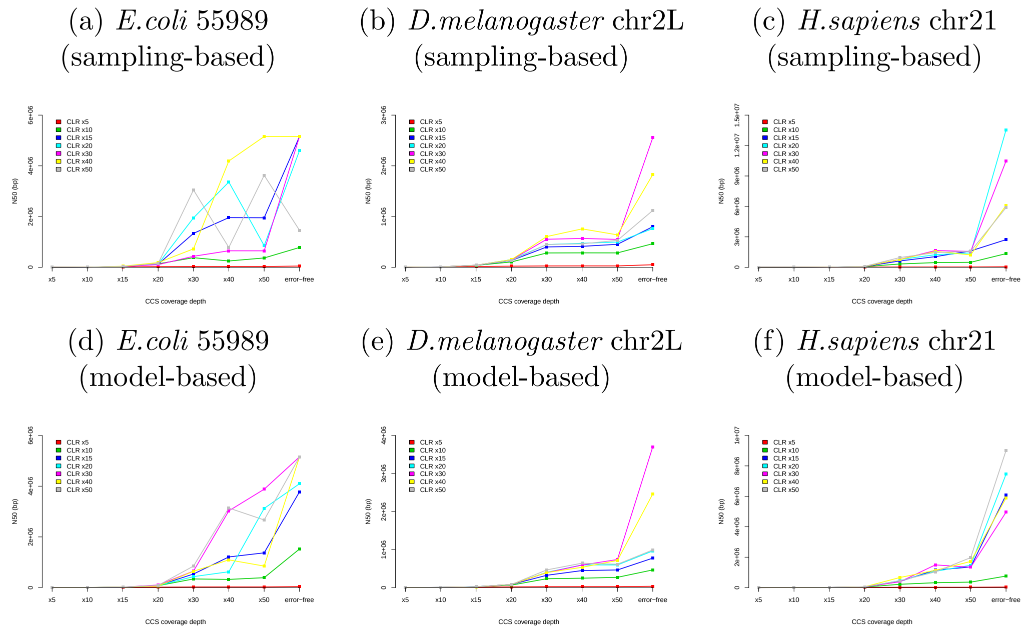
(d) *E.coli* 55989 (model-based)

(e) *D.melanogaster* chr2L (model-based)

(f) *H.sapiens* chr21 (model-based)

Aligned reference bases indicates the percentage of the reference covered by contigs. MUMmer (http://mummer.sourceforge.net/) was employed to compute this value. "error-free" at the CCS coverage depth is assembled results of error-free CLR reads.

Figure 2.16. N50 in the assembly tests.



(a) *E.coli* 55989 (sampling-based)

(b) *D.melanogaster* chr2L (sampling-based)

(c) *H.sapiens* chr21 (sampling-based)

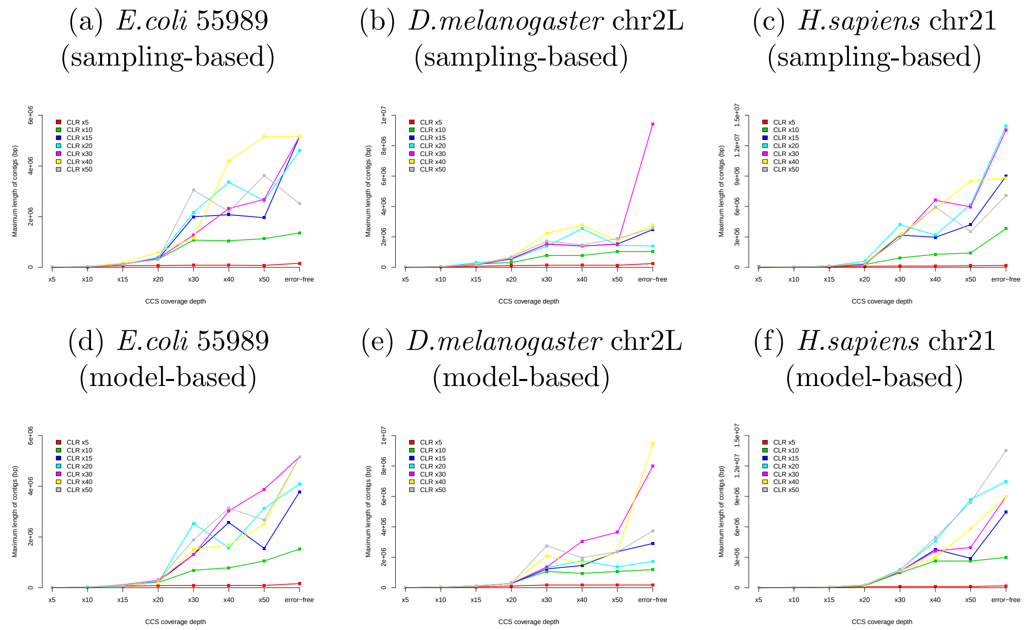
(d) *E.coli* 55989 (model-based)

(e) *D.melanogaster* chr2L (model-based)

(f) *H.sapiens* chr21 (model-based)

N50 is the contig length such that using equal or longer contigs produces half the bases of the genome. In each figure, the horizontal axis (with the exception of the label “error free”) indicates circular consensus sequencing (CCS) coverage depth and the vertical axis shows N50. Both continuous long reads (CLR) and CCS reads were simulated by using a sampling-based simulation in PBSIM for three reference sequences: (a) *E.coli* 55989, (b) *D.melanogaster* chr2L and (c) *H.sapiens* chr21 (cf. Table 2.7). The “error-free” in the horizontal axis shows the case of using only CLR with *no* error (for assembly), where the color indicates the coverage depth of CLR.

Figure 2.17. Maximum length of contigs in the assembly tests.



(a) *E.coli* 55989 (sampling-based)

(b) *D.melanogaster* chr2L (sampling-based)

(c) *H.sapiens* chr21 (sampling-based)

(d) *E.coli* 55989 (model-based)

(e) *D.melanogaster* chr2L (model-based)

(f) *H.sapiens* chr21 (model-based)

”error-free” at the CCS coverage depth is assembled results of error-free CLR reads.

Chapter 3

PBSIM2: a simulator for long read sequencers with a novel generative model of quality scores

3.1 Introduction

High-throughput DNA sequencing technology has markedly changed the style of biological research, from hypothesis-driven biology to data-driven biology. Notably, recent advances in *long* read sequencers, including Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (Nanopore), have accelerated studies on the genome [19, 47, 48, 49, 50], epigenome [51], and transcriptome [52], among others [14, 53].

It is known that reads generated by long read sequencers include more errors than those generated by short read sequencers (e.g., Illumina HiSeq), and many tools and algorithms that specifically target long read sequencers have been developed [22, 23, 24]. However, in the development of tools/algorithms for long read sequencers, it is generally difficult to evaluate those using real data. This is because real data that meets the necessary conditions cannot always be prepared; in addition, the true error information of real data is not easy to obtain. Therefore, simulators that generate reads with error information, such as alignments between reads and the reference sequences, are useful for the evaluation of new tools/algorithms. (See [10, 54] for comprehensive reviews of read simulators.) Moreover, these simulators are useful for experimental design such as estimating the depth coverage required for genome assembly and variant detection. To make this possible, it is crucial to be able to properly simulate the characteristics of real reads, especially the characteristics of errors.

PacBio sequencers have lesser systematic (or context-specific) errors (e.g., errors in high- and low-GC regions and at homopolymer runs) than that of short read sequencers, such as Illumina [12, 18, 55]. In contrast, it has been reported that PacBio reads have regional bias of error distribution within the reads, and very low quality regions are sometimes observed (For example, see Myers' report, https://dazzlerblog.wordpress.com/2015/11/06/). Low quality regions are caused by chimeras and undetected adapter sequences, as well as non-uniformity of errors. Figure 3.1 clearly shows the non-uniformity of quality scores, with the distributions of accuracy of 800 bp disjoint intervals in reads. Here, quality scores are used instead of actual errors, because it is difficult to obtain the true error information for reads, especially long reads. Note that the quality score is logarithmically related to error probability [45]. 'Random models' randomly generate quality scores according to real frequencies of quality scores, leading to a normal distribution of quality scores. Compared with random models, the distributions of real reads have broader accuracy ranges of 800 bp interval, especially for low read accuracy. My previously developed simulator, PBSIM [31], employs a random model [18], and the reads generated by it are simpler and easier to handle than real reads; this is a problem when evaluating the tools/algorithms for long read sequencers.

Currently, there are several simulators that generate long reads (see Supplementary Table 3.1 for summary). With regard to simulation of low quality regions, NanoSim [30] generates a set of read profiles from alignment-based analysis, and simulates low quality regions using the profiles. PaSS [56] adopts preset high error rates for both ends of the reads, to simulate low quality regions. Badread [57] can introduce chimeras, adapter sequences, low quality regions, and low-complex repetitive sequences into simulated reads. However, there is still room for improvement in the simulation of the non-uniformity of errors (or quality scores).

To simulate the non-uniformity of quality scores, in this study, I developed a generative model for quality scores, based on a hidden Markov Model in combination with latest model selection criteria. My computational experiments show that PBSIM2, the new version of PBSIM, simulates reads that have a tendency similar to real reads.

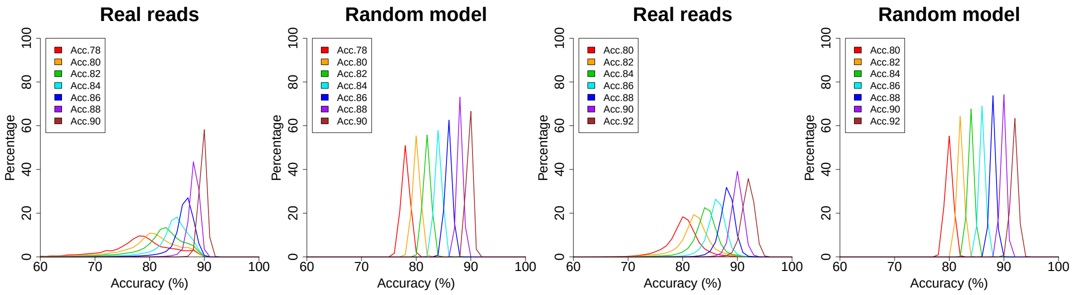
This article is organized as follows: In Section 3.2, after introducing a novel generative model for quality scores, I describe the detailed design of PBSIM2. In Section 3.3 I report comprehensive evaluations of PBSIM2 and related discussions. PBSIM2 newly added the function to simulate Nanopore reads, whereas it removed the function to simulate circular consensus sequencing (CCS also known as HiFi) reads. This is because the average accuracy of CCS exceeds 99%, which is outside the purpose of PBSIM to simulate error-prone reads. PBSIM2 is freely available from `https://github.com/yukiteruono/pbsim2`, and it will be useful for various studies using long reads.

Table 3.1. Simulators for long reads

Simulator	Long read	Error model
PBSIM [31]	PacBio	nucleotide sequence-independent error model
DAZZ_DB/simulator [a]	PacBio	nucleotide sequence-independent error model
ReadSim [58]	PacBio, Nanopore	nucleotide sequence-independent error model
SimLoRD [25]	PacBio	nucleotide sequence-independent error model
SiLiCO [59]	PacBio, Nanopore	nucleotide sequence-independent error model
LongISLND [29]	PacBio, Nanopore	entended-kmer based error model
NanoSim [30]	Nanopore	alignment-based trained model, which does not use k-mer error bias
SNaReSim [60]	Nanopore	k-mer based error model
NPBSS [61]	PacBio	nucleotide sequence-independent error model
DeepSimulator [27]	Nanopore	pore model generates raw signal, and basecaller converts raw signal into fastq
DeepSimulator1.5 [62]	Nanopore	pore model generates raw signal, and basecaller converts raw signal into fastq
Naopore SimulatION [26]	Nanopore	pore model generates raw signal, and basecaller converts raw signal into fastq
PaSS [56]	PacBio	k-mer based error model
Badread [57]	PacBio, Nanopore	k-mer based error model

[a] https://github.com/thegenemyers/DAZZ_DB/blob/master/simulator.c

Figure 3.1. Non-uniformity of quality scores for real and simulated reads.



(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R.sphaeroides*

After grouping reads by their accuracy, reads were segmented into 800 bp disjoint intervals, and accuracy of each interval was computed from quality scores. Each graph shows the distribution of averaged accuracy of 800 bp intervals, where colors of plotted lines represent read groups (e.g., 'Acc.78' refers to a read group with an accuracy of 77.5–78.4%). In random models, a house-made program randomly sampled quality scores according to the quality score distribution of each accuracy of real reads.

3.2 Methods

3.2.1 Datasets for long read sequencers

In this study, I utilized various types of datasets for PacBio (7 datasets of CLR) and Nanopore sequencers (9 datasets), as summarized in Tables 3.2 and 3.3, respectively.

Table 3.2. Datasets for PacBio sequencers

		ReadlLength		Read accuracy		
Reference	Chemistry	mean	SD	mean	SD	URL
E-coli K12 MG1655	P4–C2	5,254	3,677	81%	5%	`https://github.com/PacificBiosciences/DevNet/wiki/E.-coli-20kb-Size-Selected-Library-with-P4-C2`
S.cerevisiae	P4–C2	5,856	4,422	82%	5%	`https://github.com/PacificBiosciences/DevNet/wiki/Saccharomyces-cerevisiae-W303-Assembly-Contigs`
N.creassa	P4–C3	5,581	3,838	81%	4%	`https://github.com/PacificBiosciences/DevNet/wiki/Neurospora-Crassa-\%28Fungus\%29-Genome\%2C-Epigenome\%2C-and-Transcriptome`
H.sapiens	P5–C3	6,383	5,562	83%	3%	`https://github.com/PacificBiosciences/DevNet/wiki/H\_sapiens\_54x\_release` [a]
D.melanogaster	P5–C3	10,095	7,224	84%	2%	`https://github.com/PacificBiosciences/DevNet/wiki/Drosophila-sequence-and-assembly` [b]
E.coli K12 MG1655	P6–C4	8,582	6,953	85%	4%	`https://github.com/PacificBiosciences/DevNet/wiki/E.-coli-Bacterial-Assembly`
C.elegans	P6–C4	11,560	7,667	85%	3%	`https://github.com/PacificBiosciences/DevNet/wiki/C.-elegans-data-set` [c]

SD: standard deviation. Read accuracy was computed from quality scores.

[a] Only m130929_024849 and m130929_161837 were used.

[b] Only Dro1_24NOV2013_398 was used.

[c] Only m140928_184123, m140928_230547, m140928_033247 and m140928_075857 were used.

Table 3.3. Datasets for Oxford Nanopore sequencers

			Read length		Read accuracy		
Reference	Chemistry	Base-caller	mean	SD	mean	SD	URL
H.sapiens	R9.4	Guppy 3.6.0	16,975	38,159	80%	15%	`https://github.com/nanopore-wgs-consortium/CHM13` [a]
C.elegans	R9.4	poretools	3,962	5,250	83%	6%	`https://www.ncbi.nlm.nih.gov/sra/SRX2764157`
E.coli O127:H6	R9.4	Guppy 3.1.5	9,231	14,187	90%	3%	`https://www.ncbi.nlm.nih.gov/sra/SRX8094377`
S.cerevisiae	R9.4	poretools	12,473	14,182	86%	10%	`https://www.ncbi.nlm.nih.gov/sra/SRX2849976`
D.melanogaster	R9.5	Albacore 2.1.0	6,699	6,703	92%	6%	`https://www.ncbi.nlm.nih.gov/sra/SRX3676783`, Run=SRR6821890
P.koreensis	R9.5	Albacore 2.1.3	25,740	15,090	84%	8%	`https://www.ncbi.nlm.nih.gov/sra/SRX3923115`
R.sphaeroides	R9.5	Guppy 3.0.3	5,650	7,247	83%	8%	`https://www.ncbi.nlm.nih.gov/sra/SRX7341766`
C.armoricus	R9.5	Albacore 2.3.4	9,966	9,004	90%	3%	`https://www.ncbi.nlm.nih.gov/sra/SRX6887881`
E.coli K12 MG1655	R10.3	Guppy 3.4.5	6,397	4,708	85%	3%	`https://www.ncbi.nlm.nih.gov/sra/ERX3900444`

SD: standard deviation. Read accuracy was computed from quality scores.

[a] 100,000 reads with a length of 100 bp or more were sampled from the rel5 dataset.

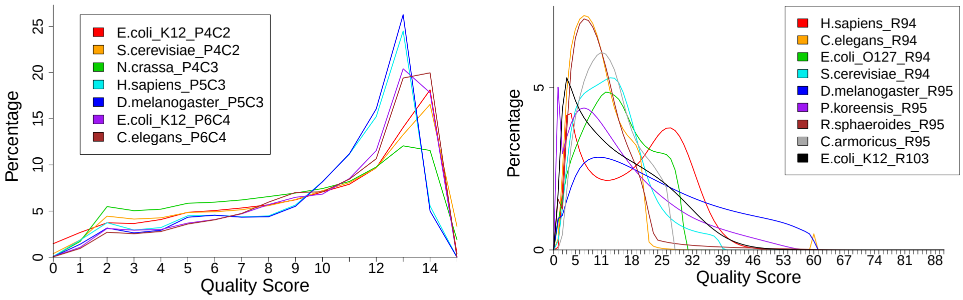
3.2.2 Basic statistics of long reads

To learn the features of long reads, I obtained basic statistics, such as read length, accuracy distribution, and quality score distribution, from the real reads in Table 3.2 and 3.3. As shown in Figure 3.2, in PacBio, quality score distributions are very similar within the same chemistry. Conversely, Nanopore has a wider range and more diverse distribution of quality scores than those of PacBio.

Additionally, I conducted local alignments of real and simulated reads to reference sequences, and got error rates from the alignment results for several analyses. These local alignments were executed by LAST version 1047 [43]. Alignments were filtered using `last-map-probs`. `lastal` was executed with parameters trained by `last-train` [63] and '`-m100 -j7`'. `lastdb`, `last-train`, and `last-map-probs` were executed using the default parameters.

Figure 3.2. Quality score distributions of real reads.

(a) PacBio (b) Nanopore



The frequency of quality scores was counted for each of the datasets in Tables 3.2 and 3.3. Colors of plotted lines represent datasets. Dataset name is species (e.g., *E. coli*_K12) + chemistry (e.g., P4C2). The horizontal axis is PHRED33 quality score defined in terms of the estimated error probability (e.g., quality scores 4, 7, and 10 represent error probabilities of 40, 20, and 10%, respectively) [45].

3.2.3 Generative model for quality scores

To construct a generative model for quality scores, I employed a hidden Markov Model (HMM), which generates observed data from hidden states that follow the Markov model. Note that HMMs are utilized in many bioinformatics tools (e.g., [64]). In my HMM, the emission probability distributions from each hidden state are provided by a categorical distribution, whose output is one of the quality scores. It should be emphasized that the parameters in categorical distribution with hidden states are different from each other.

In conventional HMM, the number of hidden states should be provided beforehand. In this study, I utilized HMM with the latest model selection criteria, called factorized information criteria (FIC-HMM) [34]. This method is theoretically sound, enabling us to train not only parameters in HMM but also the number of hidden states [33].

In this study, I adopted the model whose (lower bound of) FIC is maximum among five trials with different initial parameters, because FIC-HMM affects local optimal solutions in their training. The models were trained for each read accuracy of each chemistry (e.g., for 80% accuracy, training data comprise a read group with an accuracy of 79.5–80.4%). For read accuracy with insufficient training data, constant quality scores that match the accuracy were used.

3.2.4 Detailed design of PBSIM2

Given a reference sequence, PBSIM2 generates FASTQ file [45], including reads with quality scores, where the generative process is summarized as follows:

1. determine read length according to the read length distribution .
2. determine read accuracy according to the read accuracy distribution.
3. generate quality scores of each position in the read using the generative model, which was trained for each read accuracy of each chemistry.
4. sample a random position from the reference sequence and cut out a nucleotide sequence of the read length.
5. introduce errors (substitution, insertion, and deletion) into the nucleotide sequence according to a quality score at each position of the read and the ratio of error types.

On both PacBio and Nanopore sequencers, I utilized gamma distribution for read length, although log-normal distribution was employed in the previous version of PBSIM. This is because gamma distribution is more suitable than log-normal distribution for latest real datasets of both PacBio and Nanopore in my preliminary experiments (Figure 3.3). Note that DAZZ_DB/simulator (https://github.com/ thegenemyers/DAZZ_DB/blob/master/ simulator.c), SimLoRD [25], and NPBSS [61] employ log-normal distribution for PacBio; SiLiCO [59] employs log-normal distribution for PacBio, as well as gamma distribution for Nanopore; DeepSimulator1.5 [62] employs beta, exponential, and mixed gamma distribution for Nanopore; and Badread employs gamma distribution for both PacBio and Nanopore.

The distribution is defined by

$$f(x) = x^{k-1}\frac{\exp(-x/\theta)}{\Gamma(k)\theta^k} \tag{3.1}$$

where shape and scale parameters (k and θ) are determined by averaged length and standard deviation of reads in each dataset, respectively, which can be specified by the user as input parameters. PBSIM2 computes probability mass in each length, between the maximum and minimum length.

Both PacBio and Nanopore sequencers utilize exponential distributions for read accuracy, although normal distribution has been employed in the previous version of PBSIM. In other simulators, Badread employs beta distribution for both PacBio and Nanopore. My preliminary experiments indicated that exponential distribution was more suitable than any other distribution for latest real datasets of both PacBio and Nanopore (Figure 3.4).

Precisely, I define read accuracy distribution by

$$p(x) = \frac{f(x)}{\sum_{i=min}^{max} f(x_i)} \tag{3.2}$$

where

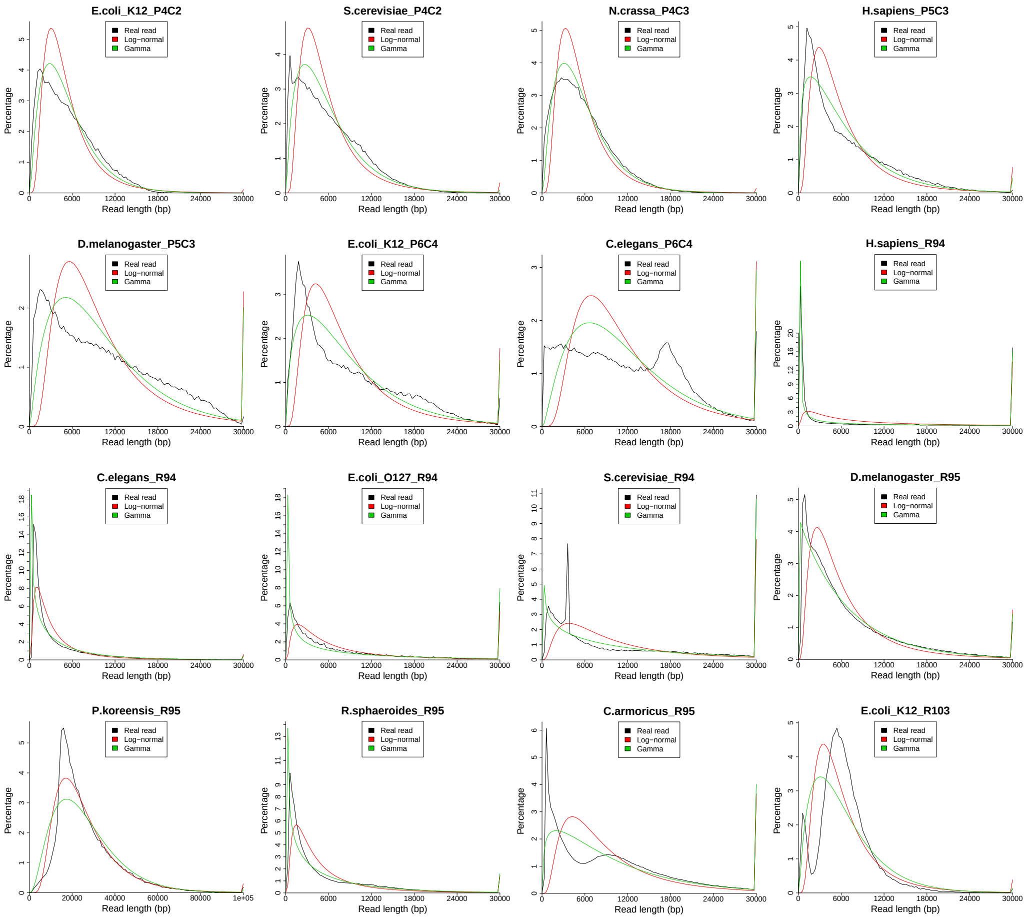
$$f(x) = \exp(0.22x) \tag{3.3}$$

and the minimum and maximum of accuracy are determined by averaged accuracy of reads, which can be specified by the user as input parameters. PBSIM2 computes probability mass in each accuracy between the maximum and minimum accuracy.

A nucleotide sequence of a read is uniformly sampled from the reference sequence, and errors are introduced into the sequence as follows: For each position of the read, all error types (substitution, insertion, and deletion) are introduced according to quality score at that position. In the previous version of PBSIM, deletion rate is uniform throughout all positions of every simulated read, but the latest datasets show that the rates of all error types are related to the quality scores (Figure 3.21). All error rates are calculated from quality scores and the ratio of error types given by the user. With regard to a deletion, there is no quality score for the deletion itself; thus, the quality score of the 5'neighbor is used. As in the previous version of PBSIM, half of the inserted nucleotides are chosen to be the same as their following nucleotides, and the other half are randomly chosen.

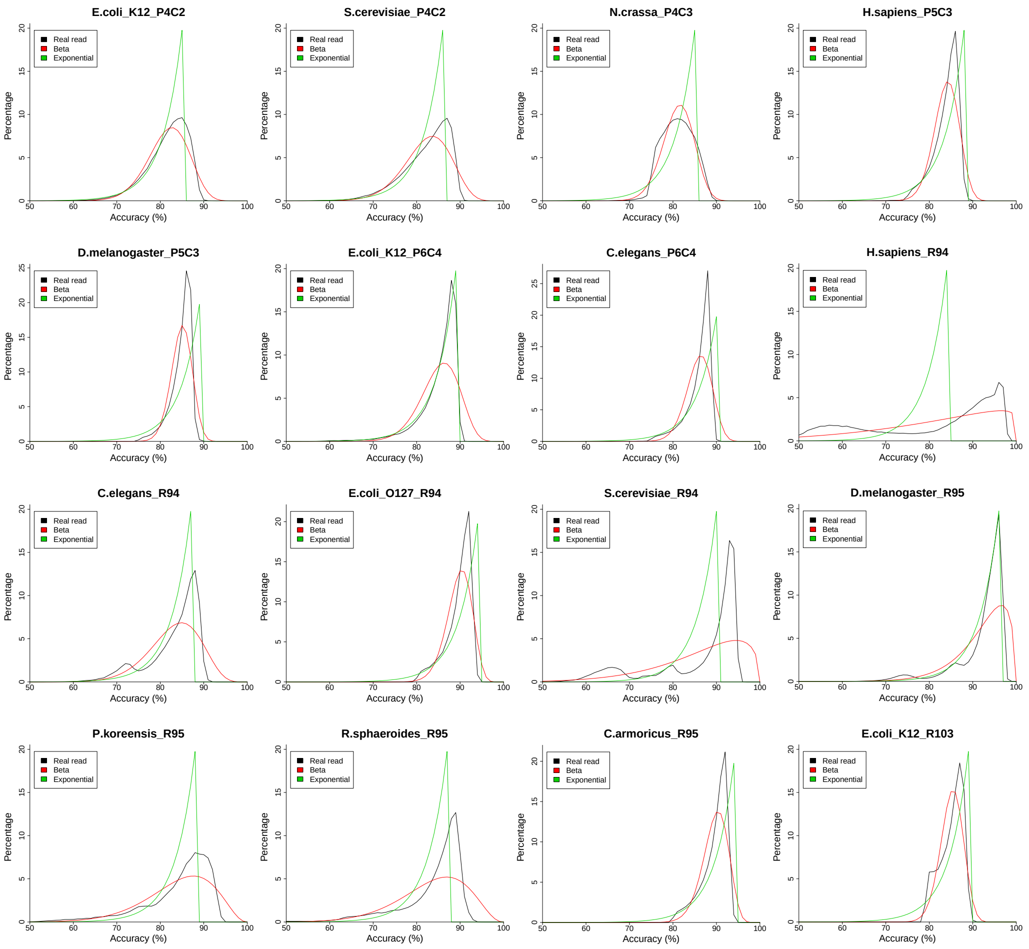
Sampling-based simulation implemented in PBSIM can also be used in PBSIM2. In this simulation, the length and quality scores of a read are randomly sampled from real data provided by the user. Subsequently, a nucleotide sequence is randomly extracted from the reference sequence, and errors are introduced in the same way as model-based simulation.

Figure 3.3. Read length distribution for each of the datasets in Tables 3.2 and 3.3



Dataset name is species (e.g., *E. coli*_K12) + chemistry (e.g., P4C2). Each graph shows distribution of real read length, as well as log-normal and gamma distributions with parameters derived from real reads.

Figure 3.4. Read accuracy distribution for each of the datasets in Tables 3.2 and 3.3

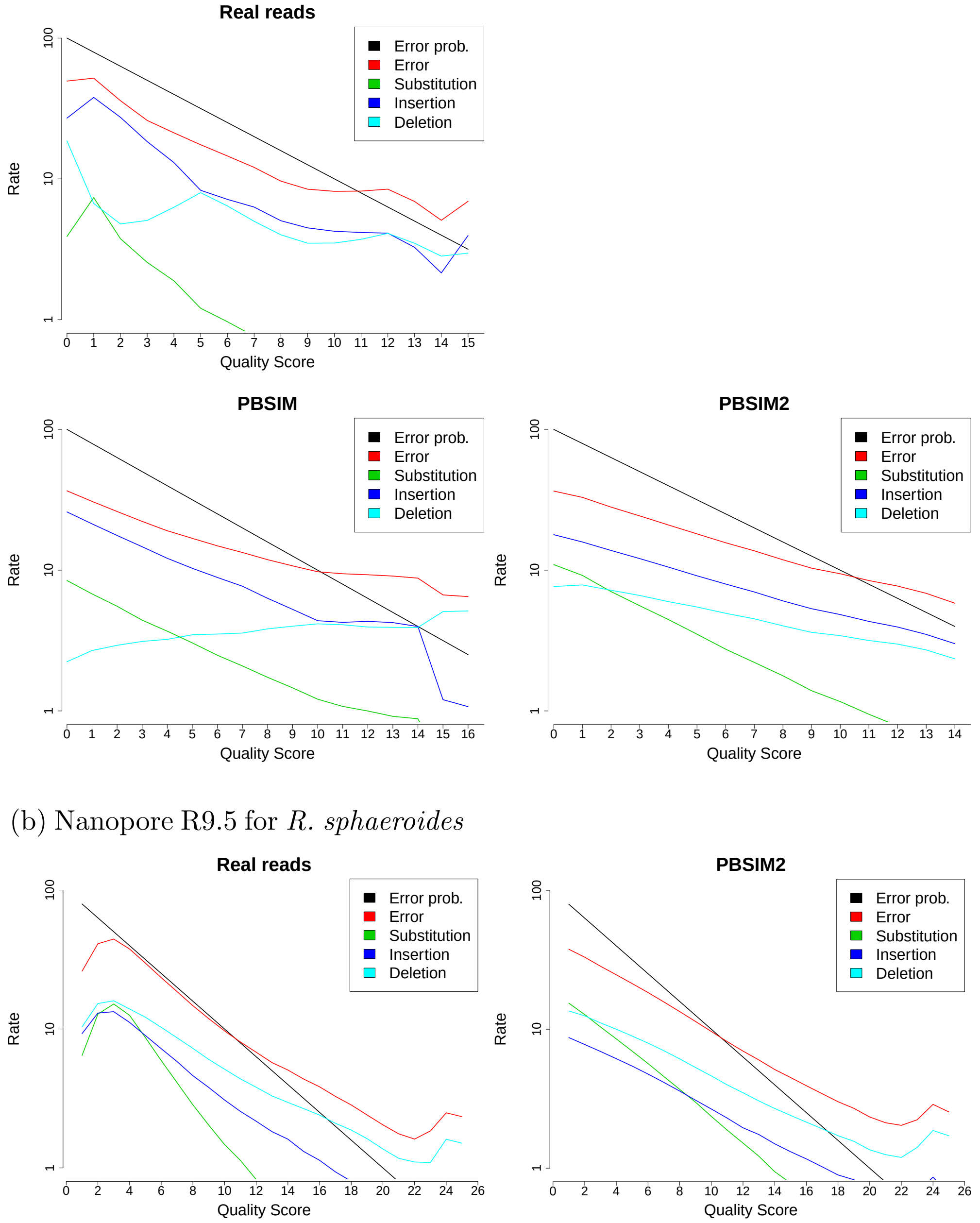


Dataset name is species (e.g., *E. coli*_K12) + chemistry (e.g., P4C2). Each graph shows distribution of real read length, as well as beta and exponential distributions with parameters derived from real reads. Read accuracy was computed from quality scores.

3.2.5 Execution of other simulators

To evaluate the ability of PBSIM2 to simulate the non-uniformity of real reads, I conducted simulations using other simulators and observed their non-uniformity. For NPBSS, I simulated PacBio CLR using the default error model. For PaSS, I simulated PacBio CLR using a prepared profile (sim.config). For LongISLND [29], I built models from real reads and simulated PacBio CLR using the models. For Badread, I built models from real reads and simulated PacBio CLR and Nanopore reads using the models. For DeepSimulator1.5, I simulated Nanopore fast5 using context-independent kmer pore model and basecalled using Guppy.

Figure 3.5. Relationship between the quality score and error rate for real reads and simulated reads


(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R. sphaeroides*

Each graph shows averaged error rate for each quality score. The horizontal axis is PHRED33, quality score defined in terms of the estimated error probability (e.g., quality scores 4, 7, and 10 represent error probabilities of 40, 20, and 10%, respectively) [45]. "Error" is the sum of the substitution, insertion, and deletion rates. Error rates were obtained from the alignment of the real and simulated reads to the reference sequences.

3.3 Result and Discussion

3.3.1 CPU time and memory consumption

For each simulator, CPU time and maximum memory usage were measured for generating a total of 100 Mbp of reads. NPBSS was executed on a Windows system equipped with Intel(R) Core(TM) CPU(i7-8565U@1.80GHz). The others were executed on the National Institute of Genetics (NIS) supercomputer system. The execution of DeepSimulator1.5 included basecalling by Guppy and checking read accuracy by Minimap2, and used ”-c 8” option (CPU number). Results are shown in Table 3.4. PBSIM is the fastest and consumes minimal memory, which enables users to simulate reads on their laptops.

Table 3.4. CPU time and maximum memory for each simulator

Simulator	CPU time (sec.)	Maximum memory (Gbyte)
PBSIM	5	0.2
PBSIM2 (this work)	7	0.2
LongISLND	565	26.7
NPBSS	1,024	0.1
DeepSimulator1.5	113,344	15.3
PaSS	14	0.8
Badread	1,498	3.5

CPU time and maximum memory usage were measured for generating a total of 100 Mbp of reads. NPBSS was executed on a Windows system equipped with Intel(R) Core(TM) CPU(i7-8565U@1.80GHz). The others were executed on the National Institute of Genetics (NIS) supercomputer system. The execution of DeepSimulator1.5 included basecalling by Guppy and assessing read accuracy by Minimap2; when using ”-c 8” option (CPU number), wall-clock time was 20,662 seconds.

3.3.2 Evaluation of a generative model of quality scores

To evaluate PBSIM2 that implemented a novel generative model of quality scores trained using FIC-HMM, I compared simulated reads of PBSIM2 with real reads in terms of non-uniformity of quality scores. PBSIM2 simulated reads with the same parameters (e.g., mean and standard deviation of read length and accuracy) as real reads. I also evaluated simulated reads of Markov Model (MM), because in Nanopore sequencing, the raw current signal is mainly influenced by 5 or 6-mer that occupies the pore simultaneously [65], and [60] showed that the strongest feature for predicting the accuracy of each k-mer was the accuracy of neighboring k-mers, one step away. MM generates quality scores by first- and second-order MM (referred to as '1st-order MM' and '2nd-order MM', respectively) of transition probabilities of quality scores of real reads.

It is clear that the non-uniformity of simulated reads of PBSIM2 is sufficiently similar to that of real reads in both PacBio and Nanopore (Figure 3.6, and Figures 3.7 and 3.8 show graphs of all the interval sizes). Figure 3.6 (b) also indicates that 1st-order MM is able to simulate the non-uniformity, as well as PBSIM2 in Nanopore. I utilized the Kullback-Leibler (KL) divergence for observing similarity between non-uniformity (see Figure 3.6). For P (real distribution) and Q (simulated distribution), the KL divergence from Q to P is defined to be;

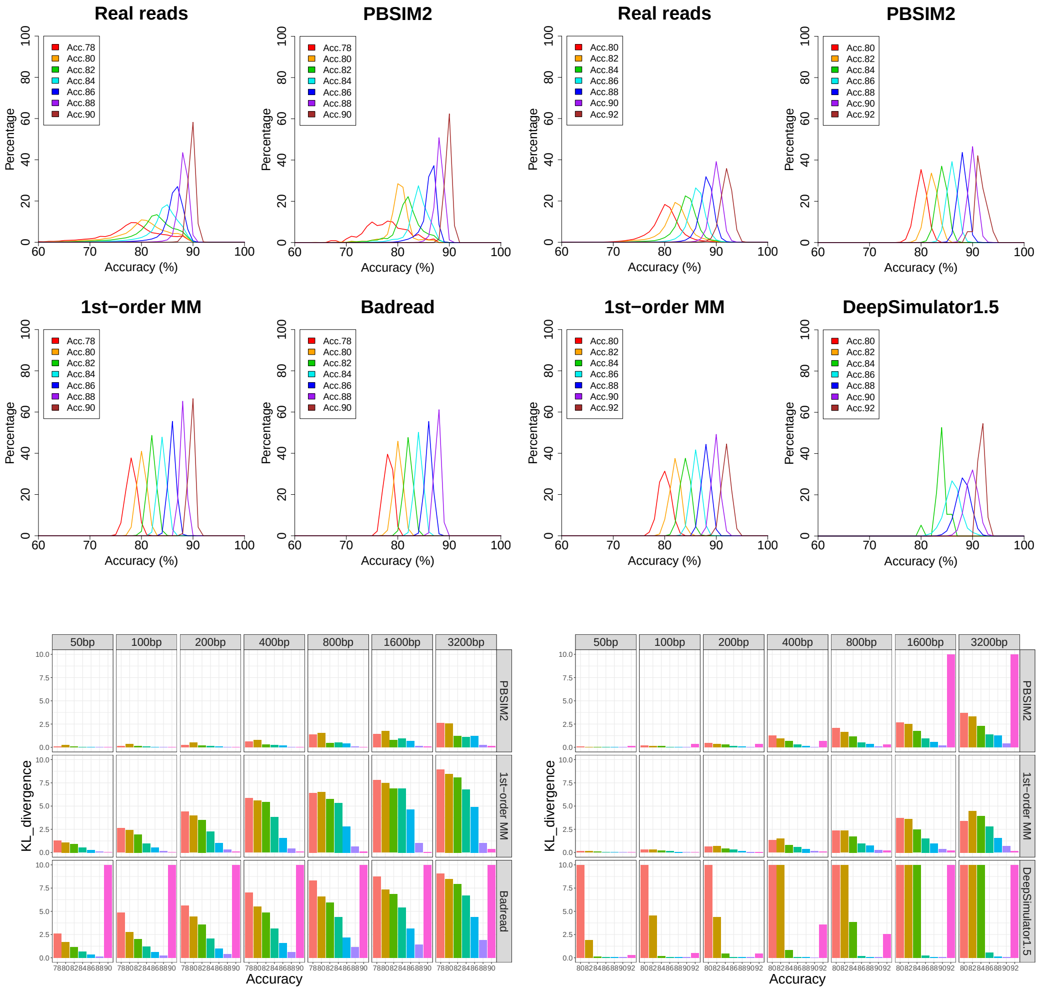
$$D_{KL}(P||Q) = \sum_{i} P(i) \log_2 \frac{P(i)}{Q(i)}.$$

Figure3.9 show that features of the transition probability matrix are clearly different between PacBio and Nanopore, and the transition ranges in Nanopore are narrower than those in PacBio. Thus, MM is successful in Nanopore. Furthermore Figures 3.10 and 3.11 show that in FIC-HMM, the transition ranges of states in Nanopore are narrower than those in PacBio, and the emission ranges of states in Nanopore are narrower and simpler than those in PacBio. These observations in MM and FIC-HMM are consistent. I decoded training data for FIC-HMM into states using the Viterbi algorithm, and examined continuous length of state (e.g., if the same state is lined up five times in a row, the continuous length is five). Figure 3.12 shows that the continuous length of state in PacBio is longer than that in Nanopore. In both PacBio and Nanopore R9.5, 2nd-order MM was a slightly better simulation than 1st-order MM (Supplementary Figures 3.13, 3.14, 3.15, and 3.16). However, in Nanopore R10.3, they were almost the same (Figures 3.17 and 3.18).

Figures 3.13-3.18 also show comparisons with other long read simulators. In simulation of PacBio reads, PBSIM2 is able to simulate the non-uniformity of real reads more than that of any other simulator (see Figure 3.14). Even in the simulation of Nanopore reads, PBSIM2 is one of the best simulators for overall read accuracy, but at 86-90% read accuracy of Figure 3.16 and at 84-88% of Figure 3.18, DeepSimulator1.5 is the best. However, DeepSimulator1.5 has narrow ranges of read accuracy.

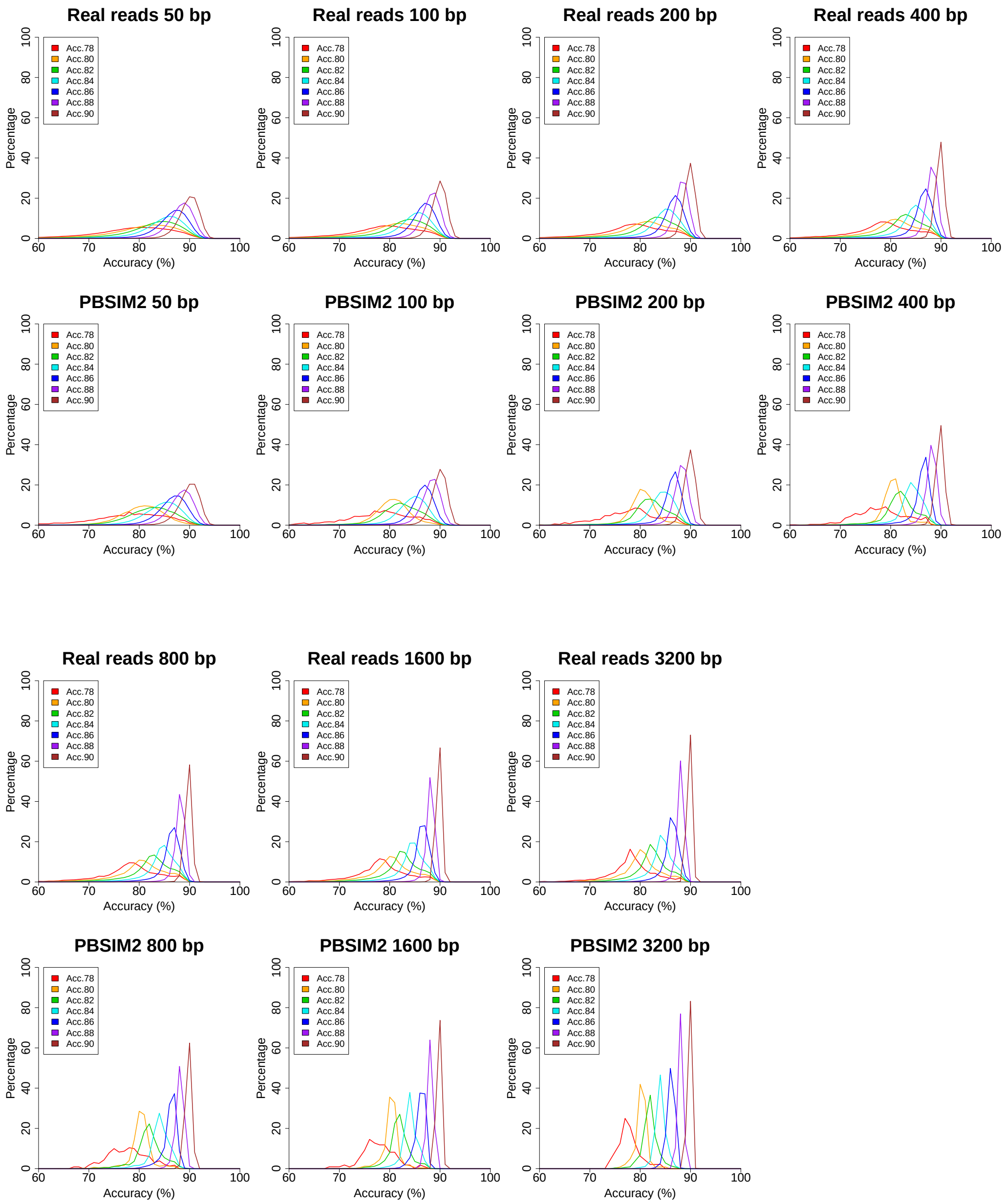
Figure 3.6. Simulation of non-uniformity of quality scores and evaluation by Kullback-Leibler (KL) divergence

(a) PacBio P6–C4 for *C.elegans* (b) Nanopore R9.5 for *R.sphaeroides*



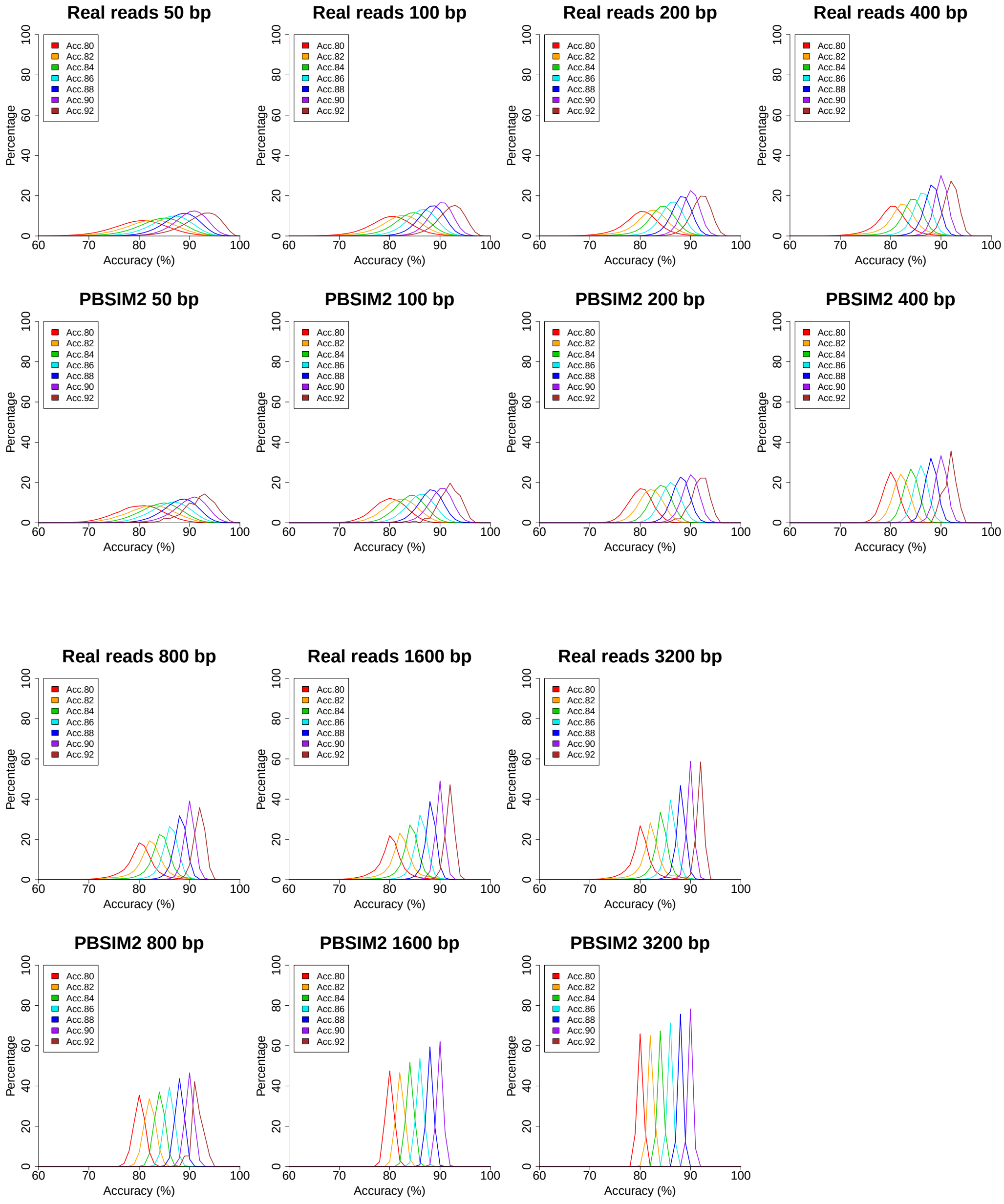
Each graph shows distributions of accuracy of 800 bp disjoint intervals in reads in the same way as Figure 3.1. Read groups (e.g., Acc.78) with insufficient data are not shown in the graphs. PBSIM2, the new version of PBSIM, generated reads using model-based simulation. '1st-order MM', our in-house software tool, generated quality scores for each read group, by a first-order Markov Model of transition probabilities of the quality score of real reads. Badread built a model and generated reads. DeepSimulator1.5 generated Nanopore fast5 using context-independent kmer pore model and basecalled using Guppy. KL divergence of distribution of accuracy of fixed size (50, 100, 200, 400, 800, 1600, and 3200 bp) intervals between real and simulated reads. Upper-limit value of KL divergence was 10.

Figure 3.7. Non-uniformity of quality scores for real and simulated reads of PacBio P6–C4 for *C. elegans*



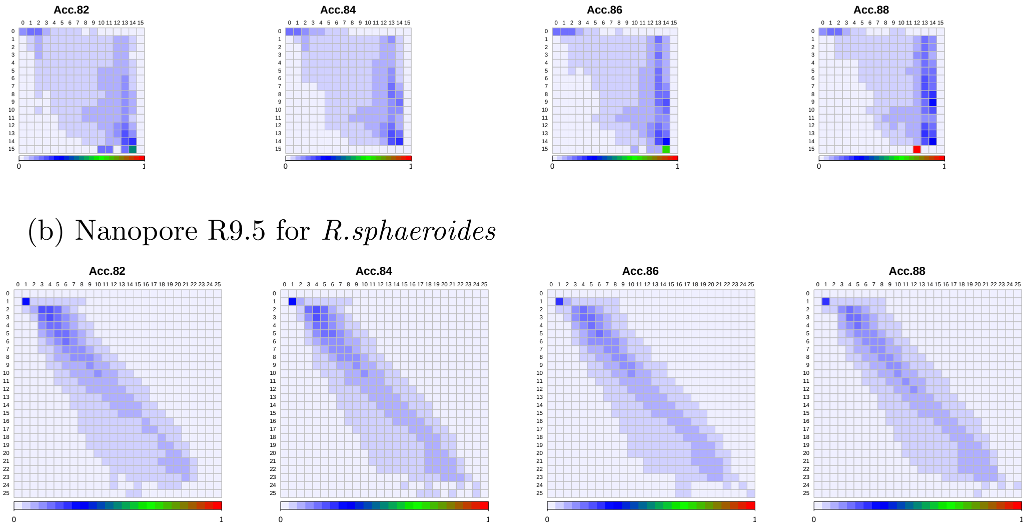
After grouping reads by their accuracy, the reads were segmented into fixed size (50, 100, 200, 400, 800, 1600, and 3200 bp) disjoint intervals, and the accuracy of each interval was computed from the quality scores. Each graph shows the distribution of averaged accuracy of each interval, where the colors of the plotted lines represent read groups (e.g., “ Acc.78 ” is a read group whose accuracy is 77.5%–78.4%). Read groups (e.g., Acc.78) with insufficient data are not shown in the graphs.

Figure 3.8. Non-uniformity of quality scores for real and simulated reads of Nanopore R9.5 for *R. sphaeroides*



Each graph shows the distribution of averaged accuracy of each interval in the same way as Figure 3.7.

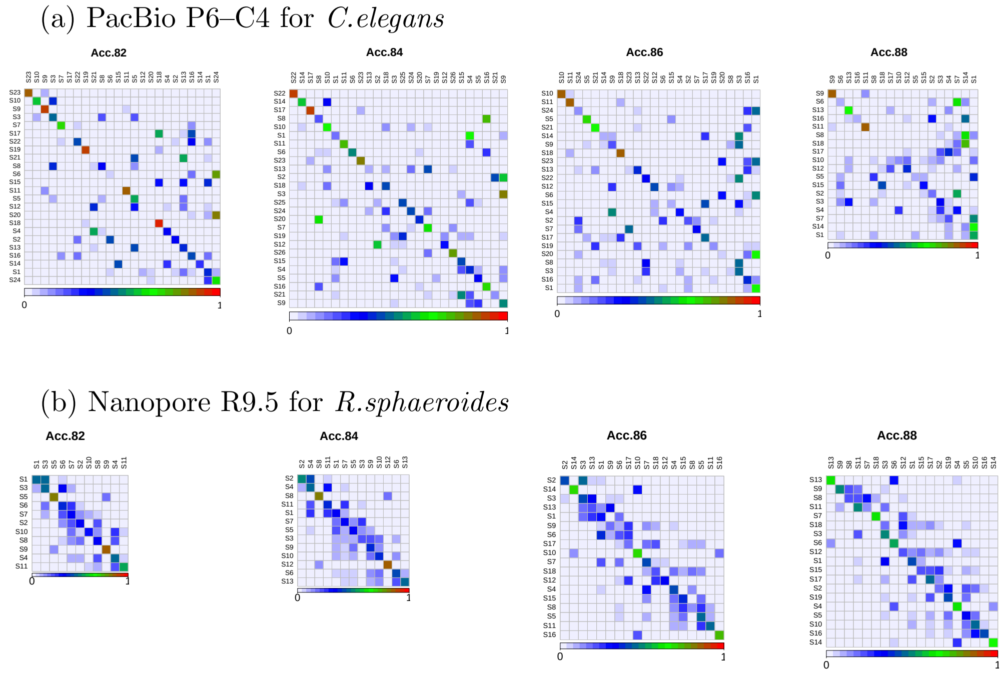
Figure 3.9. Transition probability matrices of quality scores of real reads


(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R.sphaeroides*

The vertical and horizontal axes are PHRED33 quality scores defined in terms of the estimated error probability (e.g., quality scores 4, 7, and 10 represent error probabilities of 40%, 20%, and 10%, respectively) [45]. Quality scores on the vertical axis transition to scores on the horizontal axis. The sum of transition probabilities on each quality score of the vertical axis is 100%. These are matrices of 'Acc.82'-'Acc.88' (e.g., 'Acc.84' refers to a read group with an accuracy of 83.5%-84.4%). In the Nanopore matrix, quality scores above 25 are not displayed.

Figure 3.10. Transition probability matrices of states of FIC-HMM

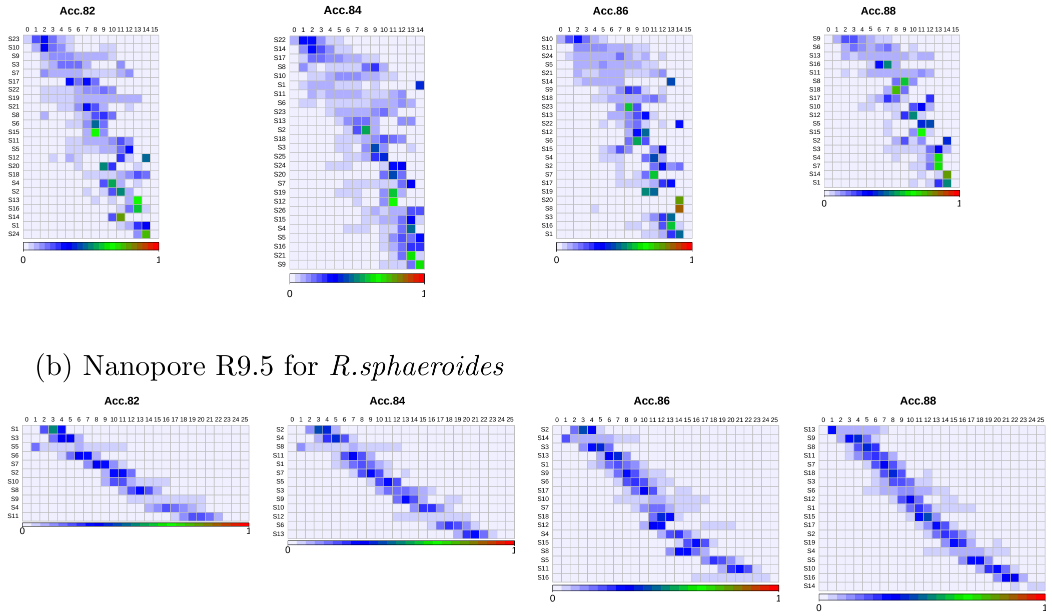


(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R.sphaeroides*

The vertical and horizontal axes represent states of FIC-HMM, which are sorted in order of the increasing averaged quality score emitted by them. States on the vertical axis transition to states on the horizontal axis. The sum of transition probabilities on each state of the vertical axis is 100%. These are matrices of 'Acc.82'-'Acc.88' (e.g., 'Acc.84' refers to a read group with an accuracy of 83.5%–84.4%).

Figure 3.11. Emission probability matrices of states of FIC-HMM


(a) PacBio P6–C4 for *C.elegans*

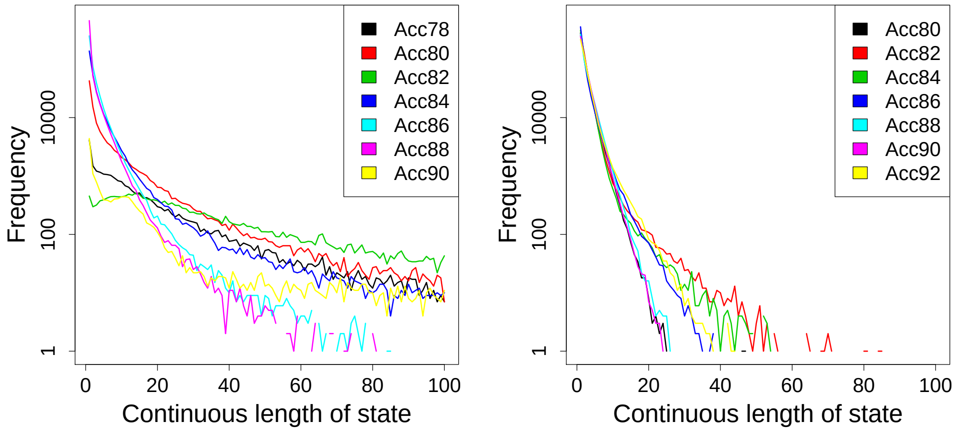
(b) Nanopore R9.5 for *R.sphaeroides*

The vertical axis represents states of FIC-HMM, which are sorted in the order of increasing averaged quality score emitted by them. The horizontal axis is PHRED33 quality score defined in terms of the estimated error probability (e.g., quality scores 4, 7, and 10 represent error probabilities of 40%, 20%, and 10%, respectively) [45]. States on the vertical axis emit quality scores on the horizontal axis. The sum of emission probabilities on each state of vertical axis is 100%. These are matrices of 'Acc.82'–'Acc.88' (e.g., 'Acc.84' refers to a read group with an accuracy of 83.5%–84.4%). In the matrix of Nanopore, quality scores above 25 are not displayed.

Figure 3.12. Distributions of continuous length of state

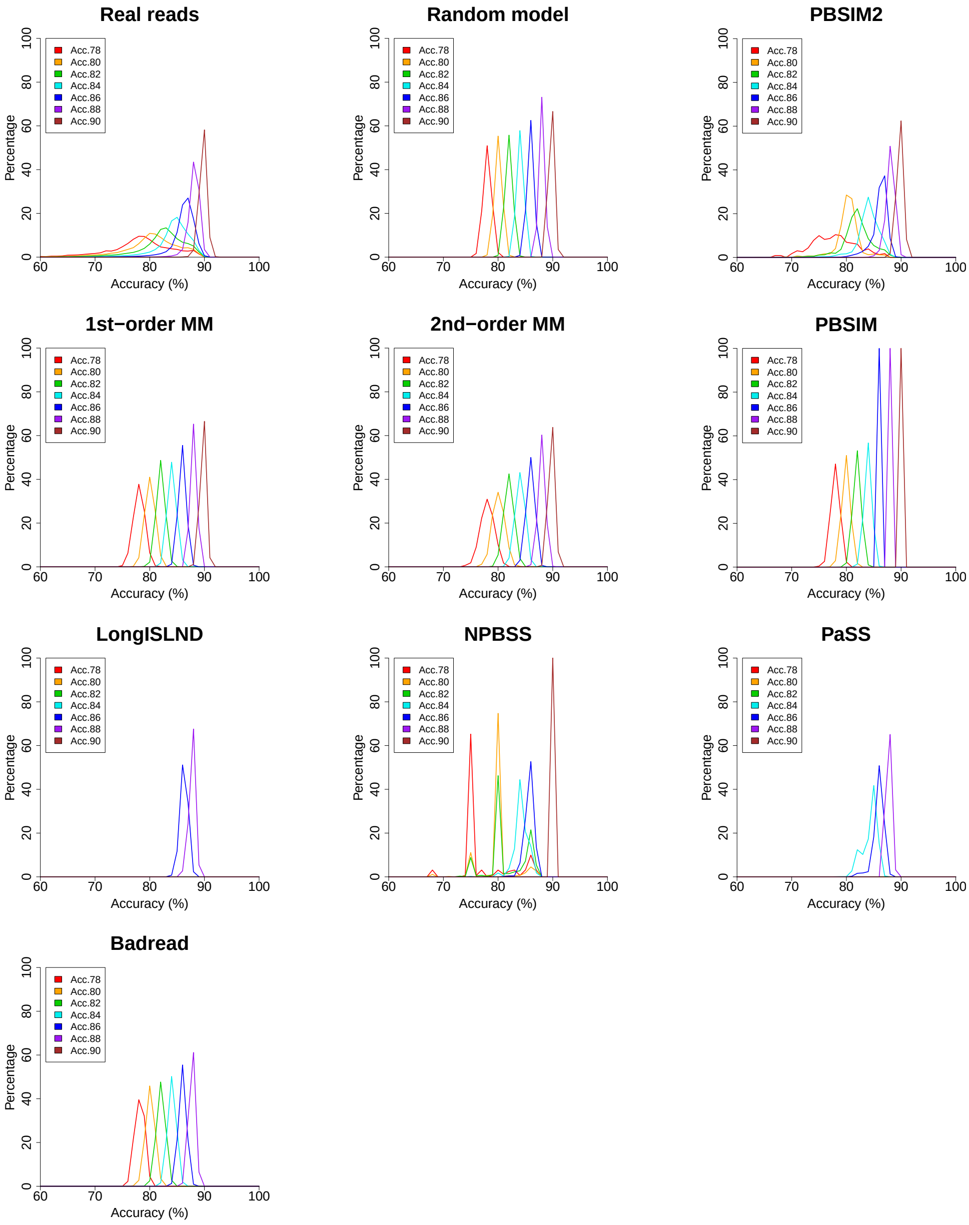
(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R.sphaeroides*



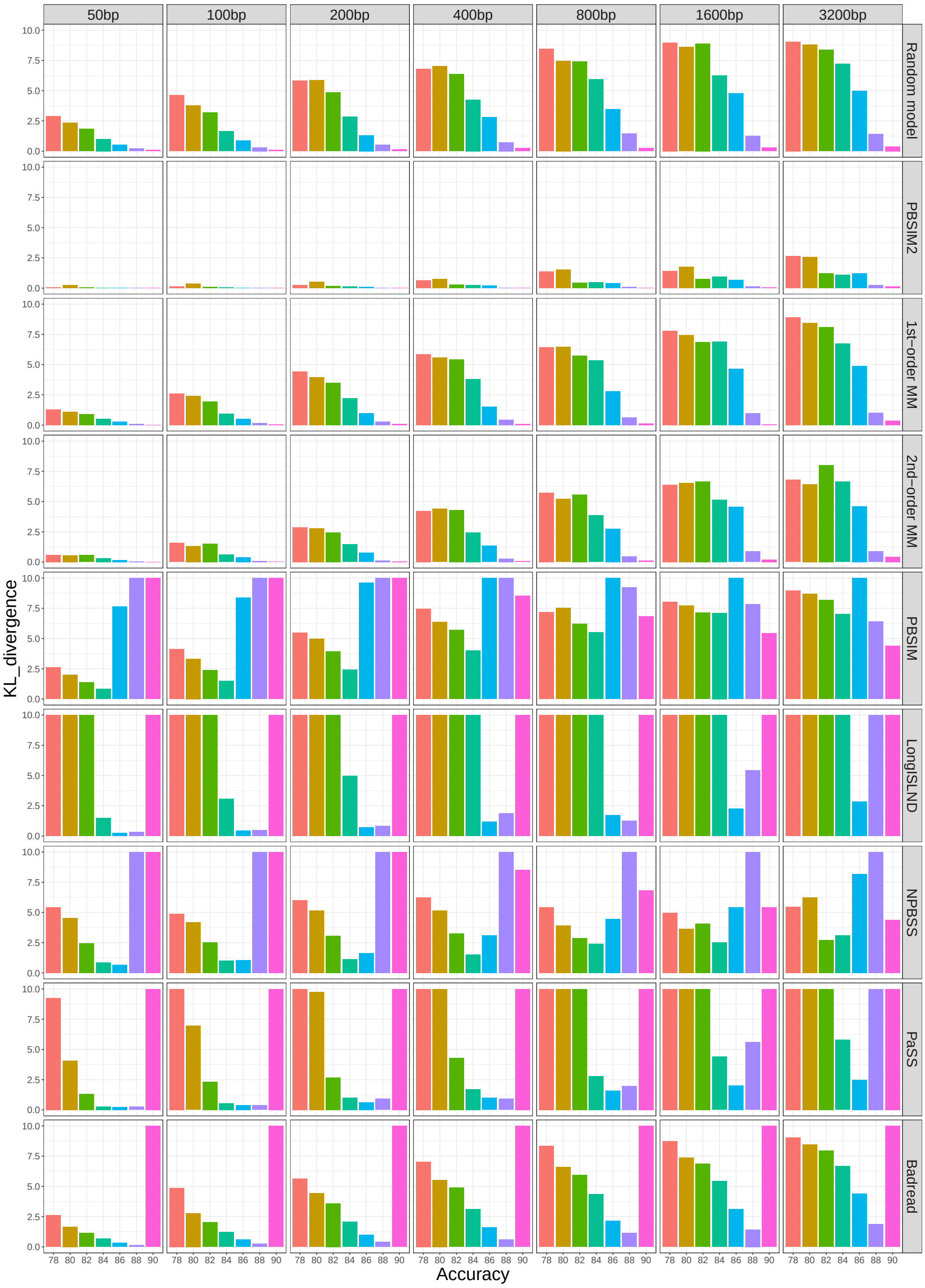
Training data for FIC-HMM was decoded into stats using the Viterbi algorithm. If the same state is lined up five times in a row, the continuous length is five. The vertical axis (log scale) is the frequency of each continuous length of state. The horizontal axis is the continuous length of state. Colors of plotted lines represent the read groups (e.g., ' Acc.78 ' refers to a read group with an accuracy of 77.5%–78.4%). Continuous lengths of state above 100 are not displayed.

Figure 3.13. Non-uniformity of quality scores for real and simulated reads of PacBio P6–C4 for *C. elegans*



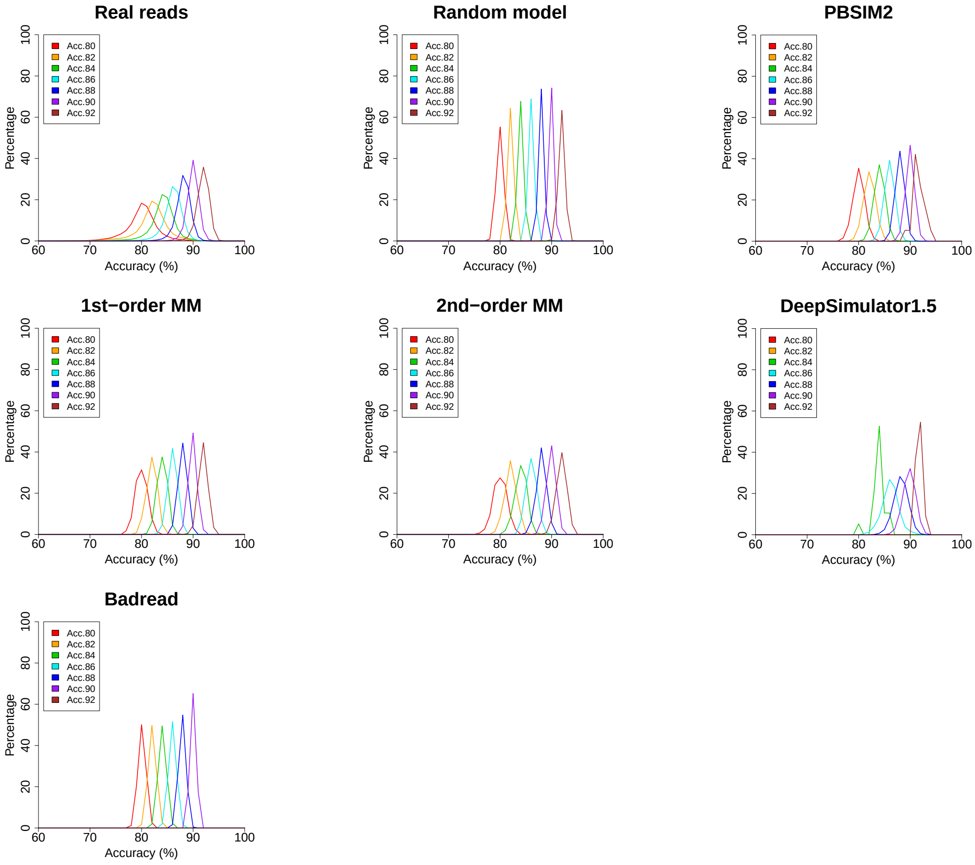
Each graph shows distributions of accuracy of 800 bp disjoint intervals in reads in the same way as Figure 3.7. PBSIM (i.e., previous version) has frequency tables of quality score for only Acc.60–85; thus, for Acc.86–90, 800 bp interval accuracy is constant.

Figure 3.14. Kullback-Leibler (KL) divergence of distribution between real and simulated reads of PacBio P6–C4 for *C. elegans*



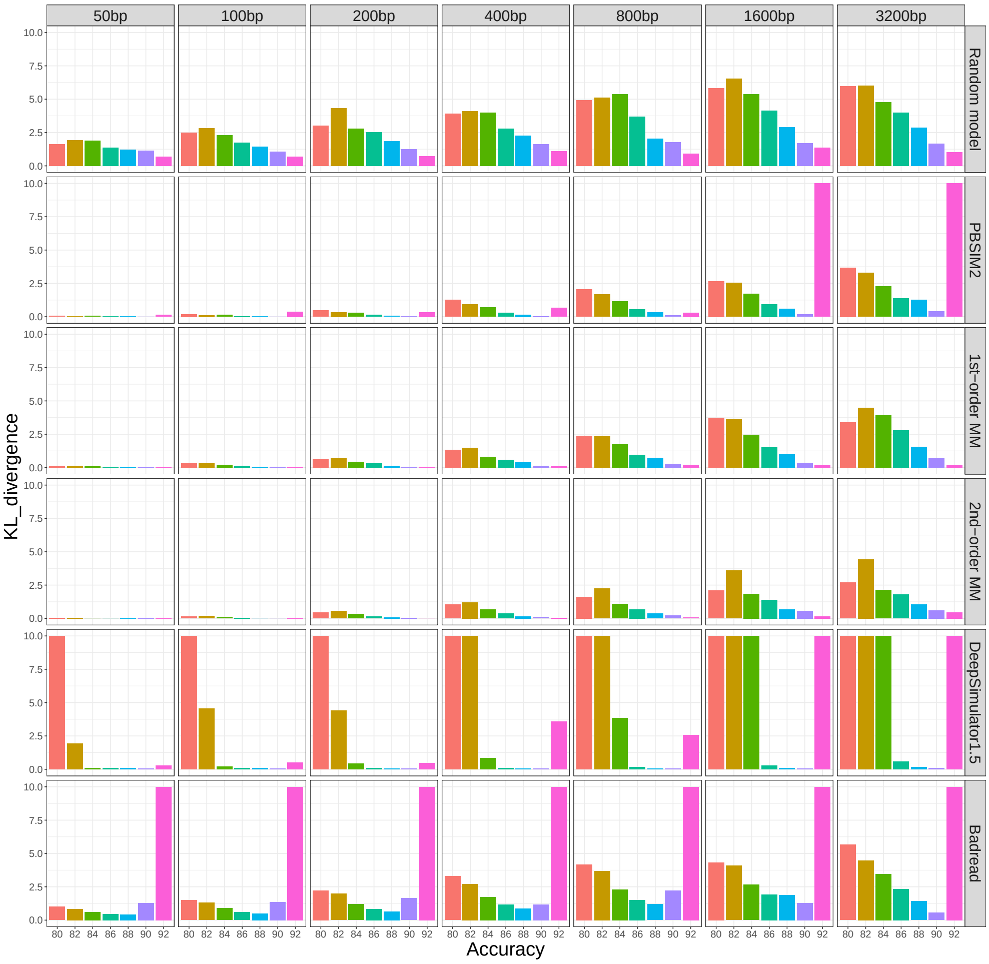
Kullback-Leibler (KL) divergence of distribution of accuracy of fixed size (50, 100, 200, 400, 800, 1600, and 3200 bp) intervals between real and simulated reads of PacBio P6–C4 for *C. elegans* in Figure 3.13. Upper-limit value of KL divergence is 10.

Figure 3.15. Non-uniformity of quality scores for real and simulated reads of Nanopore R9.5 for *R. sphaeroides*



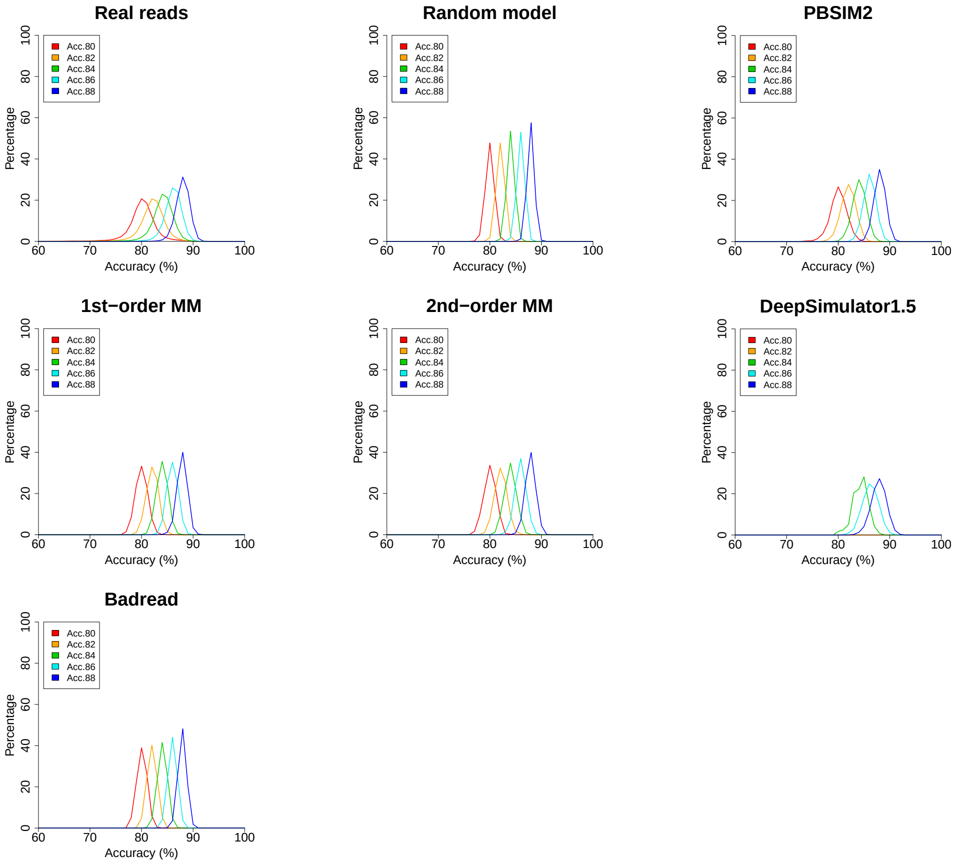
Each graph shows distributions of accuracy of 800 bp intervals in reads in the same way as Figure 3.7. Read groups (e.g., Acc.78) with insufficient data are not shown in the graphs.

Figure 3.16. Kullback-Leibler (KL) divergence of distribution between real and simulated reads of Nanopore R9.5 for *R. sphaeroides*



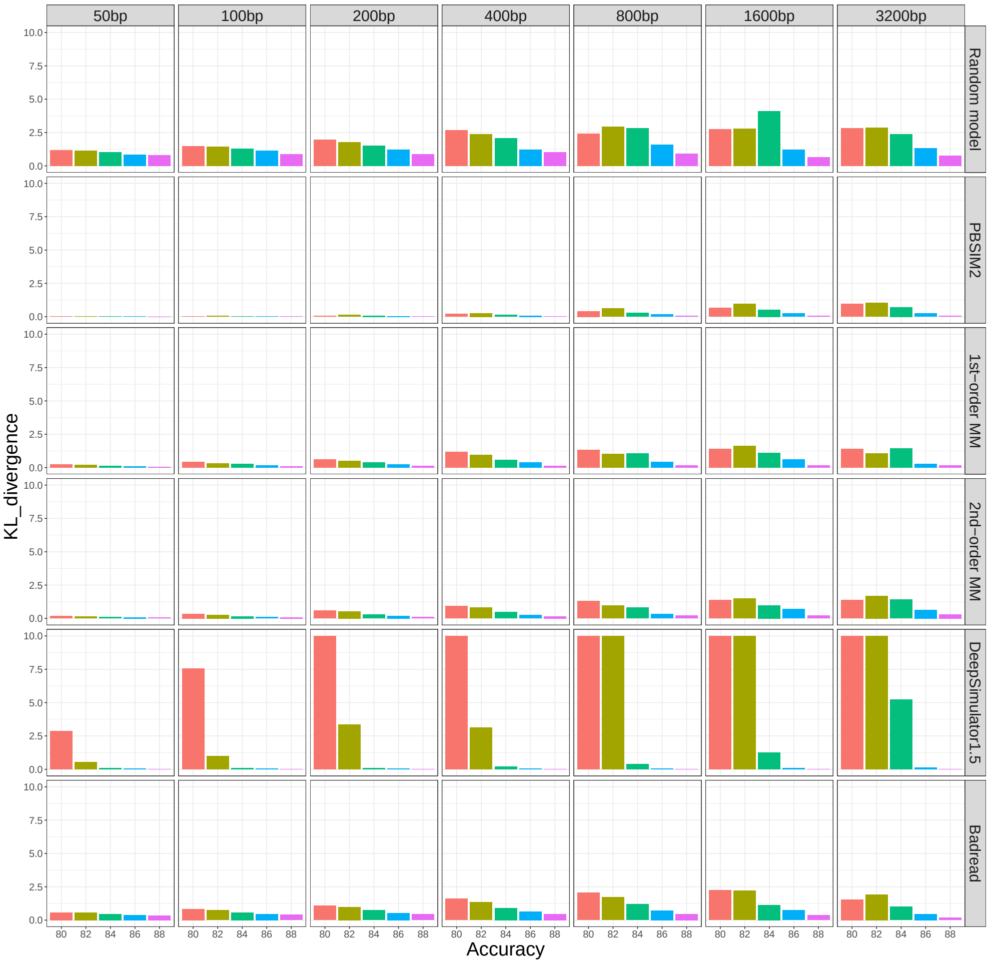
Kullback-Leibler (KL) divergence of distribution of averaged accuracy of fixed size (50, 100, 200, 400, 800, 1600, and 3200 bp) intervals between real and simulated reads of Nanopore R9.5 for *R. sphaeroides* in Figure 3.15. Upper-limit value of KL divergence is 10.

Figure 3.17. Non-uniformity of quality scores for real and simulated reads of Nanopore R10.3 for *E-coli* K12



Each graph shows distributions of accuracy of 800 bp intervals in read sequences, in the same way as Figure 3.13.

Figure 3.18. Kullback-Leibler (KL) divergence of distribution of averaged accuracy between real and simulated reads of Nanopore R10.3 for *E. coli* K12

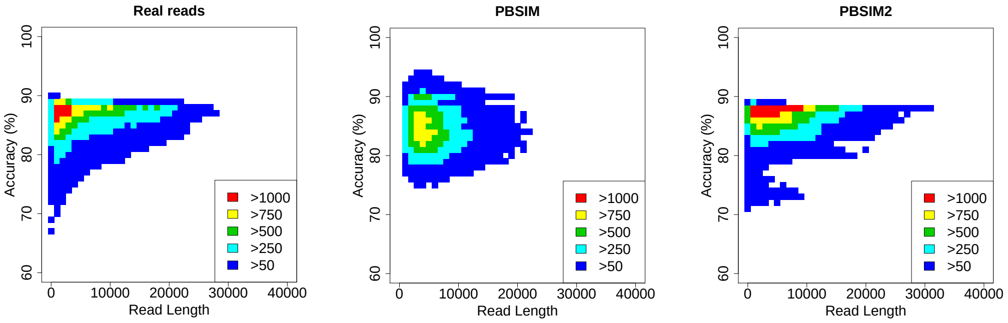


Kullback-Leibler (KL) divergence of distribution of averaged accuracy of fixed size (50, 100, 200, 400, 800, 1600, and 3200 bp) intervals between real and simulated reads of Nanopore R10.3 for *E. coli* K12 in Figure 3.17. Upper-limit value of KL divergence is 10.

3.3.3 Correlation between read length and accuracy

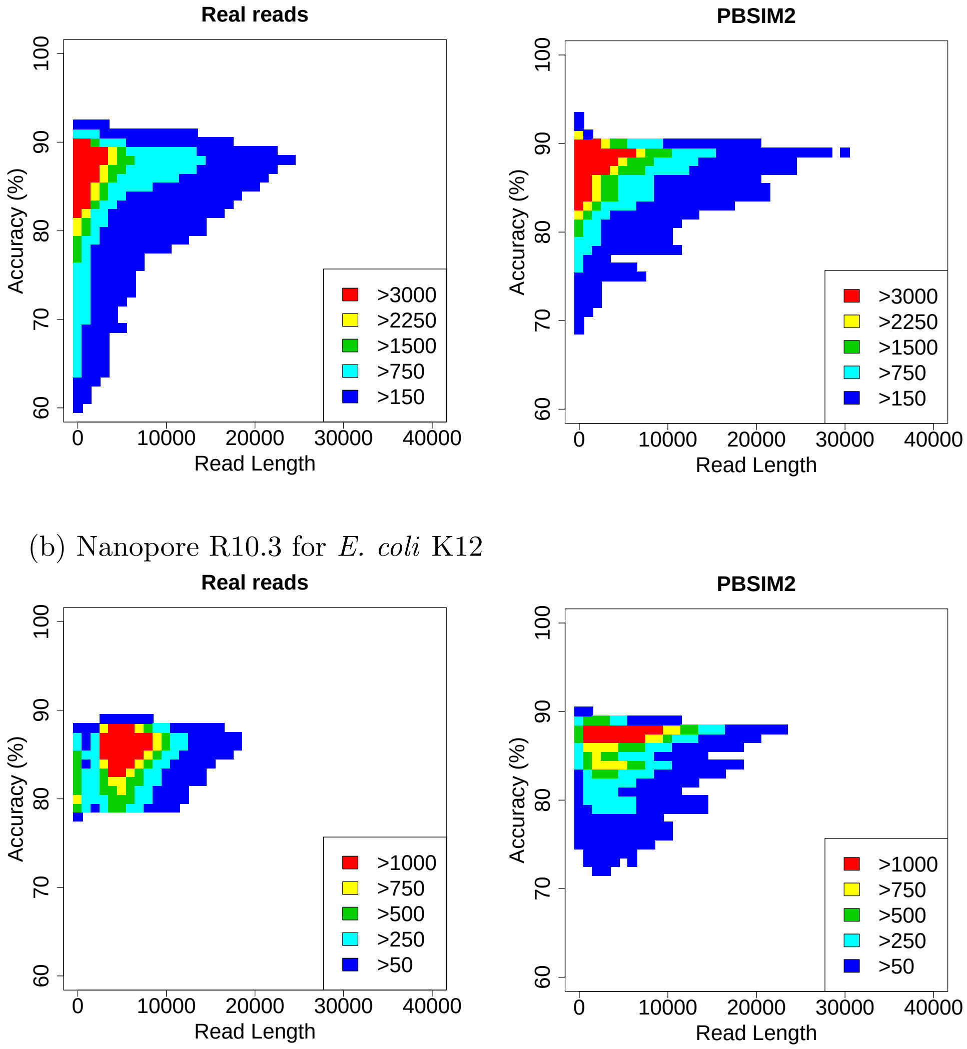
The previous version of PBSIM was unable to simulate realistic correlation between length and accuracy for each read [25, 61]. As shown in Figures 3.19 and 3.20, PBSIM2 is able to simulate realistic correlations. This improvement was mainly due to change in read accuracy distribution, as mentioned in Section 3.2.4.

Figure 3.19. Correlation between read length and accuracy for each read of PacBio P6–C4 for *E. coli* K12 MG1655



Accuracy of each read was calculated from quality scores. PBSIM and PBSIM2 simulated reads with the same parameters (e.g., mean and standard deviation of read length and accuracy) as real reads. Colors indicate the varying frequencies of each cell.

Figure 3.20. Correlation between read length and accuracy for each read



(a) Nanopore R9.5 for *R. sphaeroides*

(b) Nanopore R10.3 for *E. coli* K12

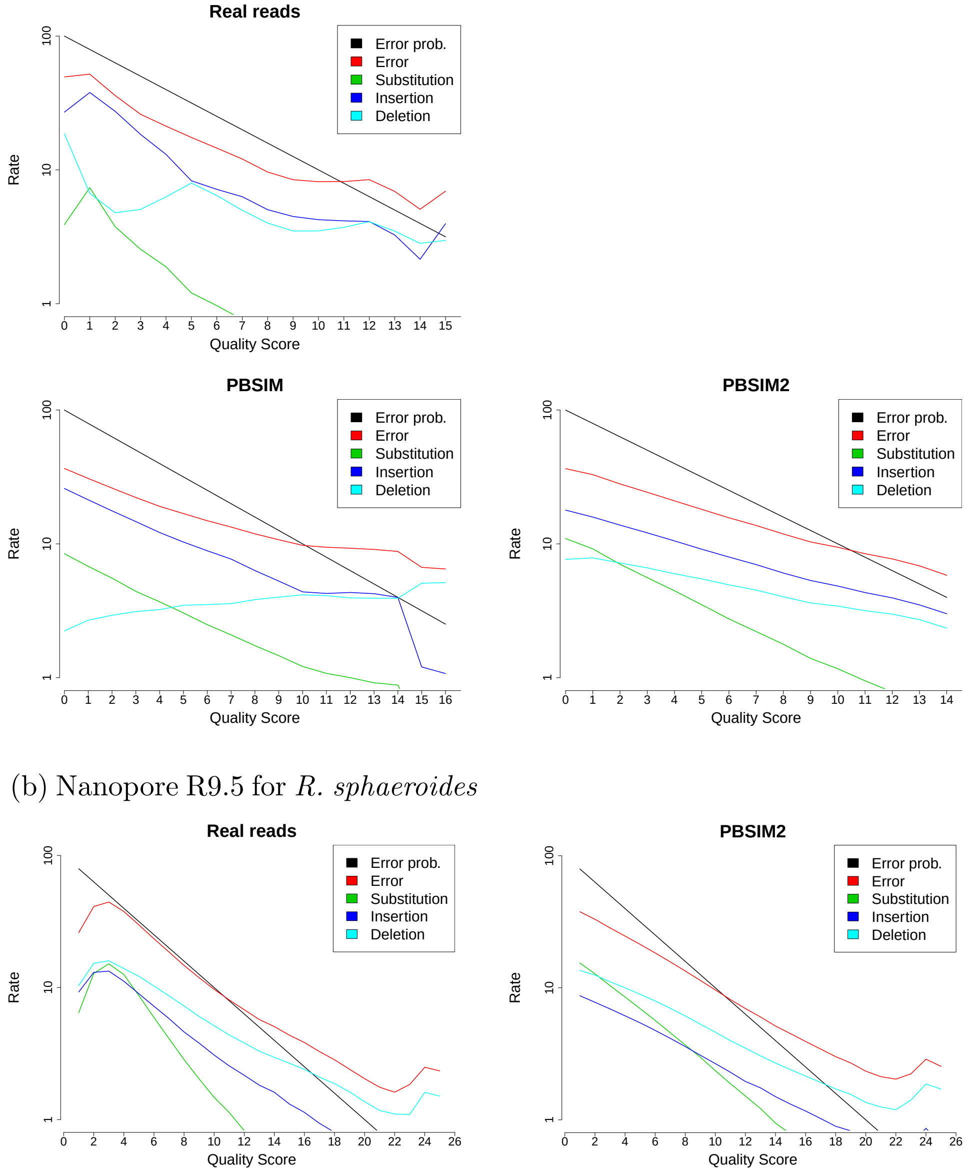
The accuracy of each read was calculated from the quality scores. PBSIM2 simulated reads with the same parameters (e.g., mean and standard deviation of read length and accuracy) as real reads. Colors indicate the different frequencies of each cell.

3.3.4 Relationship between error rate and quality scores

In the previous version of PBSIM, the relationship between error rate and quality score deviated from the correct one with increasing quality score [61]. As shown in Figure 3.21, the relationship is improved by changing the deletion rate, as mentioned in Section 3.2.4

Figure 3.21. Relationship between the quality score and error rate for real reads and simulated reads

(a) PacBio P6–C4 for *C.elegans*



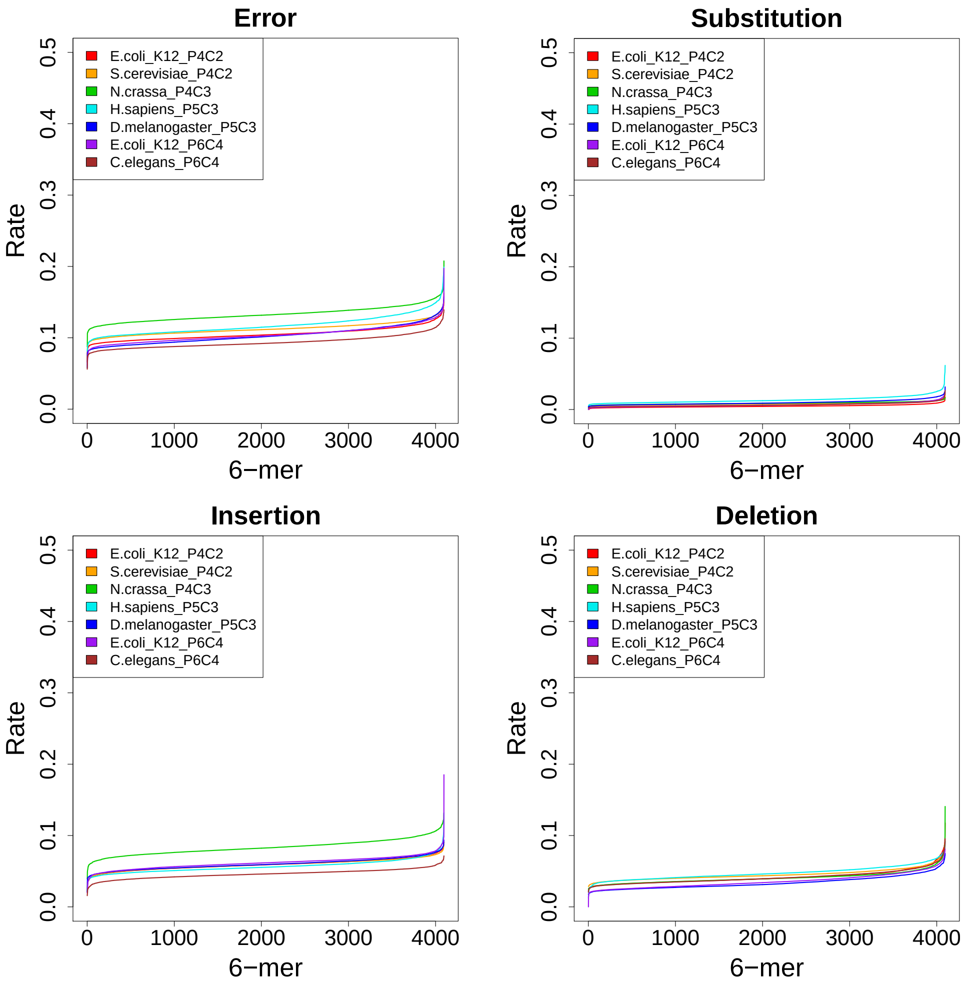
(b) Nanopore R9.5 for *R. sphaeroides*

Each graph shows averaged error rate for each quality score. The horizontal axis is PHRED33, quality score defined in terms of the estimated error probability (e.g., quality scores 4, 7, and 10 represent error probabilities of 40, 20, and 10%, respectively) [45]. ”Error” is the sum of the substitution, insertion, and deletion rates. Error rates were obtained from the alignment of the real and simulated reads to the reference sequences.

3.3.5 Nucleotide sequence-based error model

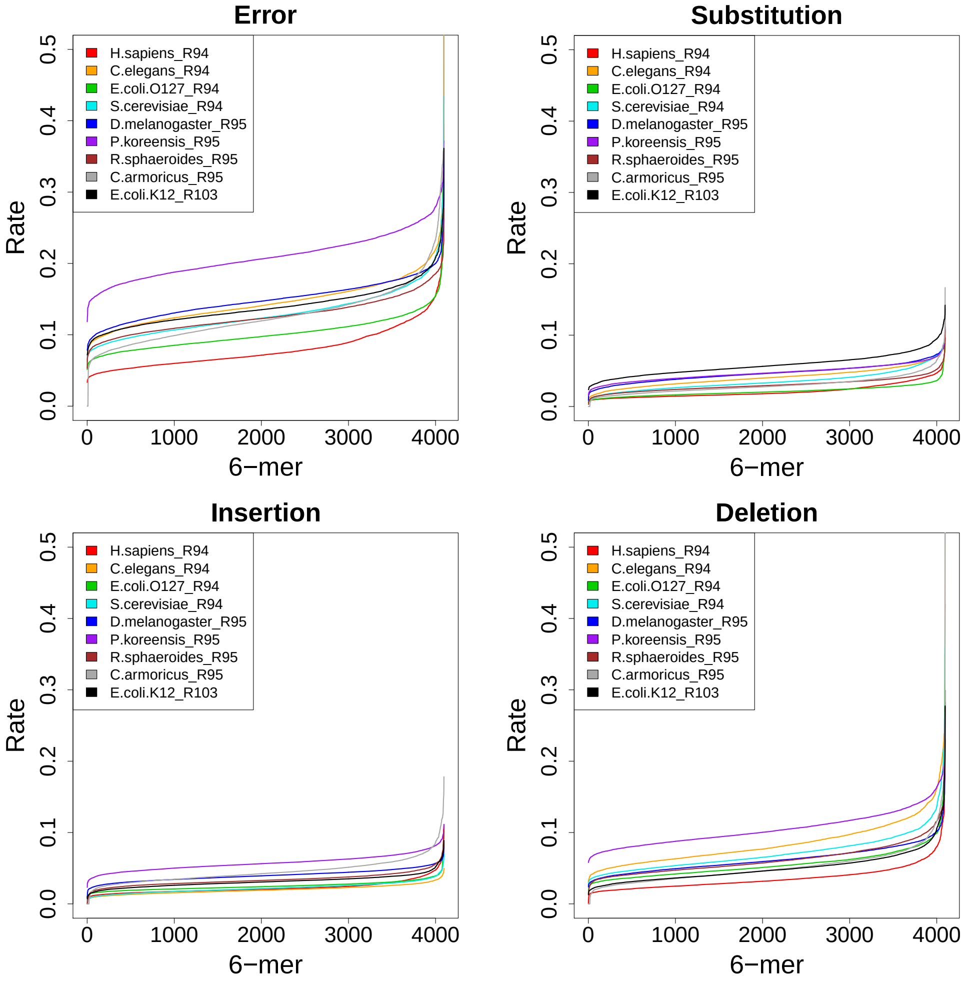
Using nucleotide sequence context (or k-mer)-based error models, generated from alignment-based analysis of real reads, several simulators are able to simulate features of real reads, such as error length [29, 56, 57, 60]. Thus, I investigated 6-mer error bias by analyzing alignments of reads to the reference sequences. As shown in Figures 3.22 and 3.23, I observed that PacBio reads had small 6-mer bias, while Nanopore reads had significant 6-mer bias, which is consistent with a previous study [60]. It has been reported that homopolymers are difficult to be called accurately by base-callers; therefore, many deletions occur at homopolymers in read sequence of Nanopore [65]. Concurrent with this report, I observed high deletion rate at homopolymers in Nanopore (see Table 3.5–3.7). I also observed that insertion and deletion (indels) were longer in Nanopore (R9.5) than those in PacBio (Figure 3.24ab). Recently, the latest Nanopore chemistry, R10, has improved resolution of homopolymeric regions [24]. Actually indels are shorter in R10 than those in R9, and the indel length distribution becomes similar to that of PBSIM2 (Figure 3.24bc), compared with PBSIM. In contrast, with regard to 6-mer error bias, R10 shows features similar to those of R9 (see Figure 3.23). Although it may be necessary to simulate 6-mer error bias, especially homopolymer-specific error, to simulate Nanopore reads accurately, this version of PBSIM2 does not address this issue because, as mentioned above, Nanopore R10 has improved for homopolymer, and basecalling software is improving for homopolymer basecalling [66].

Figure 3.22. 6-mer error bias of PacBio



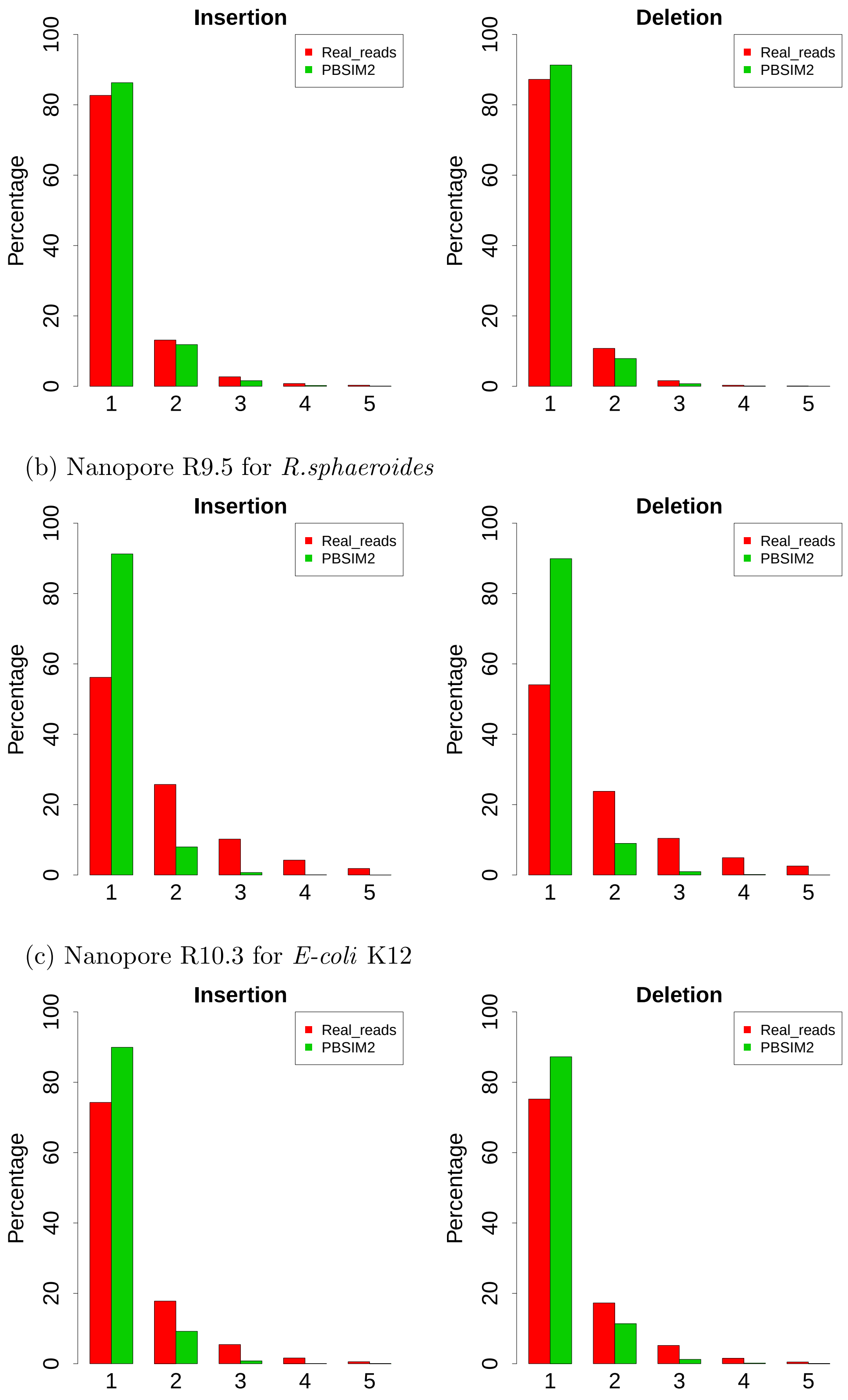
Several types of error rate were calculated for each 6-mer from the alignment of the real reads to the reference sequences. Colors of plotted lines represent each dataset in Tables 3.2. Dataset name is species (e.g., *E.coli*_K12) + chemistry (e.g., P4C2). The vertical axis represents the error rate, while the horizontal axis represents the k-mer index sorted by error rate. ”Error” is the sum of the substitution, insertion, and deletion rates.

Figure 3.23. 6-mer error bias of Nanopore



Several types of error rates were calculated for each 6-mer from the alignment of the real reads to the reference sequences. Colors of plotted lines represent each dataset in Tables 3.3. Dataset name is species (e.g., *H.sapiens*) + chemistry (e.g., R94). The vertical axis represents the error rate, while the horizontal axis represents the k-mer index sorted by error rate. ”Error” is the sum of the substitution, insertion, and deletion rates.

Figure 3.24. Distributions of insertion and deletion length (indel) for real reads and PBSIM2



(a) PacBio P6–C4 for *C.elegans*

(b) Nanopore R9.5 for *R.sphaeroides*

(c) Nanopore R10.3 for *E-coli* K12

The vertical axis represents the percentage, while the horizontal axis represents the indel length. Frequencies of indel length were obtained from the alignment of the real and simulated reads to the reference sequences.

Table 3.5. Top30 6-mer with high rate of insertion

C.elegans P6–C4		*R.sphaeroides* R9.5		*E-coli* K12 R10.3	
6-mer	rate	6-mer	rate	6-mer	rate
GTATAG	7.1%	TAGAGT	10.4%	CCTAGA	8.9%
GCACGC	7.0%	TATTAA	9.7%	CTAGAT	7.8%
GTATAC	6.9%	TTAGCT	9.1%	CCAGGA	7.8%
CACGCG	6.9%	TAATTA	9.1%	CTAGTC	7.7%
TACCGA	6.7%	CACTAA	9.0%	GCCTGG	7.6%
TACGTC	6.7%	TAAGGA	8.9%	CCAGGC	7.5%
CGGTGT	6.7%	TGTACG	8.8%	TCCTGG	7.4%
TCACGC	6.6%	ACTAAG	8.6%	CCAGGT	7.3%
GGTGTA	6.6%	GGTACA	8.1%	CCTGGT	7.3%
TACGCG	6.5%	ACACTA	7.8%	CGATCG	7.2%
ACACCG	6.5%	TAACAT	7.7%	ACCAGG	7.2%
GGCGTA	6.5%	AATAAC	7.6%	GAGATC	7.2%
GTACGT	6.5%	GAGTCA	7.5%	GATCTA	7.2%
GCGCGA	6.5%	CTGTAC	7.5%	TCTAGT	7.2%
GTATAA	6.5%	GTACAC	7.4%	ACCTGG	7.0%
TGTAGC	6.4%	GTACAT	7.3%	ATCTAG	6.8%
ACCGCG	6.4%	CCTAGT	7.2%	CGATCT	6.8%
TAACGC	6.4%	CTAACA	7.1%	AGATCG	6.8%
CGCGCA	6.4%	ATACAG	6.9%	CCCAGG	6.7%
ACGTCG	6.4%	AGTACA	6.9%	TGATCG	6.6%
GCGGTA	6.4%	CGAGTC	6.9%	GATCTC	6.6%
GTACCG	6.4%	ACTTTA	6.8%	AGATCA	6.6%
ACCGCA	6.4%	TCTAGT	6.8%	GATCAT	6.5%
TGCGGT	6.4%	ACCTTA	6.7%	GATCGG	6.5%
CGCGCG	6.3%	CATGTA	6.7%	GATCGC	6.5%
GTGTAG	6.3%	CTAGTG	6.7%	TGATCA	6.5%
GTCGTA	6.3%	TAAGAG	6.7%	GTCTAA	6.4%
ACGCGT	6.2%	GTCATA	6.6%	GATCAG	6.4%
TTATAC	6.2%	ATCTAA	6.6%	GATCGA	6.4%
GTGCAG	6.2%	TATACA	6.6%	ACGATC	6.4%

Error rates were calculated for 6 bp long sequence on the reference sequence in the alignment of real reads, while shifting the 6 bp long sequence by 1 bp from end to end. Next, the averaged error rate of each 6-mer was calculated.

Table 3.6. Top 30 6-mer with high rate of deletion

C.elegans P6–C4		*R.sphaeroides* R9.5		*E-coli* K12 R10.3	
6-mer	rate	6-mer	rate	6-mer	rate
GGGCCC	9.6%	GGGGGG	22.9%	GGGGGG	27.7%
GGGGCC	8.9%	CCCCCC	22.7%	CCCCCC	27.3%
GCCCCC	8.6%	TTTTTT	18.9%	TCCCCC	20.7%
AGGGGG	8.6%	AGGGGG	18.3%	GGGGGA	20.0%
CCCGGG	8.4%	CCCCCT	18.3%	CCCCCT	19.7%
CCCCCG	8.3%	TCCCCC	17.5%	AGGGGG	19.1%
GGGGGT	8.2%	AAAAAA	17.3%	CCCCCA	18.9%
CCCCCA	8.2%	CTTTTT	17.2%	CCCCCG	18.2%
CCCCCT	8.0%	GGGGGA	16.7%	TGGGGG	18.1%
TGGGGG	8.0%	AAAAAG	14.9%	CGGGGG	17.7%
CCGGGG	8.0%	AAGGGG	14.7%	GGGGGC	17.6%
GGGGAC	8.0%	CCCCTT	14.7%	ACCCCC	17.3%
ACCCCC	8.0%	CCTTTT	14.5%	GCCCCC	17.2%
CCCCCC	7.9%	GGGGGT	14.4%	GGGGGT	17.2%
GCCGGG	7.9%	CGGGGG	14.4%	TTCCCC	14.6%
AGGGCC	7.8%	TTAAGC	14.3%	GGGGAA	14.6%
CCCCGG	7.8%	CTCCCC	14.2%	CCCCGA	14.5%
GGCCCC	7.8%	CCCCCG	14.2%	CAGGGG	14.0%
GGGGTT	7.8%	ACCCCC	14.1%	CCCCTC	14.0%
GGGGGA	7.7%	TAAAAA	13.9%	CCGGGG	13.8%
GGGGGC	7.7%	AGGGGA	13.9%	CCCCAG	13.7%
GAGGGG	7.7%	CTCTCT	13.8%	TCGGGG	13.6%
GGCCCT	7.6%	TAAGGG	13.8%	CGGGGA	13.5%
CCGGCC	7.5%	CTTAAG	13.7%	AAAAAA	13.5%
GGCCCA	7.5%	CCCCTC	13.7%	TGCCCC	13.4%
TCCCCC	7.4%	CAAAAA	13.7%	GGGGAG	13.4%
GCCCGG	7.4%	GCCCCC	13.6%	GGGGTA	13.4%
AGGGGC	7.4%	AAAAGG	13.6%	TCCCCG	13.2%
GGGGGG	7.3%	TCCCCT	13.5%	GGGGCG	13.2%
GGGCCT	7.3%	TAAAAG	13.5%	CCCCGG	13.2%

Error rates were calculated the same as in Table 3.5.

Table 3.7. Top 30 6-mer with high rate of substitution

C.elegans P6–C4		*R.sphaeroides* R9.5		*E-coli* K12 R10.3	
6-mer	rate	6-mer	rate	6-mer	rate
GTCTGG	2.7%	CGAATC	10.8%	GATCTA	14.2%
GGGGGG	2.3%	GATTCG	10.4%	TAGATC	14.1%
AGTCTG	2.1%	ACTAGG	9.0%	CTACTA	12.8%
ATTGGG	2.1%	GTCTAG	8.5%	CAGATC	12.8%
ACGGCC	2.0%	TGTCTA	8.1%	TCGATC	12.5%
CGGGGG	2.0%	TACTAG	7.7%	GATCGA	12.3%
GGGGGT	1.9%	GACTAG	7.6%	CTAGAC	12.3%
ACCCCC	1.9%	CTAGGC	7.5%	TCTAGT	12.2%
CCCCCC	1.9%	GCCTAG	7.4%	TCTAGG	12.2%
TTGGGA	1.8%	CTAGAT	7.3%	GATCAA	12.2%
CCCCCT	1.7%	GAATCG	7.3%	CTAGTA	12.2%
GTGGGG	1.7%	CTAGTA	7.3%	CCGATC	12.2%
TGGGGG	1.7%	CGATTC	6.9%	CTAGTG	12.1%
CCCCCA	1.7%	GAATCC	6.9%	CGATCA	12.0%
CACCCC	1.6%	GGATTC	6.8%	GATCTG	11.9%
GGTACC	1.6%	AAGTTA	6.7%	TAGTAT	11.9%
AGGGGT	1.6%	AACTAG	6.7%	TACTAG	11.9%
CAGGGG	1.6%	GAATCT	6.7%	CGATCG	11.8%
GGGGTG	1.6%	GCTAGC	6.7%	GTCTAG	11.6%
GGGGTT	1.5%	TCTAGG	6.6%	GATCGG	11.6%
AGGGAC	1.5%	CTAAGT	6.5%	TGATCA	11.6%
CCCCTA	1.5%	AGATTC	6.5%	TGATCG	11.5%
CTAGGG	1.5%	CTAGCC	6.5%	TTGATC	11.5%
TGGGAA	1.5%	GTCTAC	6.4%	TAGTAG	11.3%
GGCCAG	1.5%	ACTGTA	6.3%	ACTAGA	11.2%
CCCCTG	1.5%	CTAATG	6.3%	TATCTA	11.2%
CGGCCA	1.5%	CTGTAG	6.3%	AGATCA	11.1%
GGGGTA	1.5%	GTCTAA	6.3%	ACTAGT	11.1%
ACCCCT	1.5%	CCTAGG	6.2%	CGATCT	11.0%
CCGGTC	1.4%	CAGACT	6.2%	AGATCG	11.0%

Error rates were calculated the same as in Table 3.5.

Chapter 4

Conclusions and future directions

4.1 Conclusions

In this thesis, I have developed two simulators for long read sequences to understand the characteristics of long read sequences , and to generate simulation data that accurately imitate the characteristics, which is useful for developing tools/algorithms to analyze long reads, designing sequencing experiences.

In chapter 2, my analysis of 13 PacBio datasets showed characteristic features of PacBio reads (e.g. the read length of PacBio reads follows a log-normal distribution). I have developed a read simulator, PBSIM, that captures these features using either a model-based or sampling-based method. Using PBSIM, I conducted several hybrid error correction and genome assembly tests for PacBio reads, suggesting that a continuous long reads coverage depth of at least 15 in combination with a circular consensus sequencing coverage depth of at least 30 achieved extensive assembly results.

In chapter 3, to capture characteristics of errors in reads for long read sequencers more precisely, especially to simulate the non-uniformity of quality scores, I developed a generative model for quality scores, based on a hidden Markov Model in combination with latest model selection criteria. My computational experiments show that PBSIM2, the new version of PBSIM, simulates quality scores that are more consistent with real reads of PacBio and Nanopore than other existing simulators. In addition, I improved the correlation between read length and accuracy, and the relationship between error rate and quality scores, both of which PBSIM was unable to simulate properly.

4.2 Future direction

PBSIM2 can generate better simulation data by accurately simulating low quality region. However, there are other artifacts such as chimeras and adapter sequences, which are frequently observed in long reads (see Myers ' report, https://dazzlerblog.wordpress.com/2017/04/22/1344/). These errors are the major cause of poor genome assembly. Badread [57] has previously simulated these errors, and I also plan to implement similar functions in the next version of PBSIM.

After PacBio Sequel sequencer, quality code is a fixed value and does not represent the actual error rate, so in this thesis, only RS II CLR was used as training data for quality scores. PBSIM2 is targeted at error-prone reads, so I am unsure if it can properly simulate HiFi reads. However, if a generative model of quality scores is created using the error information obtained from the alignment of reads to the reference sequences instead of the quality score, the latest PacBio Sequel II data can be used as the training data of FIC-HMM. Even though there are many problems, such as handling unaligned regions or regions where it is difficult to obtain accurate error information, including low quality regions, learning alignment by FIC-HMM is expected to significantly improve the error model of long reads.

Both PBSIM and PBSIM2 were developed for the purpose of simulating genomic sequences. By replacing genome sequence with transcriptome sequence as input data, it is possible to simulate long reads derived from the transcriptome, but it is uncertain whether the simulated read imitate real reads accurately. The demand for simulation of long read for transcriptome sequencing is increasing, but few simulators have been developed for that purpose [67]. The transcriptome is very complicated because the gene expression levels change depending on the cell type, developmental stage, and external factors, and in addition, the alternative splicing generates multiple isoforms of gene transcripts [68]. Therefore, many tools/algorithms for analyzing transcriptome have been developed [69]. I would like to develop a long read simulator for transcriptome sequencing to evaluate and improve these tools/algorithms. On the other hand, in the field of genome assembly, it is also undergoing rapid development of tools/algorithms, especially for haplotype-resolved or phased genome assembly, which provides a complete picture of genomes and their complex genetic variations [70]. I would also like to add a polypoid simulation function to PBSIM2 to evaluate and improve these tools/algorithms.

References

[1] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333, 2016.

[2] Suying Bao, Rui Jiang, WingKeung Kwan, BinBin Wang, Xu Ma, and You-Qiang Song. Evaluation of next-generation sequencing software in mapping and assembly. *Journal of human genetics*, 56(6):406–414, 2011.

[3] Stephan Pabinger, Andreas Dander, Maria Fischer, Rene Snajder, Michael Sperk, Mirjana Efremova, Birgit Krabichler, Michael R Speicher, Johannes Zschocke, and Zlatko Trajanoski. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in bioinformatics*, 15(2):256–278, 2014.

[4] Ana Conesa, Pedro Madrigal, Sonia Tarazona, David Gomez-Cabrero, Alejandra Cervera, Andrew McPherson, Michał Wojciech Szcześniak, Daniel J Gaffney, Laura L Elo, Xuegong Zhang, et al. A survey of best practices for rna-seq data analysis. *Genome biology*, 17(1):13, 2016.

[5] Jay Shendure, Shankar Balasubramanian, George M Church, Walter Gilbert, Jane Rogers, Jeffery A Schloss, and Robert H Waterston. Dna sequencing at 40: past, present and future. *Nature*, 550(7676):345–353, 2017.

[6] Michael L Metzker. Emerging technologies in dna sequencing. *Genome research*, 15(12):1767–1776, 2005.

[7] Michael C Schatz, Arthur L Delcher, and Steven L Salzberg. Assembly of large genomes using second-generation sequencing. *Genome research*, 20(9):1165–1173, 2010.

[8] Todd J Treangen and Steven L Salzberg. Repetitive dna and next-generation sequencing: computational challenges and solutions. *Nature Reviews Genetics*, 13(1):36–46, 2012.

[9] Juliane C Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput dna sequencing. *Nucleic acids research*, 36(16):e105, 2008.

[10] Merly Escalona, Sara Rocha, and David Posada. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature Reviews Genetics*, 17(8):459, 2016.

[11] Michael A Quail, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. *BMC genomics*, 13(1):1–13, 2012.

[12] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome biology*, 14(5):R51, 2013.

[13] David Sims, Ian Sudbery, Nicholas E Ilott, Andreas Heger, and Chris P Ponting. Sequencing depth and coverage: key considerations in genomic analyses. *Nature Reviews Genetics*, 15(2):121–132, 2014.

[14] Erwin L van Dijk, Yan Jaszczyszyn, Delphine Naquin, and Claude Thermes. The third revolution in sequencing technology. *Trends in Genetics*, 34(9):666–681, 2018.

[15] Anthony Rhoads and Kin Fai Au. Pacbio sequencing and its applications. *Genomics, proteomics & bioinformatics*, 13(5):278–289, 2015.

[16] Miten Jain, Hugh E Olsen, Benedict Paten, and Mark Akeson. The oxford nanopore minion: delivery of nanopore sequencing to the genomics community. *Genome biology*, 17(1):239, 2016.

[17] Richard J Roberts, Mauricio O Carneiro, and Michael C Schatz. The advantages of smrt sequencing. *Genome biology*, 14(7):1–4, 2013.

[18] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, et al. Real-time dna sequencing from single polymerase molecules. *Science*, 323(5910):133–138, 2009.

[19] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology*, 36(4):338, 2018.

[20] Sergey Koren, Michael C Schatz, Brian P Walenz, Jeffrey Martin, Jason T Howard, Ganeshkumar Ganapathy, Zhong Wang, David A Rasko, W Richard McCombie, Erich D Jarvis, et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature biotechnology*, 30(7):693–700, 2012.

[21] Chen-Shan Chin, David H Alexander, Patrick Marks, Aaron A Klammer, James Drake, Cheryl Heiner, Alicia Clum, Alex Copeland, John Huddleston, Evan E Eichler, et al. Nonhybrid, finished microbial genome assemblies from long-read smrt sequencing data. *Nature methods*, 10(6):563–569, 2013.

[22] Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, 19(6):329–346, 2018.

[23] Wojciech Makałowski and Victoria Shabardina. Bioinformatics of nanopore sequencing. *Journal of human genetics*, pages 1–7, 2019.

[24] Shanika L Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E Ritchie, and Quentin Gouil. Opportunities and challenges in long-read sequencing data analysis. *Genome biology*, 21(1):1–16, 2020.

[25] Bianca K Stöcker, Johannes Köster, and Sven Rahmann. Simlord: simulation of long read data. *Bioinformatics*, 32(17):2704–2706, 2016.

[26] Christian Rohrandt, Nadine Kraft, Pay Gießelmann, Björn Brändl, Bernhard M Schuldt, Ulrich Jetzek, and Franz-Josef Müller. Nanopore simulation–a raw data simulator for nanopore sequencing. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1–8. IEEE, 2018.

[27] Yu Li, Renmin Han, Chongwei Bi, Mo Li, Sheng Wang, and Xin Gao. Deepsimulator: a deep simulator for nanopore sequencing. *Bioinformatics*, 34(17):2899–2908, 2018.

[28] Weichun Huang, Leping Li, Jason R Myers, and Gabor T Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[29] Bayo Lau, Marghoob Mohiyuddin, John C Mu, Li Tai Fang, Narges Bani Asadi, Carolina Dallett, and Hugo YK Lam. Longislnd: in silico sequencing of lengthy and noisy datatypes. *Bioinformatics*, 32(24):3829–3832, 2016.

[30] Chen Yang, Justin Chu, René L Warren, and Inanç Birol. Nanosim: nanopore sequence read simulator based on statistical characterization. *GigaScience*, 6(4):gix010, 2017.

[31] Yukiteru Ono, Kiyoshi Asai, and Michiaki Hamada. Pbsim: Pacbio reads simulator–toward accurate genome assembly. *Bioinformatics*, 29(1):119–121, 2013.

[32] Yukiteru Ono, Kiyoshi Asai, and Michiaki Hamada. Pbsim2: a simulator for long-read sequencers with a novel generative model of quality scores. *Bioinformatics*, 2020.

[33] Ryohei Fujimaki and Kohei Hayashi. Factorized asymptotic bayesian hidden markov models. *arXiv preprint arXiv:1206.4679*, 2012.

[34] Michiaki Hamada, Yukiteru Ono, Ryohei Fujimaki, and Kiyoshi Asai. Learning chromatin states with factorized information criteria. *Bioinformatics*, 31(15):2426–2433, 2015.

[35] Chen-Shan Chin, Jon Sorenson, Jason B Harris, William P Robins, Richelle C Charles, Roger R Jean-Charles, James Bullard, Dale R Webster, Andrew Kasarskis, Paul Peluso, et al. The origin of the haitian cholera outbreak strain. *New England Journal of Medicine*, 364(1):33–42, 2011.

[36] Xuesong Hu, Jianying Yuan, Yujian Shi, Jianliang Lu, Binghang Liu, Zhenyu Li, Yanxiang Chen, Desheng Mu, Hao Zhang, Nan Li, et al. pirs: Profile-based illumina pair-end reads simulator. *Bioinformatics*, 28(11):1533–1535, 2012.

[37] Florent E Angly, Dana Willner, Forest Rohwer, Philip Hugenholtz, and Gene W Tyson. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research*, 40(12):e94–e94, 2012.

[38] Susanne Balzer, Ketil Malde, Anders Lanzén, Animesh Sharma, and Inge Jonassen. Characteristics of 454 pyrosequencing data–enabling realistic simulation with flowsim. *Bioinformatics*, 26(18):i420–i425, 2010.

[39] Daniel C Richter, Felix Ott, Alexander F Auch, Ramona Schmid, and Daniel H Huson. Metasim –a sequencing simulator for genomics and metagenomics. *PloS one*, 3(10):e3373, 2008.

[40] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and samtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[41] David A Rasko, Dale R Webster, Jason W Sahl, Ali Bashir, Nadia Boisen, Flemming Scheutz, Ellen E Paxinos, Robert Sebra, Chen-Shan Chin, Dimitris Iliopoulos, et al. Origins of the e. coli strain causing an outbreak of hemolytic–uremic syndrome in germany. *New England Journal of Medicine*, 365(8):709–717, 2011.

[42] Martin C Frith, Michiaki Hamada, and Paul Horton. Parameters for accurate genome alignment. *BMC bioinformatics*, 11(1):80, 2010.

[43] Szymon M Kiełbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C Frith. Adaptive seeds tame genomic sequence comparison. *Genome research*, 21(3):487–493, 2011.

[44] Mauricio O Carneiro, Carsten Russ, Michael G Ross, Stacey B Gabriel, Chad Nusbaum, and Mark A DePristo. Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC genomics*, 13(1):375, 2012.

[45] Peter JA Cock, Christopher J Fields, Naohisa Goto, Michael L Heuer, and Peter M Rice. The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic acids research*, 38(6):1767–1771, 2010.

[46] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

[47] Mark JP Chaisson, John Huddleston, Megan Y Dennis, Peter H Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, et al. Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, 517(7536):608–611, 2015.

[48] Jonas Korlach, Gregory Gedman, Sarah Kingan, Jason Chin, Jason Howard, Lindsey Cantin, and Erich D Jarvis. De novo pacbio long-read and phased avian genome assemblies correct and add to genes important in neuroscience research. *BioRxiv*, page 103911, 2017.

[49] Rory Bowden, Robert W Davies, Andreas Heger, Alistair T Pagnamenta, Mariateresa de Cesare, Laura E Oikkonen, Duncan Parkes, Colin Freeman, Fatima Dhalla, Smita Y Patel, et al. Sequencing of human genomes with nanopore technology. *Nature communications*, 10(1):1–9, 2019.

[50] Alexander T Dilthey, Chirag Jain, Sergey Koren, and Adam M Phillippy. Strain-level metagenomic assignment and compositional estimation for long reads with metamaps. *Nature communications*, 10(1):1–12, 2019.

[51] Jared T Simpson, Rachael E Workman, PC Zuzarte, Matei David, LJ Dursi, and Winston Timp. Detecting dna cytosine methylation using nanopore sequencing. *Nature methods*, 14(4):407, 2017.

[52] Jason L Weirather, Mariateresa de Cesare, Yunhao Wang, Paolo Piazza, Vittorio Sebastiano, Xiu-Jie Wang, David Buck, and Kin Fai Au. Comprehensive comparison of pacific biosciences and oxford nanopore technologies and their applications to transcriptome analysis. *F1000Research*, 6, 2017.

[53] Tuomo Mantere, Simone Kersten, and Alexander Hoischen. Long-read sequencing emerging in medical genetics. *Frontiers in genetics*, 10:426, 2019.

[54] Shatha Alosaimi, Armand Bandiang, Noelle van Biljon, Denis Awany, Prisca K Thami, Milaine SS Tchamga, Anmol Kiran, Olfa Messaoud, Radia Ismaeel Mohammed Hassan, Jacquiline Mugo, et al. A broad survey of dna sequence data simulation tools. *Briefings in Functional Genomics*, 19(1):49–59, 2020.

[55] David Laehnemann, Arndt Borkhardt, and Alice Carolyn McHardy. Denoising dna deep sequencing data–high-throughput sequencing errors and their correction. *Briefings in bioinformatics*, 17(1):154–179, 2016.

[56] Wenmin Zhang, Ben Jia, and Chaochun Wei. Pass: a sequencing simulator for pacbio sequencing. *BMC bioinformatics*, 20(1):1–7, 2019.

[57] Ryan Wick. Badread: simulation of error-prone long reads. *Journal of Open Source Software*, 4(36):1316, 2019.

[58] Hayan Lee, James Gurtowski, Shinjae Yoo, Shoshana Marcus, W Richard McCombie, and Michael Schatz. Error correction and assembly complexity of single molecule sequencing reads. *BioRxiv*, page 006395, 2014.

[59] Ethan Alexander Garcia Baker, Sara Goodwin, W Richard McCombie, and Olivia Mendivil Ramos. Silico: a simulator of long read sequencing in pacbio and oxford nanopore. *BioRxiv*, page 076901, 2016.

[60] Philippe C Faucon, Parithi Balachandran, and Sharon Crook. Snaresim: Synthetic nanopore read simulator. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 338–344. IEEE, 2017.

[61] Ze-Gang Wei and Shao-Wu Zhang. Npbss: a new pacbio sequencing simulator for generating the continuous long reads with an empirical model. *BMC bioinformatics*, 19(1):177, 2018.

[62] Yu Li, Sheng Wang, Chongwei Bi, Zhaowen Qiu, Mo Li, and Xin Gao. Deepsimulator1. 5: a more powerful, quicker and lighter simulator for nanopore sequencing. *Bioinformatics*, 2020.

[63] Michiaki Hamada, Yukiteru Ono, Kiyoshi Asai, and Martin C Frith. Training alignment parameters for arbitrary sequencers with last-train. *Bioinformatics*, 33(6):926–928, 2017.

[64] Byung-Jun Yoon. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6):402–415, 2009.

[65] Franka J Rang, Wigard P Kloosterman, and Jeroen de Ridder. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome biology*, 19(1):90, 2018.

[66] Ryan R Wick, Louise M Judd, and Kathryn E Holt. Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome biology*, 20(1):129, 2019.

[67] Saber Hafezqorani, Chen Yang, Theodora Lo, Ka Ming Nip, René L Warren, and Inanc Birol. Trans-nanosim characterizes and simulates nanopore rna-sequencing data. *GigaScience*, 9(6):giaa061, 2020.

[68] Bo Wang, Vivek Kumar, Andrew Olson, and Doreen Ware. Reviving the transcriptome studies: an insight into the emergence of single-molecule transcriptome sequencing. *Frontiers in genetics*, 10:384, 2019.

[69] Biswapriya B Misra, Carl Langefeld, Michael Olivier, and Laura A Cox. Integrated omics: tools, advances and future approaches. *Journal of molecular endocrinology*, 62(1):R21–R45, 2019.

[70] Xingtan Zhang, Ruoxi Wu, Yibin Wang, Jiaxin Yu, and Haibao Tang. Unzipping haplotypes in diploid and polyploid genomes. *Computational and structural biotechnology journal*, 18:66–72, 2020.