SNaReSim: Synthetic Nanopore Read Simulator

Philippe Faucon
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, AZ 85282-9709, USA
pfaucon@asu.edu

Parithi Balachandran
School of Biological and Health Systems Engineering
Arizona State University
Tempe, AZ 85287-9709, USA
pbalach1@asu.edu

Sharon Crook
School of Mathematical and Statistical Sciences
Arizona State University
Tempe, AZ 85287-9709, USA
Sharon.Crook@asu.edu

***Abstract*—Nanopores represent the first commercial technology in decades to present a significantly different technique for DNA sequencing, and one of the first technologies to propose direct RNA sequencing. Despite significant differences with previous sequencing technologies, read simulators to date make similar assumptions with respect to error profiles and their analysis. This is a great disservice to both nanopore sequencing and to computer scientists who seek to optimize their tools for the platform. Previous works have discussed the occurrence of some k-mer bias, but this discussion has been focused on homopolymers, leaving unanswered the question of whether k-mer bias exists over general k-mers, how it occurs, and what can be done to reduce the effects. In this work, we demonstrate that current read simulators fail to accurately represent k-mer error distributions, We explore the sources of k-mer bias in nanopore basecalls, and we present a model for predicting k-mers that are difficult to identify. We also propose a new SNaReSim, a new state-of-the-art simulator, and demonstrate that it provides higher accuracy with respect to 6-mer accuracy biases.**

I. Introduction

DNA sequencing has become an integral component of biological research, with applications ranging from gene network or organism identification through biological engineering. DNA sequencing has historically been dominated by synthesis-based approaches involving the replication of an existing DNA or cDNA molecule, with the attachment of fluorescent probes during synthesis for visualization of the base being added in a 4-dimensional color space. These approaches tend to have a bias in the reads selected, and tend to have reduced accuracy near the beginning and end of reads, however there tends to be minimal error bias within a single read [1]. Conversely, nanopore reads escape the read selection bias due to a lack of pre-amplification, but they provide only a single dimension of measurement, resulting in a significant within-read bias [2]. This bias is unaccounted for and lacks a published model, but it has significant implications for current generation read aligners [3]–[5] which rely on perfect or nearly perfect "seed" sequences. Unfortunately, this behavior is unaccounted for in current generation read simulators.

We demonstrate that in fact there is a read bias, that is more pronounced than the previously observed homopolymer bias, and that current generation simulators are unable to properly model this distribution. We propose a simulator based on a modified Markov chain that allows for the mutation of a reference sequence in a way that significantly improves simulation fidelity. To accomplish this task, we identify the accuracy of individual k-mers and their behavior in different contexts. We also predict key features of the error bias, and calculate their individual contributions to the final error. Our simulator is fully automated, allowing for both in silico amplification of read data, or simulation of data on completely novel genomes. Our contributions are summarized below:

- We demonstrate that while some base calling error is random there is a significant component that is systematic and unaccounted for (section IV-A).
- We develop a set of features for predicting k-mer accuracy and show that our model generally correlates well with data found empirically (section IV-B).
- We propose an algorithm for simulating reads, a variant of the Hidden Markov Model employed by Nanosim [6], with modifications to apply observed k-mer bias. We demonstrate that it models the k-mer error distribution better than other popular read simulators [6]–[8] (section IV-C).

II. Related Work

Cost, throughput, and accuracy have been major hindrances in DNA sequencing. With the development of NGS technologies cost has reduced while throughput and accuracy continue to climb. Still, simulators offer a significant benefit due to their low cost and exceptionally high throughput, allowing testing while developing new algorithms. Simulators aim to produce sequences with the most fidelity possible for their given platform. As such, simulated reads should account for biological and technical bias [9].

Simulators typically generate synthetic reads by extracting a sequence from a reference genome and then introducing errors into that sequence. Parameters required by simulators to introduce these features into the sequence are either provided at run time or are stored inside a metadata called a model or error profile. By analyzing the alignment of empirical data to a reference genome error profiles are created. Errors have been generated by first predicting a quality score [10], by base position within a simulated read [11], or by predicting an error sequence and then applying it to a read [6], [7], [12].

Modeling of third generation single-molecule reads has many advantages when compared with second generation sequencers. Polymerase chain reaction(PCR), required by second

generation sequencers during the pre-amplification step, introduces significant bias, but is not required for third generation sequencers [13]. GC bias, a secondary effect of the PCR amplification step, resulting in low base accuracy and high coverage variability, is also removed [14]. Third generation sequencing on the other hand has new challenges that must be modeled. Simulators for third generation sequencers must deal with longer reads containing significantly larger stretches of errors, and a biased error that varies with nearby nucleotides. There are two main platforms for third generation sequencing, PacBio's SMRT sequencing, and Oxford Nanopore's nanopore sequencing. Each platform has their own read simulators, but none are unable to properly model the k-mer bias of nanopore sequencing data.

NanoSim [6] is a read simulator for ONT data, modeling reads as the result of a Hidden Markov model. The model is fit by aligning empirical reads to a reference genome, then collecting a list of error subtypes, lengths, and transition probabilities. Reads can be simulated by sampling a length, then generating a sequence of errors. One major limitation of this approach is that it is unable to properly model k-mer bias; this is somewhat overcome by a post-processing step where all homopolymers of length greater than 5 (e.g. "AAAAAAA") are compressed to a length of 5.

LongISLND [8] is a read simulator developed for PacBio data. It models k-mer bias explicitly and directly by storing observed mutations for each k-mer it sees, stored as a key-value pair. It also uses an Extended K-mer (EKmer) model to aid in homopolymer error generation. An EKmer is a regular k-mer followed by an integer representing a length of homopolymer covering the middle term in the k-mer facilitating the simulation of arbitrary stretches of homopolymer without explicitly observing them. One major limitation of LongISLND is that while is is able to approximate the k-mer bias it lacks the accuracy level observed in true ONT data. This is likely due to ONT data having grouped errors (i.e. the probability of error given an error exists nearby is higher than normal) that are not properly modeled using their simulator.

III. Problem Description

DNA has a label space of $\Sigma \in \{A, C, T, G\}$ representing the different nucleotide bases. Let $s \in \Sigma^n$ be a DNA strand of length n. Using Nanopore technology, a series of discrete measurements δ is generated, representing the current that passes through the nanopore at each time step from time t_0 to t_T, where t_T reflects the total time required for s to completely transit the Nanopore. This creates a corresponding vector $\Delta \in R^d$, where $d >> n$ is the number of discrete measurements obtained from time t_0 to t_T.

The vector Δ is subsequently binned into q different bins, $B = b_1, b_2 ... b_q$, using different time intervals that capture $\approx \frac{d}{n}$ measurements per bin. The mean μ_i and standard deviation σ_i are subsequently derived for each bin b_i. An approximation strand, $\hat{s}$, of the original strand s is reconstructed or "base called" using the sequence of (μ_i, $sigma_i$) pairs using either a Recurrent Neural Network [15], or a Hidden Markov Model [16], [17] in conjunction with a lookup table from the Nanopore manufacturer that specifies the most probable k-mer base pair for the given μ_i value. Using the R9 pore from Oxford Nanopore μ_i is best explained by a sequence of 6 DNA bases, thus the k-mer length $k = 6$ is used in later analysis.

A. Error Source Identification

Naively, one may assume that error is distributed evenly over $\hat{s}$, however it is trivial to see that it is not the case. The expected mean current μ exists on a linear range, thus k-mers near the minimum current are not as influenced by δ readings lower than their expected mean, with the opposite true for readings near the top of the range. Second, if the density range of the k-mer currents is uneven, more error is to be expected in high density regions [2]. Third, some sequences of K-mers are easily identifiable due to large changes in current, while others are difficult to identify due to small changes [2].

To elucidate the drivers of k-mer bias we first calculate a list of k-mer accuracies K, and propose a set of features F, where each feature f_i confers an increase or decrease to the accuracy of each k-mer. Error cannot be negative, thus we can approximate the influence of each feature by solving Equation 1.

$$\begin{aligned} \arg\min_{x} \quad & \|Fx - B\|_2 \\ \text{subject to} \quad & x \geq 0 \end{aligned} \tag{1}$$

Where x is a vector indicating the contribution of each factor F. Fx is then the best approximation of the original k-mer bias with respect to euclidean distance.

B. Read Simulation

Read simulation occurs in 2 phases: model fitting, then simulating reads from an input genome. Fitting the model requires existing reads to be aligned to a reference genome, alignment is performed using BWA-MEM [4] with standard options. From each read, the top alignment is selected, and unaligned reads are discarded. From the remaining reads, parameters are extracted with respect to average error length for the error types {Insertion, Deletion, Mismatch}, transition probabilities, and the accuracy of all k-mers of length 6. The inverse of the k-mer accuracy is also used as a "cost" of correctly predicting a k-mer. These parameters are then stored for downstream simulation.

Reads are simulated according to a hidden-markov model (HMM) which generates transitions between correct and erroneous stretches, and the lengths of those stretches. This approach has the benefit of easy explainability, but has limitations in that it is unable to correctly model k-mer accuracy distribution. Moreover, this shortfall is difficult to overcome, as modeling k-mer accuracies as states would require an intractable number of parameters. To remedy this situation we allow errors and error lengths to be generated according to the HMM, but we adjust the lengths by using a cost function described above in a process detailed in Algorithm 1.

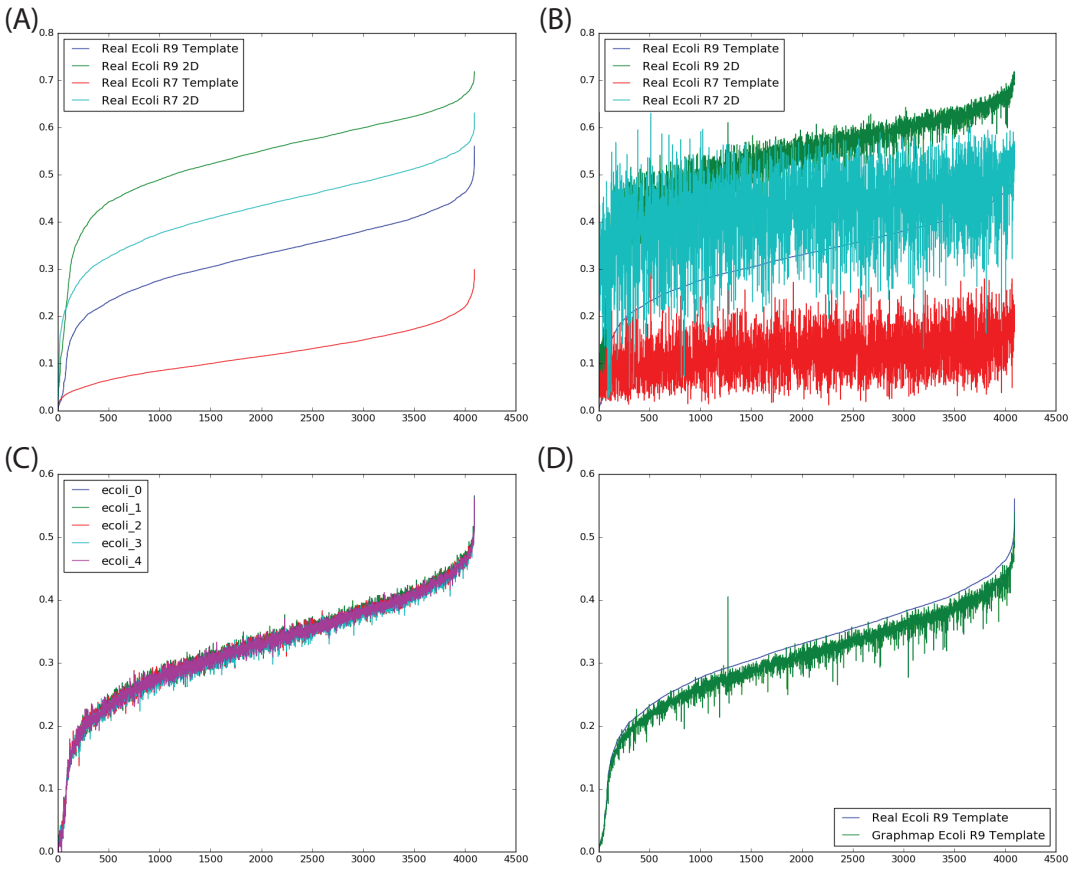

Fig. 1: The accuracies of all 6-mers with: (A) Experiments all 6-mers sorted by accuracy with reference to themselves (B) Experiments sorted with respect to Ecoli R9 2D (C) A single experiment divided randomly into 5 groups, sorted with respect to the first group (D) only R9 ecoli template results when comparing between aligners

Data: read extracted from sample genome
Result: mutated read
while $readPosition < readlength$ **do**
  scaleFactor = mean k-mer cost around readPosition;
  isError = $random * scaleFactor < .5$;
  if *isError* **then**
    errorType = transition probability from HMM;
  end
  budget = length from model based on errorType;
  while $cost < budget$ **do**
    cost += cost of k-mer at readPosition + i;
    i += 1;
  end
  if *isError* **then**
    save the mutation to a mutation list
  end
  readPosition += i;
end

Algorithm 1: Generation of mutation list for sampled read using the k-mer biased HMM model.

IV. Results

A. Modeling K-mer Bias

Before attempting to model k-mer bias in real data is it important to understand the granularity at which it occurs, and the consistency. To this end we examined 2 public datasets [1] to determine whether bias was present, and to what extent. We show that while there is a consistent k-mer bias within experiments, there are significant differences between the R7 and R9 pores in Figure 1. We attribute the majority of the differences to the pores themselves, but some influence also likely comes from different versions of ONT basecaller used on each dataset.

B. Identifying Bias Sources

After identifying the presence of k-mer bias we attempted to elucidate the source of the bias. To this end we generated a set of features with each providing a score for each k-mer, and we attempted to find a linear combination of each feature capable of explaining the bias, and providing a fractional

[1] {http://lab.loman.net/2016/07/30/nanopore-r9-data-release/,http://lab.loman.net/2014/10/01/where-can-i-get-oxford-nanopore-miniontm-data-from/}

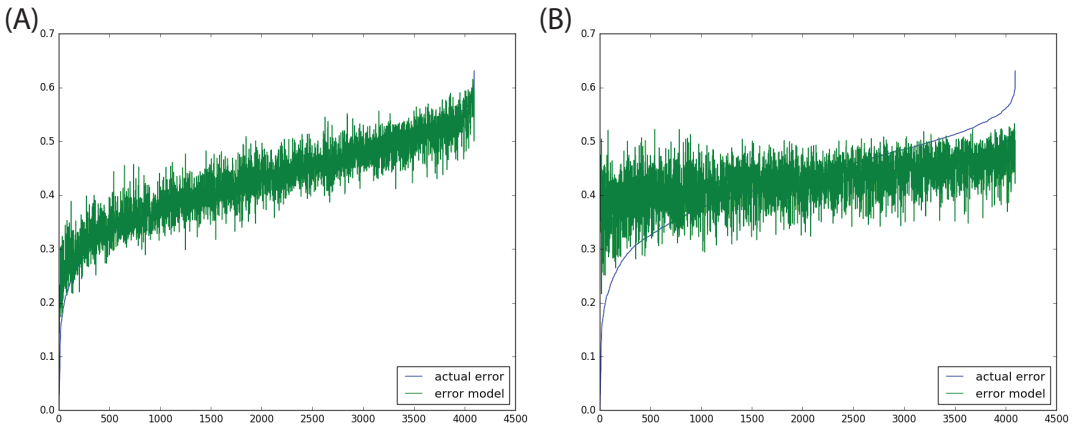

Fig. 2: The accuracy of k-mer fit to the actual distribution from the R9 pore model. (A) Including the true accuracy of neighboring k-mers as a feature (B) using only the features in Table I

Bias Source	Example Low	Example High	Contribution
Transition Identifiability	AAAAAA	ATATAT	64.6%
Sequence Identifiability	CTGTCA	TATATT	3.32%
Standard Deviation	CTAGAG	TTGAAA	1.79%
Fraction A	CGTCGT	AAAAAA	9.73%
Fraction T	CGACGA	TTTTTT	2.90%
Fraction C	AGTAGT	CCCCCC	9.33%
Fraction G	CATCAT	GGGGGG	7.03%

TABLE I: Identified error sources and their contribution to k-mer overall accuracy with respect to Figure 2

contribution of each step. As shown in Table I multiple error sources influence the accuracy fraction of each k-mer. We find that the strongest feature for predicting the accuracy is the median accuracy of neighbors at one step away. While this is not directly helpful, it does suggest that direct modeling of error sources must incorporate neighbor accuracy aspects. Overall we find that our predictive model provides a good first attempt, providing some insight, but leaving much of the error signal uncaptured, as illustrated in Figure 2. We expect that much of the remaining error signal is a result of missing features that may be identified through a more thorough analysis. We also expect that some of the error signal cannot be captured through linear combinations. For example, the original basecallers were incapable of capturing homopolymers with a length greater than 5, using linear combinations the k-mer "AAAAAA" would need to have 0 accuracy across all features, an unrealistic expectation.

C. Simulation Results

To validate our simulation results we generated read profiles for Nanosim [6], PBSim [7], LONGIslnd [8], and the two simulators we have proposed. Nanopore sequencing experiments typically yield between 10,000 and 20,000 reads [6], with measured statistics being approximately identical at even 20% of this size as shown in Figure 1(C). To measure this effect in simulations, we generated 2 data sets with each simulator: one

Source	Sample Size	SSE
Split R9	6,481	0.121
LongISLND R9	20,000	167.04
LongISLND R9	4,000	167.80
Nanosim R9	20,000	44.26
Nanosim R9	4,000	46.76
SNRG R9	20,000	9.52
SNRG R9	4,000	9.66
LongISLND R7	20,000	88.78
LongISLND R7	4,000	88.96
Nanosim R7	20,000	30.76
Nanosim R7	4,000	36.16
SNRG R7	20,000	5.24
SNRG R7	4,000	5.48

TABLE II: Sum of squares error(SSE) between k-mer simulators and their training, and between the fragmented R9 data set and the complete one(Figure 1)

with 4,000 reads, and one with 20,000 reads. These reads were then aligned using BWA-MEM [4] with standard parameters. The sum of squares error is the sum of the difference between the model 6-mer accuracy and the simulated k-mer accuracy, for all 6-mers. Results of both large and small simulations are shown in Table II, while the 20,000 read simulations are shown in Figure 3. Ultimately we find that sample size has very little influence on existing simulators, as the k-mer bias is either completely un-modeled, or has systematic difficulties capturing the k-mer error rate (LongISLND). In our simulators we find some improvement through increased sample size, though most of the remaining difference does appear to be from systematic errors.

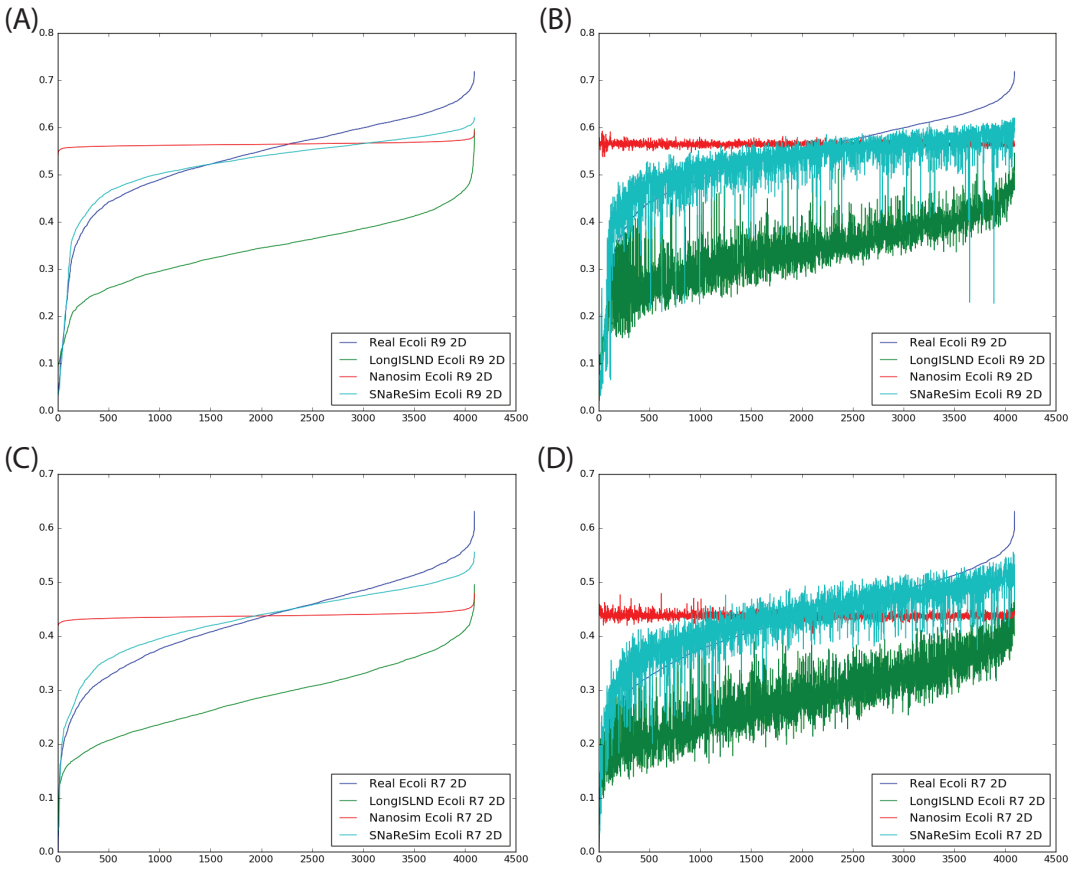

Fig. 3: Comparison of 3 previously published sequencing simulators with the new model we proposed. Fit of R9 pore model k-mer accuracies sorted internally(A), and by 2D Ecoli, the training model(B). Fit of R7 pore model k-mer accuracies sorted internally(C), and by 2D ecoli, the training model (D).

V. Conclusion

We have demonstrated the presence of biased k-mer accuracy within Oxford Nanopore Technologies sequencing platform. We show that this bias is consistent within experiments, and to a large extent between sequence aligners, but varies between pore models. This information is of significant value to sequence aligners, many of which look for perfectly matching seed sequences. Due to the bias in k-mer accuracy some seed sequences are virtually impossible to find correctly, and as such should be excluded from being chosen as seeds.

We demonstrate that this k-mer bias has some observable and predictable foundation in the mean current readings of each k-mer(i.e. the manufacturers pore model). This provides a way to estimate the read accuracy for a provided pore model without performing large sequencing runs; it also suggests that a model-guided approach for nanopore design could improve overall accuracy. Alternatively it could allow for the development of specific pores, where accurate discrimination of some k-mer subtypes is more important than others.

Finally we propose a novel nanopore read simulator capable of modeling the k-mer bias observed in this experiment. We demonstrate that our simulator has a significantly reduced sum of squares error with respect to 6-mer accuracy when compared with other simulators. This provides a more realistic benchmark for sequence aligners to compare against, specifically allowing for sequence aligners tailored for nanopore sequencing data.

The availability of high accuracy reads allows for the exploration of new applications, including; sequencing of larger organisms, organism disambiguation when sequencing a population, exact sequence detection in diploid and polyploid organisms, and the ability to scaffold genomes across exceptionally long repeat regions. Providing an understanding of accuracy in nanopore design, and development of tools to aid the alignment of the produced reads is thus critical to continued progress.

References

[1] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty *et al.*, "Characterizing and measuring bias in sequence data," vol. 14, no. 5, p. R51. [Online]. Available: http://genomebiology.biomedcentral.com.ezproxy1.lib.asu.edu/articles/10.1186/gb-2013-14-5-r51

[2] P. C. Faucon, R. Trevino, P. Balachandran, K. Standage-Beier, and X. Wang, "High accuracy base calls in nanopore sequencing," p. 126680. [Online]. Available: http://biorxiv.org/content/early/2017/04/11/126680

[3] I. Sovi, M. iki, A. Wilm, S. N. Fenlon, S. Chen, and N. Nagarajan, "Fast and sensitive mapping of nanopore sequencing reads with GraphMap," vol. 7, p. 11307. [Online]. Available: http://www.nature.com.ezproxy1.lib.asu.edu/ncomms/2016/160415/ncomms11307/full/ncomms11307.html

[4] H. Li, "Toward better understanding of artifacts in variant calling from high-coverage samples," vol. 30, no. 20, pp. 2843–2851. [Online]. Available: https://academic.oup.com/bioinformatics/article/30/20/2843/2422145/Toward-better-understanding-of-artifacts-in

[5] K. Berlin, S. Koren, C.-S. Chin, J. P. Drake, J. M. Landolin, and A. M. Phillippy, "Assembling large genomes with single-molecule sequencing and locality-sensitive hashing," vol. 33, no. 6, pp. 623–630. [Online]. Available: http://www.nature.com.ezproxy1.lib.asu.edu/nbt/journal/v33/n6/abs/nbt.3238.html

[6] C. Yang, J. Chu, Ren, e. L. Warren, and I. Birol, "NanoSim: nanopore sequence read simulator based on statistical characterization," p. 044545. [Online]. Available: http://biorxiv.org/content/early/2016/03/18/044545.1

[7] Y. Ono, K. Asai, and M. Hamada, "PBSIM: PacBio reads simulatortoward accurate genome assembly," vol. 29, no. 1, pp. 119–121. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/29/1/119

[8] B. Lau, M. Mohiyuddin, J. C. Mu, L. T. Fang, N. Bani Asadi, C. Dallett, and H. Y. K. Lam, "LongISLND: in silico sequencing of lengthy and noisy datatypes," vol. 32, no. 24, pp. 3829–3832. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5167071/

[9] M. Escalona, S. Rocha, and D. Posada, "A comparison of tools for the simulation of genomic next-generation sequencing data," vol. 17, no. 8, pp. 459–469. [Online]. Available: http://www.nature.com.ezproxy1.lib.asu.edu/nrg/journal/v17/n8/abs/nrg.2016.57.html

[10] M. Frampton and R. Houlston, "Generation of artificial FASTQ files to evaluate the performance of next-generation sequencing pipelines," vol. 7, no. 11, p. e49110. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0049110

[11] D. A. Earl, K. Bradnam, J. S. John, A. Darling, D. Lin, J. Faas *et al.*, "Assemblathon 1: A competitive assessment of de novo short read assembly methods," p. gr.126599.111. [Online]. Available: http://genome.cshlp.org/content/early/2011/09/16/gr.126599.111

[12] W. Huang, L. Li, J. R. Myers, and G. T. Marth, "ART: a next-generation sequencing read simulator," vol. 28, no. 4, pp. 593–594. [Online]. Available: https://academic.oup.com/bioinformatics/article/28/4/593/213322/ART-a-next-generation-sequencing-read-simulator

[13] J. C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer, "Substantial biases in ultra-short read data sets from high-throughput DNA sequencing," vol. 36, no. 16, p. e105. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2532726/

[14] Y.-C. Chen, T. Liu, C.-H. Yu, T.-Y. Chiang, and C.-C. Hwang, "Effects of GC bias in next-generation-sequencing data on de novo genome assembly," vol. 8, no. 4, p. e62856. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0062856

[15] V. Boa, B. Brejov, and T. Vina, "DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads." [Online]. Available: http://arxiv.org/abs/1603.09195

[16] M. David, L. J. Dursi, D. Yao, P. C. Boutros, and J. T. Simpson, "Nanocall: An open source basecaller for oxford nanopore sequencing data," p. 046086. [Online]. Available: http://biorxiv.org/content/early/2016/03/28/046086

[17] N. J. Loman, J. Quick, and J. T. Simpson, "A complete bacterial genome assembled de novo using only nanopore sequencing data," vol. 12, no. 8, pp. 733–735. [Online]. Available: http://www.nature.com.ezproxy1.lib.asu.edu/nmeth/journal/v12/n8/full/nmeth.3444.html