OXFORD

Briefings in Functional Genomics, 00(00), 2019, 1–11

doi: 10.1093/bfgp/elz033
Advance Access Publication Date:
Review paper

A broad survey of DNA sequence data simulation tools

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, Azza Ahmed, Christian D. Bope, Imane Allali, Gaston K. Mazandu, Nicola J. Mulder and Emile R. Chimusa

Corresponding author: E.R. Chimusa. Division of Human Genetics, Department of Pathology, Institute of Infectious Disease and Molecular Medicine, Faculty of Health Sciences, University of Cape Town, Cape Town, South Africa. Tel.: 27 21 406 6425; Fax: 27 21 406 6826; E-mail: emile.chimusa@uct.ac.za

Abstract

In silico DNA sequence generation is a powerful technology to evaluate and validate bioinformatics tools, and accordingly more than 35 DNA sequence simulation tools have been developed. With such a diverse array of tools to choose from, an important question is: Which tool should be used for a desired outcome? This question is largely unanswered as documentation for many of these DNA simulation tools is sparse. To address this, we performed a review of DNA sequence simulation tools developed to date and evaluated 20 state-of-art DNA sequence simulation tools on their ability to produce accurate reads based on their implemented sequence error model. We provide a succinct description of each tool and suggest which tool is most appropriate for the given different scenarios. Given the multitude of similar yet non-identical tools, researchers can use this review as a guide to inform their choice of DNA sequence simulation tool. This paves the way towards assessing existing tools in a unified framework, as well as enabling different simulation scenario analysis within the same framework.

Key words: DNA sequence; next generation sequence; simulation; genomics; bioinformatics tools

Shatha Alosaimi, MSc, at the University of Cape Town.
Armand Bandiang, PhD student at the University of Cape Town.
Noelle Van Biljon, MSc, at the University of Cape Town.
Denis Awany, PhD student in Human Genetics at the University of Cape Town.
Prisca Thami, PhD student at the Division of Human Genetics, University of Cape Town.
Milaine SS. Tchamga, PhD in Biomathematics. Postdoc at the University of Cape Town.
Anmol Kiran, Post-Doctoral Bioinformatician, Malawi-Liverpool-Wellcome Trust Clinical Research Programme, Blantyre, Malawi.
Olfa Messaoud, PhD in Human Genetics. She is an Associate Professor in the Biomedical Genomics and Oncogenetics Laboratory, Institut Pasteur de Tunis.
Radia Ismael, PhD student in Human Genetics at the University of Cape Town.
Jacquiline Mugo, PhD candidate in Bioinformatics at the University of Cape Town.
Azza Ahmed, PhD at Centre for Bioinformatics and Systems Biology, Faculty of Science, University of Khartoum.
Christian D. Bope, PhD in BioPhysics. Postdoc in Division of Human Genetics, Department of Pathology, University of Cape Town.
Imane Allali, PhD in Bioinformatics. Postdoc at the University of Cape Town.
Gaston K. Mazandu, PhD in Bioinformatics. Senior Lecturer at the Division of Human Genetics, Department of Pathology, University of Cape Town and Researcher at the African Institute for Mathematical Sciences, Cape Town.
Nicola J. Mulder, PhD in Medical Microbiology. Professor and Head of the Computational Biology Division at the University of Cape Town and PI of H3ABioNet.
Emile R. Chimusa, PhD in Bioinformatics. A/Prof at the Division of Human Genetics, Department of Pathology, University of Cape Town.

Introduction

Genomic sequence data have been at the epicentre of numerous bioscience research fields and has allowed us to gain insights into medicine, evolution, ancestry and more. The invention of deoxyribonucleic acid (DNA) sequencing tools incentivised an increase in bioinformatics tool development projects, particularly with the advancement of Next generation sequencing (NGS). NGS technologies allow massively parallel acquisition of nucleotide sequences, resulting in higher data throughput, though at the cost of shorter reads and prone to higher error rate when compared with traditional Sanger sequencing [1, 2]. This necessitates a myriad of computational algorithms and tools to interpret the raw sequence reads coming from the sequencing platforms. Despite that all sequencing platforms have the same goal to generate of DNA reads, each platform has a particular characteristic and common errors in fragmenting DNA. Effectiveness, accuracy and performance assessments of different analytical methods used to either analyse NGS data or perform variants calling are crucial in the biomedical field. Simulation tools that can accurately mimic DNA reads generation from different sequencing platforms may play a critical role in assessing the efficient NGS data analytical methods/-tools.

Prior to the completion of sequencing the first human genome, plans for genomic analysis tool development (Figure 1) and evaluation were initiated with the creation of tools such as Celsim [1], a tool that produces empirical whole genome shotgun sequencing data, and GenFrag [2, 3], a tool that uses an existing DNA sequence as a parent strand to randomly generate sequences that conform to user-specified criteria. Celsim [1] and GenFrag [2, 3] are the two first sequencing simulation tools upon which all other tools were subsequently built.

DNA sequencing simulation tool development has evolved alongside the sequencing platform development. For example, following the emergence of the Roche 454 sequencing platform, MetaSim [4] was developed in 2008 to simulate Roche 454 [5] output data and focused on applications in metagenomic studies. This tool provided the ability to generate synthetic data sets from taxonomic compositions of complex metagenomic data sets. It could not, however, produce read quality scores and poorly represented true Roche 454 data. This led to the development of Flowsim [6] in 2010. Flowsim simulates Roche 454 data by selecting substrings of the reference data and creating flowgrams for this information. These flowgrams are then analysed to call the simulated sequences and quality scores. This results in a much more accurate simulation of Roche 454 sequence data. However, Flowsim does not output a description of the simulation runs, which motivated the development of Mason [5]. Mason can also simulate Illumina and Sanger sequencing data, not only Roche 454 [5].

This pattern of development has occurred for each sequencing technology. As the characteristics of the sequence data changed, so did the simulation tools. Accordingly, more than 35 simulation technologies (Figure 2) have been developed for simulating genomic DNA sequences alone. Some of these tools are multipurpose, while others are specific to sequencing platforms (Figure 2) [7]. Although many DNA sequence simulation tools have been developed over the years, there is paucity in the number of publicly available or specialised ones, and thus, for many, the documentation and approaches are generally not extensive. Moreover, given these tools were designed based on differing sequencing platforms, cross evaluating how well they mimic DNA sequence is still challenging. The challenge is mainly related to finding common golden metrics (true parameters or true golden data) to serve as a baseline to fairly conduct such cross-assessment.

Data simulation is important and necessary for bioinformatics tool developers. It can generate a complete data set with defined parameters to benchmark bioinformatics tools [1] by allowing tool developers and users to assess the accuracy of NGS analytical approaches, such as sequencing alignment and variant calling. Data simulation also allows one to keep parameters constant and to test the effect of changing one parameter on the data output and tool performance [5]. This also allows researchers to pinpoint weaknesses of a tool and to address them more effectively [7]. There are several other applications of simulation, including evaluating the best probes for hybridisation using probe capture simulation [8], determining the accuracy of primers used in polymerase chain reaction (PCR) [9] and identifying significantly enriched sequence motifs [10]. Moreover, in most cases, using simulated data is usually more effective than real data in the absence of ground truth or gold standard (benchmark) data sets. As simulated data are artificial, there are no security requirements, allowing easy sharing of data for reproducibility testing [11]. Sequence simulation is also cheaper and less time-consuming than physical genomic sequencing [12], hence allowing the creation of large data sets in a quick and cheap manner. Simulating data cuts out the time required for transferring data which often can be more time-consuming than the actual data analysis [11]. Thus, although not the ultimate solution, simulated DNA data are critical for complementing experimental investigations based on real data.

These challenges persist, as evidenced by the observation that many studies still make use of software or web-based services that were not originally developed for the given simulation scenario. Moreover, given these tools were designed based on differing sequencing platforms, cross evaluating how well they mimic DNA sequence is challenging as there is generally no common golden metrics (true parameters or true golden data) to serve as a baseline to fairly conduct such cross-assessment.

In light of the above, we here provide a broad discussion of DNA sequence simulation approaches and guidance through a tree decision to enable researchers to determine which tool they should use, given the data and resources at their disposal and the desired result (see section below). In addition, we compared 20 state-of-art DNA sequence simulation tools on their ability to produce accurate reads based on their implemented sequence error model of 10 reached 75% of precision in generating correct reads. This review should orient users on the choice of an appropriate tool and provides an update on current advances, challenges and opportunities behind existing NGS simulation tools.

General framework of DNA sequence simulation

While the first wave of DNA sequence simulation frameworks was primarily designed for sequence analyses and testing phylogenetic hypotheses, subsequent frameworks to date have incorporated the ability to mimic DNA structures under the various NGS sequencing platforms. DNA sequence simulation tools all rely on different statistical models—each with their own assumptions—to perform the data simulation. Accordingly, various tools have different workflows and underlying processes to produce the sequence data. However, these sequence simulation tools all seem to follow a general workflow that includes defining the error models for the data set, applying these error models to the input genomic reference data and then sampling randomly from this adjusted input [14] (Figure 2).

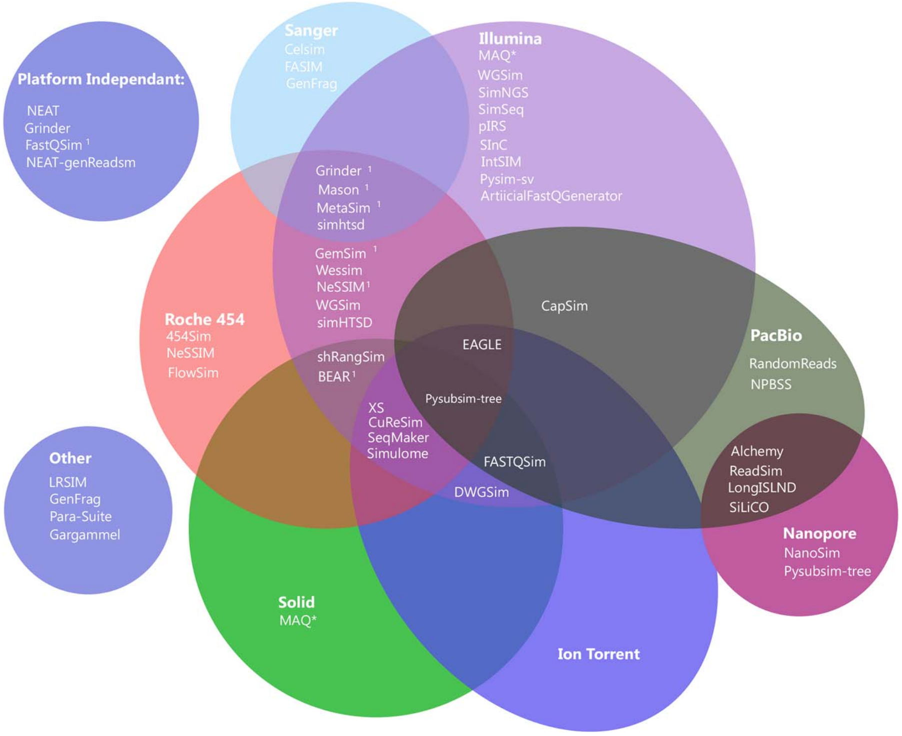

Figure 1. Venn diagram of DNA sequence reads simulation tools and respective uses.

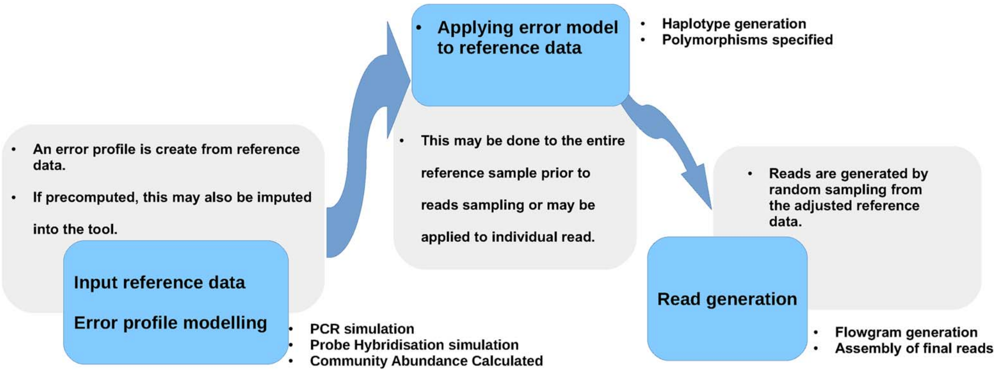
An error profile is create from reference data.
If precomputed, this may also be imputed into the tool.
Input reference data
Error profile modelling
PCR simulation
Probe Hybridisation simulation
Community Abundance Calculated
Applying error model to reference data
Haplotype generation
Polymorphisms specified
This may be done to the entire reference sample prior to reads sampling or may be applied to individual read.
Reads are generated by random sampling from the adjusted reference data.
Read generation
Flowgram generation
Assembly of final reads

Figure 2. Flow diagram of the general read simulation process. The figure displays general parameter for DNA sequence simulation.

Mimicking sequencing models and use of reference sequence

Variation between DNA simulation tools exists in the error modelling procedure. The general approach of the error models is to apply mapping that assigns error rates to each DNA base position along the length of the sequence. The type of mapping (error type, DNA base at the location where the error occurs and base preceding the position where the error occurs) then depends on the tool and on the sequencing platform from which the data are being simulated. Originally, substitution [50] errors were modelled using uniform distributions across data sets, but recently, more complex methods have been developed that allow incorporation of specific error models. These models do not only reflect sequencing errors but also incorporate PCR artefacts, experimental biases, community structures [7] and sequence context information such as insertion errors, fixed proportion of deletion errors and substitutions [17]. The development of an error model is extremely platform dependent; Box 1 describes the error model from each DNA sequencing platform.

The error models defined by the user or built into the tool are then reflected onto the reference sequence data. This may be performed on either the sampled fragments from the reference data set, or alternatively, prior to sampling and to the entire reference sequence.

The error profiles are used as input, along with the reference data, for the read simulation step [4, 21, 28, 44]. This involves random sampling from the reference sequence(s); sampling may be from one or many references, depending on the chosen simulation tool [3, 39]. This is often performed using a random number generator where a starting position is chosen from the reference sequence. Following this, read length is also chosen from a defined distribution. Using the starting position and read length information, a read is sampled from the reference sequence. Figure 2 shows the options for sequence generation which are described in Box 2. Importantly, DNA sequencing simulation approaches are sensitive and rely on how well they mimic and leverage specific data scenarios (summarized in Supplementary Tables 1 and 2). Box 3 describes current data-specific simulation approaches.

To generate synthetic data sets from sequencing platforms, both (i) mutation model and (ii) various sequencing models are being considered. Insertion models define the probability of mutation in the synthetic data set to be generated and thus models probabilities of substitutions at base positions, mutation rates of SNPs, the distribution of the indel lengths and structural variants. This model may be derived from empirical data sets to model the distribution of mutations in that data set, or derived from a simulated data. Some tools such as dwgsim [20], NEAT [21] and Gemsim [15] offer the flexibility of allowing the user to input the desired mutation model. The sequencing model captures the form of read sequence generated by sequencing platforms and consists of the quality score, read sampling and sequencing error models. The quality score model, which is generally given by the transition matrices of the time-inhomogeneous Markov model, captures the distribution for the quality of the symbols in the simulated data set. For many tools that leverage reference data sets, the sequencing error model is used to capture distribution of sequencing error observed in reference data set. Finally, the GC-content bias may be captured by the read sampling model [21–22]. Most of the current tools use the log-normal and normal distribution to model the distribution of read lengths and read accuracy, respectively.

Performance of current tools: input requirements, capabilities and limitations

The review by Escalona *et al.* [14] describes the capabilities, parameters and requirements of various DNA simulation tools. Furthermore, Escalona and colleagues describe these characteristics in extreme detail. Recent tools that have not been mentioned by Escalona *et al.* [14] are now summarised in Supplementary Tables 1 and 2. Although each of these tools is based on a combination of both empirical and statistical models, they do possess inherent strengths and weaknesses (Supplementary Tables 1 and 2). It is very important to highlight these so that the most appropriate tool can be used for the desired purpose (see Figure 1 for all possible tools) so that potential pitfalls can be avoided in order to generate a realistic simulated data set for the required outcome (Supplementary Table 1) [1–11, 13, 15–19, 21–33, 34–56]. While the first wave of DNA sequence simulation frameworks was primarily designed for sequence analyses and testing phylogenetic hypotheses, subsequent DNA sequence simulation frameworks have incorporated the ability to mimic DNA structures under a specific NGS sequencing platform. All the DNA sequence simulation tools rely on different statistical models—each with their own assumptions—to perform the data simulation.

Importantly, to produce reliable variants calling NGS data analysis tools, it is necessary to validate results obtained from these simulation tools, to assess and characterize their short falls and strengths under various scenarios. It is important to have accurate data sets or baseline parameters from sequencing platforms to benchmark these simulation tools. Sequencing platforms such as Illumina, Sanger, SOLiD and Ion torrent are private companies, which make it difficult to the researchers to access these baseline parameters, to conduct a fair comparison and to benchmark these simulation tools.

In the sequel, we discussed scenario-based contexts under which read lengths can be generated to mimic as much as possible empirical data from NGS sequencing technologies.

General (platform independent) tools

Various simulation tools are described as sequencing platform-independent. This may be advantageous when using reference data from a database that does not describe run and platform parameters in detail. These platform-independent tools allow users to input their own reference data to train the error profiles to be specific to the given data [16, 38]. However, as these tools generate error profiles from reference data by aggregating the characteristics of many reference sets, they will create error profiles that are less accurate if few samples are used [16]. Such tools include FASTQSim [16] and BEAR [38]. These tools should be used when many reference samples are available and are especially helpful if no dedicated simulation tool exists for a desired sequencing platform.

Metagenomic analysis

Tools that can be used to simulate metagenomic data are very useful as it is particularly difficult to validate metagenomic data. These tools combine population abundance profiles and reference data to simulate the variety of reads expected from metagenomic studies. These metagenomic analysis tools include MetaSim [4], Grinder [9] and NeSSM [37].

Whole exome/coding region analysis

Whole exome sequencing is cheaper and much faster than whole genome sequencing and hence has become a target for bioinformatics tool design. Probe hybridisation and capture are vital steps of whole exome and targeted sequencing that introduce biases into sequence results. Simulating this capture procedure along with the sequence simulation may facilitate the process of evaluating the effect of any biases and false sequences introduced. This also allows evaluation of hybridisation probe design. Tools that enable this evaluation feature include Wessim [36] and CapSim [8]. CapSim has a more intricate means to simulate probe dynamics than Wessim.

Ancient DNA analysis

Analysis of ancient DNA is useful for inferring history of extinct and past populations or species [47]. Ancient DNA has defining characteristics, such as extensive fragmentation, damage and contamination [47], which affect downstream analysis tools. The use of simulated data allows researchers to investigate the magnitude effect of these characteristics on the downstream tools. Gargammel tool [47] allows simulation of ancient DNA.

Illumina analysis

There are many appropriate tools available to simulate Illumina data. The error models that arise during Illumina simulation have been well studied. Accordingly, tools which are built in Illumina error models will be equally accurate as the tools that allow training any given data. Hence, newer tools that are up to date with current Illumina technology are the best choice for simulating this type of data, for example, CapSim [8] and SInC [13].

Roche 454 analysis

Error models developed for Roche 454 data are also accurate, especially for the tool Flowsim [6] that generates a flowgram during read simulation. Thus, it may not be necessary to use a tool that generates error models from given data. However, as Roche

454 sequencing has become outdated and non-competitive, it was shut down by Roche in 2013. Therefore, the use of a general platform-independent tool may be useful; otherwise, one may use tools such as NeSSM [37] and ART [22].

SOLiD analysis

Error profiles resulting from SOLiD sequencing are not as well characterised as error profiles from other sequencing platforms. DWGSim [30], ART [22] and XS [40] can simulate SOLiD sequence data with built-in error profiles. However, these tools are not recently developed and using a platform-independent tool with training data may be a better choice when simulating SOLiD data.

Ion torrent analysis

Error profiles that result from Ion Torrent sequencing are less well characterised than Illumina and Roche 454 sequencing. As a result, no dedicated Ion Torrent sequence simulation tools have been developed to date. However, if a researcher wishes to simulate Ion Torrent data, XS [40] has built-in error profiles for this. Otherwise researchers may use platform independent tools to simulate Ion Torrent data, such as BEAR [38].

Third-generation analysis

There are many third-generation sequence simulation tools. Currently, the most accurate simulation tool for PacBio data is SimLoRD [44] and for Nanopore data is SiLiCO [18]. These tools are recently developed and allow editing of parameters as the sequencing platform conditions evolve, making these tools extremely powerful.

Choice of DNA sequence simulation tool

The choice of DNA simulation tool depends on the research goal that one would like to simulate DNA data to either evaluating the ability to (i) reconstruct the DNA sequence across differing sequence alignment software, (ii) remove or marking duplication based on different marking duplication tools, (iii) calibrate and correct the bases distribution and GC content distribution and (iv) correctly discover the variants across current variants calling tools. A tree decision in Figure 3 may help and guide users and readers to make a decision on the choice of an appropriate tool for their goal.

To efficiently achieve the decision on a potential simulation tool, one may look at (i) type of simulation inputs (mainly model parameters of the simulation tool (Boxes 1 and 2) such sequence model, mutation model) and (ii) type of simulated outputs (fastq, bam or vcf). Figure 3 may provide readers with a guideline for the identification of the NGS simulation tool that are best suited for their purposes. Furthermore, the advantage of correctly choosing an appropriate DNA sequence simulation tool is that the tool will generate appropriate simulated data with fully known baseline metrics that can be used for (i) assessing the bounds of existing sequence alignment and variant calling approaches, fairly and efficiently compare how different these approaches to each other. This comparison may enable new directions toward the development of new algorithms. The simulated DNA reads data from correct DNA simulation tool can be used for teaching and for research to test hypotheses about reference genome organisation. Furthermore, these simulated tools may allow educators to generate real-looking data sets for learners to learn on.

Evaluating the sensitivity of DNA sequence simulation tool

We evaluated 20 state-of-art DNA sequence simulation tools that are available, that use reference genome and can generate reads file in standard fastq files format. In doing so we extracted chromosome 22 from Human genome reference Build 37 (GRCh37). Since all these tools capture the sequencing error model in their methods, we aimed to evaluate sensitivity of producing quality reads that can be used to accurately be mapped and reconstructed based on the genome reference data. Note that we did not insert any additional SNPs, indels or structural variants in this experiment. We have additional used a custom python script to trace each read produced from each tool to the originating location on the reference genome and the 'true' base-to-base alignment of the read to the reference genome in that location to allow us later on calculating the number of correctly mapped reads/bases.

Each DNA simulation tool was performed on its default parameters and accounting for its sequencing error model. The produced fastq (reads files) from each tool was mapped (aligned back) to the same reference genome (chromosome 22 from Human genome reference Build 37) using BWA. The reconstructed and aligned sequence was compared with the original genome reference genome (chromosome 22 from Human genome reference Build 37). We considered a *read* to be correctly mapped if (i) it gets mapped to the correct chromosome and strand and (ii) the subsequence on the reference genome the read maps to, overlaps with the 'true' mapping subsequence by at least *p* bases (two values of *p* were setup to a fixed value of 25 bp to the true alignment locus of the base). We conducted the evaluation based on (i) sensitivity which is defined as the fraction of correctly mapped bases (according to this notion of a *correct mapping*) out of the total number of bases in the reads and (ii) precision which is defined as the fraction of correctly mapped bases out of the total number of mapped bases in the reads. Using these two measures, we compared all of these 20 DNA sequence simulation tools; a summary of the result is presented in Table 1. Having taken 75% of precision as cut-off, we observed that Wessim (87.1%), NEAT (86.2%, ReadSim (85.7%), XS (82.7%, DWGsim (79.1%), WGSim (78.6%), NanoSim(78.1%), SiLiCO (77.6%), SInC (76.9%) and FastQSim (76.1%) have respectively reached precision above 75%; however, the running time seems increasing with DNA sequencing simulation tool with higher precision.

Conclusions and perspectives

Recent developments in DNA sequencing technologies have led to improvements in quality and capacity of generated sequence reads. To match this development and to complement real data in experimental research, there is need to continue to develop new DNA sequence simulation tools or refine the currently existing ones.

The DNA sequence simulation tool must be able to account for the user available data and produce result outputs in a desired format (Supplementary Tables 1 and 2). The simulation tool must be able to mimic the desired data accurately to ensure any benchmarking using such data is also accurate. Given the difficult task that DNA sequence simulation tools are required to fulfil, including flexibility of read simulation, adaptation for user-specific requirements for output data and ability to capture highly variable data, there are some shortcomings. These shortcomings are further compounded by poor documentation. Tools

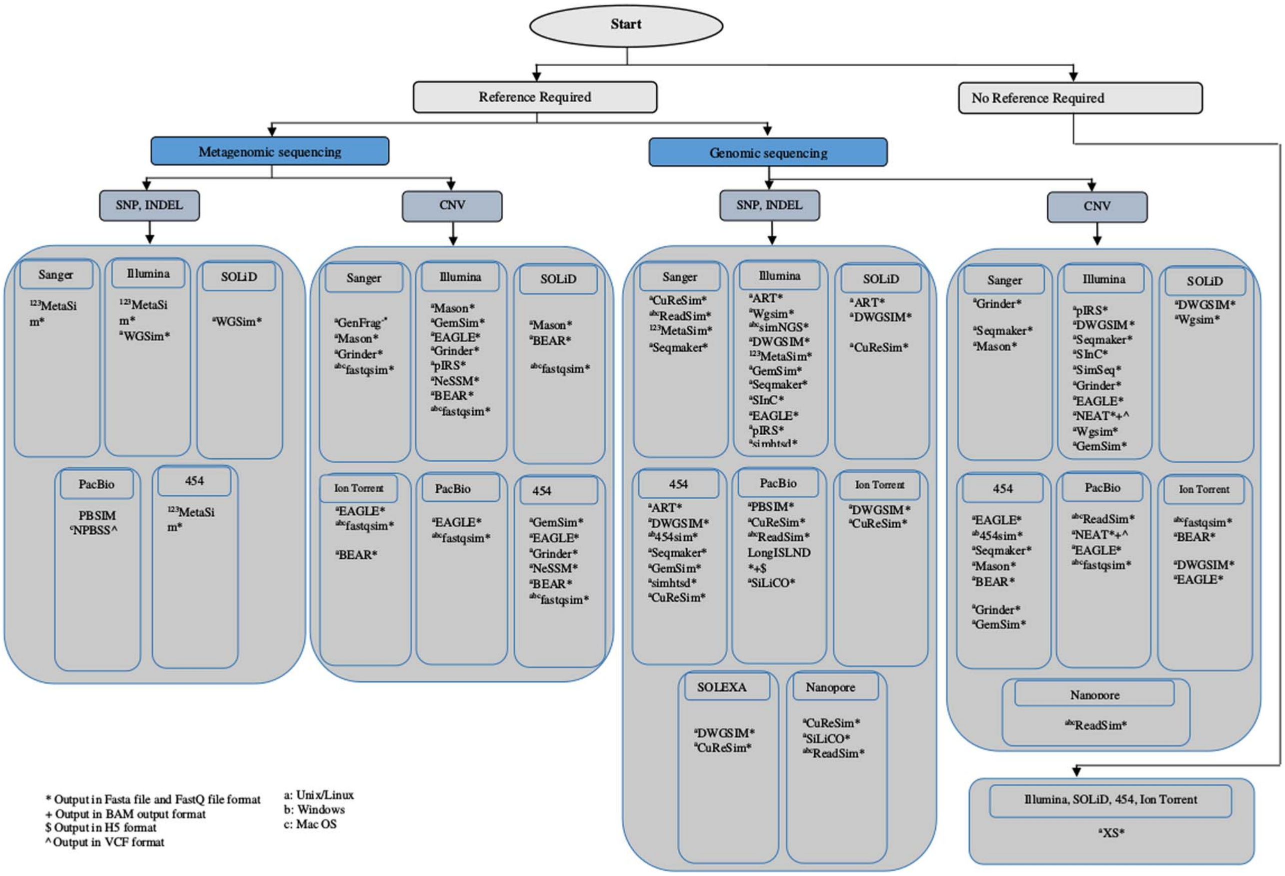

Figure 3. Decision tree for the choice of a suitable DNA simulation tool. The choice of a DNA simulation tool requires a set of sequential decisions. First, decide whether there is a reference sequence or not. Then, decide on variants to be simulated and model input parameters (sequencing error, mutation, GC content base quality distribution model). Next, specify whether genomic variants should be introduced (in addition to those that already exist in the reference sequence or sequences). Finally, determine the sequencing technology of interest. Roche 454 pyrosequencing; Nanopore, Oxford Nanopore Technologies; PacBio, Pacific Biosciences; Sanger sequencing; SOLiD, and detection (Thermo Fisher) and sequencing by oligonucleotide ligation.

Table 1. Evaluating 20 state-of-art DNA sequence simulation tools that use reference genome and can generate read files in standard fastq files format based on sensitivity and precision in producing reads

Year	Tools	Sensitivity (%)	Precision (%)	Running time (h:min:s)
2009	WGSim* **[26]**	**77.01%**	**78.6%**	**3:04:22**
	DWGsim* **[30]**	**76.37%**	**79.2%**	**3:14:09**
2012	GemSim [15]	72.55%	73.1%	1:54:19
	EAGLE* [7]	66.28%	69.8%	1:40:01
	pIRS [34]	69.12%	74.4%	1:14:42
2013	Wessim **[36]**	**86.41%**	**87.1%**	**6:23:03**
2014	BEAR [38]	73.38%	74.8%	3:24:13
	CuReSim [39]	70.24%	73.4%	4:00:22
	FastQSim **[16]**	**74.71%**	**76.1%**	**2:01:12**
	ReadSim∗ **[19]**	**76.91%**	**85.7%**	**5:41:43**
	XS **[40]**	**78.67%**	**82.7%**	**5:13:04**
	SInC **[13]**	**71.10%**	**76.9%**	**3:34:12**
2016	LongISLND [43]	71.33%	73.3%	3:34:42
	NEAT **[21]**	**85.62%**	**86.2%**	**5:52:01**
	SiLiCO **[18]**	**75.36%**	**77.6%**	**3:24:02**
2017	CapSim [8]	72.81%	74.8%	1:44:07
	LRSim [46]	74.13%	74.53%	1:44:22
	Gargammel [47]	72.52%	74.2%	2:04:02
	NanoSim **[48]**	**70.91%**	**78.1%**	**3:11:01**
	Pysim-sv [51]	61.12%	72.2%	2:04:52

that fall part of a greater package or tools that were developed for a specific research project seldom have published documentation specific to the simulation tool. As Supplementary Tables 1 and 2 show, the gaps and limitations are not necessarily due to imperfection in the tools, but rather the lack of documentation to make such a conclusion. This lack of documentation also makes these tools increasingly difficult to use for non-specialists [52].

Another shortfall regarding DNA sequence simulation is that most of existing tools are redundant and regardless of sequence models they still not reach 90% precision to produce DNA reads (Table 1). Yet, to the best of our knowledge, none of these tools incorporates the possible data scenarios and integrates strengths of all tools for all sequencing platforms. SimLoRD [44] allows adjustment of simulation conditions given sequencing platform changes (Box 1); this is an extremely powerful tool that eliminates the need for extensive tool maintenance. Maintenance constitutes an issue for many of these tools [57]. Tools that have been designed to incorporate specific error models (Box 1) for Illumina or Roche 454 sequencing become obsolete if not maintained as the sequencing platform evolves and is updated.

Another shortfall is that several DNA sequence simulation tools have not been validated or benchmarked by the developers. Hence, the question posed by Celsim [1] in 1998 still stands—how do we know if the simulated data are indeed representative of the true sequence data?

Unfortunately, addressing such questions is still challenging due to the difficulty of constructing realistic and versatile platform-independent data set that can be used as benchmarks to evaluate these DNA sequence simulation tools. There are several reasons to this, including (i) the fact that each tool employs empirical error models configured for different NGS sequencing platforms or error probabilities and (ii) the unavailability and difficulty to access baseline metrics or true parameter from these private NGS platforms. Meanwhile, for tools that leverage existing reference sequences, the simulated sequence reads will likely contain heterozygosity due to gaps and errors that likely mask true complexity. The result of all these is variability in the quality of sequence sequences generated by these various DNA sequence simulation tools, especially for those designed for different platforms. Even within the same platform, different models for sequence evolution and error probability distributions make the output of synthetic sequence (or reads) data not to be exactly identical. Thus, the lack of benchmarking data set makes it challenging for impartial intra and inter-platform comparison of the performance of the DNA sequence simulation tools. Here, for the first time, we attempted to evaluate 20 state-of-art DNA sequence simulation tools on their ability to produce accurate reads based on their implemented sequence error model (Table 1). Furthermore, we attempted to identify the literary gaps in the performance of current DNA sequence simulation tools by providing a comprehensive description of most of the DNA sequence simulation tools, highlighting the effectiveness of error models employed, read length coverage and platform context. In our opinion, there is a major need for a tool that incorporates all the advantages of the individual DNA sequence simulation tools into one framework. The ideal tool should allow training of error models on given data sets (Box 3), but also have accurate or true built-in error models for each sequencing platform (Box 1). Such a tool should permit variant simulation, PCR and probe capture simulation. Importantly, such a tool should also facilitate adjusting all sequence parameters possible to ensure the tool remains up to date with sequencing technologies. As researchers employ various tools to simulate sequence reads, it is crucial to note that data quality from NGS platforms can vary between genomic and metagenomic sequencing; thus, it is important that the statistical models by which data is simulated are modified accordingly.

In brief, we present in this review advances in the field of DNA sequence simulation tools, challenges, opportunities and perspectives. To the best of our knowledge, this is the second review addressing these objectives. We have chronologically presented advances in DNA sequence simulation tools development (Supplementary Table 1). We clustered these tools according to sequencing platforms, model parameters and input and output formats that can be found in (Supplementary Table 1). Furthermore, we provide a decision tree that can orient the user on the choice of DNA sequence simulation tools (Figure 3) and compare 20 state-of-art DNA sequence simulation tools on their ability to produce accurate reads based on their implemented sequence error model (Table 1). This review also highlights gaps in the development and implementation of DNA sequence simulation tools. This will presumably foster development of non-identical tools or maintenance of existing tools to circumvent the issue of replicating tools. As genomic research is growing rapidly, DNA sequences (reads) simulations will remain an integral part of the field. Given this, we recommend:

(i) Authors should share and provide as much documentation as possible on the developed simulation tool.
(ii) Maintenance of existing tools and upgrading to cater for advances in sequence generation.
(iii) An integrative tool that incorporates all the advantages of the best DNA sequence simulation tools into one. The tool should be flexible enough to allow the users to choose any type of simulation model. This will also offer easy comparison of output as the different outputs will be coming from the same tool. The state-of-the-art tool should also produce integrative output: reads, alignments and variants.

Key Points

(i) Discussion of issues related to efficiently mimic human DNA sequence to improve variants discovery approaches.
(ii) Dissecting current methods and tools available for simulating DNA sequence.
(iii) Discussing sensitivity of simulation parameter from existing DNA sequence to allow users to choose appropriate tool and developers to identify potential gaps to be filled.

Box 1: Error model generation per platform

Roche 454: Roche 454 sequencing allows base identification by the cyclical addition of bases to the flow chamber, and by measuring a change in emitted light. Homopolymers are identified in one run: from the emitted light intensity, the number of bases incorporated is identified. This often leads to over and under base calling. Hence, the common error models associated with simulating 454 data involve generating the resulting insertion and deletion errors [31].

Illumina: Illumina sequence data are generated by a synthesis approach—with one base added at a time. Thus, the main source of errors is substitutions due to incorrect base calling. Accordingly, simulators of Illumina sequence data will incorporate this by using base quality scores as a means to identify the likelihood of errors at each nucleotide position [31].

SOLiD: Applied Biosystems SOLiD reports nucleotide transition colour codes rather than nucleotide sequences. Hence, simulators of SOLiD data must also return transition codes [31]. From these transition codes and the quality associated, the DNA simulators identify nucleotide sequences and possible errors to incorporate in the error models.

Ion torrent: Ion torrent sequencing measures the change in system pH after the addition of a solution containing one nucleotide base per each cycle. The change in pH indicates whether a base has been incorporated or not [51]. The greater the pH changes the more bases are incorporated, hence allowing the identification of homopolymers. Consequently, the simulation process is like 454 simulations—the main source of errors will be over- and under-calling of homopolymers. Hence, insertions and deletions will need to be incorporated into these error models. It has also been shown that when using Ion torrent, the errors tend to increase towards the end of the reads [21].

Third-generation sequencing: this sequencing, also known as single molecule sequencing, results in longer sequence reads compared to NGS data. Oxford Nanopore read lengths are best fitted by the gamma distribution [45], whereas PacBio read lengths follow a lognormal distribution. Single-molecule sequencing results in errors that are randomly distributed through the sequence reads [13]. Third-generation sequencing technologies have a high error rate. However, as the process involves cyclic resequencing, the error rate decreases with each cycle of sequencing [13]. Third-generation sequence simulators produce two types of data—circular consensus sequencing (CCS) reads (which are short and have low error rates) and continuous long reads (CLR) (which are long reads with high error rates). CLR errors follow a normal distribution, while CCS reads have an error rate that exponentially increases along the read position [33]. Accordingly, this must be reflected in Third-generation read simulation tools such as PBSIM and FastQSim [21,33].

Platform Independent tools: these platform independent tools allow training on specific data sets to generate error models for one's own data. This involves inputting reference data sets for the simulation tool to aggregate across and build error profiles and more samples will result in an error profile that is more accurate [21]. Hence, these simulation tools can simulate data from any sequencing platform. However, one must choose the reference data carefully as if it does not accurately represent the sequencing platform output; the simulated data will not be representative of the true data.

Box 2: Optional steps in sequence generation

After error modelling haplotype generation may be incorporated into the final read generation. Simulation tools that incorporate this step include GemSim and Mason. This may be done using the built-in tools—or precomputed haplotype data may be added to the simulation run [4, 17].

User-specified genetic polymorphisms may also be added to the simulated data, this is very useful when assessing the accuracy of variant calling tools. Some simulation tools that allow this include: NEAT, Grinder, ReadSim, SInC, EAGLE, GemSim and Mason [7, 14, 24].

Simulation tools, such as Grinder, that simulate amplicon data sets incorporate PCR simulation into the read simulation. This includes adding sequence fragments, which may be incorporated into the library due to degenerate primers, to the simulated reads [7].

Probe hybridisation simulation is incorporated in Wessim and CapSim and allows the experimental process of whole exome sequencing to be emulated [6, 36].

For accurate 454 data simulation, flowgrams may be generated—from this base calling and quality scores are identified. The tool that incorporates this is Flowsim [16].

Simulation tools that allow metagenomic read simulations can calculate relative community abundances—performed by MetaSim, BEAR and NeSSM [15,37, 38].

Assembly may form part of the sequence simulation. Instead of FASTQ or FASTA output consisting of sequence reads an entire genome may be simulated. This can be achieved with simulation tools such as FIGG [9].

Box 3: Specific data scenario simulation.

Whole exome sequencing or targeted resequencing: this includes simulation of DNA shearing into random fragments, probe capture by hybridisation and sequencing of the selected fragments [6, 36].

Metagenomic data simulation: using taxonomy data, relative community abundances of organisms are configured. Population evolution simulation may potentially be incorporated. Error models and sequence coverage bias is estimated. Lastly, sequences for the defined data set are simulated [15, 37].

Mate-pair library simulation: reads are sampled uniformly across sequences; coverage depth is kept constant. Weighted by length, the number of reads is sampled from the reference sequence. Then several types of errors are introduced into the read—errors that would form during the mate-pair library construction [30].

Pair-end library simulation: read sampling follows the same process as for mate-pair library simulation. However, random fragments of size 150–500 bp are uniformly sampled from the reference. Fragment size follows a normal distribution [30].

Amplicon simulation: using defined forward and reverse primers, PCR amplicon output is simulated. Following this, community structure must be specified or computed.

Given the community structure and the PCR output, the amplicon reads are generated. Lastly, biological and experimental biases are then simulated [7].

Ancient DNA simulation: first DNA fragments are generated for the target and contaminant samples. Then the characteristic errors for ancient DNA are simulated—these include post-mortem fragmentation and DNA damage. Following this, experimental errors and quality scores are added [48].

Simulation from phylogenetic trees: this requires an anchor genome as reference and phylogeny information as its input. Mutations are simulated across taxa according to defined error parameters. Lastly, raw reads are simulated for each taxon [49]. A recent development in this category (53) can simulate read counts for single nucleotide variants in addition to generation of sequence reads.

Software online links

MetaSim - http://ab.inf.uni-tuebingen.de/software/metasim/
Maq - http://maq.sourceforge.net
Wgsim - https://github.com/lh3/wgsim
Flowsim- http://blog.malde.org/index.php/flowsim
Mason - http://www.seqan.de/apps/mason/
simNGS - https://www.ebi.ac.uk/goldman-srv/simNGS/
454 - https://sourceforge.net/projects/bioinfo-454sim/
DWGSIM - https://github.com/nh13/DWGSIM
makenucseq - http://emboss.sourceforge.net
SimSeq - https://cran.r-project.org/web/packages/SimSeq/
ART - https://www.niehs.nih.gov/research/resources/software/biostatistics/art/
GemSIM - https://sourceforge.net/projects/gemsim/
EAGLE - https://github.com/sequencing/EAGLE
Grinder - https://sourceforge.net/projects/biogrinder/
ArtificialFastqGenerator - https://sourceforge.net/projects/artfastqgen/
PBSIM - https://code.google.com/archive/p/pbsim/
pIRS - https://github.com/galaxy001/pirs
Alchemy - http://bix.ucsd.edu/projects/blasr/
Wessim - http://sak042.github.io/Wessim/
NeSSM - http://cbb.sjtu.edu.cn/&#x007E;ccwei/pub/software/NeSSM.php
BEAR - https://github.com/sej917/BEAR
CuReSim - http://www.pegase-biosciences.com/curesim-a-customized-read-simulator/
FASTQSim - https://sourceforge.net/projects/fastqsim/
FIGG - http://insilicogenome.sourceforge.net
ReadSim - https://sourceforge.net/projects/readsim/
XS - http://bioinformatics.ua.pt/software/xs/
SInC - https://sourceforge.net/projects/sincsimulator/
RandomReads - https://sourceforge.net/projects/bbmap/
LongISLND - http://bioinform.github.io/longislnd/
NEAT - https://github.com/zstephens/neat-genreads
NullSeq - https://github.com/amarallab/NullSeq
SiLiCo - https://github.com/ethanagbaker/SiLiCO
SimLoRD - https://bitbucket.org/genomeinformatics/simlord
SeqMaker - https://github.com/OpenGene/SeqMaker.jl
CapSim - https://github.com/Devika1/capsim
LRSim - https://github.com/aquaskyline/LRSIM
gargammel - https://grenaud.github.io/gargammel/
NanoSim - https://github.com/bcgsc/NanoSim
Simulome - https://github.com/price0416/Simulome
TreeToReads - https://github.com/snacktavish/TreeToReads
NGSPhy - http://github.com/merlyescalona/ngsphy
Pysim-sv - https://github.com/xyc0813/pysim/
IntSIM - http://intsim.sourceforge.net/
Pysubsim-tree - https://github.com/dustincys/pysubsimtree

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bfg.

Acknowledgements

We are grateful to H3ABioNet consortium members for the helpful discussions and CHPC (https://www.chpc.ac.za/) facility for providing computing platform.

Funding

This work was funded in part by DAAD, the German Academic Exchange Programme, under Reference number 91653117, and National Institutes of Health Common Fund under grant number U24HG006941 (H3ABioNet). This work was also partly supported through the National Research Foundation of South Africa for funding (NRF) [grant # RA171111285157/119056], the Sub-Saharan African Network for TB/HIV Research Excellence (SANTHE) and a DELTAS Africa Initiative [grant # DEL-15-006]. The DELTAS Africa Initiative is an independent funding scheme of the African Academy of Sciences (AAS)'s Alliance for Accelerating Excellence in Science in Africa (AESA) and supported by the New Partnership for Africa's Development Planning and Coordinating Agency (NEPAD Agency) with funding from the Wellcome Trust [grant # 107752/Z/15/Z] and the UK government. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the funders. The views expressed in this publication are those of the authors and not necessarily those of AAS, NEPAD Agency, Wellcome Trust or the UK government.

References

1. Myers GA. Dataset generator for whole genome shotgun sequencing. Proceedings. *Int. Conf. Intell. Syst. Mol. Biol.* 1999; 202–10.
2. Engle ML, Burks C. GenFrag 2.1: new features for more robust sequence fragment assembly benchmarks. *Comput. Appl. Biosci.* 1994;**10**:567–8.
3. Engle ML, Burks C. Artificially generated data sets for testing DNA sequence assembly algorithms. *Genomics* 1993;**16**: 286–8.
4. Richter DC, Ott F, Auch AF, *et al.* MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS One* 2008;**3**:e3373.
5. Holtgrewe M. Mason–a read simulator for second generation sequencing data. *Tech. Rep. FU Berlin* 2010.
6. Balzer S, Malde K, Lanzen A, *et al.* Characteristics of 454 pyrosequencing data–enabling realistic simulation with flowsim. *Bioinformatics* 2010;**26**:i420–5.
7. Brinda K. Novel computational techniques for mapping and classifying Next-Generation Sequencing data. *PhD Thesis*,

Univ. Paris-Est Marne-la-Vallée 2016; https://hal.archives-ouvertes.fr/tel-01484198v1/document (2 August 2018, date last accessed).

8. Janin L. GitHub - sequencing/EAGLE: Enhanced Artificial Genome Engine: next generation sequencing reads simulator. 2014; https://github.com/sequencing/EAGLE (2 August 2017, date last accessed).
9. Cao MD, Ganesamoorthy D, Zhou C, *et al*. Simulating the dynamics of targeted capture sequencing with CapSim. *Bioinformatics* 2018;**34**:873–4.
10. Angly FE, Willner D, Rohwer F, *et al*. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Res.* 2012;**40**:e94.
11. Liu SS, Hockenberry AJ, Lancichinetti A, *et al*. NullSeq: a tool for generating random coding sequences with desired amino acid and GC contents. *PLoS Comput. Biol.* 2016;**12**:e1005184.
12. Killcoyne S, del Sol AFIGG. Simulating populations of whole genome sequences for heterogeneous data analyses. *BMC Bioinformatics* 2014;**15**:149.
13. Pattnaik S, Gupta S, Rao AA, *et al*. SInC: an accurate and fast error-model based simulator for SNPs, Indels and CNVs coupled with a read generator for short-read sequence data. *BMC Bioinformatics* 2014;**15**:40.
14. Escalona M, Rocha S, Posada D. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nat. Rev. Genet.* 2016;**17**:459–69.
15. McElroy KE, Luciani F, Thomas T. GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics* 2012;**13**:74.
16. Shcherbina A. FASTQSim: platform-independent data characterization and in silico read generation for NGS datasets. *BMC Res. Notes* 2014;**7**:533.
17. Lamprecht A-L, Naujokat S, Margaria T, *et al*. Semantics-based composition of EMBOSS services. *J. Biomed. Semantics* 2011;**2**(Suppl 1):S5.
18. Baker EAG, Goodwin S, McCombie WR, *et al*. SiLiCO: a simulator of Long read sequencing in PacBio and Oxford Nanopore. *bioRxiv* 2016;76901.
19. Lee H, Gurtowski J, Yoo S, *et al*. Error correction and assembly complexity of single molecule sequencing reads. *BioRxiv* 2014;6395.
20. Chen GK, Marjoram P, Wall JD. Fast and flexible simulation of DNA sequence data. *Genome research* 2009;**19**(1):136–42.
21. Stephens ZD, Hudson ME, Mainzer LS, *et al*. Simulating next-generation sequencing datasets from empirical mutation and sequencing models. *PLoS One* 2016;**11**:e0167047.
22. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. *Bioinformatics* 2011;**28**(4):593–4.
23. Hur C, Kim S, Kim CH, *et al*. FASIM: fragments assembly simulation using biased-sampling model and assembly simulation for microbial genome shotgun sequencing. *J. Microbiol. Biotechnol.* 2006;**16**:683.
24. Li H, Durbin R. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics* 2009;**25**:1754–60.
25. Li H. Manual Page - maq(1). http://maq.sourceforge.net/maq-manpage.shtml (2 August 2018, date last accessed).
26. Li H. GitHub - lh3/wgsim: Reads simulator. 2017; https://github.com/lh3/wgsim (3 August 2018, date last accessed).
27. SimHTSD - Simulate High-Throughput Sequencing Data. https://sourceforge.net/projects/simhtsd/ (10 August 2018, date last accessed).
28. simNGS and simLibrary – Software for Simulating Next-Gen Sequencing. https://www.ebi.ac.uk/goldman-srv/simNGS/ (3 August 2018, date accessed).
29. Lysholm F, Andersson B, Persson B. An efficient simulator of 454 data using configurable statistical models. *BMC Res. Notes* 2011;**4**:449.
30. Homer N. Simulating Reads with DWGSIM · nh13/DWGSIM Wiki · GitHub. 2011; https://github.com/nh13/DWGSIM/wiki/Simulating-Reads-with-DWGSIM (3 August 2018, date last accessed).
31. Benidt S, Nettleton D. SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. *Bioinformatics* 2015;**31**(13):2131–40.
32. Frampton M, Houlston R. Generation of artificial FASTQ files to evaluate the performance of next-generation sequencing pipelines. *PLoS One* 2012;**7**:e49110.
33. Ono Y, Asai K, Hamada M. PBSIM: PacBio reads simulator—toward accurate genome assembly. *Bioinformatics* 2012;**29**(1):119–21.
34. Hu X, Yuan J, Shi Y, *et al*. pIRS: profile-based Illumina pair-end reads simulator. *Bioinformatics* 2012;**28**(11):1533–5.
35. Chaisson MJ, Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC Bioinformatics* 2012;**13**:238.
36. Kim S, Jeong K, Bafna V. Wessim: a whole-exome sequencing simulator based on in silico exome capture. *Bioinformatics* 2013;**29**:1076–7.
37. Jia B, Xuan L, Cai K, *et al*. NeSSM: a next-generation sequencing simulator for metagenomics. *PLoS One* 2013;**8**:e75448.
38. Johnson S, Trost B, Long JR, *et al*. A better sequence-read simulator program for metagenomics. In BMC bioinformatics. *BioMed Central* 2014, September;**15**(9):S14.
39. Caboche S, Audebert C, Lemoine Y, *et al*. Comparison of mapping algorithms used in high-throughput sequencing: application to ion torrent data. *BMC Genomics* 2014;**15**.
40. Pratas D, Pinho AJ, Rodrigues JMOSXS. A FASTQ read simulator. *BMC Res. Notes* 2014;**7**:40.
41. BBMap. SourceForge.net. https://sourceforge.net/projects/bbmap/ (03 August 2018, date last accessed).
42. BioInfoTools/BBMap. https://github.com/BioInfoTools/BBMap ().
43. Lau B, Mohiyuddin M, Mu JC, *et al*. LongISLND: in silico sequencing of lengthy and noisy datatypes. *Bioinformatics* 2016;**32**(24):3829–32.
44. Stöcker BK, Köster J, Rahmann S. SimLoRD: simulation of long read data. *Bioinformatics* 2016;**32**:2704–6.
45. Chen S, Han Y, Guo L, *et al*. SeqMaker: a next generation sequencing simulator with variations, sequencing errors and amplification bias integrated. Bioinforma. Biomed. (BIBM), 2016. *IEEE Int. Conf.* 2016;835–40.
46. Luo R, Sedlazeck FJ, Darby CA, *et al*. LRSim: a linked-reads simulator generating insights for better genome partitioning. *Computational and structural biotechnology journal* 2017;**15**:478–84.
47. Renaud G, Hanghøj K, Willerslev E, Orlando L. Gargammel: a sequence simulator for ancient DNA. *Bioinformatics* 2016;**33**(4):577–9.
48. Yang C, Chu J, Warren RL, *et al*. NanoSim: nanopore sequence read simulator based on statistical characterization. *Gigascience* 2017;**6**:1–6.
49. Price A, Gibas C. Simulome: a genome sequence and variant simulator. *Bioinformatics* 2017;**33**(12):1876–8.

50. McTavish EJ, Pettengill J, Davis S, *et al.* TreeToReads-a pipeline for simulating raw reads from phylogenies. *BMC bioinformatics* 2017;**18**(1):178.
51. Xia Y, Liu Y, Deng M, *et al.* Pysim-sv: a package for simulating structural variation data with GC-biases. *BMC Bioinformatics* 2017;**18**:53.
52. Yuan X, Zhang J, IntSIM YL. An integrated simulator of next-generation sequencing data. *IEEE Trans. Biomed. Eng.* 2017;**64**:441–51.
53. Chu Y, Wang L, Wang R, *et al.* Pysubsim-tree: a package for simulating tumor genomes according to tumor evolution history. Bioinforma. Biomed. (BIBM), 2017. *IEEE Int. Conf.* 2017;2195–7.
54. Escalona M, Rocha S, Posada D. NGSphy: phylogenomic simulation of next-generation sequencing data. *Bioinformatics* 2018;**34**(14):2506–7.
55. Wei ZG, Zhang SW. NPBSS: a new PacBio sequencing simulator for generating the continuous long reads with an empirical model. *BMC Bioinformatics* 2018;**19**(1): 177.
56. Boenn M. ShRangeSim: simulation of single nucleotide polymorphism clusters in next-generation sequencing data. *J. Comput. Biol.* 2018;**25**:613–22.
57. Dupanloup I, Schneider S, Excoffier L. A simulated annealing approach to define the genetic structure of populations. *Mol. Ecol.* 2002;**11**:2571–81.