COMPUTATIONAL TOOLS

A comparison of tools for the simulation of genomic next-generation sequencing data

Merly Escalona[1], Sara Rocha[1] and David Posada[1,2]

Abstract | Computer simulation of genomic data has become increasingly popular for assessing and validating biological models or for gaining an understanding of specific data sets. Several computational tools for the simulation of next-generation sequencing (NGS) data have been developed in recent years, which could be used to compare existing and new NGS analytical pipelines. Here we review 23 of these tools, highlighting their distinct functionality, requirements and potential applications. We also provide a decision tree for the informed selection of an appropriate NGS simulation tool for the specific question at hand.

Coverage bias
A bias in the amount of reads for a particular region. For example, sequencing depth increases in regions of elevated GC content.

Next-generation sequencing (NGS) techniques are the current standard for the generation of genomic data, producing ever-increasing amounts of information rapidly and at a low cost. These techniques allow us to sequence DNA and RNA very quickly, facilitating the acquisition of massive genomic, transcriptomic, DNA–protein interaction and epigenomic data sets; they are also radically changing the way we look at genomes[1–3]. Given their higher degree of parallelism and smaller reaction volumes compared to conventional Sanger sequencing, NGS methods offer larger amounts of data, shorter sequencing times and reduced costs, albeit at the expense of increased error rates and shorter read lengths[4]. NGS clearly facilitates the accumulation of large data sets, but the downstream processing of these data is still an important bottleneck[5]. Not surprisingly, NGS data give rise to numerous bioinformatics challenges, including storage, transmission, manipulation and analysis. Improved computational methods and more efficient software tools are constantly being developed to provide faster processing and more accurate inferences. However, it is essential that these methods are benchmarked against existing tools with similar functionality, to show their superiority at least in some aspect.

In general, computational methods can be benchmarked using empirical and/or simulated data. Although validation with empirical data is essential, as it represents real scenarios, the true process underlying it is usually unknown, thus complicating its use for the assessment of accuracy (that is, how close the estimated value is to the 'true' value). Alternatively, *in silico* data allow us to generate as much data as desired and under controlled scenarios with predefined parameters for which the true values are known, thus nicely complementing validation with real data[6,7]. Therefore, computer simulation of genetic and genomic data has become increasingly popular for assessing and validating biological models or for gaining an understanding of specific data sets. Simulations alone can be used as guidance for the development of new computational tools[8], for debugging and for evaluating software performance[9,10]. Computer simulations also allow us to generate new hypotheses[11], help in the design of sequencing projects[12,13] and are essential to verify distinct inferences, such as the correctness of an assembly[14], the accuracy of gene prediction[15] or the power to reconstruct accurate genotypes and haplotypes[2,16]. Several computational tools for the simulation of NGS data have been developed in the past few years. These tools have very diverse input requirements and functionalities, which makes it quite difficult to choose the most appropriate one for the problem at hand.

Here we present, to our knowledge, the first review of available software tools for the simulation of genomic NGS data. Note that we focus on the simulation of DNA sequences and do not discuss RNA sequencing (RNA-seq) simulation, which has its own characteristics. We review 23 NGS simulation tools that were either recently published or developed, that were — in most cases — still maintained and that were freely available. We discuss their various features, such as the required input, the interaction with the user, the sequencing platforms, the type of reads, the error models, the possibility of introducing coverage bias, the simulation of genomic variants and the output provided. This is done within the framework of potential applications, providing readers with guidelines for the identification of the NGS simulators that are best suited for their purposes (FIG. 1).

[1]*Department of Biochemistry, Genetics and Immunology, University of Vigo, Vigo 36310, Spain.*
[2]*Institute of Biomedical Research of Vigo (IBIV), University of Vigo, Vigo 36310, Spain.*
Correspondence to D.P.
dposada@uvigo.es

doi:10.1038/nrg.2016.57
Published online 20 Jun 2016

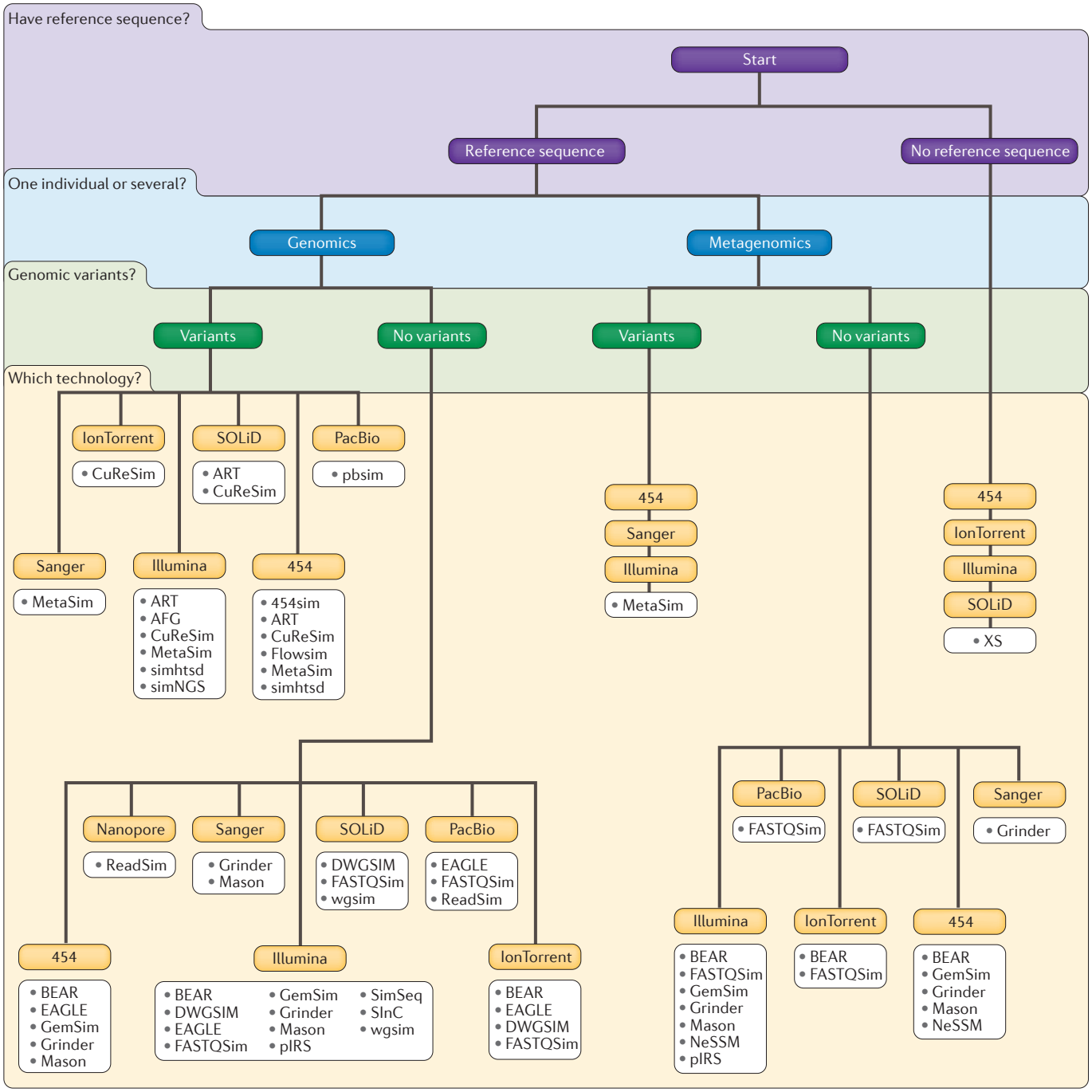


Figure 1 | **Decision tree for the selection of a suitable NGS genomic simulator.** The selection of a next-generation sequencing (NGS) simulator requires a set of sequential decisions. First, decide whether there is a reference sequence or not. Then, decide whether reads should be simulated from one or several organisms. 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. 454, 454 pyrosequencing (Roche); Nanopore, Oxford Nanopore Technologies; PacBio, Pacific Biosciences; Sanger, Sanger sequencing; SOLiD, sequencing by oligonucleotide ligation and detection (Thermo Fisher).

An overview of current NGS technologies

The most popular NGS technologies on the market are Illumina's sequencing by synthesis (which is probably the most widely used platform at present)[17], Roche's 454 pyrosequencing (454), Thermo Fisher's sequencing by oligonucleotide ligation and detection (SOLiD), Thermo Fisher's IonTorrent semiconductor sequencing[18], Pacific Biosciences' (PacBio) single-molecule, real-time (SMRT) sequencing [19] and Oxford Nanopore Technologies' (Nanopore) single-cell DNA

Single end
Reads generated by single-read sequencing, which involves sequencing DNA fragments from only one end.

Paired end
In paired-end sequencing, a single fragment is sequenced from both the 5′ and 3′ ends, giving rise to reads in both forward and reverse orientations, in which read one is the forward read and read two is the reverse. The sequenced fragments may be separated by a certain number of bases (depending on insert size and read length) or overlapping.

Mate pair
Mate-pair sequencing means generating long-insert paired-end DNA libraries. The inserts are circularized and fragmented, and the labelled fragments (corresponding to the ends of the original DNA ligated together) are purified, ligated to another set of adapters and finally sequenced at the paired end. The resulting inserts include two DNA segments that were originally separated by 2–5 kb, facilitating mapping and assembly.

Reference sequence
A particular genomic region, multiple genomic regions concatenated, a chromosome or a complete genome from which next-generation sequencing reads will be generated.

Profile
A set of biological (GC content, insertions and deletions, and substitution rates) and/or technological (insert sizes, read lengths, error rates and quality scores) parameter distributions or values that will be used in a specific simulation.

Abundance profile
A set of probabilities that represent the proportion of taxa within a community (and data set).

Quality scores
(Also known as Phred *Q* scores). Predictions of the probability of an error in a base call.

template strand sequencing. These strategies can differ, for example, regarding the type of reads they produce or the kind of sequencing errors they introduce (TABLE 1). Only two of the current technologies (Illumina and SOLiD) are capable of producing all three sequencing read types — single end, paired end and mate pair. Read length is also dependent on the machine and the kit used; in platforms like Illumina, SOLiD or IonTorrent it is possible to specify the number of desired base pairs per read. According to the sequencing run type selected, it is possible to obtain reads with maximum lengths of 75 bp (SOLiD), 300 bp (Illumina) or 400 bp (IonTorrent). Conversely, in platforms like 454, Nanopore or PacBio, information is only given about the mean and maximum read lengths that can be obtained, with average lengths of 700 bp, 10 kb and 15 kb, and maximum lengths of 1 kb, 10 kb and 15 kb, respectively. Error rates vary depending on the platform, from ≤1% in Illumina to ~30% in Nanopore. Further overviews and comparisons of NGS strategies can be found in REFS 5,20–22.

Simulation parameters

Existing sequencing platforms use distinct protocols that result in data sets with different characteristics[1]. The simulators take into account many of these attributes (FIG. 2), but none of the available tools incorporates all possible variations. The main characteristics of the 23 simulators considered here are summarized in TABLES 2,3. These tools differ in several aspects, such as sequencing technology, input requirements or output format, but have several aspects in common. With some exceptions, all programs need a reference sequence, multiple parameter values that indicate the characteristics of the sequencing experiment to be simulated (read length, error distribution, type of variation to be generated, if any, and so on) and/or a profile (a set of parameter values, conditions and/or data used for controlling the simulation), which can be provided by the simulator or estimated *de novo* from empirical data. The outcome will be aligned or unaligned reads in different standard file formats, such as FASTQ, FASTA or BAM. An overview of the NGS data simulation process is represented in FIG. 3. In the following sections we delve into the different steps involved.

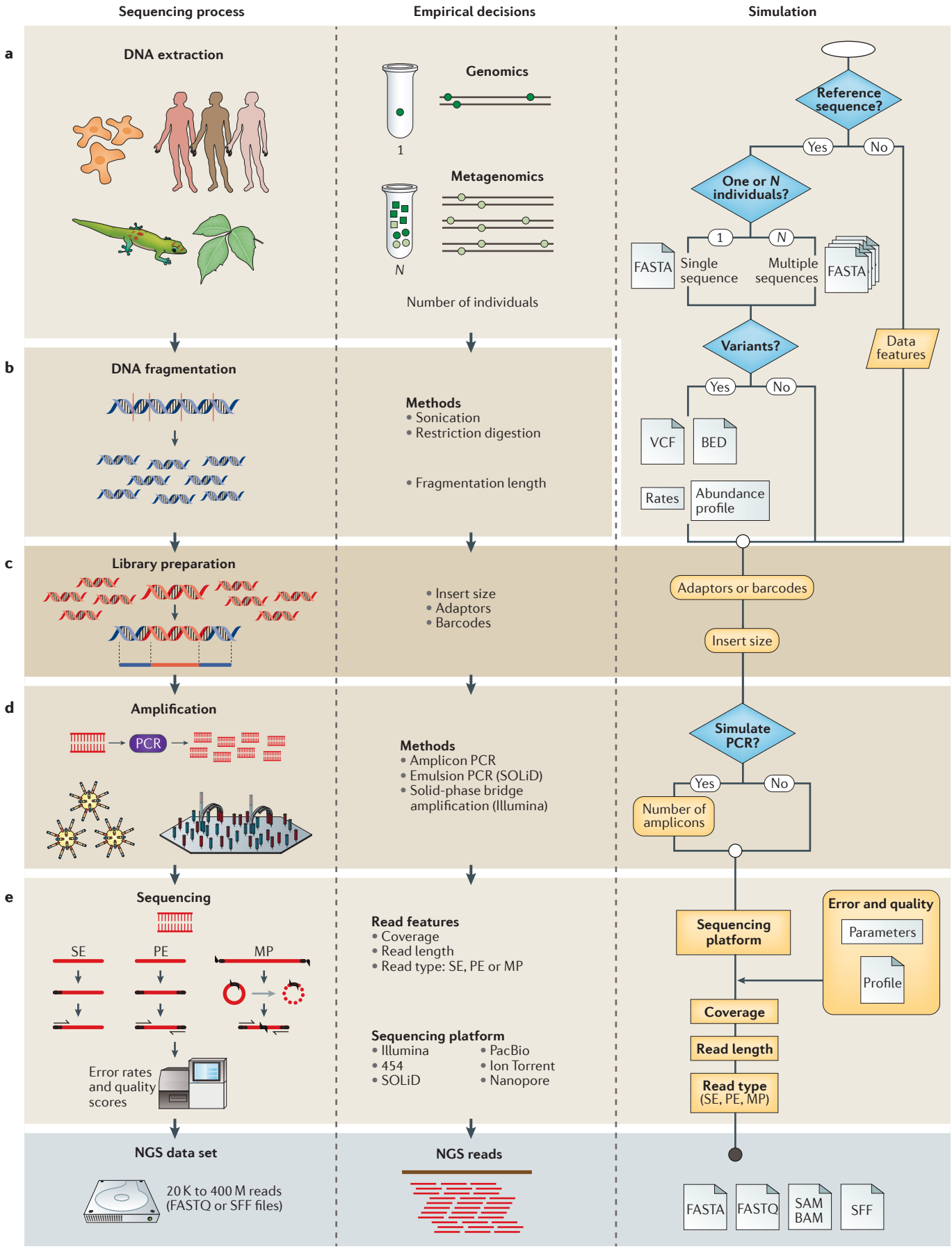
Reference sequence. Most NGS simulators require a reference sequence from which they will generate the simulated reads. This reference sequence can be a particular genomic region, multiple genomic regions that are concatenated, a chromosome or a complete genome. The only exception in this regard is the XS read simulator[23], which only requires the read length, sequencing technology and nucleotide composition to generate completely *de novo* reads. Most of the current NGS simulators use a haploid genome as the reference sequence. Some tools, such as EAGLE, pIRS[9], ReadSim[24] and SimSeq[25], simulate reads from different ploidy levels. In EAGLE and ReadSim, one can specify any ploidy (or even a specific chromosome for EAGLE), whereas pIRS and SimSeq simulate reads from diploid genomes given a haploid reference. Furthermore, several tools are able to generate pools of reads from multiple reference sequences, in some cases using an abundance profile that defines the proportion of reads that are generated from each sequence.

Profiles. Most simulators require the setting of many parameters, which can be done in the command line and/or by using a profile. Profiles can specify parameter distributions or discrete values for different biological features (for example, GC content, insertion and deletion (indel) rates and substitution rates) and/or technological features (for example, insert sizes, read lengths, error rates and quality scores). Note that there are no standard formats for profiles, and the information they include can change for the different tools. Because it might be difficult for many users to decide on particular parameter values or to construct their own profile, some simulators provide default profiles. Alternatively, many tools offer a way to estimate *de novo* profiles from empirical data. Several simulators are able to generate new profiles from alignments of reads mapped to a reference genome (SAM and BAM files) or from real sequencing data from a previous sequencing run (FASTQ files). Thus, BEAR[26], NeSSM[27] and pIRS provide guidelines for the use of alignment and mapping tools, such as BWA[28], BLAST[29], SOAP[30] or SOAP2 (REF. 31), and for error estimation programs, such as DRISEE[32], together with other scripts for parsing the data or for other tasks.

Table 1 | **Main characteristics of current NGS technologies**

Technology	Run type			Maximum read length	Quality scores	Error rates	Refs
	Single end	Paired end	Mate pair				
Illumina	Yes	Yes	Yes	300 bp	>30	0.0034–1%	59
SOLiD	Yes	Yes	Yes	75 bp	>30	0.01–1%	60
IonTorrent	Yes	Yes	No	400 bp	~20	1.78%	22
454	Yes	Yes	No	~700 bp (up to 1 kb)	>20	1.07–1.7%	53,61
Nanopore	Yes	No	No	5.4–10 kb	NA	10–40%	62–66
PacBio	Yes	No	No	~15 kb (up to 40 kb)	<10	5–10%	22,67–69

454, 454 pyrosequencing (Roche); NA, not applicable; Nanopore, Oxford Nanopore Technologies; NGS, next-generation sequencing; PacBio, Pacific Biosciences; SOLiD, sequencing by oligonucleotide ligation and detection (Thermo Fisher).



Amplicon
A piece of DNA or RNA resulting from a natural or artificial amplification event (for example, PCR).

K-mers
The possible sub-sequences of length k that can be obtained from a given sequence.

Coverage
The number of times a certain nucleotide has been sequenced.

Base calling
The analysis of the information obtained from the machine sensors during next-generation sequencing and posterior prediction of the individual bases. This converts the signal into actual sequence data with quality scores.

The ART[8], FASTQsim[13], GemSim[16], SimSeq[25] and SInC[17] packages provide their own standalone tools for the generation of error, quality and/or abundance profiles. ART and SInC generate quality profiles based on specific error models and/or the quality-score distribution that is extracted from empirical data. NeSSM generates quality and error profiles. The quality profiles define the quality score given to each base along the read and are estimated based on an existing set of reads. The error profiles define the proportion of the different error types (substitutions and indels) and are estimated with specific scripts. pIRS generates quality profiles using mapped reads and known variations from re-sequencing data. The program BEAR (which is focused on metagenomics) generates error, quality and abundance profiles. For the generation of the error profile it uses a modified version of DRISEE to infer error rates by clustering artefactual duplicate reads in the supplied data set. For the quality profile it uses the output of the error model to determine the average quality score assigned to erroneous nucleotides per position per read[26]. In addition, it generates an abundance profile from the relative frequency of the different taxa in a metagenomic data set.

Finally, other simulation programs such as ArtificialFastqGenerator[33] and CuReSim[10] do not use a profile; their simulation parameter values are specified directly through the command-line.

Accounting for PCR amplification

DNA amplification by PCR is currently a necessary step in the preparation of libraries for the Illumina, 454, IonTorrent and SOLiD[34,35] sequencing platforms. One may be interested, therefore, in modelling the bias introduced by PCR[1,36,37], as done by ART, Flowsim[38,39] and Grinder[7].

ART, which simulates reads for Illumina, 454 and SOLiD, can mimic PCR bias by specifying the number of reads (single end or paired end) generated per amplicon[8]. Flowsim is a suite of executable modules that simulate the entire 454 pyrosequencing process; using the 'kitsim' module, one can simulate the attachment of adaptors to the end of each amplicon, which then serve as primers for PCR amplification simulated by the 'duplicator' module[38,39].

Grinder was specifically developed to simulate amplicon sequencing from user-supplied PCR primer collections by introducing known experimental artefacts, such as chimaeras[37] and spurious CNVs (copy number variants). Grinder can generate chimaeras in two ways: by appending consecutive segments at given breakpoints, in which both amplicon sequences and breakpoints are randomly selected; and by concatenating fragments at breakpoints determined by specific k-mers that must be shared by the amplicons. In addition, the presence of several gene copies in a genome may affect the composition of the amplicon library, contributing extra amplicon reads. Grinder models this bias by sampling species proportionally to their relative abundance and to the number of copies of the amplicon in their genome[7].

Read features

In an NGS experiment, the number, length and type of reads are determined by the specific sequencing machine and the library preparation. It is possible to simulate a specific amount of reads with different lengths and types according to the sequencing technology assumed. The number of reads can be specified or estimated according to the desired coverage. It is also possible to select a fixed length — the length of the longest read or a length distribution. The read type can be specified directly or indirectly by defining particular insert sizes. By default, most simulators assume single-end reads.

Base-calling errors

NGS technologies rely on a complex interplay between chemistry, hardware and optical sensors. Adding to this complexity is the software that analyses the sensor data and predicts the individual bases; this is referred to as base calling[40]. Base calling converts the signals into actual sequence data with quality scores (also known as Phred *Q* scores[41,42]). The different sequencing platforms usually assume an explicit error model in order to assign a measure of uncertainty to each base call[43].

Error-rate models determine the probability of erroneous substitutions or indels at a given position within a read[26,44]. For the generation of realistic reads, it is necessary to understand and incorporate as much as possible of the different sources of sequencing error. Each sequencing platform has a specific error rate (TABLE 1), which can also vary within the same technology and among reads[16]. The importance of taking this into account and simulating sequencing data based on specific error models should not be underestimated.

Simulators may generate sequence errors in different ways: based on the quality scores (ArtificialFastqGenerator); by introducing particular

◀ Figure 2 | **General overview of the sequencing process and steps that can be parameterized in the simulations.** Next-generation sequencing (NGS) simulators try to imitate the real sequencing process as closely as possible by considering all the steps that could influence the characteristics of the reads. **a** | NGS simulators do not take into account the effect of the different DNA extraction protocols on the resulting data. However, they can consider whether the sample to be sequenced includes one or more individuals, from the same or different organisms (for example, pooled sequencing and metagenomics). Pools of related genomes can be simulated by replicating the reference sequence and introducing variants into the resulting genomes. Some tools can also simulate metagenomes with distinct taxa abundance. **b** | Simulators can try to mimic the length range of DNA fragmentation (empirically obtained by sonication or digestion protocols) or assume a fixed amplicon length. **c** | Library preparation involves the ligation of sequencing-platform-dependent adaptors or barcodes (blue) to the selected DNA fragments (red). Some simulators can control the insert size and produce reads with adaptors or barcodes. **d** | Most NGS techniques include an amplification step for the preparation of libraries. Several simulators can take this step into account (for example, by introducing errors and/or chimaeras), with the possibility of specifying the number of reads per amplicons. **e** | Sequencing runs imply a decision about coverage, read length, read type (single end (SE), paired end (PE) or mate pair (MP)) and a given platform (with their specific errors and biases). Simulators exist for the different platforms, and they can use particular parameter profiles, often estimated from real data. 454, 454 pyrosequencing (Roche); K, thousand; M, million; Nanopore, Oxford Nanopore Technologies; PacBio, Pacific Biosciences; SFF, standard flowgram format; SOLiD, sequencing by oligonucleotide ligation and detection (Thermo Fisher); VCF, variant call format.

Table 2 | **General information about 23 NGS genomic simulators***

Simulator	Technology	G vs M	Run types	Ref seq	Characterization							Processes			Outputs		
					Input			Profile process									
					PA	RE	PR	DF	PA	GU	SW	PCR	GV	QS	RE	AL	FO
454sim	454	G	SE	Yes	No	No	Yes	Yes	No	No	No	No	No	Yes	Yes	No	SFF
ART	454, Illumina and SOLiD	G	SE, PE and MP	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	SFF and FQ
ArtificialFastqGenerator	Illumina	G	PE	Yes	Yes	Yes	No	No	Yes	No	No	No	No	Yes	Yes	No	FQ
BEAR	454, Illumina and IonTorrent	G and M	SE and PE	Yes	No	Yes	No	No	No	Yes	No	No	Yes	Yes	Yes	No	FQ
CuReSim	454, Illumina, SOLiD and IonTorrent	G	SE	Yes	Yes	No	No	No	Yes	No	No	No	No	Yes	No	No	FQ
DWGSIM (DNA analysis)	Illumina, SOLiD and IonTorrent	G	SE, PE and MP	Yes	Yes	No	Yes	No	Yes	No	No	No	Yes	Yes	Yes	No	FQ
EAGLE	454, Illumina, PacBio and IonTorrent	G	SE and PE	Yes	No	No	Yes	Yes	No	No	No	No	Yes	Yes	Yes	Yes	FQ
FASTQSim	Illumina, SOLiD, PacBio and IonTorrent	G and M	SE	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	FQ
Flowsim	454	G	SE and PE	Yes	Yes	No	Yes	Yes	Yes	No	No	Yes	No	Yes	Yes	No	SFF
GemSim	454 and Illumina	G and M	SE and PE	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	No	FQ
Grinder	454, Illumina and Sanger	G and M	SE, PE and MP	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No	FQ
Mason	454, Illumina and Sanger	G	SE, PE and MP	Yes	Yes	No	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	FA and FQ
MetaSim	454, Illumina and Sanger	G and M	SE, PE and MP	Yes	Yes	No	No	No	Yes	No	No	No	No	No	Yes	No	FA
NeSSM	454 and Illumina	M	SE and PE	Yes	No	No	Yes	No	No	Yes	No	No	Yes	Yes	Yes	No	FQ
pbsim	PacBio	G	CLR and CCS	Yes	Yes	No	No	Yes	No	No	No	No	No	Yes	Yes	Yes	FQ
pIRS	Illumina	G and M	PE	Yes	Yes	No	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	No	FQ
ReadSim	PacBio and Nanopore	G	SE	Yes	Yes	No	No	No	Yes	No	No	No	Yes	Yes	Yes	No	FQ
simhtsd	454 and Illumina	G	SE and PE	Yes	Yes	No	No	No	Yes	No	No	No	No	Yes	No	No	FQ
simNGS	Illumina	G	SE and PE	Yes	Yes	No	Yes	Yes	No	No	No	No	No	Yes	Yes	No	FQ
SimSeq	Illumina	G	SE, PE and MP	Yes	Yes	No	Yes	Yes	No	No	Yes	No	Yes	Yes	No	Yes	SAM and BAM‡
SInC	Illumina	G	PE	Yes	No	Yes	Yes	No	No	No	Yes	No	Yes	Yes	Yes	No	FQ
wgsim	Illumina and SOLiD	G	SE	Yes	Yes	No	No	No	Yes	No	No	No	Yes	Yes	Yes	No	FQ
XS	454, Illumina, SOLiD and IonTorrent	G	SE and PE	No	Yes	No	No	No	Yes	No	No	No	No	Yes	Yes	No	FQ

454, 454 pyrosequencing; AL, alignments; CCS, circular consensus sequencing; CLR, continuous long read; DF, default profile; FA, FASTA; FO, format; FQ, FASTQ; G, genomics; GU, guide to generate profiles; GV, genomic variants; M, metagenomics; MP, mate pair; Nanopore, Oxford Nanopore Technologies; NGS, next-generation sequencing; PA, parameter; PacBio, Pacific Biosciences; PE, paired end; PR, profile; QS, quality score; RE, reads; Ref seq, reference sequence; Sanger, Sanger sequencing; SE, single end; SFF, standard flowgram format; SOLiD, sequencing by oligonucleotide ligation and detection; SW, specific software to generate profile. *An extended version of this table is available at http://darwin.uvigo.es/ngs-simulators. ‡Compressed SAM file.

Homopolymers
Sequences of multiple identical nucleotides.

errors at specific positions (as in SimSeq); by using specific error parameters for each platform or technology, which can be either user defined (as in ART, Mason[6] and pIRS) or fixed by the program (as in DWGSIM and FASTQsim); using variable error rates within reads (as in simhtsd and wgsim); using error distributions (Grinder); or generating specific errors along with some noise (as in simNGS). In the following subsections we describe in more detail the different errors that are modelled, their occurrence in sequencing platforms, and how they are implemented on the different simulators.

Indel errors. It has been reported that Illumina platforms rarely contain indel errors[9], whereas indels are actually the main source of error for 454 and IonTorrent, although they occur at very low rates[45]. However, in 454, assessing the correct number of polynucleotide sites (homopolymers) is often quite difficult because changes in the light signals among homopolymers with similar lengths can be undetectable[5,46–50]. PacBio yields long single-molecule reads that are prone to false indels from non-fluorescing nucleotides[46,48], which are stochastically modelled by the PacBio reads simulator (pbsim)[51]. With Nanopore it is also possible to have indel errors; insertions occur when the strand slips back and forth so that a given position is read more than once, and deletions occur when the rate of strand displacement in the pore sensor exceeds the rate of data acquisition[51]. ReadSim, which is so far the only simulator available for Nanopore, assumes fixed error rates for indels and

Table 3 | **Technical information about 23 NGS genomic simulators**

Simulators	Programming language	Operating system	Interface	Processing	License	Open source?
454sim	C++ and Perl	Winows, Linux and Mac OS	CLI	NP and P	GNU GPL v1	Yes
ART	C++ and Perl	Windows, Linux and Mac OS	CLI	P	GNU GPL	Yes
ArtificialFastqGenerator	Java	Windows, Linux and Mac OS	CLI	P	GNU GPL v3	Yes
BEAR	Python and Perl	Linux	CLI	P	AU	Yes
CuReSim	Java	Windows, Linux and Mac OS	CLI	P	NA	No
DWGSIM (DNA analysis)	C, Perl and Python	Linux	CLI	P	GNU GPL v2	Yes
EAGLE	C++	Linux	CLI	NP and P	BSD	Yes
FASTQSim	Bash and Python	Linux	CLI	NP and P	GNU GPL v3	Yes
Flowsim	Haskell	Linux	CLI	P	GNU	Yes
GemSim	Python	Windows, Linux and Mac OS	CLI	P	GNU GPL v3	Yes
Grinder	Perl	Windows, Linux and Mac OS	CLI, GUI and API	P	GPL	Yes
Mason	C++	Windows, Linux and Mac OS	CLI	P	GNU GPL and GNU LGPL	Yes
MetaSim	Java	Windows, Linux and Mac OS	CLI and GUI	P	PRO and AU	No
NeSSM	C, CUDA and Perl	Linux	CLI	NP and P	AU	Yes
pbsim	C++	Linux	CLI	P	GNU GPL v2	Yes
pIRS	C++ and Perl	Linux	CLI	NP and P	GNU GPL v2	Yes
ReadSim	Python	Windows, Linux and Mac OS	CLI	P	NA	Yes
simhtsd	Perl	Linux	CLI	P	GNU GPL v3	Yes
simNGS	C	Linux and Mac OS	CLI	P	GNU GPL v3	Yes
SimSeq	Java	Linux	CLI	P	MIT	Yes
SInC	C++	Linux	CLI	NP and P	CCANCL V2.0	No
wgsim	C	Linux	CLI	P	MIT	Yes
XS	C++	Linux	CLI	P	GNU GPL v3	Yes

API, application programming interface; AU, academic use only; BSD, Berkeley software distribution; CCANCL, creative commons attribution non-commercial license; CLI, command line interface; GPL, general public license; GUI, graphical user interface; LGPL, lesser general public license; MIT, Massachusetts Institute of Technology; NA, not applicable; NGS, next-generation sequencing; NP, no parallel processing; P, parallel processing that accepts multi-threading; PRO, proprietary software.

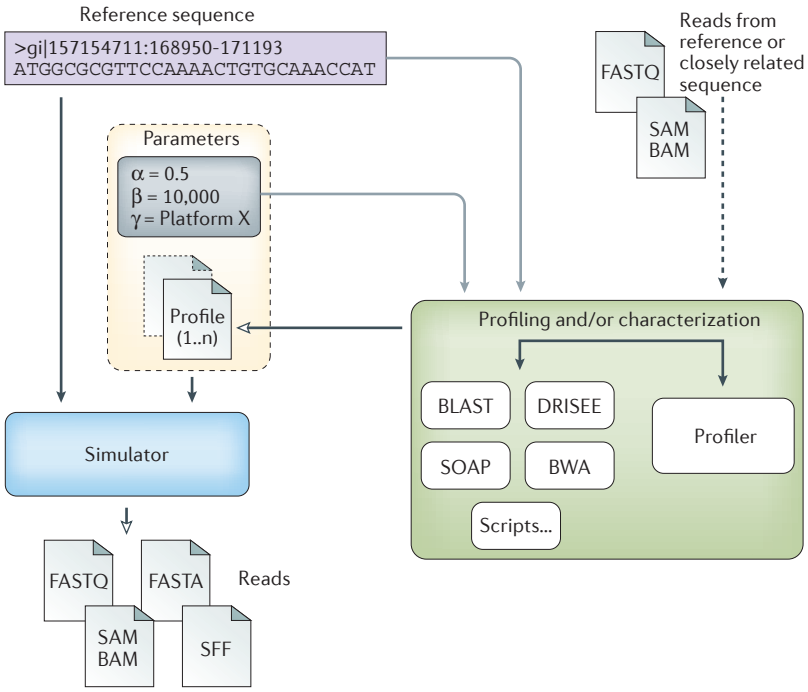


Figure 3 | **General overview of NGS simulation.** The next-generation sequencing (NGS) simulation process begins with the input of a reference sequence (in most cases) and simulation parameters. Some of the parameters can be given by using a profile that is estimated (by the simulator or other tools) from other reads or alignments. The outcome of this process may be reads (with or without quality information) or genome alignments in different formats. SFF, standard flowgram format.

substitutions. Indel rates can be specified through the command line or by using a configuration profile, such as for ART, CuReSim, Grinder, Mason, MetaSim[52], NeSSM, pbsim, ReadSim, SInC and XS. Some programs, such as BEAR, EAGLE and GemSim, include utilities or use external tools like DRISEE to estimate indel rates from FASTQ or SAM files. Conversely, 454 and IonTorrent homopolymer-specific errors[53] may be extracted from a profile that determines the position and corresponding error rate (as in ART), or they may be introduced under the form of homopolymeric stretches using a specified empirical model (as in MetaSim, Flowsim or Grinder).

Substitution errors. Substitution errors are dominant in Illumina and SOLiD platforms. These may occur when incorrect bases are introduced during clonal amplification of templates (for example, by PCR)[9,48,54] or when the optical signals are translated into bases. In the latter process, a green laser is used to detect G and T nucleotides at the same time, afterwards using a filter to distinguish between G and T. A and C nucleotides are detected in a similar way but using a red laser. Thus, base-calling errors may arise because of insufficient discrimination of the respective base emission spectra[45]. It is also known that SOLiD sequencers are unable to read through palindromic regions, presumably owing to the formation of hairpin structures, and therefore such regions are interpreted as miscellaneous random sequences. ART simulates this kind of error. As with indels, substitution error rates have to be defined in the command line or within a profile.

Some NGS platforms can produce position-specific substitution errors, with reads having significantly lower quality in the later cycles. In Illumina, these type of errors possibly arise from either single-strand DNA folding or sequence-specific alterations in enzyme preference[1,46,48,54] and can be modelled by GemSim and pIRS. Similar errors can be observed for 454 platforms[53]. Flowsim, 454sim and MetaSim can simulate two kinds of sequencing flows with a degradation model. The positive flow, interpreted as the occurrence of one or more bases, is modelled as a normal distribution; the negative flow, with no base or noise, is modelled as a log-normal distribution. The degradation model is introduced as a standard deviation that gradually increases the probability of error along the sequence.

Quality scores

The quality score is a prediction of the probability of an error in a base call[41,42,44,55]. The distribution of base quality scores is position dependent, and the mean quality score decreases as a function of increasing base position for most of the available technologies[8]. Some NGS read simulators separate the quality score from sequencing error, even though they are correlated measurements. Several strategies can be used to simulate the quality scores, in most cases using empirical information. 454sim, EAGLE, Flowsim and simNGS use fixed quality score profiles that are based on previous studies. ART, ArtificialFastqGenerator, BEAR, FASTQsim, GemSim, NeSSM, SimSeq and SInC also include utilities that allow the user to derive quality profiles from FASTQ files. Conversely, pIRS determines both the base and quality score in relation to the cycle number and to the base position on the simulated read, using empirical parameters. Alternatively, the distribution of the quality scores can be controlled by the user. Some programs use a simple parameter that determines a fixed quality score for every read, such as ArtificalFastqGenerator, CuReSim, DWGSIM, ReadSim, simhtsd, wgsim and XS. Grinder assigns two quality scores, depending on whether the simulated base call is correct or not. More complex and realistic simulators use a Gaussian distribution (as in XS) or a position-specific normal distribution (as in Mason) with mean, standard deviation and quality standard deviation for the first and last base. For PacBio, the distribution of errors is considered to be constant along the chromosomes[22], and programs like pbsim use a uniform distribution to assign the quality scores. In Illumina, each paired-end read can have equal or different quality scores. The simulators ArtificialFastqGenerator, DWGSIM, EAGLE, SimSeq and SInC explicitly allow two different quality distributions for paired-end reads.

Sequencing depth

Sequencing depth or coverage is not continuous along genomes. This can be due to chance[56] but also to the GC bias introduced during DNA amplification by PCR[57,58],

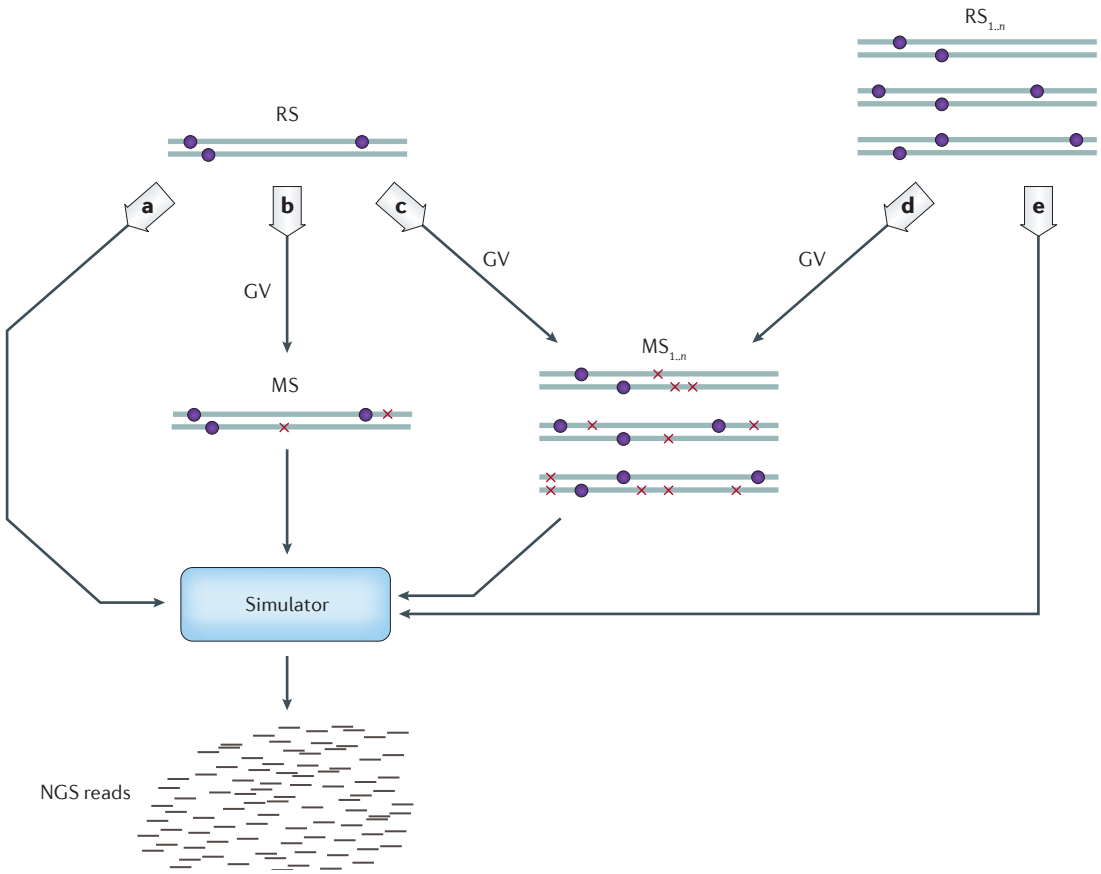


Figure 4 | **Flows available to generate reads with and without genomic variation.** Dots represent variants that are present in the reference sequence (RS), and crosses represent the newly introduced variants (mutated sequences; MSs). **a** | Simulation of reads from a single RS without adding new genomic variants (GVs). **b** | Generation of reads from a single MS generated from a single RS. **c** | Reads generated from a set of MSs that were generated from a single reference sequence. **d** | Generation of reads from a set of MSs obtained from a set of RSs. **e** | Reads obtained directly from a set of RSs without introducing additional GVs. NGS, next-generation sequencing.

as sequencing depth increases in regions with elevated GC content[36,45]. This coverage bias is taken into account by ArtificialFastqGenerator, BEAR, EAGLE, NeSSM and pIRS. ArtificialFastqGenerator calculates the GC content of different genomic regions from the reference sequence and then samples coverage levels for these regions from a normal distribution. BEAR, EAGLE, NeSSM and pIRS use data from previous studies to determine the variation of the GC content along the reference sequence, resulting in the simulation of variable regional coverage.

Simulating genomic variants

Apart from sequencing errors (FIG. 4), many tools can also introduce different types of genomic variants in the simulated reads[17], such as SNPs (single-nucleotide polymorphisms), indels, inversions, translocations, CNVs and short tandem repeats (STRs) (TABLE 4).

The general strategy is to create a mutated sequence by introducing genomic variants in the reference sequence before the generation of reads (FIG. 4). In most cases, these variants are simulated using a given mutation rate, so the mutated sequence differs by a given percentage from the reference sequence; however, programs like DWGSIM and EAGLE require a file with known mutations (in plain text, variant cell format (VCF) or BED-like format). FASTQsim includes a separate tool that builds a mutation file from real data, using an NGS data set (FASTQ files) and a reference genome, and is best suited for re-sequencing.

Some programs are capable of generating population-level diversity by creating several mutated sequences from a single reference sequence (FIG. 4). Programs like GemSim and Mason can generate sets of related haplotypes differing by at least one SNP from the reference sequence. In GemSim, users may also create their own tab-delimited haplotype file providing the specific position and mutation introduced.

Tools like GemSim, BEAR, Grinder and NeSSM can introduce genomic variants in a given set of reference sequences belonging to different taxa to create a set of mutated genomes that resemble a metagenomic community (FIG. 4). As mentioned above, these programs use an abundance profile so the reads are generated from these sequences with a probability proportional to 'taxa' abundance.

Table 4 | **Genomic variants**

Simulators	Genomic variants*							
	MGC	PLO	SNPs	Indels	INVs	TRA	CNVs	STRs
BEAR	Yes	No	No	No	No	No	No	No
DWGSIM (DNA analysis)	No	Yes	Yes	Yes	Yes	Yes	No	No
EAGLE	No	Yes	Yes	Yes	Yes	Yes	Yes	No
FASTQSim	No	No	Yes	Yes	No	No	No	Yes
GemSim	Yes	No	Yes	Yes	No	No	No	No
Grinder	Yes	No	Yes	Yes	No	No	No	No
Mason	No	No	Yes	Yes	No	No	No	No
NeSSM	Yes	No	No	No	No	No	No	No
pIRS	No	Yes	Yes	Yes	Yes	No	No	No
ReadSim	No	Yes	Yes	Yes	Yes	No	No	No
SimSeq	No	Yes	No	No	No	No	No	No
SInC	No	No	Yes	Yes	No	No	Yes	No
wgsim	No	Yes	Yes	Yes	No	No	No	No

CNVs, copy number variants; indels, insertions and/or deletions; INVs, inversions; MGC, metagenomic community; NA, not applicable; PLO, ploidy; SNPs, single-nucleotide polymorphisms; STRs, short tandem repeats; TRA, translocation. *Variation that can be introduced in the reference sequences.

Output

The generated NGS reads may be stored in different file formats. According to the specific NGS technology simulated, one can obtain SFF (standard flowgram format) files from 454 platforms (such as 454sim and Flowsim) and FASTA or FASTQ files from IonTorrent, Illumina, PacBio, SOLiD and Nanopore. Other possible output files include alignment files, either in MAF (multiple alignment format) or SAM and BAM formats. These can be outputted by default (as in Mason, pbsim and SimSeq) or as an option, complementary to the simulated reads (as in ART).

Conclusions

NGS is having a big influence in a broad range of areas that benefit from genetic information, from medical genomics and phylogenetic and population genomics to the reconstruction of ancient genomes, epigenomics and environmental barcoding. These applications include approaches such as *de novo* sequencing, re-sequencing, target sequencing or genome reduction methods. In all cases, caution is necessary in choosing a proper sequencing design and/or a reliable analytical approach for the specific biological question of interest. The simulation of NGS data can be extremely useful for planning experiments, testing hypotheses, benchmarking tools and evaluating particular results. Given a reference genome or data set, for instance, one can investigate an array of sequencing technologies to choose the best-suited technology and parameters for a particular goal, possibly optimizing time and costs. Yet, this is still not the standard practice and researchers often base their choices on practical considerations, such as technology and availability of money. As shown throughout this Review, the simulation of NGS data from known genomes or transcriptomes can be extremely useful when evaluating assembly, mapping, phasing or genotyping algorithms[2,7,10,13,58], exposing their advantages and drawbacks under different circumstances.

Altogether, current NGS simulators consider most, if not all, of the important features regarding the generation of NGS data; however, they are not problem-free. The different simulators are largely redundant, implementing the same or very similar procedures. In our opinion, many are poorly documented and can be difficult to use for non-experts, and some of them are no longer maintained. Most importantly, they have largely not been benchmarked or validated. Remarkably, among the 23 tools considered here, only 13 have been described in dedicated application notes, 3 have been mentioned as add-ons in the methods section of bigger articles and 5 have never been referenced in a journal. Indeed, peer-reviewed publication of these tools in dedicated articles would be highly desirable. Although this would not definitively guarantee quality, at least it would encourage authors to reach minimum standards in terms of validation, benchmarking and documentation. Collaborative efforts like the Assemblathon[25] or iEvo might also be a source of inspiration. Meanwhile, we hope that the decision tree presented in FIG. 1 helps users in making appropriate choices.

1. Metzker, M. L. Sequencing technologies — the next generation. *Nat. Rev. Genet.* **11**, 31–46 (2010).
2. Nielsen, R., Paul, J. S., Albrechtsen, A. & Song, Y. S. Genotype and SNP calling from next-generation sequencing data. *Nat. Rev. Genet.* **12**, 443–451 (2011).
3. Koboldt, D. C., Steinberg, K. M., Larson, D. E., Wilson, R. K. & Mardis, E. R. The next-generation sequencing revolution and its impact on genomics. *Cell* **155**, 27–38 (2013).
4. Wang, X. V., Blades, N., Ding, J., Sultana, R. & Parmigiani, G. Estimation of sequencing error rates in short reads. *BMC Bioinformatics* **13**, 185 (2012).
5. Liu, L. *et al.* Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.* **2012**, 1–11 (2012).
6. Holtgrewe, M. Mason — a read simulator for second generation sequencing data. http://publications.mi.fu-berlin.de/962 (FU Berlin, 2010).
7. Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P. & Tyson, G. W. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Res.* **40**, e94 (2012).
8. Huang, W., Li, L., Myers, J. R. & Marth, G. T. ART: a next-generation sequencing read simulator. *Bioinformatics* **28**, 593–594 (2012).
This paper describes probably the most popular NGS simulator nowadays, with well-supported and detailed documentation.
9. Hu, X. *et al.* pIRS: profile-based Illumina pair-end reads simulator. *Bioinformatics* **28**, 1533–1535 (2012).
10. Caboche, S., Audebert, C., Lemoine, Y. & Hot, D. Comparison of mapping algorithms used in high-throughput sequencing: application to Ion Torrent data. *BMC Genomics* **15**, 264 (2014).
11. Hoban, S., Bertorelle, G. & Gaggiotti, O. E. Computer simulations: tools for population and evolutionary genetics. *Nat. Rev. Genet.* **13**, 110–122 (2012).
12. Shendure, J. & Aiden, E. L. The expanding scope of DNA sequencing. *Nat. Biotechnol.* **30**, 1084–1094 (2012).
13. Shcherbina, A. FASTQSim: platform-independent data characterization and *in silico* read generation for NGS datasets. *BMC Res. Notes* **7**, 533 (2014).
14. Knudsen, B., Forsberg, R. & Miyamoto, M. M. A computer simulator for assessing different challenges and strategies of *de novo* sequence assembly. *Genes* **1**, 263–282 (2010).
15. Mavromatis, K. *et al.* Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nat. Methods* **4**, 495–500 (2007).
This paper describes the use of NGS simulations for benchmarking NGS analytical methods.
16. McElroy, K. E., Luciani, F. & Thomas, T. GemSIM: general, error-model based simulator of next-generation sequencing data. *BMC Genomics* **13**, 74 (2012).
17. Pattnaik, S., Gupta, S., Rao, A. A. & Panda, B. 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* **15**, 40 (2014).
18. Rothberg, J. M. *et al.* An integrated semiconductor device enabling non-optical genome sequencing. *Nature* **475**, 348–352 (2011).

19. Eid, J. *et al.* Real-time DNA sequencing from single polymerase molecules. *Science* **323**, 133–138 (2009).
20. Shendure, J. & Ji, H. Next-generation DNA sequencing. *Nat. Biotechnol.* **26**, 1135–1145 (2008).
21. Shendure, J., Mitra, R. D., Varma, C. & Church, G. M. Advanced sequencing technologies: methods and goals. *Nat. Rev. Genet.* **5**, 335–344 (2004).
22. Quail, M. *et al.* A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics* **13**, 341 (2012).
23. Pratas, D., Pinho, A. J. & O. S. Rodrigues, J. M. XS: a FASTQ read simulator. *BMC Res. Notes* **7**, 40 (2014).
24. Lee, H. *et al.* Error correction and assembly complexity of single molecule sequencing reads. *bioRxiv* http://dx.doi.org/10.1101/006395 (2014).
25. Earl, D. *et al.* Assemblathon 1: a competitive assessment of *de novo* short read assembly methods. *Genome Res.* **21**, 2224–2241 (2011).
26. Johnson, S., Trost, B., Long, J. R., Pittet, V. & Kusalik, A. A better sequence-read simulator program for metagenomics. *BMC Bioinformatics* **15**, S14 (2014).
27. Jia, B. *et al.* NeSSM: a next-generation sequencing simulator for metagenomics. *PLoS ONE* **8**, e75448 (2013).
28. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
29. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
30. Li, R., Li, Y., Kristiansen, K. & Wang, J. SOAP: short oligonucleotide alignment program. *Bioinformatics* **24**, 713–714 (2008).
31. Li, R. *et al.* SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* **25**, 1966–1967 (2009).
32. Keegan, K. P. *et al.* A platform-independent method for detecting errors in metagenomic sequencing data: DRISEE. *PLoS Comput. Biol.* **8**, e1002541 (2012).
33. Frampton, M. & Houlston, R. Generation of artificial FASTQ files to evaluate the performance of next-generation sequencing pipelines. *PLoS ONE* **7**, e49110 (2012).
34. Mardis, E. R. The impact of next-generation sequencing technology on genetics. *Trends Genet.* **24**, 133–141 (2008).
35. Morozova, O. & Marra, M. A. Applications of next-generation sequencing technologies in functional genomics. *Genomics* **92**, 255–264 (2008).
36. Aird, D. *et al.* Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**, R18 (2011).
37. Haas, B. J. *et al.* Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.* **21**, 494–504 (2011).
38. Balzer, S., Malde, K., Lanzén, A., Sharma, A. & Jonassen, I. Characteristics of 454 pyrosequencing data — enabling realistic simulation with flowsim. *Bioinformatics* **27**, i420–i425 (2010).
This paper presents one of the most popular simulators for 454 pyrosequencing long reads.
39. Balzer, S., Malde, K. & Jonassen, I. Systematic exploration of error sources in pyrosequencing flowgram data. *Bioinformatics* **27**, 304–309 (2011).
40. Ledergerber, C. & Dessimoz, C. Base-calling for next-generation sequencing platforms. *Brief. Bioinform.* **12**, 489–497 (2011).
41. Ewing, B. *et al.* Base-calling of automated sequencer traces using *phred*. I. Accuracy assessment. *Genome Res.* **8**, 175–185 (1998).
42. Ewing, B. *et al.* Base-calling of automated sequencer traces using *phred.* II. Error probabilities. *Genome Res.* **8**, 186–194 (1998).
43. Kao, W.-C., Stevens, K. & Song, Y. S. BayesCall: a model-based base-calling algorithm for high-throughput short-read sequencing. *Genome Res.* **19**, 1884–1895 (2009).
44. Illumina. Technical note: Sequencing. Quality scores for next-generation sequencing: assessing sequencing accuracy using Phred quality scoring. *Illumina* http://www.illumina.com/documents/products/technotes/technote_Q-Scores.pdf (2011).
45. Dohm, J. C., Lottaz, C., Borodina, T. & Himmelbauer, H. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.* **36**, e105 (2008).
This paper describes the most relevant biases that affect the generation of NGS data.
46. Kircher, M. & Kelso, J. High-throughput DNA sequencing - concepts and limitations. *BioEssays* **32**, 524–536 (2010).
47. Loman, N. J. *et al.* Performance comparison of benchtop high-throughput sequencing platforms. *Nat. Biotechnol.* **30**, 434–439 (2012).
48. Robasky, K., Lewis, N. E. & Church, G. M. The role of replicates for error mitigation in next-generation sequencing. *Nat. Rev. Genet.* **15**, 56–62 (2013).
49. Yang, X., Chockalingam, S. P. & Aluru, S. A survey of error-correction methods for next-generation sequencing. *Brief. Bioinform.* **14**, 56–66 (2013).
50. Ekblom, R., Smeds, L. & Ellegren, H. Patterns of sequencing coverage bias revealed by ultra-deep sequencing of vertebrate mitochondria. *BMC Genomics* **15**, 467 (2014).
51. Ono, Y., Asai, K. & Hamada, M. PBSIM: PacBio reads simulator — toward accurate genome assembly. *Bioinformatics* **29**, 119–121 (2013).
This paper presents one of the most popular simulators for the PacBio sequencing platform.
52. Richter, D. C., Ott, F., Auch, A. F., Schmid, R. & Huson, D. H. MetaSim — a sequencing simulator for genomics and metagenomics. *PLoS ONE* **3**, e3373 (2008).
53. Margulies, M. *et al.* Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437**, 376–380 (2005).
54. Nakamura, K. *et al.* Sequence-specific error profile of Illumina sequencers. *Nucleic Acids Res.* **39**, e90 (2011).
55. Kwon, S., Park, S., Lee, B. & Yoon, S. In-depth analysis of interrelation between quality scores and real errors in Illumina reads. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* **2013**, 635–638 (2013).
56. Lander, E. S. & Waterman, M. S. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics* **2**, 231–239 (1988).
57. Sims, D., Sudbery, I., Ilott, N. E., Heger, A. & Ponting, C. P. Sequencing depth and coverage: key considerations in genomic analyses. *Nat. Rev. Genet.* **15**, 121–132 (2014).
58. Li, B. *et al.* Evaluation of *de novo* transcriptome assemblies from RNA-Seq data. *Genome Biol.* **15**, 553 (2014).
59. Ross, M. G. *et al.* Characterizing and measuring bias in sequence data. *Genome Biol.* **14**, R51 (2013).
60. Glenn, T. C. Field guide to next-generation DNA sequencers. *Mol. Ecol. Resour.* **11**, 759–769 (2011).
61. Gilles, A. *et al.* Accuracy and quality assessment of 454 GS-FLX Titanium pyrosequencing. *BMC Genomics* **12**, 245 (2011).
62. Quick, J., Quinlan, A. R. & Loman, N. J. A reference bacterial genome dataset generated on the MinION portable single-molecule nanopore sequencer. *GigaScience* **3**, 22 (2014).
63. Loman, N. J., Quick, J. & Simpson, J. T. A complete bacterial genome assembled de novo using only nanopore sequencing data. *bioRxiv* http://dx.doi.org/10.1101/015552 (2015).
64. Jain, M. *et al.* Improved data analysis for the MinION nanopore sequencer. *Nat. Methods* **12**, 351–356 (2015).
65. Laver, T. *et al.* Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomol. Detect. Quantif.* **3**, 1–8 (2015).
66. Madoui, M.-A. *et al.* Genome assembly using Nanopore-guided long and error-free DNA reads. *BMC Genomics* **16**, 327 (2015).
67. Carneiro, M. O. *et al.* Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC Genomics* **13**, 375 (2012).
68. Koren, S. *et al.* Hybrid error correction and *de novo* assembly of single-molecule sequencing reads. *Nat. Biotechnol.* **30**, 693–700 (2012).
69. Salmela, L. & Rivals, E. LoRDEC: accurate and efficient long read error correction. *Bioinformatics* **30**, 3506–3514 (2014).

Acknowledgements
This work was supported by the European Research Council (ERC-617457- PHYLOCANCER to D.P.) and the Spanish Government (research grants BFU2012-33038 and BFU2015-63774-P to D.P.; Research Personnel Training (FPI) graduate fellowship BES-2013-067181 to M.E.; and a Juan de la Cierva postdoctoral fellowship (FPDI-2013-17503 to S.R.). The authors thank two anonymous reviewers and members of the phylogenomics laboratory for their comments.

Competing interests statement
The authors declare no competing interests.

FURTHER INFORMATION
454sim: http://sourceforge.net/projects/bioinfo-454sim
ART: http://www.niehs.nih.gov/research/resources/software/biostatistics/art
ArtificialFastqGenerator: http://sourceforge.net/projects/artfastqgen
BEAR: https://github.com/sej917/BEAR
CuReSim: http://www.pegase-biosciences.com/curesim-a-customized-read-simulator
DWGSIM: https://github.com/nh13/DWGSIM
EAGLE: https://github.com/sequencing/EAGLE
FastqSim: http://sourceforge.net/projects/fastqsim
Flowsim: http://biohaskell.org/Applications/FlowSim
GemSim: http://sourceforge.net/projects/gemsim
Grinder: http://sourceforge.net/projects/biogrinder
iEvo: http://www.ievobio.org
Mason: http://www.seqan.de/projects/mason
MetaSim: http://ab.inf.uni-tuebingen.de/software/metasim
NeSSM: http://cbb.sjtu.edu.cn/~ccwei/pub/software/NeSSM.php
NGS simulators: http://darwin.uvigo.es/ngs-simulators
PacBio reads simulator: https://code.google.com/archive/p/pbsim
pIRS: https://github.com/galaxy001/pirs
ReadSim: http://sourceforge.net/projects/readsim
simhtsd: http://sourceforge.net/projects/simhtsd
simNGS: http://www.ebi.ac.uk/goldman-srv/simNGS
SimSeq: https://github.com/jstjohn/SimSeq
SInC: http://sourceforge.net/projects/sincsimulator
Wgsim: http://github.com/lh3/wgsim
XS: http://bioinformatics.ua.pt/software/xs

ALL LINKS ARE ACTIVE IN THE ONLINE PDF