Published online 19 March 2012

Nucleic Acids Research, 2012, Vol. 40, No. 12 ***e94***
doi:10.1093/nar/gks251

Grinder: a versatile amplicon and shotgun sequence simulator

Florent E. Angly[1,*], Dana Willner[1,2], Forest Rohwer[3], Philip Hugenholtz[1,4] and Gene W. Tyson[1,5]

[1]Australian Centre for Ecogenomics, School of Chemistry and Molecular Biosciences, [2]Diamantina Institute, The University of Queensland, St Lucia, QLD 4072, Australia, [3]Biology Department, San Diego State University, San Diego, CA 92182, USA, [4]Institute for Molecular Bioscience, and [5]Advanced Water Management Centre, The University of Queensland, St Lucia, QLD 4072, Australia

Received November 23, 2011; Revised February 11, 2012; Accepted March 5, 2012

ABSTRACT

We introduce Grinder (http://sourceforge.net/projects/biogrinder/), an open-source bioinformatic tool to simulate amplicon and shotgun (genomic, metagenomic, transcriptomic and metatranscriptomic) datasets from reference sequences. This is the first tool to simulate amplicon datasets (e.g. 16S rRNA) widely used by microbial ecologists. Grinder can create sequence libraries with a specific community structure, α and β diversities and experimental biases (e.g. chimeras, gene copy number variation) for commonly used sequencing platforms. This versatility allows the creation of simple to complex read datasets necessary for hypothesis testing when developing bioinformatic software, benchmarking existing tools or designing sequence-based experiments. Grinder is particularly useful for simulating clinical or environmental microbial communities and complements the use of *in vitro* mock communities.

INTRODUCTION

The rapid development of high-throughput sequencing technologies such as 454 and Illumina has made large-scale sequencing projects both feasible and affordable. Bioinformatic tools are constantly being developed to manage and analyze data generated by these new sequencing platforms. Rigorous evaluation of the accuracy of these tools requires either the sequencing of synthetic communities of known composition created *in vitro* or the generation of simulated datasets *in silico*, which can account for both community structure and technical aspects of sequencing such as read length and errors. The construction of artificial *in vitro* communities and nucleic acid pools in the laboratory is both expensive and labor intensive, which limits the number of sequence libraries that can be produced (1–6). Mavromatis *et al*. (7) circumvented the need for *in vitro* manipulations when assembling the FAMES artificial metagenomes using DNA reads from existing single-genome shotgun sequencing projects. While both approaches produce realistic datasets, they are limited by confounding factors such as genome length bias (8,9), DNA amplification bias (10,11) and sequencing artifacts (12,13), the extent of which is generally unknown and can compromise interpretation of the sequence data. In contrast, bioinformatic tools that produce simulated reads based on reference sequences *in silico* allow users to rapidly generate large numbers of sequence libraries with controlled and predefined parameters.

Recently, characterization of microbial communities by 16S rRNA gene amplicon sequencing has experienced a renaissance, largely owing to the advent of high-throughput sequencing (14). This has spurred the development of an unprecedented number of tools and pipelines for the analysis of 16S rRNA amplicon sequences, but microbial ecologists lack a read simulator capable of generating synthetic amplicon libraries to validate existing and upcoming bioinformatic tools.

To address this limitation and also to expand upon existing shotgun sequence simulators, we present Grinder, an open-source software package that generates *in silico* simulated amplicon and shotgun (genomic, metagenomic, transcriptomic and metatranscriptomic) libraries from reference sequences. Grinder incorporates error models for a variety of sequencing platforms, can generate paired-end reads with variable insert size, and libraries with a user-specified species composition. Grinder libraries can also be designed based on α diversity metrics and model-based community structures, while sets of related libraries can be created by providing their β diversity. Unlike existing read simulators, Grinder can simulate the multiplexed PCR process to produce

*To whom correspondence should be addressed. Tel: +61 4 3365 4957; Fax: +61 7 3365 4273; Email: florent.angly@gmail.com

barcoded amplicon reads for any gene of interest, while also introducing experimental artifacts such as chimeras and biological biases due to variations in gene copy number between different species.

MATERIALS AND METHODS

Grinder implementation

Overview

Grinder is a platform-independent software package implemented in Perl and uses the Bioperl toolkit (15). Grinder is designed to run on a standard desktop computer and can be installed using a Perl module installer or a Debian package. Grinder includes a full test suite that automatically validates all components during installation. Grinder uses the Mersenne Twister algorithm (16) to generate random numbers because the default random number generation routines in many packages, such as Java, are below simulation grade (17).

The read simulation in Grinder generates amplicon (Figure 1A), or shotgun (Figure 1B) reads. While most steps in read simulation are common to shotgun and amplicon libraries, there is an additional initial step in amplicon simulation that identifies and extracts full-length amplicons in the input reference sequences based on the provided PCR primers (Figure 1, Step i). For both amplicon and shotgun read simulations, species relative abundance (which defines community structure) is calculated from rank-abundance models, α and β diversity (Figure 1, Step ii). Reads are selected from the community either from the beginning of the full-length reference amplicon (for amplicon datasets) or randomly in the reference shotgun sequences (for shotgun datasets) (Figure 1, Step iii). Finally, sequencing errors (indels, substitutions, homopolymers) are introduced in the reads in a position-specific manner (Figure 1, Step iv). An exhaustive list of options that affect these steps can be obtained at the command line using the standard help function (Grinder–help) and all specific parameters used for a particular execution of Grinder can be put in a profile file to allow the easy reuse of complex custom configurations. A subset of the available options and features are described in detail below.

Input and output sequences for simulated datasets

Publicly available FASTA-formatted databases can readily be used in Grinder. For example, the curated microbial and viral genome sequences in the NCBI RefSeq collection (18) are suitable to produce artificial genomic, metagenomic or amplicon libraries. While reads can be taken from a reference sequence and its reverse complement, for example to simulate (meta)genomic data, strand-specific datasets such as some transcriptomes (19) can be put together by taking reads from only one strand, either forward or reverse, of the reference sequences. Curated gene-specific sequence databases such as Greengenes (20), Silva (21) and PseudoMLSA (22) can also be used to simulate amplicon datasets.

Simulated read libraries are output as FASTA files with optional QUAL and FASTQ files as well as accompanying text files describing library content and community rank-abundances. Grinder offers many options to adjust the read characteristics. For example, read length can have a fixed value or follow a uniform or normal distribution and insert length for mate pairs or paired-end datasets can be specified in the same way. Detailed information for each read including its source, location on the reference sequence and introduced errors are provided in its description line, making reads entirely traceable for downstream analyses and applications (Supplementary Figure S1).

PCR simulation

A unique feature of Grinder relative to other read simulators is that a PCR simulation is performed when an amplicon read library is requested. The forward and reverse primers provided in a FASTA file by the user can contain degenerate residues following the IUPAC convention. In cases where PCR primers match different positions of a genome, several full-length amplicons will be extracted, except if these amplicons overlap, in which case only the smallest one will be extracted to mimic the PCR process (Figure 2). In subsequent Grinder steps, simulated amplicon reads are taken from the start of each full-length PCR amplicon, forward primer included.

Community structure, diversity and multiplexed identifiers

Community structure for simulated shotgun or amplicon libraries can be specified in a text file listing species and their relative abundances. Unlike most read simulators, Grinder can alternatively generate community structures based on a specified community richness (α diversity) and a deterministic rank-abundance model (uniform, linear, power law, logarithmic or exponential), with species selected randomly during library construction.

Another novel feature of Grinder is the simultaneous production of multiple read libraries (shotgun or amplicon) with related characteristics, allowing the user to vary the percentage of species shared between libraries and the percent of dominant species with different rank abundances (β diversity) (23). Multiplexed libraries consisting of individual barcoded samples pooled and sequenced on the same sequencing run can also be simulated by appending multiplexed identifiers (MIDs) given in a FASTA file to the beginning of each read. Optional MIDs are added to the reads prior to applying sequencing errors, so that MIDs may contain errors, as in real read libraries.

Simulation of biological and experimental biases

Sequencing errors such as substitutions, indels (insertions and deletions) or homopolymers can be introduced in Grinder-simulated reads by specifying position-specific models (uniform, linear or polynomial). Sanger reads can be simulated by increasing the number of substitutions and indels linearly along the reads, from 1% at its 5′ end to 2% at its 3′ end (24,25) (Supplementary Figure S2A). A fourth-degree polynomial model was implemented to reflect the accrued error rate (e) of substitutions at the 3′ end of Illumina reads (26): $e = 3.10^{-3} + 3.3.10^{-8}.i^{4}$, where i is the position from the 5′ end (in bp) (Supplementary

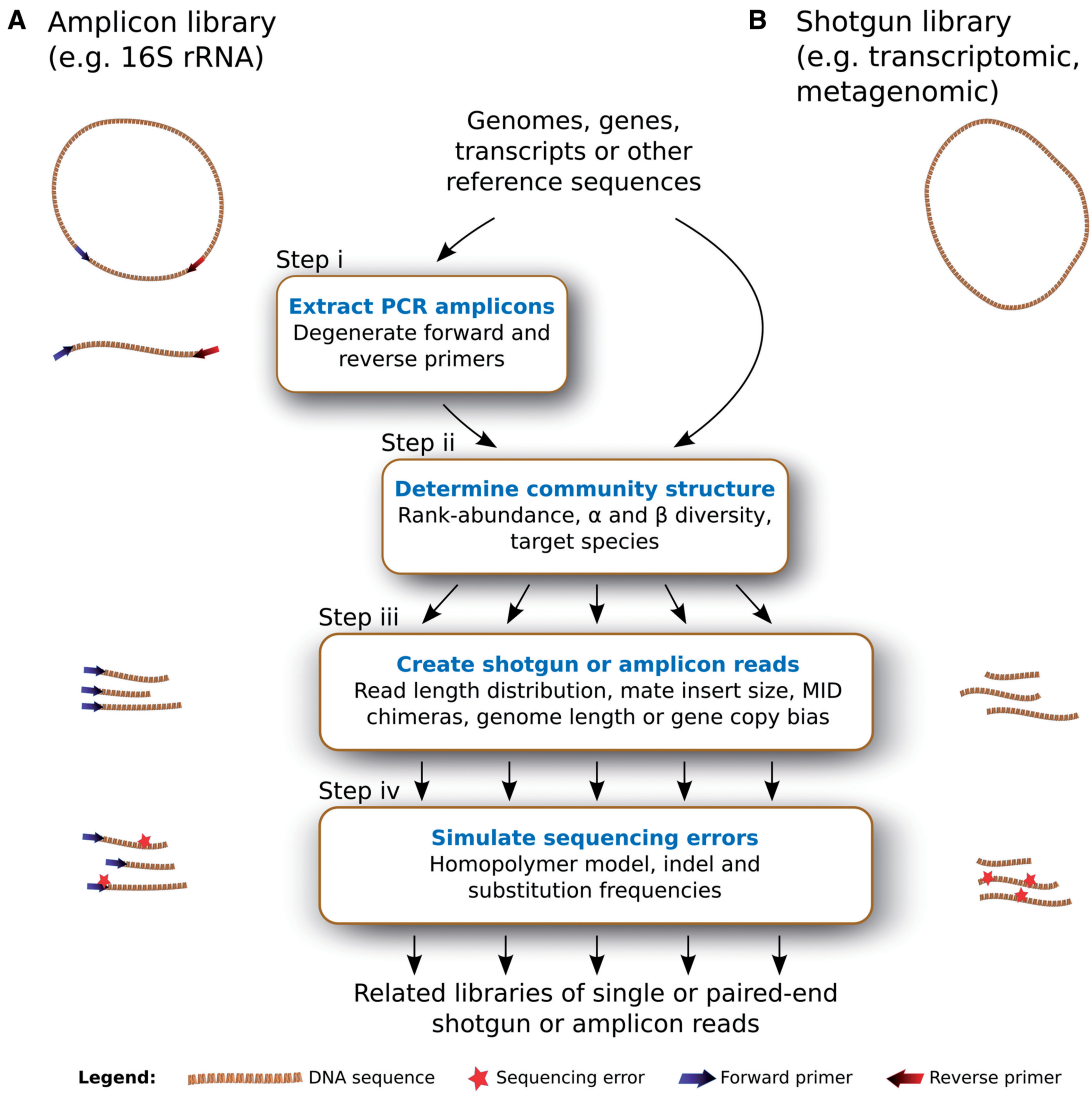

Figure 1. Flowchart of the processes and parameters used by Grinder to generate related (**A**) amplicon and (**B**) shotgun libraries.

Figure S2A). Grinder also uses several deterministic models to simulate the homopolymer errors typical of 454 pyrosequencing (25,27,28). The recent empirical homopolymer model described by Balzer *et al.* inserts more errors as the length of the homopolymeric region increases (27). This is achieved by assigning each homopolymer a new length (n') that is normally distributed around the actual length n, but with a standard deviation that increases linearly with homopolymer length: $n' \sim \mathrm{N}(n,\ 0.03494 + 0.06856n)$, for $n \geq 6$ (Supplementary Figure S2B).

Quality files (FASTQ or QUAL) can be generated based on two user-specified values, one for low (e.g. 10) and one for high (e.g. 30) quality bases. Grinder assigns the low-quality score to introduced errors and the high-quality score to all other bases. Users requiring 454 pyrosequencing libraries with more realistic quality files (in native SFF format) can run Flowsim (27) on the reads generated by Grinder.

A known issue with amplicon sequencing is the formation of chimeras, spurious sequences formed during co-amplification of homologous genes (1,29). The most common type of chimera is a bimera, which results from the fusion of two amplicon template sequences. Higher order chimeras such as trimeras and quadrameras can also occur in amplicon read datasets, albeit at lower frequencies (30). In Grinder, chimeras are generated in one of two ways. In the first method, amplicon sequences and breakpoints are randomly selected in frame. The chimeras are generated by appending consecutive amplicon segments at the breakpoint. The second method is similar to that used by CHsim (31), i.e. chimeras are produced by concatenating two or more amplicon sequences, split at particular break points. The chosen breakpoints are k-mers, or short sequence stretches of k bp, shared by two amplicons and are more likely to be chosen if the amplicons are abundant and more similar to each other.

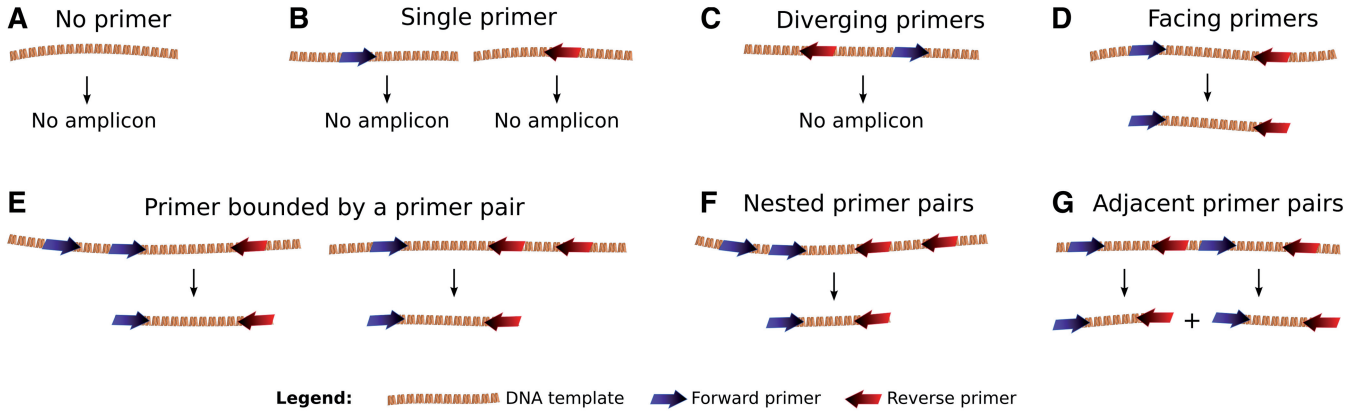

Figure 2. Grinder PCR amplicon selection process. All possible combinations of degenerate primer matches on the template DNA are considered. By default, Grinder will extract the shortest amplicon.

Finally, biological bias affects sequence libraries. Similar to the bias described in metagenomes arising from genome length differences (8), the presence of several gene copies in a genome may affect the composition of an amplicon library (32). When complete genomes are used as input, the effect of variable gene copy number in different genomes is modeled in Grinder by sampling species proportionally to their relative abundance and to the number of copies of the amplicon in its genome, instead of proportionally to their relative abundance only.

User interfaces

Grinder provides a command-line interface (CLI), graphical user interface (GUI) and application programming interface (API). The CLI can be used in a terminal and permits the automated generation of the many replicate datasets needed for statistical validation of bioinformatic tools. We have also implemented a GUI for Grinder on the Galaxy platform (Supplementary Figure S3) (33), which makes it possible to run Grinder through a web browser on any local desktop, remote server equipped with Galaxy or even on distributed computers (34). Unlike previous read simulators, Grinder also provides an object-oriented Perl API, which technical users can take advantage of when writing Perl pipelines. When using the API (Supplementary Figure S4), a Grinder factory has to be created first by using the *new()* method, which accepts the same options as the CLI. From there, the *next_lib()* method allows the user to proceed to the next sequence library and *next_read()* generates the next simulated read of that library. Each read produced is a Bio::Seq::SimulatedRead object (implemented in a Perl module written for Grinder and contributed to Bioperl) that has methods to query its nucleotide sequence, position, errors and other tracking information (Supplementary Figure S2).

I6S rRNA amplicon case study

Eight amplicon libraries, each with a unique MID sequence, were generated from the Greengenes database of named isolates (http://greengenes.lbl.gov/Download/Sequence_Data/Fasta_data_files/Isolated_named_strains_16S_aligned.fasta) using the universal primer set for 16S rRNA: 926F and 1492R (35). For each library, 454 GS-FLX Titanium pyrosequencing was simulated by requesting 5000 reads with normally distributed lengths (mean: 400 bp, standard deviation: 50 bp) and homopolymer errors (Balzer model) (27). Two additional libraries were constructed without homopolymer errors. All libraries were designed to contain 100 unique phylotypes following a power law rank-abundance curve (with parameter value of 2) and to have 80% of their phylotypes in common. The resulting Grinder files are available in Supplementary Dataset S1. FASTA and QUAL files for all libraries were concatenated prior to analysis to mimic the output of multiplexed sequencing. QIIME (36) was used to separate the libraries based on their MID, to cluster the reads at 100% and 97% identity and to assign taxonomy by comparing sequences to the Greengenes database using BLAST. A normal distribution was also fit to the empirical distribution of sequence lengths in each library by the R function fit_distr (37).

RESULTS AND DISCUSSION

Recent advances in DNA sequencing technology have allowed for the rapid generation of large sequence datasets, ushering in the age of genomics and metagenomics. Platforms and chemistries evolve quickly, engendering newer generations of sequencing that rapidly replace old methods and require the development and refinement of bioinformatic tools for analysis. Proper algorithm design and implementation requires large amounts of sequence data. However, such data may not be publicly accessible or exist in the volume necessary for rigorous testing. *In silico* simulated datasets overcome these limitations and also allow for optimization of study parameters, which may depend on sequencing depth and quality (e.g. sample size) in advance.

Table 1. Comparison of existing sequencing read simulators

Name	References	Lic.	Homepage	Lang.	Interf.	Dataset types	Paired-end	Sequencing technologies	Qual. scores	Distinguishing features
Grinder	Angly *et al.* 2012 (this article)	GPL	sf.net/projects/biogrinder	Perl	CLI, API, GUI	Amplicon, (meta)genomic, (meta) transcript-omic	Yes	Sanger, 454, Illumina	Yes	Species abundance models, α and β diversity, MIDs, FASTQ output, multimeras, genome length and gene copy number bias
GemSIM	McElroy KE (unpublished data)	GPL	sf.net/projects/gemsim	Python	CLI	(Meta)genomic	Yes	Sanger, 454, Illumina	Yes	Haplotypes, FASTQ and SAM output
Mason	Holtgrewe (44)	GPL	www.seqan.de/projects/mason.html	C++	CLI	Genomic	Yes	Sanger, 454, Illumina	Yes	Haplotypes, speed-focused
Flowsim	Balzer *et al.* (27)	GPL	biohaskell.org/Applications/FlowSim	Haskell	CLI	Genomic	No	454	Yes	Targets 454 simulation: SFF flowgram output, artificial replicates
MetaSim	Richter *et al.* (25)	Prop.	ab.inf.uni-tuebingen.de/software/metasim	Java	CLI, GUI	(Meta)genomic	Yes	Sanger, 454, Illumina	No	Genome evolution model
FASIM	Hur *et al.* (45)	Prop.	www.gem.re.kr/fasim	C	CLI	Genomic	No	Sanger	No	Biased sampling model, chimeras, chromatograms
CelSim	Myers (24)	Prop.	–	Awk, Perl	CLI	Genomic	No	Sanger	No	Repeat and variants generation
GenFrag	Engle and Burks (46,47)	Prop.	–	C	CLI	Genomic	No	Sanger	No	First read simulator

Lic, License; Prop, proprietary; Lang, Programming language; Interf, Interfaces; Sim, Simulation; Qual, Quality.

Grinder for shotgun dataset simulation

Grinder incorporates many common features of existing read simulators (Table 1) including deterministic error profiles, support for paired-end reads and the generation of sequences characteristic of particular sequencing technologies. Similar to other modern read simulators, Grinder provides sequencing errors, allowing users to flexibly specify their own error models or use preset values corresponding to known error profiles for the Sanger, 454, and Illumina platforms (Table 1). For example, Grinder was used with the Balzer error model (27) to test different short read alignment methods to improve PaPaRa (38).

Grinder also includes unique features such as the ability to specify a community structure based on a given richness (number of species) and ecologically-realistic species-abundance models (39). Multiple libraries representing communities with a specified structure and α and β diversity can be generated simultaneously. The β diversity feature in Grinder was recently used to establish empirical cutoffs for statistically significant differences between viral metagenomes (40). Grinder also provides parameters to introduce sampling biases inherent in metagenomic studies into sequence libraries. The development and benchmarking of GAAS (8) relied on the unique capability of Grinder to account for how the different length of genomes in a microbial or viral community affects the number of reads obtained from these genomes in a metagenome.

Grinder for amplicon dataset simulation

Grinder is the first read simulator to generate amplicon datasets (Table 1). Amplicon sequencing has most commonly been used for the characterization of bacterial and archaeal communities, but its applications are rapidly expanding to include characterization of fungal (41) and viral populations (42) as well as HLA class I genotyping (43). Amplicon libraries can be created in Grinder both with and without copy number bias, i.e. correction for the presence of multiple amplicons in a single reference sequence, and also with and without multiplex identifiers. Grinder uses an input set of PCR primers to find amplicons in reference sequences (Figure 2), and thus can be applied to any desired target gene or sequence.

To demonstrate the use of Grinder for amplicon reads, MID-barcoded 16S rRNA libraries with and without pyrosequencing errors were simulated. Grinder faithfully produced 5000 simulated amplicon reads with MIDs in accordance with the input specifications: normal read distribution (Figure 3A), power law rank-abundance and richness (Figure 3B), β diversity (Figure 3C). All libraries were processed with QIIME and a total of 22 411 operational taxonomic units (OTUs) at 100% identity clustering, nearly 100 times the expected number. Kunin *et al.* (48) reported similar results for 454 amplicon pyrosequencing of *Escherichia coli*, demonstrating a 40- to 150-fold increase in the expected number of 100% OTUs depending on the type of quality filtering used. An approximately 100-fold increase in 100% OTUs due to homopolymer errors was also observed by Quince *et al.*

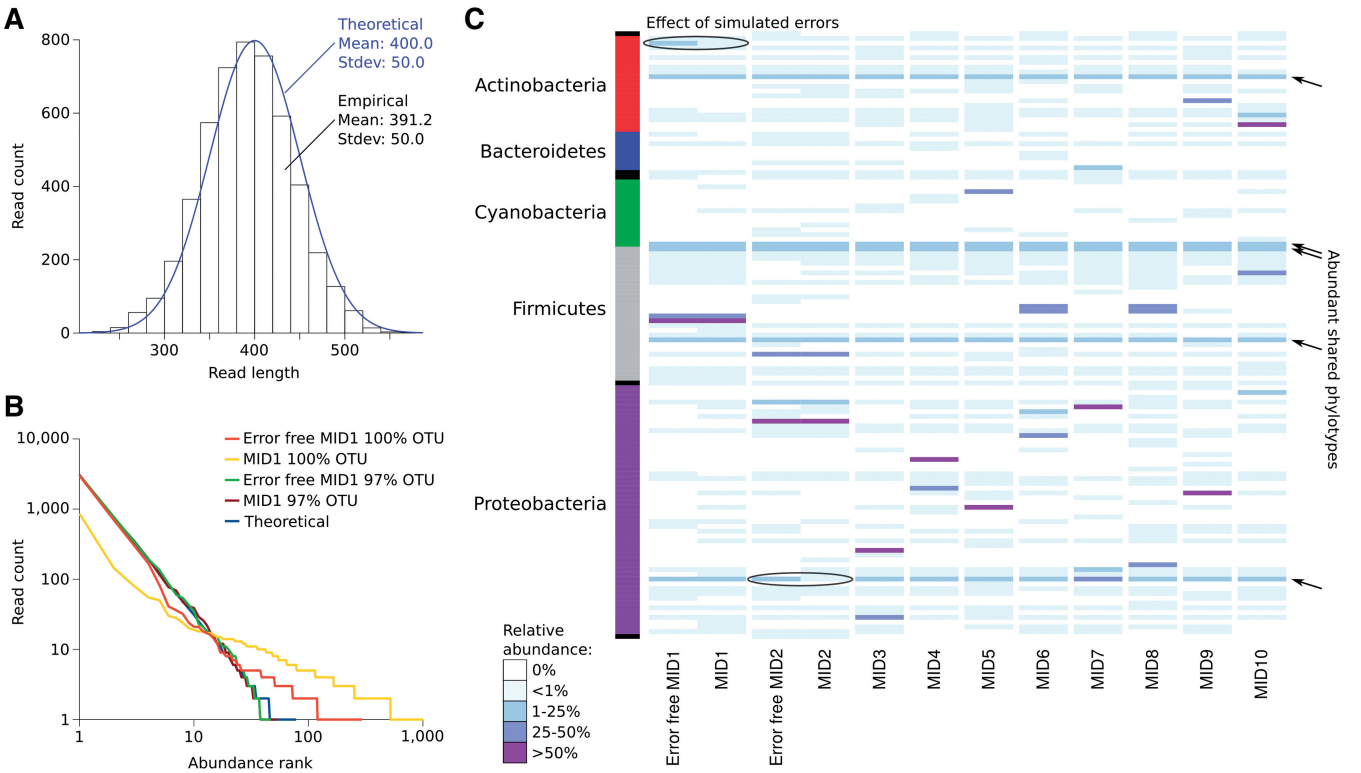

Figure 3. Analysis of 10 MID-containing 16S rRNA gene amplicon libraries generated by Grinder that share 20% of their phylotypes. (**A**) Histogram of read lengths for the libraries and curve representing their expected normal distribution. (**B**) Log-log plot of phylotype rank-abundance in the MID1 amplicon library, with and without simulated sequencing errors, using 97% and 100% identity for OTU clustering in QIIME. (**C**) Heatmap comparison of the OTU distribution in the amplicon libraries analyzed with QIIME at 97% identity OTU clustering.

(4). Consistent with the empirical observation of Kunin *et al.*, 97% identity clustering reduced the number of OTUs, resulting in a rank-abundance distribution approaching the theoretical values (Figure 3B).

Comparison of the error-free libraries with their counterparts demonstrated changes in relative abundance for some OTUs, the introduction of 21 novel OTUs and the elimination of 15 others due to homopolymer errors (Figure 3C). While most of the discrepancies occurred for OTUs at a low abundance level (<1%), as previously reported (4,48,49), the decrease of two OTUs from a medium abundance level (1–25%) to a low abundance level (<1%) shows that care should be taken when analyzing amplicon data that contain sequencing errors. The simulated errors mostly affected low-abundance OTUs, artificially inflating the size of the rare biosphere (4,48,50). Overall, this example illustrates that Grinder is capable of creating realistic amplicon libraries and modeling the effects of 454 homopolymer errors on microbial community profiling using the 16S rRNA gene.

CONCLUSION

Grinder is a read simulator that generates shotgun and amplicon libraries for software benchmarking, algorithm development, statistical testing and educational purposes. Grinder has been used in this capacity to simulate large volumes of environmental and clinical sequence data (8,38,40,51). Grinder libraries can be given a variety of community structures by specifying an ecological species-abundance distribution and α diversity or β diversity and MIDs when multiple libraries are created simultaneously. As demonstrated here Grinder has the unique ability to generate realistic 16S rRNA amplicon reads *in silico* with 454 homopolymer errors. The errors of current sequencing technologies can be flexibly specified in Grinder by combining several deterministic models. Sequencing technologies evolve rapidly, but the open-source nature of Grinder will facilitate the addition of new technologies such as IonTorrent (52) as their error profiles become available. By helping test hypotheses, create better bioinformatic tools and enhance data interpretation, the more systematic use of read simulators has the potential to accelerate the rate of biological discoveries. In this context, we believe that Grinder will be a valuable tool for bioinformaticians and biologists alike.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online: Supplementary Figures 1–4 and Supplementary Dataset 1.

ACKNOWLEDGEMENTS

We thank people who have tested Grinder and offered comments, suggestions and support, specifically Barry

Cayford, Paul Dennis, Mike Imelfort, Steve Rayhawk, Robert Schmieder, Ramzi Temanni and Albert Villela.

FUNDING

QEII Fellowship from the Australian Research Council, [DP1093175 (to G.W.T.)]; University of Queensland strategic funding of the Australian Centre for Ecogenomics. Funding for open access charge: F.E.A's Discovery Early Career Research Award.

REFERENCES

1. Haas,B.J., Gevers,D., Earl,A.M., Feldgarden,M., Ward,D.V., Giannoukos,G., Ciulla,D., Tabbaa,D., Highlander,S.K., Sodergren,E. *et al.* (2011) Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.*, **21**, 494–504.
2. Turnbaugh,P.J., Quince,C., Faith,J.J., McHardy,A.C., Yatsunenko,T., Niazi,F., Affourtit,J., Egholm,M., Henrissat,B., Knight,R. *et al.* (2010) Organismal, genetic, and transcriptional variation in the deeply sequenced gut microbiomes of identical twins. *Proc. Natl Acad. Sci. USA*, **107**, 7503–7508.
3. Sun,Y., Cai,Y., Liu,L., Yu,F., Farrell,M.L., McKendree,W. and Farmerie,W. (2009) ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. *Nucleic Acids Res.*, **37**, e76.
4. Quince,C., Lanzén,A., Curtis,T.P., Davenport,R.J., Hall,N., Head,I.M., Read,L.F. and Sloan,W.T. (2009) Accurate determination of microbial diversity from 454 pyrosequencing data. *Nat. Methods.*, **6**, 639–641.
5. Henn,M.R., Sullivan,M.B., Stange-Thomann,N., Osburne,M.S., Berlin,A.M., Kelly,L., Yandava,C., Kodira,C., Zeng,Q., Weiand,M. *et al.* (2010) Analysis of high-throughput sequencing and annotation strategies for phage genomes. *PLoS One*, **5**, e9083.
6. Kan,J., Hanson,T.E., Ginter,J.M., Wang,K. and Chen,F. (2005) Metaproteomic analysis of Chesapeake Bay microbial communities. *Saline Syst.*, **1**, 7.
7. Mavromatis,K., Ivanova,N., Barry,K., Shapiro,H., Goltsman,E., McHardy,A.C., Rigoutsos,I., Salamov,A., Korzeniewski,F., Land,M. *et al.* (2007) Use of simulated data sets to evaluate the fidelity of metagenomic processing methods. *Nat. Methods.*, **4**, 495–500.
8. Angly,F.E., Willner,D., Prieto-Davó,A., Edwards,R.A., Schmieder,R., Vega-Thurber,R., Antonopoulos,D.A., Barott,K., Cottrell,M.T., Desnues,C. *et al.* (2009) The GAAS metagenomic tool and its estimations of viral and microbial average genome size in four major biomes. *PLoS Comput. Biol.*, **5**, e1000593.
9. Beszteri,B., Temperton,B., Frickenhaus,S. and Giovannoni,S.J. (2010) Average genome size: a potential source of bias in comparative metagenomics. *ISME J.*, **4**, 1075–1077.
10. Yilmaz,S., Allgaier,M. and Hugenholtz,P. (2010) Multiple displacement amplification compromises quantitative analysis of metagenomes. *Nat. Methods.*, **7**, 943–944.
11. Pinard,R., de Winter,A., Sarkis,G., Gerstein,M., Tartaro,K., Plant,R., Egholm,M., Rothberg,J. and Leamon,J. (2006) Assessment of whole genome amplification-induced bias through high-throughput, massively parallel whole genome sequencing. *BMC Genomics*, **7**, 216.
12. Gomez-Alvarez,V., Teal,T.K. and Schmidt,T.M. (2009) Systematic artifacts in metagenomes from complex microbial communities. *ISME J.*, **3**, 1314–1317.
13. Huse,S.M., Huber,J.A., Morrison,H.G., Sogin,M.L. and Welch,D. (2007) Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol.*, **8**, R143.
14. Tringe,S.G. and Hugenholtz,P. (2008) A renaissance for the pioneering 16S rRNA gene. *Curr. Opin. Microbiol.*, **11**, 442–446.
15. Stajich,J.E., Block,D., Boulez,K., Brenner,S.E., Chervitz,S.A., Dagdigian,C., Fuellen,G., Gilbert,J.G., Korf,I., Lapp,H. *et al.* (2002) The Bioperl toolkit: Perl modules for the life sciences. *Genome Res.*, **12**, 1611–1618.
16. Matsumoto,M. and Nishimura,T. (1998) Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model Comput. Simul.*, **8**, 3–30.
17. L'Ecuyer,P. and Simard,R. (2007) TestU01: A C library for empirical testing of random number generators. *ACM Trans. Math. Softw.*, **33**, Article 22.
18. Pruitt,K.D., Tatusova,T. and Maglott,D.R. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res.*, **35**, D61–D65.
19. Vivancos,A.P., Güell,M., Dohm,J.C., Serrano,L. and Himmelbauer,H. (2010) Strand-specific deep sequencing of the transcriptome. *Genome Res.*, **20**, 989–999.
20. DeSantis,T.Z., Hugenholtz,P., Larsen,N., Rojas,M., Brodie,E.L., Keller,K., Huber,T., Dalevi,D., Hu,P. and Andersen,G.L. (2006) Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol.*, **72**, 5069–5072.
21. Pruesse,E., Quast,C., Knittel,K., Fuchs,B.M., Ludwig,W., Peplies,J. and Glöckner,F.O. (2007) SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Res.*, **35**, 7188–7196.
22. Bennasar,A., Mulet,M., Lalucat,J. and García-Valdés,E. (2010) PseudoMLSA: a database for multigenic sequence analysis of Pseudomonas species. *BMC Microbiol.*, **10**, 118.
23. Angly,F.E., Felts,B., Breitbart,M., Salamon,P., Edwards,R.A., Carlson,C., Chan,A.M., Haynes,M., Kelley,S., Liu,H. *et al.* (2006) The marine viromes of four oceanic regions. *PLoS Biol.*, **4**, e368.
24. Myers,G. (1999) A dataset generator for whole genome shotgun sequencing. *Proc. Int. Conf. Intell. Syst. Mol. Biol*, 202–10.
25. Richter,D.C., Ott,F., Auch,A.F., Schmid,R. and Huson,D.H. (2008) MetaSim: a sequencing simulator for genomics and metagenomics. *PLoS One*, **3**, e3373.
26. Korbel,J.O., Abyzov,A., Mu,X.J., Carriero,N., Cayting,P., Zhang,Z., Snyder,M. and Gerstein,M.B. (2009) PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.*, **10**, R23.
27. Balzer,S., Malde,K., Lanzén,A., Sharma,A. and Jonassen,I. (2010) Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim. *Bioinformatics*, **26**, i420–i425.
28. Margulies,M., Egholm,M., Altman,W.E., Attiya,S., Bader,J.S., Bemben,L.A., Berka,J., Braverman,M.S., Chen,Y.J., Chen,Z. *et al.* (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.
29. Wang,G.C. and Wang,Y. (1996) The frequency of chimeric molecules as a consequence of PCR co-amplification of 16S rRNA genes from different bacterial species. *Microbiology*, **142**, 1107–1114.
30. Quince,C., Lanzen,A., Davenport,R. and Turnbaugh,P. (2011) Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, **12**, 38.
31. Edgar,R.C., Haas,B.J., Clemente,J.C., Quince,C. and Knight,R. (2011) UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, **27**, 2194–2200.
32. Crosby,L.D. and Criddle,C.S. (2003) Understanding bias in microbial community analysis techniques due to rrn operon copy number heterogeneity. *BioTechniques*, **34**, 790–794, 796, 798 passim.
33. Giardine,B., Riemer,C., Hardison,R.C., Burhans,R., Elnitski,L., Shah,P., Zhang,Y., Blankenberg,D., Albert,I., Taylor,J. *et al.* (2005) Galaxy: A platform for interactive large-scale genome analysis. *Genome Res.*, **15**, 1451–1455.
34. Afgan,E., Baker,D., Coraor,N., Chapman,B., Nekrutenko,A. and Taylor,J. (2010) Galaxy CloudMan: delivering cloud compute clusters. *BMC Bioinformatics*, **11**, S4.
35. Ochman,H., Worobey,M., Kuo,C.H., Ndjango,J.B., Peeters,M., Hahn,B.H. and Hugenholtz,P. (2010) Evolutionary relationships of wild hominids recapitulated by gut microbial communities. *PLoS Biol.*, **8**, e1000546.

36. Caporaso,J.G., Kuczynski,J., Stombaugh,J., Bittinger,K., Bushman,F.D., Costello,E.K., Fierer,N., Peña,A.G., Goodrich,J.K., Gordon,J.I. *et al.* (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods.*, **7**, 335–336.
37. R development core team. R. *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
38. Berger,S.A. and Stamatakis,A. (2011) Aligning short reads to reference alignments and trees. *Bioinformatics*, **27**, 2068–2075.
39. Ulrich,W. (2001) Models of relative abundance distributions I: model fitting by stochastic models. *Pol. J. Ecol.*, **49**, 145–157.
40. Willner,D., Haynes,M.R., Furlan,M., Hanson,N., Kirby,B., Lim,Y.W., Rainey,P.B., Schmieder,R., Youle,M., Conrad,D. *et al.* (2011) Case studies of the spatial heterogeneity of DNA viruses in the cystic fibrosis lung. *Am. J. Respir. Cell Mol. Biol.*, 10.1165/rcmb.2011-0253OC.
41. Ghannoum,M.A., Jurevic,R.J., Mukherjee,P.K., Cui,F., Sikaroodi,M., Naqvi,A. and Gillevet,P.M. (2010) Characterization of the oral fungal microbiome (mycobiome) in healthy individuals. *PLoS Pathog.*, **6**, e1000713.
42. Simons,J., Egholm,M., Lanza,J., Turenchalk,G., Desany,B., Ronan,M., Knight,J., Du,L., Leamon,J., Rothberg,J. *et al.* (2005) Ultra-deep sequencing of HIV from drug-resistant patients. *Antivir. Ther.*, **10**, S157.
43. Lank,S.M., Wiseman,R.W., Dudley,D.M. and O'Connor,D.H. (2010) A novel single cDNA amplicon pyrosequencing method for high-throughput, cost-effective sequence-based HLA class I genotyping. *Hum. Immunol.*, **71**, 1011–1017.
44. Holtgrewe,M. (2010) *Mason - A Read Simulator for Second Generation Sequencing Data*, Institut für Mathematik und Informatik, Freie Universität Berlin.
45. Hur,C.-G., Kim,S., Kim,C.-H., Yoon,S.-H., In,Y.-H., Kim,C.-M. and Cho,H.-G. (2006) FASIM: Fragments assembly simulation using biased-sampling model and assembly simulation for microbial genome shotgun sequencing. *J. Microbiol. Biotechn.*, **16**, 683–688.
46. Engle,M.L. and Burks,C. (1993) Artificially generated data sets for testing DNA sequence assembly algorithms. *Genomics*, **16**, 286–288.
47. Engle,M.L. and Burks,C. (1994) GenFrag 2.1: new features for more robust fragment assembly benchmarks. *Comput. Appl. Biosci. (CABIOS)*, **10**, 567–568.
48. Kunin,V., Engelbrektson,A., Ochman,H. and Hugenholtz,P. (2010) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environ. Microbiol.*, **12**, 118–123.
49. Balzer,S., Malde,K. and Jonassen,I. (2011) Systematic exploration of error sources in pyrosequencing flowgram data. *Bioinformatics*, **27**, i304–i309.
50. Sogin,M.L., Morrison,H.G., Huber,J.A., Welch,D., Huse,S.M., Neal,P.R., Arrieta,J.M. and Herndl,G.J. (2006) Microbial diversity in the deep sea and the underexplored "rare biosphere.". *Proc. Natl Acad. Sci. USA*, **103**, 12115–12120.
51. Treangen,T.J., Sommer,D.D., Angly,F.E., Koren,S. and Pop,M. (2011) Next generation sequence assembly with AMOS. *Curr. Protoc. Bioinformatics*, **33**, 11.8.1–11.8.18.
52. Rothberg,J.M., Hinz,W., Rearick,T.M., Schultz,J., Mileski,W., Davey,M., Leamon,J.H., Johnson,K., Milgrew,M.J., Edwards,M. *et al.* (2011) An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, **475**, 348–352.