A better sequence-read generator program for metagenomics

Stephen Eric Johnson, Brett Trost, Jeffrey R Long, Anthony Kusalik✉
University of Saskatchewan, Saskatoon, Canada

Motivation and Objectives

There are many programs available for generating simulated metagenomic sequence reads. The data generated by these programs follow rigid models, which limits the use of a given program to the author's original intentions. For example, many popular simulator programs only generate reads that follow uniform or normal distributions. To our knowledge, there are no programs that allow a user to generate simulated data following non-parametric read-length distributions and quality profiles based on empirical next-generation sequencing (NGS) data.

We present BEAR (Better Emulation for Artificial Reads), a program that uses a machine learning approach to generate reads with lengths and quality values mimicking empirically derived distributions. BEAR is able to emulate reads from various NGS platforms, including Illumina, 454 and Ion Torrent. BEAR requires minimal user input, as it automatically determines appropriate internal parameter settings.

Methods

Multiple popular sequence simulator programs were tested to gauge their ability to emulate real data available to our lab. The tested programs were SimSeq (Earl *et al.*, 2011), MetaSim (Richter *et al.,* 2008), Grinder (Angly *et al.,* 2012), and 454sim (Lysholm *et al.*, 2011). Shortcomings were identified and used to guide the development of our improved sequence simulator.

BEAR was written using a combination of Perl and Python scripts. An advantage of BEAR is that it requires only three files as input:

1. the training set, a multi-FASTA file that exhibits the desired read-length and quality distributions;
2. organism database, a multi-FASTA file containing the genomes from which reads will be generated. Each genome is a single sequence;
3. a tab-delimited file containing a sequence identifier for each sequence in the database and the desired relative abundance of that sequence.

BEAR uses a two-step process: in the first step, the organism database and abundance file are used to generate a simulated metagenomic dataset containing reads of uniform quality and length. In the second step, a model of the distribution of read lengths is generated from the training data, and a Markov chain is created based on the quality scores in the training data. The reads from step 1 are then trimmed using a Monte Carlo process based on the read length distribution and have quality scores generated using the Markov chain. The quality score of the current position and the average quality of the five previous positions determine the quality score of the next position.

To evaluate BEAR, actual read data from various sequencer technologies were obtained. Each of the aforementioned programs was used to generate simulated data for these technologies, using appropriate parameter settings. In the case of BEAR, the real data was used as input. (Not all programs could generate all types of simulated reads.) Since metagenomic data was the goal, a list of species and abundances as outlined by Pignatelli and Moya (2011) was used. The characteristics of simulated reads from each program were then determined and compared to the characteristics from the real data.

BEAR is available from the authors upon request. It requires that the user have BioPerl and BioPython installed.

Results and Discussion

As shown in Figure 1a, modern sequencing simulators are limited in their ability to model actual read lengths, being restricted to uniform or normal distributions. In addition, Figure 1b demonstrates the inflexibility of quality score generation within these programs. In contrast, BEAR is better able to mimic the read length and quality distribution characteristics of reads from next-generation sequencing technologies. For example, Figure 1a shows that BEAR more accurately emulates the read distribution of the Ion Torrent data when compared to Grinder, the program that was second closest to matching the read-length

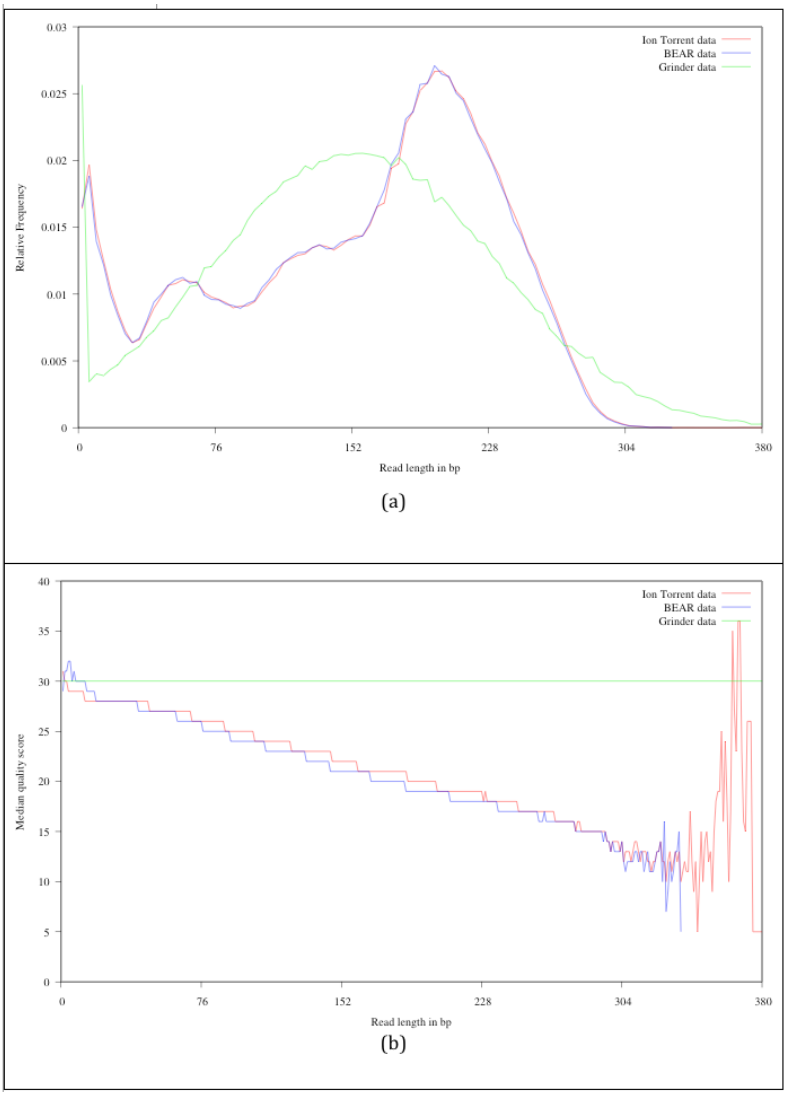

Figure 1. Top: The read length distribution of real Ion Torrent data (red) is compared to the trained data from BEAR (blue) and the second-most accurate program Grinder (green). Bottom: The median quality scores for real Ion Torrent data (red) are compared to the trained data from BEAR (blue) and Grinder (green). As evident in the figure, the longest read generated by BEAR is 338 bp, while the longest read in the Ion Torrent data is 380bp. This is due to the fact that reads longer than 338bp comprise less than 0.0005% of all Ion Torrent reads in our training data.

distribution of the Ion Torrent data. A plot of the median quality scores for the Ion Torrent, BEAR, and Grinder data (Figure 1b) suggests that BEAR generates reads that better emulate the quality profile of real data. Similar results were observed for 454 and Illumina data, suggesting that BEAR is a versatile tool for emulating various NGS platforms.

Acknowledgements

This work was supported by MAVEN, a project funded by Western Economic Diversitification Canada and Enterprise Saskatchewan. Oversight of the MAVEN project is provided by Genome Prairie and Dr. Reno Pontarollo.

References

Angly FE, Willner D, Rohwer F, Hugenholtz P, Tyson, GW. (2012) Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research* **40**(12), e94-e94. doi:10.1093/nar/gks251.

Earl D, Bradnam K, St. John J, Darling A, Lin D, et al. (2011) Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res* **21**(12), 2224-41. doi:10.1101/gr.126599.111.

Lysholm F, Anderson B, Persson B. (2011) An efficient simulator of 454 data using configurable statistical models. *BMC Research Notes* **4**, 449. doi:10.1186/1765-0500-4-449.

Pignatelli M, Moya A. (2011) Evaluating the fidelity of de novo short read metagenomic assembly using simulated data. *PLoS One* **6**(5), e19984. doi:10.1371/journal.pone.0019984.

Richter DC, Ott F, Auch AF, Schmid R, Huson DH. (2008) MetaSim--A Sequencing Simulator for Genomics and Metagenomics. *PLoS ONE* **3**(10), e3373.