Simulome: a genome sequence and variant simulator

Adam Price and Cynthia Gibas*

Department of Bioinformatics and Genomics, University of North Carolina at Charlotte, Charlotte, NC 28223-0001, USA

*To whom correspondence should be addressed.

Associate Editor: Bonnie Berger

Received on November 14, 2016; revised on January 24, 2016; editorial decision on February 6, 2017; accepted on February 8, 2017

Abstract

Summary: Simulome provides a powerful and easy to use tool for creating prokaryotic pseudo-genomes. It provides options that can be used in combination to create mutated variants of the simulated genome, which allows for controlled testing of specific genomic conditions. Simulome can be used in combination with real reads generated from next-generation sequencing platforms, or with simulated reads.

Availability and Implementation: Simulome is available for download at https://github.com/price0416/Simulome and is free for use under the MIT license.

Contact: cgibas@uncc.edu or price0416@gmail.com

1 Introduction

As new data types and methodologies for analysis of biological data are developed, simulation tools are becoming increasingly necessary to the development, testing and benchmarking of bioinformatics research. Simulations provide a valuable control case in many contexts, such as the identification of read mapping bias (Degner et al., 2009), correction of read bias in RNA-seq mapping (Satya et al., 2012) and analysis of the accuracy of gene expression profiling (Hirsch et al., 2015). It is therefore necessary to develop simulation software that is flexible, accessible and able to model a wide range of different genomic conditions.

Currently, several tools exist for simulating read data, such as ART (Huang et al., 2012) and Mason (http://seqan.de/projects/mason.html). Simulation tools such as these approach the problem of simulation from the perspective of reads, and are capable of producing impressive read sets that model variation and sequence errors for a variety of sequencing platforms. Simulome, however, approaches the problem of simulation from the perspective of the reference genome, which expands on the set of potential problems that can be addressed using genomic simulation. When Simulome is used in combination with read simulation tools like ART and Mason, entirely simulated experimental scenarios become possible.

2 Features and methods

Simulome is a python-based tool that incorporates biopython (Cock et al., 2009) to generate synthetic prokaryotic reference genomes by sampling and restructuring existing genomic sequence data. Simulome provides the ability to create pseudo-reference genomes with a specified gene set and controlled intergenic regions, as well as versions of the simulated genome that contain user controlled mutation events. This functionality makes it possible to analyze the effect of specific mutation types on a large scale, providing researchers with the ability to investigate the efficacy of analysis methodologies on a large number of genes that contain similar mutation events, while providing a control genome to which comparisons can be made. Simulome’s variant simulations can also be applied to whole prokaryotic genomes, allowing researchers to create variants of existing genomes with mutations introduced according to user specifications.

Simulome features four different run modes for simulating mutation events: SNP mode, synonymous/nonsynonymous mutation mode, insertion/deletion mode and duplication mode. These run modes can be combined to produce variant genomes containing any combination of mutation events. Further customization of both the control and variant genome is possible through additional optional arguments.
2.1 Reference genome simulation

Creation of a simulated reference genome requires a nucleotide FASTA file containing the sequence of an existing organism, and an associated annotation file in GFF/GTF3 format. Basing the simulation on the properties of a real genome ensures that simulated genomes accurately model natural sequence variation. and allows genomes with different properties to serve as the base for simulation. For example, a researcher studying the effects of GC bias in RNA-seq data may be interested in simulating genomes with low or high GC content, and could do so by providing a low or high GC content genome to Simulome as input. Users can either use the provided genomic data to create a pseudo-genome of arbitrary size and its mutated variant, or simply create a mutated variant of the entire existing genome. If a pseudo-genome is to be created, gene data is extracted from the base genome and a user-specified number of genes are randomly selected to be used in the simulation. These genes are sampled so as to follow a normal distribution of gene length. The mean length and the standard deviation of all genes in the base genome are calculated, and genes are then sampled such that the mean and standard deviation of the simulated reference genome approximates that of the base genome. Randomness is determined by a seed value, and in the event that a user wishes to repeat a simulation, an exact replication can be produced by reusing the same random seed. Once target genes are selected, the simulated genome is created by interspersing intergenic regions and coding regions. Intergenic regions can be simulated in a variety of ways. Users can either specify the use of randomly generated intergenic regions, or can use real intergenic sequence data from the base genome. Random sequences are generated such that each base has a 25% chance of being selected for any given position. When real intergenic regions are used, all intergenic regions for the base genome are extracted and segments are randomly selected when needed. In both cases, users can specify intergenic length or allow randomly sized intergenic regions to be selected. BLAST searches can optionally be performed on each simulated intergenic region to ensure that there are no unintended duplicate regions in the simulation (Altschul et al., 1990). The simulated genome will be output in FASTA and GFF/GTF3 format. Additional properties, such as operon inclusion and frequency, can be simulated by specifying optional arguments, which are covered in more depth in the Simulome manual.

2.2 Variant genome simulation

Four run modes are available for simulating SNPs, synonymous/ nonsynonymous mutations, indels and/or duplication events. These can be used in any combination and are applied to each gene in the simulated genome. For example, it is possible to simulate a reference genome and/or a variant genome in which each gene contains a specific number of SNPs and deletions. Each run mode can be configured to introduce either an exact number of mutations in each gene, or otherwise to simulate variants in a range based on a Gaussian distribution with user-defined means and standard deviations. Once mutation events are simulated, sequence and annotation files representing the variant simulated genome are written, as well as an additional file containing meta-data for the introduced variants.

2.2.1 SNP run mode

In SNP run mode, a specified number of SNP events are introduced at random locations for each gene in the variant genome. By default, polymorphisms can occur at any position in a gene and no base will mutated more than once. However, additional control is provided for SNP mode to allow for the control of SNP density. Users can specify a window size in which SNP events can occur. To simulate the effect of SNPs on read alignment, a user might create simulations with regions of increasingly dense SNPs. By specifying the same number of SNPs in a decreasingly small window size, it would be possible to quantify the effect of locally clustered SNPs on read alignment on a large number of samples.

2.2.2 Synonymous/nonsynonymous run mode

In Synonymous/Nonsynonymous run mode, mutations are performed such that a specified percentage of mutations will be synonymous mutations. This run mode assumes that the first position of each gene begins the open reading frame and requires that the user provide distribution parameters to determine the number of mutations that will occur in each gene.

2.2.3 Indel run mode

Insertion and deletion run mode allows users to simulate indel events for each gene in the simulated reference genome. Users can choose to include insertion events, deletion events, or both, and can specify the number and length of each event. When both insertions and deletion events are specified, deletion events are performed first in order to preserve mutation integrity across all genes.

2.2.4 Duplication run mode

Duplication run mode allows the user to simulate scenarios where multiple possible mapping locations for a read are present in a genome. In duplication mode, a duplication percentage is specified by the user. A duplicate region of the specified size (e.g. 1000 nt if 10% of a base genome of 10000 nt is specified) is simulated and added to the base genome. Genes are randomly selected for duplication, along with appropriate intergenic regions, until the desired duplication level is reached.

2.3 Performance

Simulations were performed for reference genomes and mutated variants using *Escherichia coli* strain K12. These tests were performed on the Red Hat Enterprise Linux 7.2 operating system, with an Intel Xeon 2.53 GHz processor. Results of performance testing can be seen in Table 1.

| Genes | Simulation modes | Time |
|-------|-----------------|------|
| 1000  | Reference only  | 3 m 5 s |
| 1000  | Reference + SNP variant | 3 m 12 s |
| 1000  | Reference + Indel variant | 3 m 18 s |
| 1000  | Reference + Synonymous/Indel variant | 3 m 24 s |
| 1000  | Reference + SNP/Indel/duplication variant | 3 m 41 s |
| 2000  | Reference + SNP/Indel/duplication variant | 10 m 5 s |
| 3000  | Reference + SNP/Indel/duplication variant | 16 m 40 s |
| E.coli (Full) | Reference + SNP/Indel/duplication variant | 0 m 48 s |
| E.coli (Full) | Reference + Synonymous/Indel variant | 0 m 40 s |
| E.coli (Full) | Reference + SNP/Indel variant | 0 m 41 s |

2.4 Acknowledgements

The authors of this paper would like to thank Dr. Jennifer Weller and Dr. Ehsan Tabari for their valuable discussions that have contributed to the development of this package.
Funding
This work was supported by the Charlotte Research Institute (to CJG) and the Lucille P. and Edward C. Giles Dissertation-Year Fellowship (to AP).
Conflict of Interest: none declared.

References
Altschul, S.F. et al. (1990) Basic local alignment search tool. J. Mol. Biol., 215, 403–410.
Cock, P.J. et al. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics, Bioinformatics, 25, 1422–1423.
Cock, P.J. et al. (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucl. Acids Res., 38, 1767–1771.
Degner, J.F. et al. (2009) Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. Bioinformatics, 25, 3207–3212.
Hirsch, C.D. et al. (2015) Genomic limitations to RNA sequencing expression profiling. Plant J., 84, 491–503.
Huang, W. et al. (2012) ART: A next generation sequencing read simulator. Bioinformatics, 28, 593–594.
Satya, R.V. et al. (2012) A new strategy to reduce allelic bias in RNA-Seq read-mapping. Nucl. Acids Res., 40.