


Sequence analysis

SCSsim: an integrated tool for simulating single-cell genome sequencing data

Zhenhua Yu ^{1,*}, Fang Du¹, Xuehong Sun¹ and Ao Li^{2,*}¹Department of Software Engineering, Ningxia University, Yinchuan 750021, China and ²Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027, China

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on June 18, 2019; revised on August 20, 2019; editorial decision on September 12, 2019; accepted on September 15, 2019

Abstract

Motivation: Allele dropout (ADO) and unbalanced amplification of alleles are main technical issues of single-cell sequencing (SCS), and effectively emulating these issues is necessary for reliably benchmarking SCS-based bioinformatics tools. Unfortunately, currently available sequencing simulators are free of whole-genome amplification involved in SCS technique and therefore not suited for generating SCS datasets. We develop a new software package (SCSsim) that can efficiently simulate SCS datasets in a parallel fashion with minimal user intervention. SCSsim first constructs the genome sequence of single cell by mimicking a complement of genomic variations under user-controlled manner, and then amplifies the genome according to MALBAC technique and finally yields sequencing reads from the amplified products based on inferred sequencing profiles. Comprehensive evaluation in simulating different ADO rates, variation detection efficiency and genome coverage demonstrates that SCSsim is a very useful tool in mimicking single-cell sequencing data with high efficiency.

Availability and implementation: SCSsim is freely available at <https://github.com/qasimyu/scssim>.

Contact: zhyu@nxu.edu.cn or aoli@ustc.edu.cn

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

With the development of single cell isolation (Brasko *et al.*, 2018) and whole-genome amplification (WGA) (Zong *et al.*, 2012) techniques, single cell sequencing (SCS) has become a powerful means to identify cellular diversity at single cell resolution. Compared to bulk-sequencing, there are main technical issues for SCS data such as allele dropout (ADO) and unbalanced amplification of alleles. Although novel computational methods are continuously developed to deliver reliable profiles from SCS data (Ross and Markowitz, 2016; Singer *et al.*, 2018; Zafar *et al.*, 2016, 2017), the inference performance of these tools may be underestimated or overestimated due to limited evaluation on small datasets (Miura *et al.*, 2018). On the other hand, currently available SCS datasets are far from sufficient for comprehensive benchmarking analysis, giving rise to the necessity of constructing plentiful baseline SCS datasets. Computer simulation is an efficient way to yield as much as desired SCS datasets under a controlled manner, enabling reliable and effective evaluation of competitive methods (Escalona *et al.*, 2016; Yuan *et al.*, 2017). Despite the fact that an arsenal of simulators is available for bulk-sequencing (Escalona *et al.*, 2016), they are free of the single cell genome pre-amplification involved in SCS technique and cannot effectively mimic the specific issues introduced in WGA procedure. Moreover, to the best of our knowledge there are no currently available tools that integrate WGA and read simulation

functionality into a single framework for facilitating end-to-end simulation of SCS data.

Here we present a new software called SCSsim to enable efficient simulation of SCS datasets. First, single cell genome is constructed by inserting a complement of genomic variations into a given reference sequence under user-controlled scenarios. Second, the WGA procedure is implemented as dividing the single cell genome into variable-size fragments and amplifying the fragments by emulating MALBAC (Zong *et al.*, 2012) technique. Third, the amplified products from WGA are used as templates to yield reads based on the sequencing profiles that are inferred from real sequencing data. By comprehensively evaluating the abilities of the proposed method in simulation of different ADO rates, variation detection efficiency and genome coverage, we demonstrate SCSsim is an effective SCS simulator.

2 Materials and methods

We develop three functional modules named as ‘learnProfile’, ‘simuVars’ and ‘genReads’ in the SCSsim framework, each of which is responsible for a specific aspect of SCS technique (Supplementary Fig. S1). The pipeline of SCSsim includes: (i) if no sequencing profiles are available for a given sequencing platform, the profiles are learned from real sequencing data; (ii) single cell genome is

constructed by tuning input reference sequence; (iii) genome is amplified to produce full amplicons from which reads are generated.

The ‘learnProfile’ component takes three inputs: a FASTA file representing a reference sequence, a non-tumor BAM file containing real sequencing data, and a VCF file generated from the BAM file to define germline heterozygous SNPs. From the BAM file, each exactly mapped read with >20 mapping quality is extracted to construct a triad (S, B, Q) , where S denotes the underlying source sequence from which the read sequence B and Phred quality sequence Q are derived. Based on all the triples (S, B, Q) , three profiles including indel error rates, base substitution probabilities, Phred quality score distributions are inferred by estimating histograms as described in [Supplementary Methods](#). In addition, GC-content bias is estimated by fitting the relationship between read counts and GC-content with a locally weighted linear regression.

The ‘simuVars’ module is used to generate mutated genome by inserting various types of variations into the input reference sequence. Due to distinct complement of genomic variations exist among single cells, the utility is designed to emulate cell-specific variations. Such functionality is essential for SCS based bioinformatics analysis such as cellular lineage inference. The types and locations of all variations are deterministic and defined by users in a file following a specific format (more details are provided in the [Supplementary Methods](#)), which enables the single cell sequencing data to be simulated under controlled scenarios. The produced sequence data is written into a file in FASTA format and used as template for MALBAC amplification.

The ‘genReads’ utility consists of two functionally independent procedures: genome amplification followed by read simulation. To amplify single cell genomes, we emulate the experimental steps of MALBAC technique ([Supplementary Algorithm S1](#)). A specific parameter γ is introduced to control the number of primers binding to DNA templates. Semi and full amplicons are generated from the DNA templates and further processed to introduce amplification errors. The produced full amplicons are used as the source sequences to yield reads. Here we take an assumption that the probability of sampling a read from an amplicon is proportional to the weighted length of the amplicon ([Supplementary Algorithm S3](#)). Reads are randomly sampled from the amplicons and further fine-tuned to introduce sequencing errors. All refined reads are saved into single FASTQ file for single-end sequencing or two FASTQ files for pair-end sequencing.

3 Results

We assess the effects of parameter γ and sequencing coverage c on SNV detection efficiency, ADO rate and genome coverage by simulating SCS datasets under different conditions. The sequencing data is generated from a 1MB-length sequence that is randomly captured from chromosome 20 of hg19. SNVs are detected using GATK ([McKenna et al., 2010](#)) HaplotypeCaller, and genome coverage is measured using SAMtools ([Li et al., 2009](#)). The results show SNV detection efficiency increases with γ and c ([Fig. 1A and B](#)). For instance, when γ is larger than 2.6×10^{-10} , at least 92 and 56% sensitivity are achieved at $15 \times$ coverage for homozygous and heterozygous SNVs, respectively. The overall SNV detection sensitivity is given in [Supplementary Figure S2](#). As expected, the ADO rate tends to decrease with γ and c ([Fig. 1C](#)), while the genome coverage consistently increases with γ and c ([Fig. 1D](#)). In addition, most of the measurements are basically saturated when γ increases to 9.2×10^{-10} , and slight fluctuations are observed at larger γ values. These results provide helpful guidance for selecting appropriate values of parameter γ and sequencing coverage in different applications.

We also evaluate the reliability of simulation datasets by analyzing the similarity of base quality distributions, and the results manifest that simulated data presents very close distribution to the real data ([Supplementary Figs S4 and S5](#)). Further analysis of GC-content bias shows that similar patterns are shared across real and

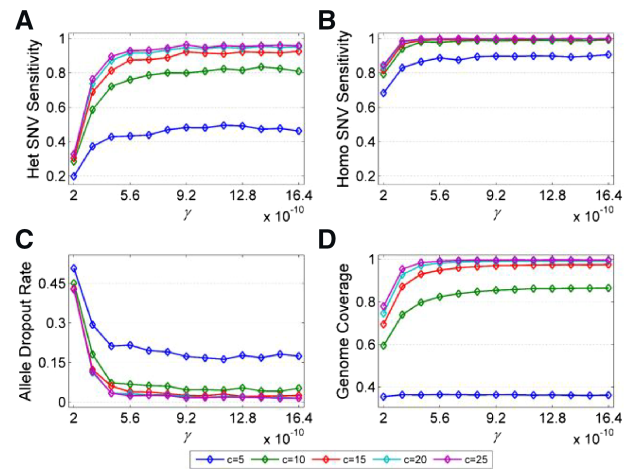


Fig. 1. Comparative analysis of different parameter configurations. The effects of parameter γ and sequencing coverage c on heterozygous SNVs detection efficiency (A), homozygous SNVs detection efficiency (B), allele dropout rate (C) and genome coverage (D) are evaluated

simulation datasets ([Supplementary Fig. S6](#)). When compared to bulk-sequencing simulators, SCSSim can reflect the whole picture of single cell sequencing, and is able to provide similar results with those generated by bulk-sequencing simulators at proper parameter settings ([Supplementary Figs S7 and S8](#)). These results demonstrate that SCSSim is a very useful tool in mimicking single-cell sequencing data. Moreover, runtime performance evaluation suggests that SCSSim has high efficiency under different computational constraints ([Supplementary Table S1](#)).

Funding

This work was supported by the Science and Technique Research Foundation of Ningxia Institutions of Higher Education (NGY2018-54); and the National Natural Science Foundation of China (61901238, 61571414 and 61971393).

Conflict of Interest: none declared.

References

- Brasko, C. et al. (2018) Intelligent image-based in situ single-cell isolation. *Nat. Commun.*, **9**, 226.
- Escalona, M. et al. (2016) A comparison of tools for the simulation of genomic next-generation sequencing data. *Nat. Rev. Genet.*, **17**, 459.
- Li, H. et al. (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.
- McKenna, A. et al. (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.*, **20**, 1297–1303.
- Miura, S. et al. (2018) Computational enhancement of single-cell sequences for inferring tumor evolution. *Bioinformatics*, **34**, i917–i926.
- Ross, E.M. and Markowitz, F. (2016) OncoNEM: inferring tumor evolution from single-cell sequencing data. *Genome Biol.*, **17**, 69.
- Singer, J. et al. (2018) Single-cell mutation identification via phylogenetic inference. *Nat. Commun.*, **9**, 5144.
- Yuan, X. et al. (2017) IntSIM: an integrated simulator of next-generation sequencing data. *IEEE Trans. Biomed. Eng.*, **64**, 441–451.
- Zafar, H. et al. (2016) Monovar: single-nucleotide variant detection in single cells. *Nat. Methods*, **13**, 505.
- Zafar, H. et al. (2017) SiFit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome Biol.*, **18**, 178.
- Zong, C. et al. (2012) Genome-wide detection of single-nucleotide and copy-number variations of a single human cell. *Science*, **338**, 1622–1626.