bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

simuG: a general-purpose genome simulator

Jia-Xing Yue[1*] and Gianni Liti[1*]

[1] Université Côte d'Azur, CNRS, INSERM, IRCAN, Nice, France.

corresponding author: yuejiaxing@gmail.com and gianni.liti@unice.fr

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Abstract

Summary:

Simulated genomes with pre-defined and random genomic variants can be very useful for benchmarking genomic and bioinformatics analyses. Here we introduce simuG, a light-weighted tool for simulating the full-spectrum of genomic variants. The simplicity and versatility of simuG makes it a unique general purpose genome simulator for a wide-range of simulation-based applications.

Availability and implementation: Code in Perl along with user manual and testing data is available at https://github.com/yjx1217/simuG. This software is free for use under the MIT license.

1 Introduction

Along with the rapid progressing of genome sequencing technologies, many bioinformatics tools have been developed for characterizing genomic variants based on genome sequencing data. While there is an increasing availability of experimentally validated gold-standard genome sequencing data set from real biological samples, *in silico* simulation remains a powerful approach for gauging and comparing the performance of bioinformatics tools. Correspondingly, many read simulators have been developed for different sequencing technologies, such as ART (Huang *et al.*, 2012) for Illumina and 454, SimLoRD (Stöcker *et al.*, 2016) for PacBio, and DeepSimulator (Li *et al.*, 2018) for Oxford Nanopore. However, when it comes to tools for simulating genome sequences with embeded variants, the choices

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

appear much limited. The current available tools are either too simple or too specialized. For example, SInC (Pattnaik *et al.*, 2014) can introduce random single nucleotide polymorphisms (SNPs), Insertion/Deletions (INDELs), and copy number variants (CNVs) into a user-provided reference genome but lacks the ability to simulate pre-defined variants, which is actually highly relevant in some simulation applications. Simulome (Price *et al.*, 2017) is another random variant simulator that provides finer control options, but it is designed for prokaryote genome only. More sophisticated tools exist, such as VarSim (Mu *et al.*, 2015) and Xome-Blender (Semeraro *et al.*, 2018), but these tools are majorly tailored for human cancer genome simulation and often require additional third-party databases. Therefore, we feel there is need for a genome simulator that strikes a balance between simplicity and versatility. With this in mind, we developed a general-purpose genome simulator simuG, which is versatile enough to simulate both small (i.e. SNPs and INDELs) and large (i.e. CNVs, inversions, and translocations) genomic variants while staying light weighted with no extra dependency and minimal input requirements. These features together make simuG highly amenable to a wide range of application scenarios.

2 Description and feature highlight

simuG is a command-line tool written in Perl and supports all mainstream operating systems. It takes the user-supplied reference genome as the working template to introduce non-overlapping genomic variants of all major types (i.e. SNPs, INDELs, CNVs, inversions, and translocations). SNP and INDELs can be introduced in the same time, whereas CNVs (implemented as segmental duplications and deletions), inversions, and translocations can be introduced with independent runs. For each variant type, simuG can simulate pre-defined or random variants depending on specified options. For pre-defined variants, a user-supplied

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

VCF file that specifies all desired variants is needed, based on which simuG will operate on the input reference genome to introduce the corresponding variants. For random variants, simuG provides a rich array of options for fine-grained controls, such as '-titv_ratio' for specifying the transition/transversion ratio of SNPs, '-indel_size_powerlaw_alpha' and '-indel_size_powerlaw_constant' for specifying the size distribution of INDELs, '-cnv_gain_loss_ratio' for specifying the ratio of segmental duplication and segmental deletion for CNVs, and '-centromere_gff' for specifying the location of centromeres so that simulated random CNVs, inversions, and translocations will not disrupt the specified centromeres. An ancillary script vcf2model.pl is further provided to directly calculate the best parameter combinations for the random SNP/INDEL simulation based on real data. Moreover, given the strong association between gross chromosomal rearrangement breakpoints and repetitive sequences (e.g. transposable elements) observed in empirical studies (Zhang *et al.*, 2011; Yue *et al.*, 2017), simuG can simulate random inversions and translocations by only sampling from user-defined breakpoints (by specifying the '-inversion_breakpoint_gff' and '-translocation_breakpoint_gff' options). The specific feature type and strand information of these user-defined breakpoints will be considered during the breakpoint sampling. For example, the breakpoint pairs that can trigger inversion should belong to the same feature type but from opposite strands (e.g. inverted repeats). Also, when specified, centromere will be given special consideration in random translocation simulation so that translocations leading to dicentric chromosomes will not be sampled. Finally, when needed, users can also define a list of chromosome(s) to be excluded from variant introduction. Upon the completion of the simulation, three files will be produced: 1) a simulated genome bearing introduced variants in FASTA format, 2) a tabular file showing the genomic locations of all introduced variants relative to both the reference genome and the simulated genome, 3) a VCF file

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

showing the genomic locations of all introduced variants relative to the reference genome. Since simuG's major input/output formats (e.g. FASTA, VCF, and GFF3) are all widely used in the field, it should be fairly straightforward to connect simuG with other computational tools both upstream and downstream in any user-specific simulation study design. Please note that when comparing the VCF outputs from simuG and other tools, all VCF files used for the such comparison should be normalized by tools like vt (Tan *et al.*, 2015) beforehand.

3 Application demonstration

To demonstrate the application of simuG in a real case scenario, we ran simuG with the budding yeast *Saccharomyces cerevisiae* S288C (R64-2-1) reference genome to generate five simulated genomes: 1) with 1000 SNPs + 100 random INDELs, 2) with 10 random inversions, 3) with 5 random inversions triggered by breakpoints sampled from pre-specified transposable elements (TEs), 4) with 2 random translocation, 5) with 2 random translocation triggered by breakpoints sampled from pre-specified TEs. Based on each simulated genome, 50X 150-bp Illumina paired-end reads were simulated with ART (Huang *et al.*, 2012) and mapped to the reference genome by BWA (Li and Durbin, 2009). With this setup, we evaluated the performance of different variant calling tools for both small and large variants (Table 1 and Supplementary Note). For small-variants (i.e. SNP and INDELs), we found freebayes (Garrison and Marth, 2012) and GATK4's HaplotypeCaller (Poplin *et al.*, 2018) both performed well, with the latter one edged out in INDEL calling. For large variants like inversions and translocations, we found both Delly (Rausch *et al.*, 2012) and Manta (Chen *et al.*, 2016) were able to identify simulated events when no TEs were associated with the breakpoints, although the exact breakpoint could be slightly off sometimes, especially with

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Delly. In contrast, for simulated inversions and translocations with TE breakpoints, both tools failed to detect most events in our test.

Variant type	Variant caller	Precision	Recall	F_1 score
SNP (n = 1000)	freebayes	0.997	0.969	0.983
	GATK4	1.000	0.969	0.984
INDEL (n = 100)	freebayes	0.929	0.910	0.919
	GATK4	1.000	0.970	0.984
inversion (n = 10)	Delly	1.000	1.000	1.000
	Manta	1.000	1.000	1.000
inversion with TE breakpoints (n = 5)	Delly	1.000	0.200	0.333
	Manta	1.000	0.200	0.333
translocation (n = 2)	Delly	1.000	1.000	1.000
	Manta	1.000	1.000	1.000
translocation with TE breakpoints (n = 2)	Delly	NA	0.000	NA
	Manta	NA	0.000	NA

Table 1. Benchmarking popular variant callers with the small and large genomic variants simulated by simuG. For each variant type, number of introduced variants are shown in parentheses. TE: transposable elements (*S. cerevisiae* full-length Ty-1 in this case). Precision = true positive/(true positive + false positive). Recall = true positive/(true positive + false negative). F_1 score = 2 * (recall * precision)/(recall + precision).

4 Conclusions

We developed simuG, a simple, flexible, and powerful tool to simulate genome sequences with both pre-defined and random genomic variants. Simple as it is, simuG is highly versatile

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

to handle the full spectrum of genomic variants, which makes it very useful to serve the purpose of various simulation studies.

Funding

This work was supported by Agence Nationale de la Recherche (ANR-16- CE12-0019). J.-X. Yue was supported by a postdoctoral fellowship from Fondation ARC pour la Recherche sur le Cancer (PDF20150602803).

Conflict of Interest: none declared.

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

References

Chen,X. *et al.* (2016) Manta: Rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, **32**, 1220–1222.

Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short-read sequencing. *arXiv Prepr. arXiv1207.3907*, 9.

Huang,W. *et al.* (2012) ART: A next-generation sequencing read simulator. *Bioinformatics*, **15**, 593–594.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li,Y. *et al.* (2018) DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics*, **34**, 2899–2908.

Mu,J.C. *et al.* (2015) VarSim: a high-fidelity simulation and validation framework for high-throughput genome sequencing with cancer applications. *Bioinformatics*, **31**, 1469–1471.

Pattnaik,S. *et al.* (2014) 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.

Poplin,R. *et al.* (2018) Scaling accurate genetic variant discovery to tens of thousands of samples. *bioRxiv*, 201178.

Price,A. *et al.* (2017) Simulome: a genome sequence and variant simulator. *Bioinformatics*, **33**, 1876–1878.

Rausch,T. *et al.* (2012) DELLY: Structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, **28**, i333–i339.

Semeraro,R. *et al.* (2018) Xome-Blender: A novel cancer genome simulator. *PLoS One*, **13**,

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted December 9, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

e0194472.

Stöcker,B.K. *et al.* (2016) SimLoRD: Simulation of Long Read Data. *Bioinformatics*, **32**, 2704–2706.

Tan,A. *et al.* (2015) Unified representation of genetic variants. *Bioinformatics*, **31**, 2202–2204.

Yue,J.-X. *et al.* (2017) Contrasting evolutionary genome dynamics between domesticated and wild yeasts. *Nat. Genet.*, **49**.

Zhang,J. *et al.* (2011) Transposable Elements as Catalysts for Chromosome Rearrangements. In, *Plant Chromosome Engineering*. Humana Press, Totowa, NJ, pp. 315–326.