bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder.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 April 30, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder.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 lightweight tool for simulating the full-spectrum of genomic variants (SNPs, INDELs, CNVs, inversions and translocations) for any organisms (including human). 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 progress 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

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder.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.

it comes to tools for simulating genome sequences with embedded variants, the choices appear much more 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 known 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 genomes only. More sophisticated tools exist, such as VarSim (Mu *et al.*, 2015) and Xome-Blender (Semeraro *et al.*, 2018), but these tools are mostly 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 lightweight with no extra dependency and minimal input requirements. In addition, simuG provides a rich array of fine-grained controls, such as simulating SNPs in different coding partitions (e.g. coding sites, noncoding sites, 4-fold degenerate sites, or 2-fold degenerate sites); simulating CNVs with different formation mechanisms (e.g. segmental deletions, dispersed duplications, and tandem duplications); and simulating inversions and translocations with specific types of breakpoints. These features together make simuG highly amenable to a wide range of application scenarios.

2 Description and feature highlights

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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 is a command-line tool written in Perl and supports all mainstream operating systems. It takes the user-supplied reference genome (in FASTA format) 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 simultaneously, whereas CNVs (implemented as segmental duplications and deletions), inversions, and translocations can be introduced with separated runs. For each variant type, simuG can simulate pre-defined or random variants depending on specified options. For pre-defined variants, a user-supplied 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 supports a wide-spectrum of fine control options, 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 versus segmental deletion, "-duplication_tandem_dispersed_ratio" for specifying the ratio of tandem versus dispersed duplications, 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 restrict random inversions and translocations to only use user-defined breakpoints (by specifying the '-inversion_breakpoint_gff' or '-translocation_breakpoint_gff' option). The specific feature type and strand information of

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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.

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, centromeres 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 chromosomes (e.g. mtDNA) 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 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. 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* (version R64-2-1) and human (version GRCh38) reference genomes to generate nine simulated genomes for each organism: A) with 10000 SNPs, B) with 1000 random INDELs, C) with 10 random CNV due to segmental deletions, D) with 10 random CNV due to dispersed duplications, E) with 10 random CNV due to tandem duplications, F) with 5 random inversions, G) with 5 random inversions triggered by breakpoints sampled from pre-specified transposable elements (TEs), H) with 5 random

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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.

translocation, I) with 5 random translocation triggered by breakpoints sampled from pre-specified TEs. Based on each simulated genome, 50X 150-bp Illumina paired-end reads and 25X PacBio reads were simulated with ART (Huang *et al.*, 2012) and SimLoRd (Stöcker *et al.*, 2016) respectively and subsequently mapped to the yeast and human reference genomes. The read mapping was performed by BWA (Li and Durbin, 2009) for Illumina reads and by minimap2 (Li, 2018) for PacBio reads. With this setup, we evaluated the performance of different variant callers 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 the GATK4 HaplotypeCaller (Poplin *et al.*, 2018) both performed well, with the latter one marginally won out in INDEL calling. For large structural variants like CNVs, inversions, and translocations, we found both the short-read-based callers Delly (Rausch *et al.*, 2012) and Manta (Chen *et al.*, 2016) and the long-read-based caller Sniffles (Sedlazeck *et al.*, 2018) were able to identify most simulated events, especially when no TEs were associated with the breakpoints. The long-read caller Sniffles showed superior accuracy in resolving the exact breakpoints to the basepair resolution than short-read-based callers by taking advantage of the long-reads, even with half of the sequencing coverage. Between the two short-read-based callers, Manta outperformed Delly in terms of breakpoint accuracy at the basepair level.

		Yeast			Human		
Variant type	**Variant caller**	**Precision**	**Recall**	**F_1 score**	**Precision**	**Recall**	**F_1 score**
SNP	freebayes	1.000	0.971	0.985	0.999	0.981	0.990
(n = 10000)	GATK4	1.000	0.970	0.985	1.000	0.977	0.988
INDEL	freebayes	0.954	0.931	0.942	0.939	0.930	0.935

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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.

(n = 1000)	GATK4	1.000	0.969	0.984	1.000	0.976	0.988
CNV:	Delly	1.000	1.000	1.000	1.000	1.000	1.000
segmental deletion	Manta	1.000	1.000	1.000	1.000	1.000	1.000
(n = 10)	Sniffles	1.000	1.000	1.000	1.000	1.000	1.000
CNV:	Delly	1.000	0.875	0.933	1.000	0.906	0.951
dispersed duplication	Manta	1.000	1.906	0.951	1.000	0.906	0.951
(n = 10)	Sniffles	1.000	0.875	0.933	1.000	0.906	0.951
CNV:	Delly	1.000	1.000	1.000	1.000	0.700	0.824
tandem duplication	Manta	1.000	1.000	1.000	1.000	0.700	0.824
(n = 10)	Sniffles	1.000	1.000	1.000	1.000	0.800	0.889
Inversion	Delly	1.000	1.000	1.000	1.000	1.000	1.000
(n = 5)	Manta	1.000	1.000	1.000	1.000	1.000	1.000
	Sniffles	1.000	1.000	1.000	1.000	1.000	1.000
Inversion with TE	Delly	1.000	0.200	0.333	1.000	1.000	1.000
breakpoints	Manta	1.000	0.200	0.333	1.000	1.000	1.000
(n = 5)	Sniffles	1.000	0.200	0.333	1.000	1.000	1.000
Translocation	Delly	1.000	1.000	1.000	0.800	0.800	0.800
(n = 5)	Manta	1.000	1.000	1.000	1.000	1.000	1.000
	Sniffles	1.000	1.000	1.000	1.000	1.000	1.000
Translocation with TE	Delly	NA	0.000	NA	1.000	1,000	1.000
breakpoints	Manta	NA	0.000	NA	1.000	1.000	1.000
(n = 5)	Sniffles	NA	0.000	NA	1.000	1.000	1.000

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 (full-length Ty1 for *S. cerevisiae* and full-length intact L1 for human). Precision = true positive/(true positive + false positive). Recall = true positive/(true positive + false negative). F_1 score = 2 * (recall * precision)/(recall + precision). For a single CNV derived from dispersed duplication, there could be multiple duplicated copies inserted to different genomic locations, making it tricky to calculate

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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.

accuracy, precision, and F_1 score by measuring the number of recovered CNV events. Therefore, we calculated these values based on the number of recovered breakpoints instead in this case.

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 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 and ANR-15-IDEX-01). J.-X. Yue was supported by a postdoctoral fellowship from Fondation ARC pour la Recherche sur le Cancer (PDF20150602803). Part of computation involved in this work was performed via the Extreme Science and Engineering Discovery Environment (XSEDE) (TG-BIO170065).

Conflict of Interest: none declared.

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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. (2018) Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*, **34**, 3094–3100.

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

bioRxiv preprint doi: https://doi.org/10.1101/491498; this version posted April 30, 2019. 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.

split-read analysis. *Bioinformatics*, **28**, i333–i339.

Sedlazeck,F.J. *et al.* (2018) Accurate detection of complex structural variations using single-molecule sequencing. *Nat. Methods*, **15**, 461–468.

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