

1

2

3 simuG: a general-purpose genome simulator

4

5

6

7

8 Jia-Xing Yue^{1*} and Gianni Liti^{1*}

9

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

11

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

13

14

15

16

17

1 **Abstract**

2

3 **Summary:**

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

9

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

13

14 **1 Introduction**

15 Along with the rapid progressing of genome sequencing technologies, many bioinformatics
16 tools have been developed for characterizing genomic variants based on genome sequencing
17 data. While there is an increasing availability of experimentally validated gold-standard
18 genome sequencing data set from real biological samples, *in silico* simulation remains a
19 powerful approach for gauging and comparing the performance of bioinformatics tools.
20 Correspondingly, many read simulators have been developed for different sequencing
21 technologies, such as ART (Huang *et al.*, 2012) for Illumina and 454, SimLoRD (Stöcker *et al.*,
22 2016) for PacBio, and DeepSimulator (Li *et al.*, 2018) for Oxford Nanopore. However, when
23 it comes to tools for simulating genome sequences with embedded variants, the choices

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

16

17 **2 Description and feature highlight**

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

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

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

7

8 **3 Application demonstration**

9 To demonstrate the application of simuG in a real case scenario, we ran simuG with the
10 budding yeast *Saccharomyces cerevisiae* S288C (R64-2-1) reference genome to generate five
11 simulated genomes: 1) with 1000 SNPs + 100 random INDELS, 2) with 10 random inversions,
12 3) with 5 random inversions triggered by breakpoints sampled from pre-specified
13 transposable elements (TEs), 4) with 2 random translocation, 5) with 2 random translocation
14 triggered by breakpoints sampled from pre-specified TEs. Based on each simulated genome,
15 50X 150-bp Illumina paired-end reads were simulated with ART (Huang *et al.*, 2012) and
16 mapped to the reference genome by BWA (Li and Durbin, 2009). With this setup, we
17 evaluated the performance of different variant calling tools for both small and large variants
18 (Table 1 and Supplementary Note). For small-variants (i.e. SNP and INDELS), we found
19 freebayes (Garrison and Marth, 2012) and GATK4's HaplotypeCaller (Poplin *et al.*, 2018) both
20 performed well, with the latter one edged out in INDEL calling. For large variants like
21 inversions and translocations, we found both Delly (Rausch *et al.*, 2012) and Manta (Chen *et*
22 *al.*, 2016) were able to identify simulated events when no TEs were associated with the
23 breakpoints, although the exact breakpoint could be slightly off sometimes, especially with

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

3

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

4

5 **Table 1. Benchmarking popular variant callers with the small and large genomic variants simulated by**
 6 **simuG.** For each variant type, number of introduced variants are shown in parentheses. TE: transposable
 7 elements (*S. cerevisiae* full-length Ty-1 in this case). Precision = true positive/(true positive + false positive).
 8 Recall = true positive/(true positive + false negative). F₁ score = 2 * (recall * precision)/(recall + precision).

9

10 4 Conclusions

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

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

3

4 **Funding**

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

8

9

10 Conflict of Interest: none declared.

11

1 **References**

- 2 Chen,X. *et al.* (2016) Manta: Rapid detection of structural variants and indels for germline
3 and cancer sequencing applications. *Bioinformatics*, **32**, 1220–1222.
- 4 Garrison,E. and Marth,G. (2012) Haplotype-based variant detection from short-read
5 sequencing. *arXiv Prepr. arXiv1207.3907*, 9.
- 6 Huang,W. *et al.* (2012) ART: A next-generation sequencing read simulator. *Bioinformatics*,
7 **15**, 593–594.
- 8 Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler
9 transform. *Bioinformatics*, **25**, 1754–1760.
- 10 Li,Y. *et al.* (2018) DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics*,
11 **34**, 2899–2908.
- 12 Mu,J.C. *et al.* (2015) VarSim: a high-fidelity simulation and validation framework for high-
13 throughput genome sequencing with cancer applications. *Bioinformatics*, **31**, 1469–
14 1471.
- 15 Pattnaik,S. *et al.* (2014) SInC: an accurate and fast error-model based simulator for SNPs,
16 Indels and CNVs coupled with a read generator for short-read sequence data. *BMC*
17 *Bioinformatics*, **15**, 40.
- 18 Poplin,R. *et al.* (2018) Scaling accurate genetic variant discovery to tens of thousands of
19 samples. *bioRxiv*, 201178.
- 20 Price,A. *et al.* (2017) Simulome: a genome sequence and variant simulator. *Bioinformatics*,
21 **33**, 1876–1878.
- 22 Rausch,T. *et al.* (2012) DELLY: Structural variant discovery by integrated paired-end and
23 split-read analysis. *Bioinformatics*, **28**, i333–i339.
- 24 Semeraro,R. *et al.* (2018) Xome-Blender: A novel cancer genome simulator. *PLoS One*, **13**,

- 1 e0194472.
- 2 Stöcker, B.K. *et al.* (2016) SimLoRD: Simulation of Long Read Data. *Bioinformatics*, **32**, 2704–
- 3 2706.
- 4 Tan, A. *et al.* (2015) Unified representation of genetic variants. *Bioinformatics*, **31**, 2202–
- 5 2204.
- 6 Yue, J.-X. *et al.* (2017) Contrasting evolutionary genome dynamics between domesticated
- 7 and wild yeasts. *Nat. Genet.*, **49**.
- 8 Zhang, J. *et al.* (2011) Transposable Elements as Catalysts for Chromosome Rearrangements.
- 9 In, *Plant Chromosome Engineering*. Humana Press, Totowa, NJ, pp. 315–326.
- 10