OncoSimulR: genetic simulation of cancer progression with arbitrary epistasis and mutator genes

Ramon Diaz-Uriarte

Department of Biochemistry, Universidad Autónoma de Madrid, Instituto de Investigaciones Biomédicas "Alberto Sols" (UAM-CSIC), Madrid, Spain

ramon.diaz@iib.uam.es, http://ligarto.org/rdiaz

1. Introduction

Forward-in-time genetic simulations are widely used in population genetics and cancer research to verify analytic results, to assess the performance of statistical methods, and to examine mathematically intractable models (reviews and examples in [1–5]). Many forward-in-time simulators are available (see https://popmodels.cancercontrol.cancer.gov/gsr/).

Do we need another forward-in-time genetics simulator?
Yes, if we want several of the features in Boxes 1 or 2.

Box 1. Fitness, mutation, and mutator specifications with OncoSimulR

1. Flexibility to specify **genotype's fitness** using a variety of combinable procedures:
   a. Specifying the fitness of each genotype.
   b. Using a lego system to combine:
      i. Effects of individual genes and **epistatic effects** of any order that involve an arbitrary number of genes.
      ii. **Order effects** involving an arbitrary number of genes. With order effects, recently discovered in myeloma [6], the fitness of a genotype with genes A and B mutated differs depending on whether A or B mutated first.
      iii. **Directed acyclic graphs (DAGs)**, as used in Oncogenetic Trees and Conjunctive Bayesian Networks [7], to represent restrictions in the order of accumulation of mutations.
      - Conjunctions in the DAG can represent AND, OR, or XOR relationships (Figure 1).
   c. **"Modules"**: effects of mutations on fitness (or mutation rates) depend not on the specific genes, but on modules or pathways defined as collections of genes (Figure 1).
2. **Gene-specific mutation rates** or common genome-wide mutation rates.
3. **Mutator/antimutator genes**, genes whose mutation leads to an increase/decrease in the mutation rates of other genes [8].
   - Effects specified using 1.b.i and 1.c.
   - Genes with (anti)mutator effects can also have direct effects on fitness.
4. **Large genomes** (> 50000 genes).

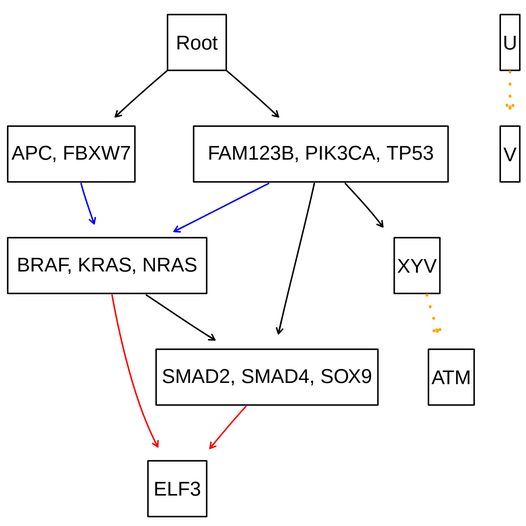

Fig. 1: Fictitious fitness specification using a DAG with AND (black), OR (blue) and XOR (red) dependencies, order effects (orange), and modules (boxes with > 1 gene).

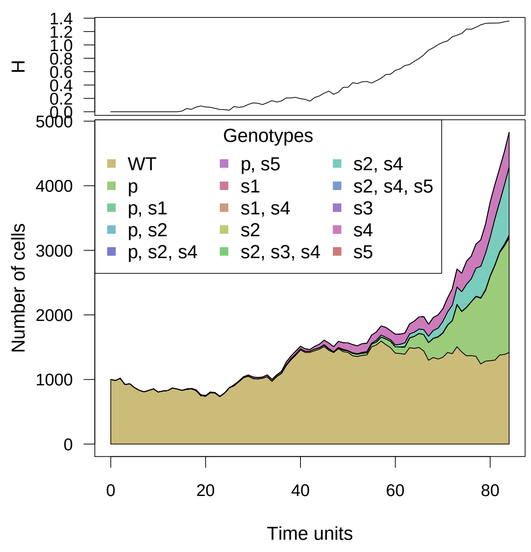

Fig. 2: Genotype abundances and diversity (top panel) over time for one simulation under Bauer's model of epistasis between driver and passengers [9], using exponential growth.

Box 2. Simulations: starting, stopping, sampling with OncoSimulR

5. Running simulations:
   a. Continuous time, using BNB algorithm [10]. Growth models: **exponential** (Figure 2) and logistic-like with **carrying capacity** [11].
   b. Multiple simulations **parallelized**.
   c. **Large populations** ($> 10^{13}$ cells).
   d. Any genotype can be used as the **starting genotype**.
6. Stopping the simulations:
   a. Population size, time periods, or mutated driver genes larger than user-specified values.
   b. **Probability of tumor detection: increasing function of tumor size**.
7. Tracking and sampling:
   a. Tracking and plotting the **complete history of all clones** (Figure 3).
   b. Sampling time: arbitrary times, uniformly, or as a function of population size.
   c. Sampling granularity: **whole tumor** or **single-cell**.

ABSTRACT

OncoSimulR implements forward-in-time genetic simulations of diallelic loci in asexual populations, with special focus on cancer progression. Fitness can be defined as an arbitrary function of genetic interactions between multiple genes or modules of genes, including epistasis, restrictions in the order of accumulation of mutations, and order effects. Mutation rates can be made to differ between genes, and can be affected by (anti)mutator genes. Also available are sampling from single or multiple simulations, including single-cell sampling, plotting the parent-child relationships of the clones, and generating and plotting random fitness landscapes. OncoSimulR is implemented in R and C++. Version $\geq$ 2.3.12 available from **BioConductor** (http://www.bioconductor.org/packages/devel/bioc/html/OncoSimulR.html) for Linux, Mac, and Windows under the GNU GPL license. GitHub repository at: https://github.com/rdiaz02/OncoSimul.

2. How fast is it?

Simulation code in C++. Depending on evolutionary model, stopping conditions, etc, 100 simulations in 0.05 to 10 seconds.

3. Tests, code coverage, documentation

- Over 2000 tests run at every check: **95% code coverage** (C++ and R). Another 500 long-running tests.
- Vignette (> 140 pp.) with examples.

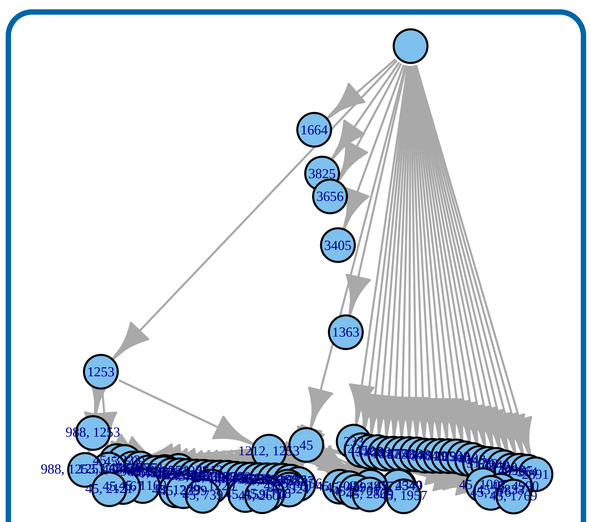

Fig. 3: Parent-child relationships of all clones (≥ 1 cells at end) for one simulation. Numbers: id of mutated gene.

Box 3. OncoSimulR: other functionality

8. Generating random fitness landscapes (Rough Mount Fuji, House of Cards, additive models: see [12]).
9. Plotting fitness landscapes, inspired by MAGELLAN [13] (Figure 4).
10. Generating random DAGs of restrictions in the order of mutations (as for 1.b.iii).

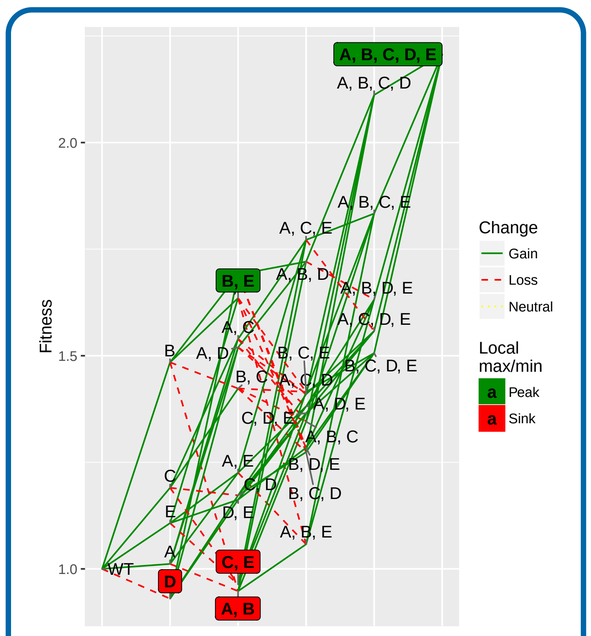

Fig. 4: Fitness landscape plot (*à la* MAGELLAN: [13]) of a randomly generated Rough Mount Fuji fitness landscape.

4. Conclusions

OncoSimulR can be used to address a **wide range of questions**, from the effects of mutator genes to the interplay between fitness landscapes and mutation rates. OncoSimulR also adds features **convenient for examining cancer evolution models**, such as modules, order effects, or flexibility for sampling and stopping the simulations. Some tools closest to provide the functionality in Boxes 1 and 2 are simuPOP [1], fwdpp [3] and FFPopSim [5] but they miss flexible ways to specify fitness, mutator and order effects, gene-specific mutation rates, or sampling schemes. TTP [14] is limited to four genes. OncoSimulR's unique characteristics therefore make it of interest for a broad scientific readership that covers from population and evolutionary geneticists to computational oncologists.

5. Funding

Supported by BFU2015-67302-R (MINECO/FEDER, EU).

6. References

[1] Peng, B., et al. (2012). *Forward-time population genetics simulations: methods, implementation, and applications*. Wiley-Blackwell.
[2] Diaz-Uriarte, R. (2015). *BMC Bioinformatics*, **16**, 1–36.
[3] Thornton, K. R. (2014). *Genetics*, **198**, 157–166.
[4] Yuan, X., et al. (2012). *J. Comp. Biol.*, **19**, 42–54.
[5] Zanini, F. and Neher, R. A. (2012). *Bioinformatics*, **28**, 3332–3333.
[6] Ortmann, C. A., et al. (2015). *NEJM*, **372**, 601–612.
[7] Beerenwinkel, N., et al. (2014). *Syst. Biol.*, **64**, e1–e25.
[8] Gerrish, P. J., et al. (2007). *PNAS*, **104**, 6266–6271.
[9] Bauer, B. et al. (2014). *J. Theor. Biol.*, 358:52–60.
[10] Mather, W. H., et al. (2012). *Bioinformatics*, **28**, 1230–1238.
[11] McFarland, C. D., et al. (2013). *PNAS*, **110**, 2910–2915.
[12] Szendro, I. G., et al. (2013). *J. Stat. Mech. Theor. Exp.*, **2013**, P01005.
[13] Brouillet, S., et al. (2015). *bioRxiv*, doi:10.1101/031583.
[14] Reiter, J., et al. (2013). In N. Sharygina and H. Veith, eds., *Lecture Notes in Computer Science*, pp. 101–106. Springer-Verlag.