

CancerSim: A Cancer Simulation Package for Python 3

Luka Opasic¹, Jacob G. Scott², Arne Traulsen¹, and Carsten Fortmann-Grote¹

¹ Max-Planck-Institute for Evolutionary Biology, Plön, Germany ² Cleveland Clinic, Cleveland, OH, US

DOI: [10.21105/joss.02436](https://doi.org/10.21105/joss.02436)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Pierre de Buyl](#) ↗

Reviewers:

- [@jenzopr](#)
- [@blakeaw](#)

Submitted: 25 May 2020

Published: 23 September 2020

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Cancer is a group of complex diseases characterized by excessive cell proliferation, invasion, and destruction of the surrounding tissue (Vinay Kumar, 2017). Its high division and mutation rates lead to excessive genetic diversity among tumour cells (intra tumour genetic heterogeneity). As a consequence, tumours can adapt very efficiently to environmental pressures, in particular to cancer therapy (Turajlic, Sottoriva, Graham, & Swanton, 2019). This process is known as somatic evolution of cancer.

Throughout most of its existence a tumour is inaccessible to direct observation and experimental evaluation through genetic sequencing of tumour samples. Therefore, computational modelling can be useful to study many aspects of cancer. Examples where theoretical models can be of great use include (i) early carcinogenesis, as lesions are clinically observable when they already contain millions of cells, (ii) seeding of metastases, and (iii) cancer cell dormancy (Altrock, Liu, & Michor, 2015).

Statement of Need

Advanced cancer simulation software (Waclaw et al., 2015) often exhibit a prohibitively steep learning curve especially for new students in the field of somatic evolution of cancer. A software package that is accessible, simple to use, and yet covers the essential biological processes of cancer growth is needed to provide an entry point for students and newcomers to mathematical oncology.

Cancer growth model

Here we present CancerSim, a software that simulates somatic evolution of tumours. The software produces virtual spatial tumours with variable extent of intra tumour genetic heterogeneity and realistic mutational profiles (i.e., the order of appearance of mutations and their distribution among tumour cells). Simulated tumours can be virtually sampled at random or in specified positions. By binning the frequency of mutations in each sample and convolving the resulting histogram with a beta-binomial distribution to add realistic sequencing noise (Williams, Werner, Barnes, Graham, & Sottoriva, 2016), the simulation produces mutation profiles from different tumour regions. This makes the software useful for studying various sampling strategies in clinical cancer diagnostics such as needle biopsy sampling or liquid biopsy sampling. An early version of this cancer evolution model was used to simulate tumours subjected to sampling for classification of mutations based on their abundance (Opasic, Zhou, Werner, Dingli, & Traulsen, 2019).

Our model is abstract, not specific to any neoplasm type, and does not consider a variety of biological features commonly found in the neoplasm, such as blood vessels, immune cells,

nutrient availability, and architecture of the tumour surroundings. It resembles the most superficially spreading tumours like carcinoma *in-situ*, skin cancers, or gastric cancers, but it can be used to model any tumour on this abstract level.

The tumour is simulated using a two-dimensional, on-lattice, agent-based model. The tumour lattice structure is established by a sparse matrix whose non-zero elements correspond to the individual cells. Each cell is surrounded by eight neighbouring cells (Moore neighbourhood). The value of the matrix element is an index pointing to the last mutation the cell acquired in the list of mutations which is updated in each simulation step.

Simulation parameters

The simulation advances in discrete time-steps. In each simulation step, every tumour cell on the lattice that has an unoccupied neighbour can divide with a certain probability (controlled through the parameter `division_probability`). The daughter cell resulting from a cell division inherits all mutations from the parent cell and acquires a new mutation with a given probability (`mutation_probability`). A new mutation that changes death and/or birth probabilities can be introduced into a random cell at a specific time step defined by `adv_mutation_wait_time`. By changing the fitness parameters of a mutant cell `adv_mutant_division_probability` and `adv_mutant_death_probability` one can model various evolutionary processes like emergence of a faster dividing sub-clone or selective effects of drug treatment.

The simulation allows the acquisition of more than one mutational event per cell (`number_of_mutations_per_division`). In that case, variable amounts of sequencing noise (Williams et al., 2016) can be added to make the output data biologically more realistic. The key parameters `number_of_generations`, `division_probability` and `death_probability` determine the final size of the tumour, while the degree of intra tumour heterogeneity can be varied by changing the `mutation_probability` parameter. For neutral tumour evolution, parameter `adv_mutant_division_probability` and `adv_mutant_death_probability` must be the same as `division_probability` and `death_probability`.

Throughout the cancer growth phase, `CancerSim` stores information about the parent cell and a designation of newly acquired mutations for every cell. Complete mutational profiles of cells are reconstructed a posteriori based on the stored lineage information.

The division rules which allow only cells with empty neighbouring nodes to divide, cause exclusively peripheral growth and complete absence of dynamics in the tumour centre. To allow for variable degrees of growth inside the tumour, we introduced a death process. At every time step, after all cells attempt their division, a number of random cells die according to `death_probability` and `adv_mutant_death_probability` and yield their position to host a new cancer cell in a subsequent time step.

Simulation results

After the simulation, the tumour matrix, and the lists of lineages and frequencies of each mutation in the tumour are exported to files. Furthermore, the virtual tumour can be sampled and a histogram over the frequency of mutations will be visualised. Alternatively, a saved tumour can be loaded from file and then subjected to the sampling process.

Outlook

Future investigations and expansions of `CancerSim` will focus on sampling of tumour specimens in a specific spatial pattern. Additionally, the effects of chemotherapy can be modelled by the introduction of different modes of cell death. Our model only considers sequencing noise for mutated cells. Future improvements of our model will also consider other sources of

sequencing noise, e.g., read errors from non-mutated sites which would lead to false-positive cancer detections.

Download and Installation

CancerSim is written in Python (version >3.5). We recommend to install it directly from the source code hosted at github https://github.com/mpievbio-scicomp/cancer_sim.

Detailed instructions including creation of a conda environment are given in the online documentation at <https://cancer-sim.readthedocs.io/en/master/include/README.html#installation>. After installation, the software is available as a python module `casim`.

High-level functionality

Setting up the cancer simulation parameters

The parameters of the cancer simulation are specified in a python module or programmatically via the `CancerSimulationParameters` class. The table below lists all parameters, their function and acceptable values. [...] denotes a range of permissive parameter values.

Parameter name	function	valid options
<code>matrix_size</code>	Number of mesh points in each dimension	>0
<code>number_of_generations</code>	Number of generations to simulate	>0
<code>division_probability</code>	Probability of cell division per generation	[0,1]
<code>adv_mutant_division_probability</code>	Probability of division for cells with advantageous mutation	[0,1]
<code>death_probability</code>	Fraction of cells that die per generation	[0,1]
<code>adv_mutant_death_probability</code>	Fraction of cells with advantageous mutation that die per generation	[0,1]
<code>mutation_probability</code>	Probability of mutations	[0,1]
<code>adv_mutant_mutation_probability</code>	Mutation probability for the adv. cells	[0,1]
<code>number_of_mutations_per_division</code>	Number of mutations per cell division	>=0
<code>adv_mutation_wait_time</code>	Number of generations after which adv. mutation occurs	>0
<code>number_of_initial_mutations</code>	Number of mutations present in first cancer cell	>=0

Parameter name	function	valid options
tumour_multiplicity	Tumour multiplicity	“single”, “double”
read_depth	Sequencing read depth	read length * number of reads / genome length
sampling_fraction	Fraction of cells to be sampled	[0,1]
sampling_positions	A list of [(x,y)] coordinates specifying the grid points where to take samples from the tumour. If unset or None, a single random position is chosen.	[0, matrix_size-1]
plot_tumour_growth	Plot the tumour growth curve	True, False
export_tumour	Export the tumour growth data to file	True, False

The file [params.py](#) can serve as a template to setup a simulation with all above parameters. Users should start with the template and adjust the parameters as needed for their application by setting experimentally or theoretically known values or by calibrating the simulation output against experiments or other models.

Run the simulation

The simulation is started either from the command line, through a python script or from within an interactive python session. The command line interface accepts as arguments a random seed, a path for the output directory and a log level. More detailed instructions are given in the [online documentation](#) and in the [reference manual](#).

For the impatient, we also provide a jupyter notebook with a more condensed version of the above example (grid size 20x20) at docs/source/include/notebooks/quickstart_example.ipynb. An interactive version can be launched on the [Binder service](#).

A further notebook demonstrates the possibility to dump a simulation to disk, reload it and continue the simulation with optionally changed parameters (docs/source/include/notebooks/run_dump_reload_continue.ipynb). This feature could be exploited to simulate cancer dormancy with CancerSim.

Output

After the run has finished, all output is found in the specified output directory. Simulation output consists of the mutation list and a sample file representing the sampled portion of the tumour. Two pdf files are generated showing the mutation histograms of the whole tumour and the sampled portion, respectively. If requested by the parameter `plot_tumour_growth`, a third pdf is generated showing the tumour growth curve (number of tumour cells over time). Furthermore, the output contains serialized versions (“pickles”) of the tumour geometry as a 2D matrix, the death list, and the mutation list. Another directory contains all logged information from the run. The amount of logged information depends on the chosen log level.

As an example and possible starting point for further investigations beyond the produced plots, the quick start example notebook demonstrates how to load the tumour matrix into memory and how to visualize the mutational profile as a heat map as shown in Figure 1.

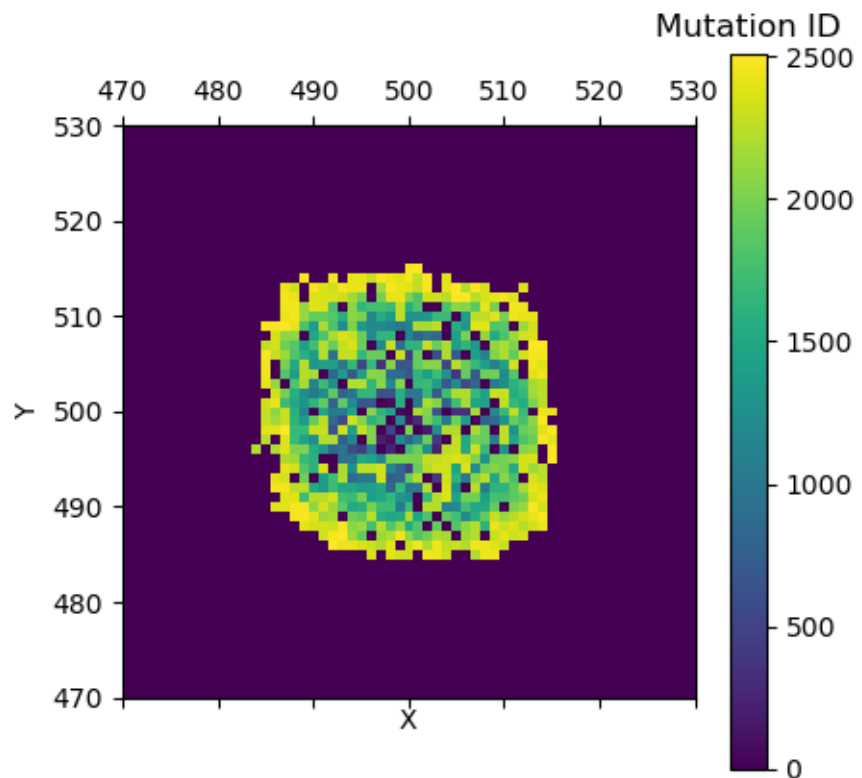


Figure 1: Tumour matrix as a 2D grid with colored cells indicating the order in which mutations appear.

The figure shows the central part of the tumour matrix that results from the template parameter file `params.py` on the X-Y grid with colors indicating the order in which mutations appear. 0 (deep purple) indicates no mutation in this cell.

Documentation and support

The API reference manual and community guidelines including directions for contributors and bug reports are given in the online documentation at <https://cancer-sim.readthedocs.io>.

Testing and continuous integration

Each commit to the code base triggers a build and test run on <https://travis-ci.org>. The build and test status can be observed at https://travis-ci.org/mpievolbio-scicomp/cancer_sim.

References

- Altrock, P. M., Liu, L. L., & Michor, F. (2015). The mathematics of cancer: Integrating quantitative models. *Nat Rev Cancer*, *15*(12), 730–45. doi:[10.1038/nrc4029](https://doi.org/10.1038/nrc4029)
- Opasic, L., Zhou, D., Werner, B., Dingli, D., & Traulsen, A. (2019). How many samples are needed to infer truly clonal mutations from heterogenous tumours? *BMC Cancer*, *19*, 403. doi:[10.1186/s12885-019-5597-1](https://doi.org/10.1186/s12885-019-5597-1)
- Turajlic, S., Sottoriva, A., Graham, T., & Swanton, C. (2019). Resolving genetic heterogeneity in cancer. *Nat Rev Genet*. doi:[10.1038/s41576-019-0114-6](https://doi.org/10.1038/s41576-019-0114-6)
- Vinay Kumar, J. C. A., Abul K. Abbas. (2017). *Robbins basic pathology*. Robbins pathology (10th ed.). Elsevier. ISBN: [978-0-323-48054-3](https://doi.org/10.1016/B978-0-323-48054-3)
- Waclaw, B., Bozic, I., Pittman, M. E., Hruban, R. H., Vogelstein, B., & Nowak, M. A. (2015). A spatial model predicts that dispersal and cell turnover limit intratumour heterogeneity. *Nature*, *525*(7568), 261–264. doi:[10.1038/nature14971](https://doi.org/10.1038/nature14971)
- Williams, M. J., Werner, B., Barnes, C. P., Graham, T. A., & Sottoriva, A. (2016). Identification of neutral tumor evolution across cancer types. *Nature Genetics*, *48*, 238–244. doi:[10.1038/ng.3489](https://doi.org/10.1038/ng.3489)