
CancerSim

Release 1.0.0

Mar 17, 2020

Contents:

1	Background	3
2	Installation	5
2.1	Alternative 1: Conda	5
2.2	Alternative 2: Using pip	6
3	Testing	7
4	High-level functionality	9
4.1	Example 1	10
4.2	Example 2	10
5	Examples	11
6	References	13
7	Quickstart example for cancer simulation	15
7.1	Import modules	15
7.2	Setup parameters	15
7.3	Setup the simulation engine.	16
7.4	Run the simulation	16
7.5	Display the growth curve.	16
8	Reference Manual	19
9	Indices and tables	21

Build Status Documentation Status Binder

Cancer is a group of complex diseases characterized by excessive cell proliferation, invasion, and destruction of the surrounding tissue [1]. Its high division and mutation rates lead to excessive intratumour genetic heterogeneity which makes cancer highly adaptable to environmental pressures such as therapy [2]. Throughout most of its existence tumour is inaccessible to direct observation and experimental evaluation. Therefore, computational modelling can be useful to study many aspects of cancer. Some examples where theoretical models can be of great use include early carcinogenesis, as lesions are clinically observable when they already contain millions of cells, seeding of metastases, and cancer cell dormancy [3].

Here, we present CancerSim, a software that simulates somatic evolution of tumours. The software produces virtual spatial tumours with variable extent of intratumour genetic heterogeneity and realistic mutational profiles. Simulated tumours can be subjected to multi-region sampling to obtain mutation profiles that are realistic representation of the sequencing data. This makes the software useful for studying various sampling strategies in clinical cancer diagnostics. An early version of this cancer evolution model was used to simulate tumours subjected to sampling for classification of mutations based on their abundance [4]. Target users are scientists working in the field of mathematical oncology and students with interest in studying somatic evolution of cancer.

Our model is abstract, not specific to any neoplasm type and does not consider a variety of biological features commonly found in neoplasm such as vasculature, immune contexture, availability of nutrients, and architecture of the tumour surroundings. It resembles the most to 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 cell acquired in the list of mutations which is updated in each simulation step.

The simulation advances in discrete time-steps. In each simulation step, every tumour cell in the tumour that has an unoccupied neighbour can divide with a certain probability (`params.div__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 (`params.mut_prob`). Different division probabilities can be introduced for some cells in order to simulate variability in fitness of cells that acquired a beneficial or deleterious mutation. The simulation allows the acquisition of more than one mutational event per cell (`params.mut_per_division`). In that case, variable amounts of sequencing noise [5] can be added to make the output data more biologically realistic.

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 degree 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 and yield their position to host a new cancer cell in a subsequent time step.

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.

CHAPTER 2

Installation

CancerSim is written in Python (version >3.5). We recommend to install it directly from the source code. To download the code:

EITHER clone the repository:

```
$> git clone https://github.com/mpievolbio-scicomp/cancer_sim.git
```

OR download the source code archive:

```
$> wget https://github.com/mpievolbio-scicomp/cancer_sim/archive/master.zip
$> unzip master.zip
$> mv cancer_sim-master cancer_sim
```

Change into the source code directory

```
$> cd cancer_sim
```

We provide for two alternatives to install the software after it was downloaded:

2.1 Alternative 1: Conda

2.1.1 New conda environment

We provide an `environment.yml` to be consumed by `conda`. To create a fully self-contained conda environment (named `casim`):

```
$> conda env create -n casim --file environment.yml
```

This will also install the cancer simulation code into the new environment.

To activate the new conda environment:

```
$> source activate casim
```

or

```
$> conda activate casim
```

if you have set up conda appropriately.

2.1.2 Install into existing and activated conda environment

To install the software into an already existing environment:

```
$> conda activate <name_of_existing_conda_environment>  
$> conda env update --file environment.yml
```

2.2 Alternative 2: Using pip

The file `requirements.txt` is meant to be consumed by pip:

```
$> pip install -r requirements.txt [--user]
```

The option `--user` is needed to install without admin privileges.

Although not strictly required, we recommend to run the test suite after installation. Simply execute the `run_tests.sh` shell script:

```
$> ./run_tests.sh
```

This will generate a test log named `casim_test@<timestamp>.log` with `<timestamp>` being the date and time when the test was run. You should see an OK at the bottom of the log. If instead errors or failures are reported, something is wrong with the installation or the code itself. Feel free to open a github issue at https://github.com/mpievolbio-scicomp/cancer_sim/issues and attach the test log plus any information that may be useful to reproduce the error (version hash, computer hardware, operating system, python version, a dump of `conda env export` if applicable, ...).

The test suite is automatically run after each commit to the code base. Results are published on travis-ci.org.

CHAPTER 4

High-level functionality

The parameters of the cancer simulation are given via a python module or programmatically via the `CancerSimulationParameters` class. A documented example `params.py` is included in the source code (under `test/params.py`) and reproduced here:

```
$> cat test/params.py
# Number of mesh points in each dimension
matrix_size                = 100

# Number of generations to simulate.
num_of_generations         = 20

# Number of divisions per generation.
div_probability            = 1

# Number of division for cells with mutation.
fitness_advantage_div_prob = 1

# Fraction of cells that die per generation.
dying_fraction            = 0.1

# Fraction of cells with mutation that die per generation.
fitness_advantage_death_prob = 0.0

# Rate of mutations.
mutation_probability       = 1

# Mutation probability for the adv. cells.
advantageous_mut_prob      = 1

# Number of mutations per cell division.
mut_per_division          = 10

# Time after which adv. mutations occur.
time_of_adv_mut           = 10
```

(continues on next page)

(continued from previous page)

```
# Number of mutations present in first cancer cell.
num_of_clonal                = 150

# Tumour multiplicity.
tumour_multiplicity          = None

# Sequencing read depth.
read_depth                   = 100

# Fraction of cells to be sampled.
# sampling_fraction          = 0.9

# Plot the tumour growth curve.
plot_tumour_growth           = True

# Export the tumour growth data to file.
export_tumour                 = True
```

The simulation is started from the command line. The syntax is

```
$> python -m casim.casim [-h] [-o DIR] seed
```

The mandatory command line argument `seed` is the random seed. Using the same seed in two simulation runs with identical parameters results in identical results. This feature may be used for testing and debugging. The optional argument `DIR` specifies the directory where to store the simulation log and output data. If not given, output will be stored in the directory `casim_out` in the current directory. For each seed, a subdirectory `cancer_SEED` will be created. If that subdirectory already exists because an earlier run used the same seed, the run will abort. This is a safety catch to avoid overwriting data from previous runs.

4.1 Example 1

```
$> python -m casim.casim 1
```

4.2 Example 2

```
$> mkdir sim_out
$> python -m casim.casim -o sim_out 2
```

Results will be stored in the newly created directory `sim_out/`.

CHAPTER 5

Examples

See our quickstart example in `docs/source/include/notebooks/quickstart_example.ipynb` or use the following link to [launch it in Binder](#).

CHAPTER 6

References

- [1] J. C. A. Vinay Kumar Abul K. Abbas, *Robbins Basic Pathology*, 10th ed. (Elsevier, 2017). ISBN: 9780323353175.
- [2] S. Turajlic, A. Sottoriva, T. Graham, and C. Swanton, *Nat Rev Genet* (2019). DOI: [10.1038/s41576-019-0114-6](https://doi.org/10.1038/s41576-019-0114-6)
- [3] P. M. Altrock, L. L. Liu, and F. Michor, *Nat Rev Cancer* **15**, 730 (2015). DOI: [10.1038/nrc4029](https://doi.org/10.1038/nrc4029)
- [4] L. Opasic, D. Zhou, B. Werner, D. Dingli, and A. Traulsen, *BMC Cancer* **19**, 403 (2019). DOI: [10.1186/s12885-019-5597-1](https://doi.org/10.1186/s12885-019-5597-1)
- [5] M. J. Williams, B. Werner, C. P. Barnes, T. A. Graham, and A. Sottoriva, *Nature Genetics* **48**, 238 (2016). DOI: [10.1038/ng.3489](https://doi.org/10.1038/ng.3489)

Quickstart example for cancer simulation

7.1 Import modules

```
[1]: ### Uncomment and execute if the next cell fails with "module not found".` Then,
      ↪ reload the kernel.
      # %cd ../../../../..
      # %pip install -e .
      # %cd -
```

```
/opt/data/repos/cancer_software
Obtaining file:///opt/data/repos/cancer_software
Installing collected packages: CaSim
  Found existing installation: CaSim 1.0.0
    Uninstalling CaSim-1.0.0:
      Successfully uninstalled CaSim-1.0.0
  Running setup.py develop for CaSim
Successfully installed CaSim
Note: you may need to restart the kernel to use updated packages.
/opt/data/repos/cancer_software/docs/source/include/notebooks
```

```
[2]: # The cancer simulation module.
      from casim import casim
```

```
[3]: # 3rd party modules.
      import os, shutil
      from wand.image import Image as WImage
```

7.2 Setup parameters

```
[3]: parameters=casim.CancerSimulatorParameters(
      matrix_size=20,
```

(continues on next page)

(continued from previous page)

```
number_of_generations=10,  
division_probability=0.5,  
advantageous_division_probability=0.8,  
death_probability=0.1,  
advantageous_death_probability=0.4,  
mutation_probability=0.2,  
advantageous_mutation_probability=0.8,  
mutations_per_division=10,  
time_of_advantageous_mutation=30000,  
number_of_clonal=2,  
plot_tumour_growth=True,  
export_tumour=True  
)
```

7.3 Setup the simulation engine.

```
[4]: if os.path.isdir('out'):  
      shutil.rmtree('out')
```

```
[5]: cancer_sim = casim.CancerSimulator(parameters, seed=1, outdir='out/')
```

7.4 Run the simulation

```
[6]: cancer_sim.run()
```

```
[6]: 0
```

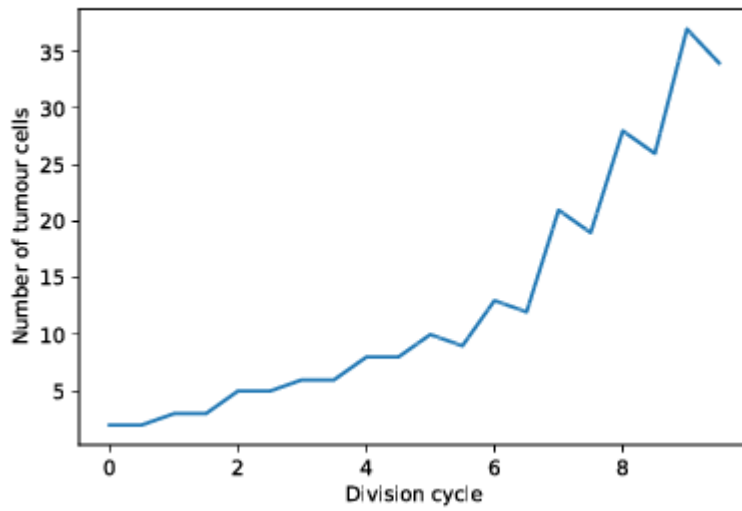
```
<Figure size 432x288 with 0 Axes>
```

7.5 Display the growth curve.

```
[7]: image_path=os.path.join(cancer_sim.outdir, 'cancer_1', 'simOutput', 'growthCurve.pdf')
```

```
[8]: img = WImage(filename=image_path)
```

```
[9]: display(img)
```



[]:

CHAPTER 8

Reference Manual

CHAPTER 9

Indices and tables

- `genindex`
- `modindex`
- `search`