

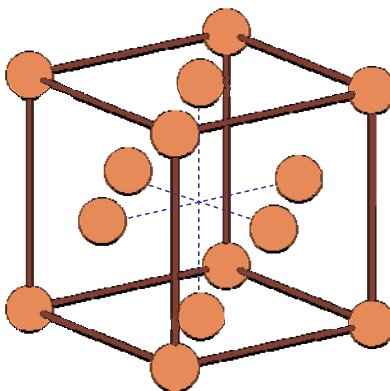
## *In silico* evolutionary dynamics of tumour virotherapy

Carlos L. Reis, Jorge M. Pacheco, Matthew K. Ennis and David Dingli

### ELECTRONIC SUPPLEMENTARY MATERIAL

#### Model implementation

We discretize real 3D space into a regular mesh, a Face Centred Cubic (FCC) lattice. The FCC lattice is a close-packing structure where each site has 12 nearest neighbours.



**Figure 1.** The *Face Centred Cubic* (FCC) lattice represented in the conventional cubic cell.

On such a lattice, we keep track of the location of cancer cells (**C**) and virus infected cancer cells (**V**), assuming all remaining sites are occupied by normal tissue. The model is implemented as a cellular automaton evolving in time according to the following rules which are repeated for as many generations as desired:

- Start with **C**<sub>0</sub> and **V**<sub>0</sub> cells in a given initial spatial structure (see section below).
- In each generation do **C** + **V** times:
- Choose an occupied site at random; the site may be occupied by a **C**-cell or a **V**-cell

- **C**-cell-case:

With probability  $r$  the cell will replicate.

With probability  $\beta$  this cell will become infected, i.e. it will become a **V**-cell, to the extent that, at least one of its nearest-neighbours is a **V**-cell.

With probability  $d$  the cell dies.

- **V**-cell-case:

With probability  $\delta$  the cell dies. **V**-cells do not replicate [1, 2].

We have enforced a limit to the maximum number of **C**-cells, i.e. a carrying capacity  $K$ , which is used to adjust the  $r$  and  $d$  probabilities:

$$r = r_i(1 - C / K)$$

$$d = d_i(1 - C / K)$$

where  $r_i$  is the initial **C**-cell replication probability and  $d_i$  is the initial **V**-cell death probability. This results in a logistic growth in time  $t$  governed by a Verhulst equation:

$$\frac{dC}{dt} = \bar{r}C(1 - C / K)$$

with solution

$$C(t) = KC_0 e^{\bar{r}t} [K + C_0(e^{\bar{r}t} - 1)]^{-1}$$

where  $\bar{r} = r_i - d_i$ .  $C(t)$  satisfies

$$C(t) \xrightarrow{t \rightarrow \infty} K$$

The parameters  $r_i$  and  $d_i$  were set to 0.12 and 0.02, respectively, for a maximum carrying capacity  $K = 10000$ ;  $\beta$  was set to 0.5. These values are chosen as representatives of a typical evolution in a structureless model of tumour development. When tumour structure is explicitly included, one can observe the impact of structure compared to the reference model. However, we also found out that changing these parameters does not change qualitatively the overall dynamics of tumour development and viral infection, and consequently we adopted them as representative after exploring the parameter space. It is generally accepted that tumour growth follows a sigmoidal curve and at some point reaches a carrying capacity. The size of the carrying capacity depends on the host. We chose a carrying capacity of 10,000 for computational convenience. While it is possible to have ‘larger’ tumours, this will not alter the outcome of our simulations while increasing computational demands. The absolute value of the carrying capacity used here is such that the qualitative behavior does not depend on the specific value adopted. Our choice is a compromise between accurate results and memory demands of the computer simulations. With respect to the chosen values of  $r_i$  and  $d_i$ , the result is a tumor with a doubling time of approximately 7 days which is similar to what is observed in relevant animal models of tumor growth. The value of  $\beta$  is an estimate, since to our knowledge there is no experimental determination of such a parameter.

### **Initial tumour configuration**

The initial spatial distribution of cells can be arbitrary. We studied two main cases:

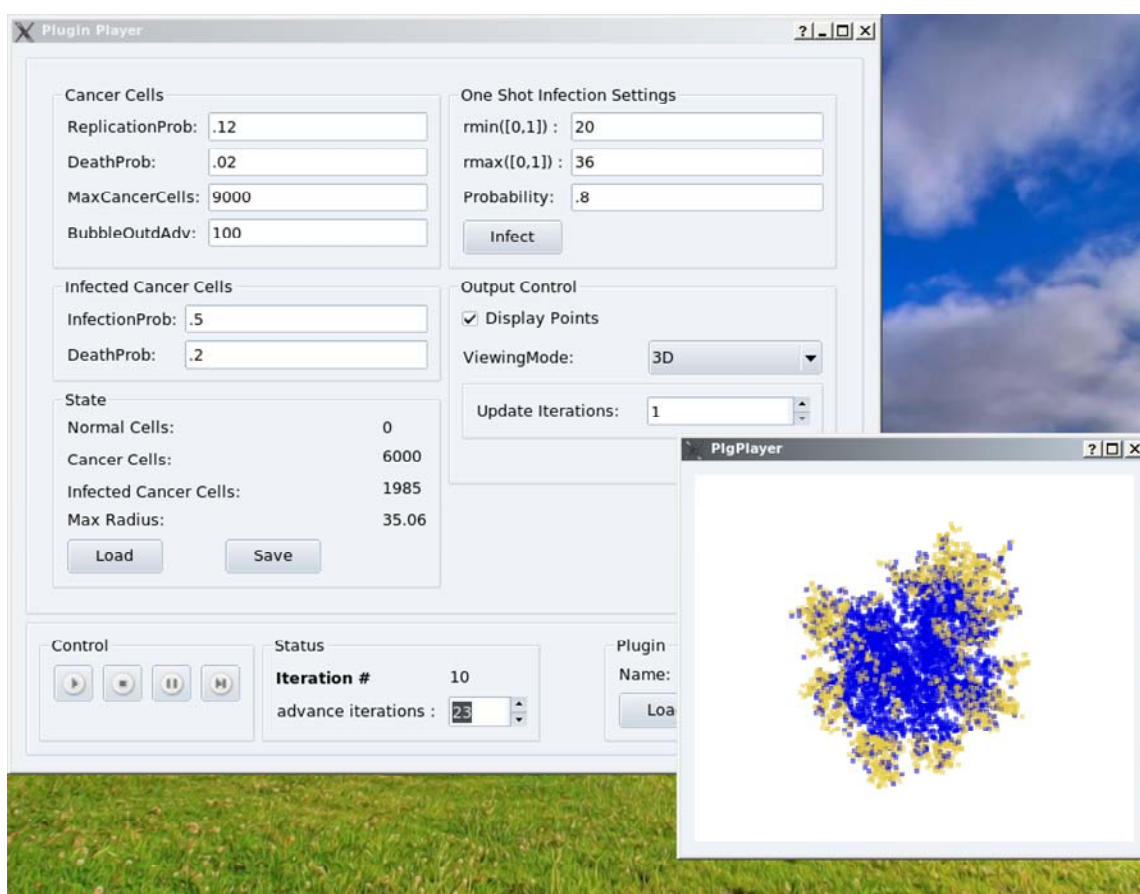
- 1) *Homogeneous*: An initial single tumour cell and subsequent offspring are

- allowed to replicate up to a pre-defined fraction of the carrying capacity. In its compact form, whenever a tumour cell reproduces, it creates a replica of itself which will fill a site if it is empty or will ‘push’ a cell away, hence generating pressure within the tumour. By imposing that the previously resident cell displaces itself, this will start a random, domino-like effect, in which cells “push” other cells away until one reaches normal tissue — that is, some external edge of the tumour — where this cell dynamics stops
- 2) *Fractal like*: In this case we use an initial cell distribution generated by a Diffusion-Limited-Aggregation (DLA) algorithm [3]. In DLA one starts from a single tumour cell fixed in space. This fixed tumour cell has the property of binding to it any moving tumour cell that occupies a neighboring site. Motion of tumour cells is random in space. Once a tumour cell becomes fixed it can also bind other tumour cells to it. Hence, tumour growth is associated with the pool of cells which get fixed. This aggregation dynamics is repeated until a pre-defined fraction of the carrying capacity is reached.

The death of a tumour cell creates an empty space which will eventually be occupied by another tumour cell ( $C$ ,  $V$ ) or a normal tissue cell. This empty space or “bubble” creates a pressure imbalance within the tumour; consequently, it will move randomly in space, displacing other cells, until it reaches normal tissue.

Several initial tumour sizes with a variable percentage of infected cells were generated. We also considered two initial infection schemes: Infection that starts close to surface of the tumour (resembling intravenous delivery of the virus) and another that starts near the centre (closer to intratumoral injection of the virus).

We implemented the algorithm in C++ taking advantage of Object-Oriented-Programming. The model algorithm was implemented separately and compiled as a dynamic link library. Using this scheme we can use the same code as a standalone executable or combined with a Graphic User Interface (GUI, see Figure 2). The GUI was implemented using the well known QT graphics library [4], with 3D real-time rendering of cell distribution in OpenGL [5].



**Figure 2.** Screenshot of the model simulation using a Graphical User Interface.

This numerical setup provides the ideal framework for model development. When coupled to the GUI interface, one can easily visualize tumour-virus dynamics and get a feeling of the relevant parameters. When the *exact same code* is executed as a

standalone code, simulations can be performed remotely in massively parallel computers, which allow an efficient and more exhaustive exploration of the model parameter space, as well as the production of results with good statistics.

## References

- [1] Z. Bajzer *et al.*, Journal of theoretical biology **252**, 109 (2008).
- [2] D. Dingli *et al.*, Cancer gene therapy **In Press** (2009).
- [3] T. A. J. Witten, and L. M. Sander, Phys Rev Lett **47**, 1400 (1984).
- [4] <http://qt.nokia.com/>
- [5] <http://www.opengl.org/>