

A Parameter Adjustment Tool for CompuCell3D

Xuefeng Gao, Sabin Tabirca

Abstract

The aim of this paper is to present a graphic user interface (GUI) tool for parameter adjustment in CompuCell3D, an open source multi-cell modeling framework and software packages. We provide an overview of this software and its underlying *Glazier-Graner-Hogeweg* (GGH) algorithm. To enhance the parameter adjustment approach in CompuCell3D simulations, we develop an easy-to-use tool - ParaAdjuster - using Python and PyQt. We briefly present a case study of using ParaAdjuster combined with CompuCell3D to simulate a solid tumor development.

1 Introduction

Mathematical and computational models have become more accepted by the biological community both as a means to motivate experimentation but also as a route to integrate multiple experimental measurements to generate testable predictions. CompuCell3D is able to facilitate biological simulations by defining spatially-extended *generalized cells*, which can represent clusters of cells, single cells, sub-compartments of single cells or small subdomains of noncellular materials [1]. A significant advantage of this flexible definition is to allow tuning of the level of detail in a simulation from intracellular to tissue without switching framework to examine the effect of changing the level of detail on a macroscopic outcome.

In this paper, we present an overview of the CompuCell3D modeling framework and explore the use of a parameter adjustment tool, ParaAdjuster, for the purpose of modifying parameters of CompuCell3D models. Both CompuCell3D and ParaAdjuster use Python (www.python.org), which is a popular scripting language for developing interactive scientific applications [3], and PyQt toolkit (wiki.python.org/moin/PyQt) in the user interface design.

2 CompuCell3D

CompuCell3D (www.compuCell3d.org) is an open-source, multi-cell, multi-scale modeling framework for developing biomedical simulations [1]. CompuCell3D is widely used in modeling a broad range of phenomena in biology and physics, e.g. biological cells, foams and ferromagnetism. It implements *Glazier-Graner-Hogeweg* (GGH) model (a.k.a. Cellular Potts Model), and operates on a regular lattice in two-dimensional or three-dimensional. GGH follows an *energy-minimization* philosophy and has *generalized cells* as its most primitive objects. Users can define the properties and behaviors of a *generalized cell* type to mimic a real cell or cell-like object in the real world, e.g. a cancer cell. A typical CompuCell3D simulation includes a list of objects, a description of their interactions and dynamics, and appropriate initial conditions. Cells evolving in time by interacting with each other as well as the surrounding environment, which tends to minimize the total *effective energy*. Additional fields can also be defined that may affect the dynamics of the *generalized cells*. These fields can represent chemical diffusants and

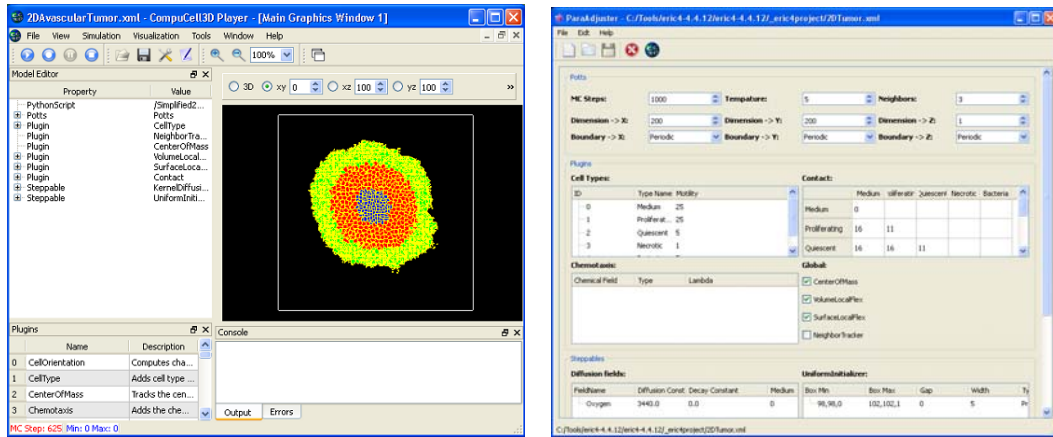


Figure 1: (a) An interactive CompuCell3D application shows a solid tumor growth model. (b) The ParaAdjuster GUI displaying the tumor model parameters using different widgets according to specific data type.

non-diffusing extracellular matrix.

CompuCell3D can be invoked either as an interactive application displaying the evolution of cells and the underlying fields in a GUI (see Figure 1(a)), or running a batch program that outputs raw data that can be post-processed. A CompuCell3D model is created textually using either an eXtensible Markup Language (XML) or Python script, in which a user can define the cell types, contact energy, fields and the global environmental parameters. The system dynamics can only be defined in a Python file, wherein a user can define the mechanisms controlling the cell behaviors, cell-cell and cell-environment interactions. To explore the effects of changing parameters in a model a user must modify the XML or Python files textually, which is inefficient and lack of error control. One may expect a tool to manage the parameters in a compact view with only the parameters themselves but all the tags in an XML file. It is this functionality that we wish to address in ParaAdjuster.

3 ParaAdjuster

ParaAdjuster is a light tool for CompuCell3D users to manage parameters in a graphical way (see Figure 1(b)). Similar tools like VisTrails, known as a scientific workflow and provenance management system [7], offers parameter exploration and data management for CompuCell3D. VisTrails sweeps parameters in a CompuCell3D model through the input and output ports of a module that executes the related CompuCell3D simulation. Compared to VisTrails, ParaAdjuster offers a direct modification feature with less additional effort. The workflow of using ParaAdjuster for an existed ComCell3D project is as follows:

1. Load a XML file that contains CompuCell3D definitions.
2. ParaAdjuster parses the loaded file and displays the data on a GUI using different widgets (e.g. spinner, text box, tree, etc.) according to the data types (e.g. integer, string, double, etc.).
3. Modify parameters in the widget and lunch a CompuCell3D simulation.

ParaAdjuster is designed to read and parse the CompuCell3D XML configuration file, and allow users to modify the basic simulation structure in a GUI. In an CompuCell3D XML file, each tag contains some descriptions of the model objects (e.g. CellType plugin uses the parameter syntax `<CellType TypeName="Medium" TypeId="0" />` map verbose *generalized-cell* type names to numeric cell TypeIds for all *generalized cell* types) or environment configurations (e.g. `<Boundary_x>`, `<Boundary_y>` and `<Boundary_z>` tag pairs define the boundary conditions in the system). Whenever users need to modify some parameters in their simulations, they have to locate the target parameters in the XML and then

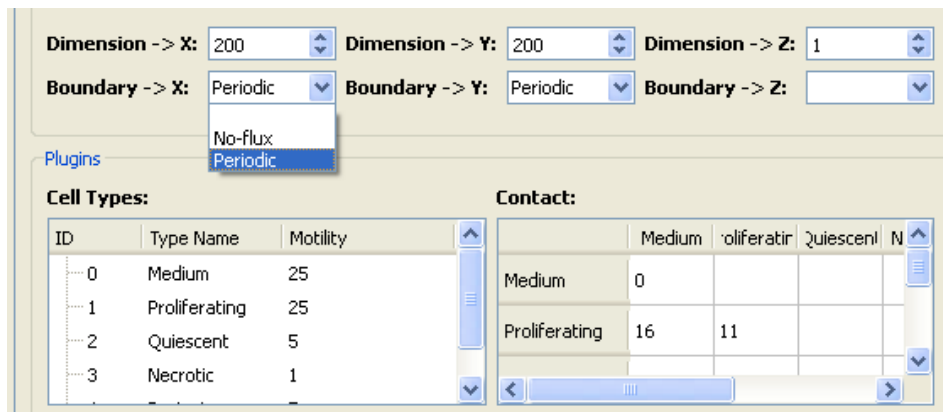


Figure 2: Widgets show CellType and Boundary Condition plugins in ParaAdjuster.

change it textually. This can bring some potential issues into the simulation including misspelling, variable type and value range errors etc.. In ParaAdjuster, all the parameters are presented using GUI widgets (see Figure 2), which are bound with a set of constraints to protect against some potential input errors. Take the TypeId in CellType plugin for instance, all CellType data are contained in a tree widget, in which each node element stands for a cell type showing a cell type ID, Name and Motility. A user can assign a unique TypeId for each cell type in the positive integer format that ranges from 0 - 1000. Therefore a repeated number, negative number, or non-integer data type is not allowed to input in the TypeId widget. The boundary condition in each direction is contained in a drop-down list widget, where users can choose either *no-flux* or *periodic*.

Users can also create a new CompuCell3D definition file based on a default template offered by ParaAdjuster, and further modify the objects, fields and related parameters to make a specific model. Starting a CompuCell3D simulation of the current loaded model in ParaAdjuster is as simple as clicking on the 'CC3D' button. We present a case study in next section, in which we use ParaAdjuster combined with CompuCell3D to simulate nutrient limited avascular tumor growth.

4 Case study: Avascular Tumor Growth

Our primary goal in using ParaAdjuster here is to perform parameter control for CompuCell3D models. We present a simplified version of a tumor growth model based on our previous work [5], [4]. This model contains three cancer cell types: proliferating, quiescent, and necrotic. Each cell on the grid behaves according to its genotype and the microenvironmental factors, which is oxygen in this case. Proliferating cells consume oxygen to support their activities, e.g. migration and mitosis, and become quiescent when experiencing hypoxia (oxygen deficiency region, wherein the oxygen concentration is below some threshold). A quiescent cell consumes less oxygen (1/5 of the base consumption rate), has low motility, and does not perform mitosis due to the cell-cycle arrest in G_0 phase [2]. Both proliferative and quiescent cells will turn into necrosis when the oxygen concentration further drops below another threshold value.

Each simulation is started with a single proliferating cell at the center of the lattice, and the initial oxygen concentration is homogeneous across the tissue. The formation of a layered-structure tumor can be observed with a central necrotic region that is surrounded by a layer of quiescent cells and an outer rim of proliferating cells. The oxygen field is described as a partial differential equation (PDE) in CompuCell3D XML file, and the related parameters including diffusion constant, consumption rate and decay rate per cell type, are parsed in ParaAdjuster and displayed in a tree widget, in which each tree element has a constraint relating to its data contained. Take oxygen constant for instance, its value is confined between 0 - 10000 in double data type; integer input will be converted into the double type; and character input is not allowed.

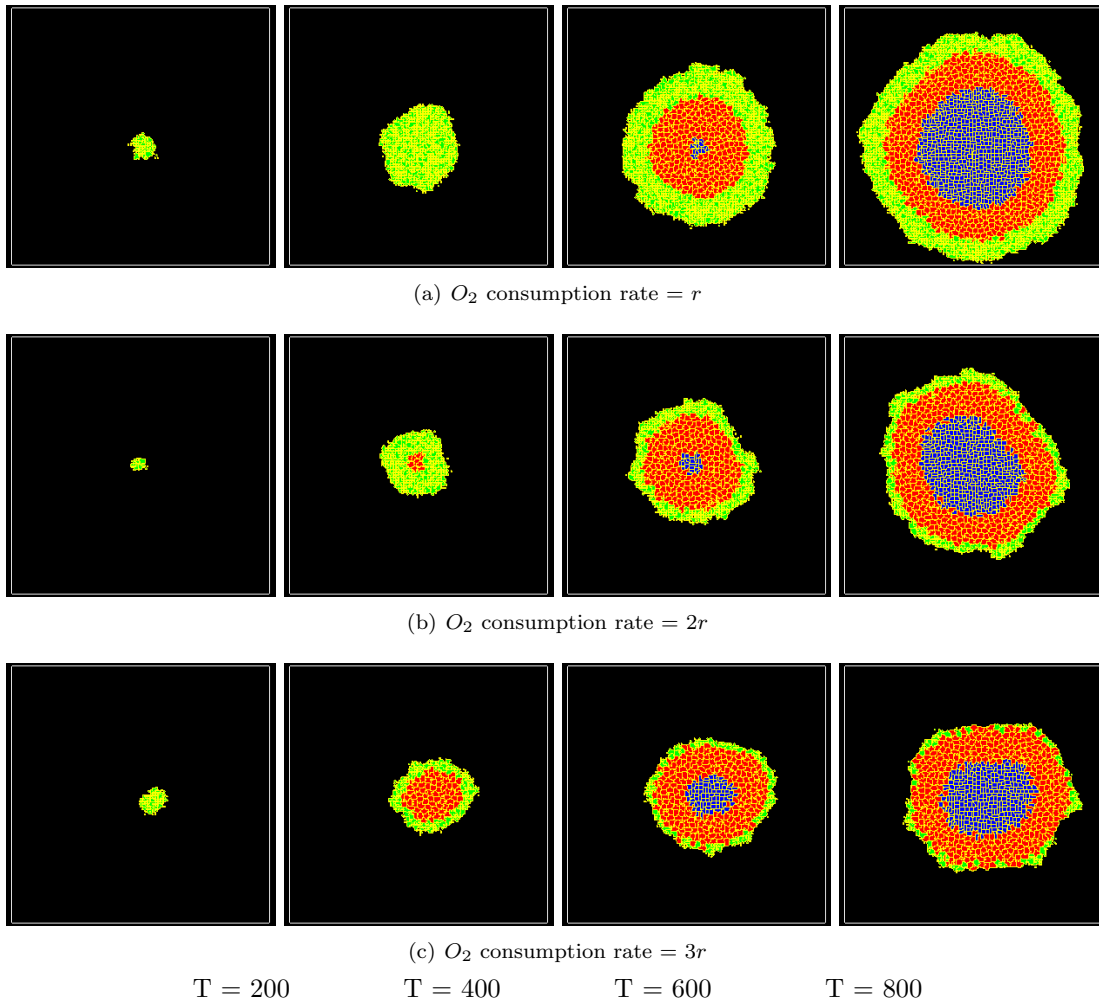


Figure 3: Exported images at regular intervals for development of a solid tumor. Proliferating cells are colored green, quiescent red, and necrotic blue. With a normal oxygen consumption rate, the tumor grows rapidly with larger final size, while with higher oxygen consumption rate the tumor forms a quiescent layer and a necrotic core earlier and appears a decrease in the final size.

Simulation results are captured at regular time intervals, 200 *Monte Carlo Steps*. Figure 3(a) shows an avascular tumor growing under a normal oxygen consumption rate. We then modified the oxygen consumption rate using ParaAdjuster, and the consequences of simulations are shown in Figure 3(b), and (c). Tumor growth is regulated through modulation of the consumption of oxygen, which appears to be a rate limiting substrate for tumor proliferation. From Figure 3 we can see that the increased oxygen consumption rate slows down tumor growth. This outcome is in agreement with the *in vivo* observations reported by Chen *et al.* [6]. Another important observation is that higher oxygen consumption rate can induce an earlier emergence of the quiescent and necrotic region. This decides the cell type distribution in a tumor, which will affect the tumor morphology in avascular phase and the following angiogenesis stage (data not shown).

It is important to note that the efficiency of metabolism has not been considered in this model, because there is no oxygen utilization contributing to individual cell's growth. Hence the future improvement of this model is to consider single cell growth and division speed by utilizing the constantly changing oxygen.

5 Conclusions and Future Work

ParaAdjuster is an easy-to-use tool, which offers graphically control parameters for CompuCell3D models with a compact view. Users can modify model parameters by using GUI widgets rather than textually edit the XML file. This approach can help users avoid some potential mistakes, for example typing mistakes and data type errors. Clearly, it is more friendly to users who are non-professional programmers and/or not familiar with the markup language like XML. In addition, users can launch a CompuCell3D simulation from ParaAdjuster, which can avoid some operation steps and directly see the consequent results after the parameter modification.

ParaAdjuster has not yet supported Python file, which is an important format for system dynamics definition in a CompuCell3D model. The first step in our future work is to integrate Python format as another type of entity in ParaAdjuster. Furthermore, we would like to enhance ParaAdjuster by displaying the relationships of cell-cell and cell-microenvironment by using widgets, such as maps.

Acknowledgement: This work was funded by IRCSET-Embark (Irish Research Council for Science, Engineering and Technology).

References

- [1] M. H. Swat, S. D. Hester, R. W. Heiland, B. L. Zaitlen and J. A. Glazier, Multi-Cell Simulations of Development and Disease Using the CompuCell3D Simulation Environment. *Methods Mol Biol.*, 500, 361-428, 2009.
- [2] C. J. Sherr, Cancer Cell Cycles. *Science*, 274, 1672-1677, 1996.
- [3] T. E. Oliphant, Python for Scientific Computing. *Computing in Science and Engineering*, 9, 10-20, 2007.
- [4] X. Gao, C. K. Baban, M. Tangney and S. Tabirca, Computer simulation of hypoxia regulates avascular tumor growth through p27 expression. *Complex Medical Engineering (CME), 2011 IEEE/ICME International Conference*, 15-22, 2011.
- [5] X. Gao, S. Tabirca and M. Tangney, A Multiscale Model for Hypoxia-induced Avascular Tumor Growth. *2011 International Conference on Bioscience, Biochemistry and Bioinformatics*, 53-58, 2011.
- [6] Y. Chen, R. Cairns, I. Papandreou, A. Koong, N. Denko and C. Nicholas, Oxygen Consumption Can Regulate the Growth of Tumors, a New Perspective on the Warburg Effect. *PLoS ONE*, e7033, 2009.
- [7] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva and H. T. Vo, VisTrails: visualization meets data management. *Proceedings of the 2006 ACM SIGMOD*, 745-747, 2006.

Xuefeng Gao
University College Cork
Dept. of Computer Science
WGB, UCC, Western Rd., Cork
IRELAND
E-mail: xfg1@cs.ucc.ie

Sabin Tabirca
University College Cork
Dept. of Computer Science
WGB, UCC, Western Rd., Cork
IRELAND
E-mail: s.tabirca@cs.ucc.ie