Instructions to run the single cell CompuCell3D Project simulation

CompuCell3D basics and project simulation files and contents

CompuCell3D (**CC3D**) is an open source modeling framework, which executes under most PC, Mac and Linux operating systems. Downloads and installation instructions are available from http://www.compucell3d.org/. This site provides one-button installers for most PC and Mac configurations. It also provides source code and binaries for some flavors of Linux. The site also archives older executables of CC3D to allow control for changes in program behavior during revision. The simulations, published in Fortuna et al. 2020, (https://doi.org/10.1016/j.bpj.2020.04.024) were executed using CC3D version 3.6.2, but execute and reproduce the original results in CompuCell3D version 3.7.9, available at https://sourceforge.net/projects/cc3d/files/3.7.9/windows/.

CC3D executables consist of two tightly-coupled Graphical User Interfaces (**GUI**s), Twedit++ (a simulation editor, which allows users to modify simulation specifications) and Player (which executes simulation specifications and displays and stores simulation results). A CompuCell3D simulation specification consists of a hierarchical group of files and folders. Most simulation specifications combine scripts written in Python and various XMLs. Specifically, a CC3D simulation ***<project>*** contains at least four files: an XML **project manager**, called *<project>.cc3d*, a Python **control file**, called *<project>.py*, a Python **steppable file**, called *<project_steppables>*.py, and an XML **parameter scan file**, called *ParameterScanSpecs.xml*. CompuCell3D stores the project manager file in a folder that usually shares the project name *<project>*, and stores the three other files in a subfolder called Simulation (*<project>/Simulation)*. If you are new to CC3D simulations, please watch the presentation http://compucell3d.org/BinDoc/cc3d_binaries/Presentations/Introduction_To_CompuCell/CompuCell_intro_2014_Hamner.pdf.

After installing CC3D, unzip the file SF1_Code.zip in an empty folder. Then, go to CC3D root folder. To execute a simulation, launch Twedit++ by clicking in *twedit++.bat*. Select *Open CC3D Project* from the *CC3D Project* pulldown menu, locate, and open the project manager file (file **CellMig3D.cc3d**). Execute the simulation by selecting the CellMig3D.cc3d file in the left-hand subwindow and "right clicking" and selecting *Open in Player*. Alternatively, launch Player, select *Open Simulation* from the *File* pulldown menu and select the **CellMig3D.cc3d** file. In either case, the simulation should now execute. You can use the *Windows* pulldown menu *Tile* option to adjust the display layout to improve your visualization of the executing simulation.

In what follows, we describe briefly each of the four files:

The file *<project>.cc3d* tells CC3D which files Player should run and where to find the parameters for the simulation(s). Our project is called *CellMig3D*, so the name of the project manager is **CellMig3D.cc3d**. CellMig3D.cc3d contains the lines

```
<Simulation version="3.5.1">
  <PythonScript Type="PythonScript">Simulation/CellMig3D.py</PythonScript>
  <Resource Type="Python">Simulation/CellMig3D_Steppables.py</Resource>
  <ParameterScan Type="ParameterScan">Simulation/ParameterScanSpecs.xml</ParameterScan>
</Simulation>
```

The Python *<project>.py* control file specifies the simulation components and environment definitions: cell sizes and types, lattice size, interaction energies, calculation frequencies, simulation duration, chemical fields, *etc*…. It also calls the calculation subroutines (**CC3D plugins**): center-of-mass position, neighbor tracker, *etc*… In our project, the name of the control file is **CellMig3D.py**.

The steppable file, **CellMig3D_Steppables.py**, specifies any temporally dynamic components of the simulation structure and parameters, performs the simulation analysis and displays and stores data, at intervals specified by the control file. Our project has two output files: one contains information about cell compartments' center-of-mass displacements, and the other contains information about cell symmetry breaking. The simulation updates these output files at the frequency defined in the control file. The simulation stores output files in a subfolder of the *<project>* folder.

Finally, the XML **ParameterScanSpecs.xml file** contains the list of the parameters we will sweep in the simulation, with their ranges (values) of variation. All the values we have used in this work are listed in it. Of course, the simulations can be grouped in sets of parameters. The file provided in the supplementary material generates 10 replicas for each parameter set, where the parameter sets have three possible cell radii, four values of phiF and seven values of lambCHEM. So executing the file runs a set of simulations consisting of a total of $10 \times 3 \times 4 \times 7 = 830$ individual simulations. This file generates all of the individual simulations used in the paper:

```
  <ParameterScan version="3.7.0">
    <OutputDirectory>CellMig3D_ParameterScan</OutputDirectory>
    <ParameterList Resource="Simulation/CellMig3D.py">
      <Parameter CurrentIteration="0" Name="RANDOM_SEED" Type="PYTHON_GLOBAL"
ValueType="int">
        <Values>68721, 198463, 206497, 211561, 217236, 240803, 353789, 380866, 404317,
410770</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="deltaT" Type="PYTHON_GLOBAL" ValueType="int">
        <Values>50</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="cellRad" Type="PYTHON_GLOBAL" ValueType="float">
        <Values>10., 15., 20.</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="phiF" Type="PYTHON_GLOBAL" ValueType="float">
        <Values>0.05, 0.1, 0.2, 0.3</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="lambCHEM" Type="PYTHON_GLOBAL"
ValueType="float">
        <Values>-75., -100., -125., -150., -175., -200., -250</Values>
      </Parameter>
    </ParameterList>
  </ParameterScan>
```

The values in the list for the parameter RANDOM_SEED define the number of replicas for each parameter set (deltaT, cellRad, phiF, lambCHEM). Each time a simulation replica starts, the CurrentIteration variable for each parameter increments by 1. For example, the replica that runs with RANDOM_SEED = 198463, deltaT = 50, cellRad = 15.0, phiF = 0.1, and lambCHEM = -100.0, will be replica number 161 (the replica counter starts at "0"). These values will be also used to assemble the associated output filenames. Identifying cellRad as "R", phiF as "pF", lambCHEM as "lC", and deltaT as "dT", sample filenames are:

- *161_R15.0_pF0.1_lC-175.0_dT50_Displacement.dat*
  This file contains 13 columns: time (mcs), the center-of-mass coordinates of the cell's three compartments C, F, and N, and the center-of-mass coordinates of the entire cell.
- *161_R15.0_pF0.1_lC-175.0_dT50_SBAn.dat*,
  This file contains 8 columns: time (mcs), the distance between the center-of-mass of the F compartment and the center of mass of the combined C and N compartments, the z-coordinate of the N compartment, the area of the boundary between the C and F

compartments, the volume of the C compartment, the volume of the F compartment, the volume of the N compartment, and the volume of the entire cell. All lengths and areas are in units of lattice spacing to the appropriate power.

CC3D simulations have a natural time unit of a Monte Carlo Step (**mcs**). The calculations for mean-squared-displacement rescale both experiment and simulation times by the measured persistence times. The ratio of these two persistence times converts mcs into experimental time units.

To generate different simulations using the **ParameterScanSpecs.xml** file, you must change the numbers between the appropriate tag pairs of form **<Values> … </Values>**. For example, to run 5 replicas of a simulation with deltaT = 50, cellRad = 20.0, phiF = 0.2, and lambCHEM = -150.0, the modified **ParameterScanSpecs.xml** file reads:

```
  <ParameterScan version="3.7.0">
    <OutputDirectory>CellMig3D_ParameterScan</OutputDirectory>
    <ParameterList Resource="Simulation/CellMig3D.py">
          <Parameter CurrentIteration="0" Name="RANDOM_SEED" Type="PYTHON_GLOBAL" ValueType="int">
        <Values>68721, 198463, 206497, 211561, 217236</Values>
      </Parameter>
  <Parameter CurrentIteration="0" Name="deltaT" Type="PYTHON_GLOBAL" ValueType="int">
        <Values>50</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="cellRad" Type="PYTHON_GLOBAL" ValueType="float">
        <Values>20.</Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="phiF" Type="PYTHON_GLOBAL" ValueType="float">
        <Values> 0.2 </Values>
      </Parameter>
      <Parameter CurrentIteration="0" Name="lambCHEM" Type="PYTHON_GLOBAL" ValueType="float">
        <Values>-150. </Values>
      </Parameter>
    </ParameterList>
  </ParameterScan>
```

where **bold face** indicates lines changed from the version of the **ParameterScanSpecs.xml** provided.

How to modify and run the project simulation using CC3D

First, download the appropriate CC3D installer or binary package from http://compucell3d.org/ and install it. On Windows computers, we recommend installing to the "Desktop" rather than the "Programs" directory to avoid permission clashes. Download the compressed project file and unpack it to a folder in your workspace.

Launch the CC3D project editor/creator **Twedit++** using the method appropriate to your operating system. Click on *CC3d Project* and then on *Open CC3D Project*. Go to the folder where you unpacked the project files and open the .cc3d file. This selected project will now show in **Twedit++**´s leftmost project structure panel, which displays the file hierarchy of open projects. Clicking on the project will display the project's component files and will open both the Python files described in the previous section, as tabs in Twedit++'s right editing panel.

If you want to run the simulation for a specific set of parameters, click on *ParameterScan* in Twedit++'s leftmost project structure panel to open the **ParameterScanSpecs.xml** file for editing.

ParameterScanSpecs.xml specifies all externally-controlled simulation parameters. Make any changes desired to the number of replicas, choices of "deltaT" or other swept parameters. **CellMig3D_Steppables.py** specifies all other simulation parameters, as given in Table 2 of the main text. These parameters are left the same in each simulation replica. You can change any of these values by clicking on *CellMig3D_Steppables* in Twedit++'s leftmost project structure panel to open the **CellMig3D_Steppables.py** file for editing.

Save all files using the *Save All* button or the *Save CC3D Project As* menu item in the *CC3D Project* pulldown menu, then right click on the project name in Twedit++'s leftmost project structure panel. Click on *Open in Player*. The CC3D player will open and start the series of simulations specified in **ParameterScanSpecs.xml**.

Simulation initial configuration

Our simulations use a 3D lattice with periodic boundary conditions, of size (L_x, L_y, L_z), defined in units of the of cell radius, R_{cell}. Initially, the number of lamellipodium sites is zero and the cell is spherical. The cell flattens on contact with the substrate and the lamellipodium forms and spreads rapidly due to the target-volume effective energy term. The lamellipodium target volume is proportional to $R_{cell}{}^{3}$and ϕ_l. Consequently, the horizontal dimensions of the cell-lattice must increase with R_{cell} and ϕ_l. The following cell-lattice dimensions sufficed to prevent the cell spanning any single cell-lattice dimension and causing an artifact due to the periodic boundary conditions:

$$L_z = 2.1\, R_{cell},$$

$$L_x = L_y = \begin{cases} 8\, R_{cell}, \text{if } R_{cell} < 20 \text{ lattice sites}^{1/3} \text{ or } \phi_l < 0.20 \\ 10\, R_{cell}, \text{if } 20 \text{ lattice sites}^{1/3} \leq R_{cell} < 30 \text{ lattice sites}^{1/3} \text{ and } \phi_l \geq 0.20 \\ 12\, R_{cell}\ , \text{if } 30 \text{ lattice sites}^{1/3} \leq R_{cell} < 40 \text{ lattice sites}^{1/3} \text{ and } \phi_l \geq 0.20 \\ 14\, R_{cell}\ , \text{if } 40 \text{ lattice sites}^{1/3} \leq R_{cell} \text{ and } \phi_l \geq 0.20 \end{cases} \quad (S1)$$

Inside the lattice, the lattice sites with coordinates at $(x, y, z = 0)$ are frozen and set to a generalized cell of type substrate. All other lattice sites not in a cell compartment are set to a generalized cell of type medium. Initially the cell is a sphere centered on coordinates $\left(\frac{L_x}{2}, \frac{L_y}{2}, \frac{L_z}{2}\right)$ consisting of two concentric compartments, a central sphere, of cell type nucleus and a surrounding spherical shell, of cell type cytoplasm. When lattice sites of type cytoplasm come into contact with lattice sites of generalized cell type substrate they create lattice sites of cell type lamellipodium, as illustrated in the main text.