A Spatial Tumor–Immune Model

Factors modulating malignant growth

Vedang Narain

30 April 2019

Summary

The morphology of a malignant tumor is influenced by the interactions between its cells and those of the immune system via the process of immunoediting. The administration of chemotherapeutic treatments — which are toxic to both cancerous cells as well as to the cells of the immune system — may explain the impaired immune responses exhibited by patients treated with cytotoxic compounds. By designing and implementing a spatial model of tumor growth in different immunoediting contexts, this project aims to computationally evaluate the hypothesis that chemotherapeutic intervention can disrupt the outcome of immunoediting. Simulation results indicated that the outcome of chemotherapy could depend upon the immunoediting context at the time of administration.

Introduction

Tumor Morphology

A malignant tumor is an abnormal mass of cancerous tissue that results from the uncontrolled proliferation of cells. It consists of a proliferating layer (high growth), a quiescent layer (little/no growth), and a necrotic (dying) core.

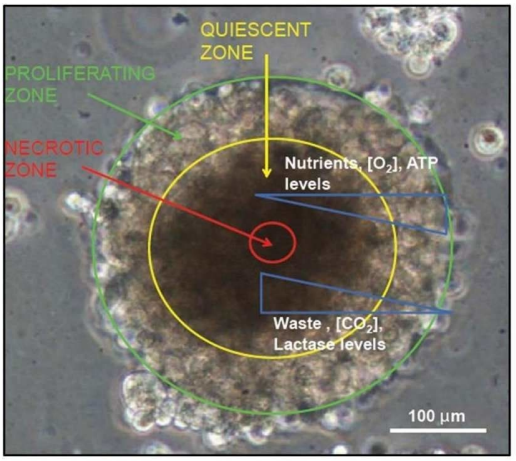

Figure 1. Concentric layers of a tumor spheroid form as a result of variations in nutrient availability [1]

Immunoediting

Immunoediting is the biological process by which the immune system alters the progression of tumor growth [2]. The process of immunoediting features three phases.

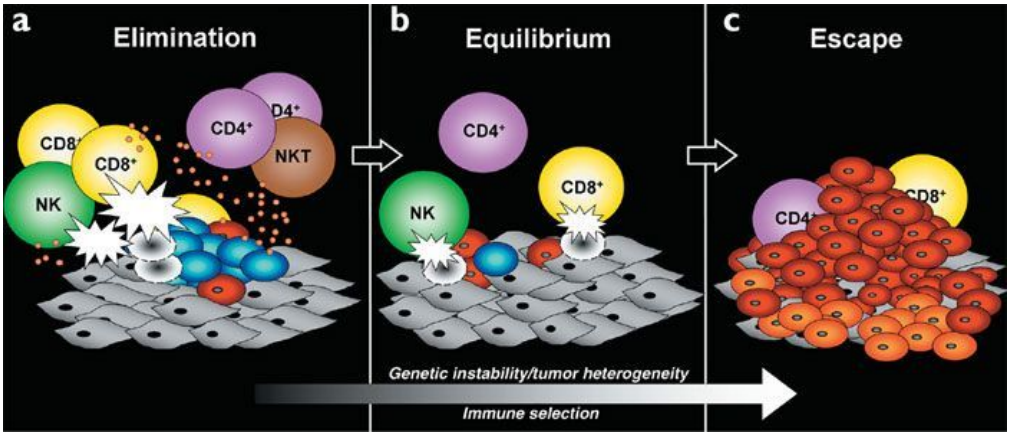

Figure 2. Developing tumor cells (blue), tumor cell variants (red and orange), and underlying stroma (grey) in the microenvironment [3]

In the first phase, ‘elimination’, tumor cells are destroyed by the immune system. The surviving population of cells consists of a mutated genetic makeup that resists the action of the immune cells. This population enters an ‘equilibrium’ phase. Due to the inability of the immune system to check the growth of these mutated cells, a final ‘escape’ phase features the circumvention of the immune response and the establishment of an immunosuppressive microenvironment [4].

Chemotherapy

Chemotherapy is the use of synthetic or naturally–occurring chemicals for the treatment of diseases. Chemotherapeutic medication targets rapidly dividing cells. While this helps eliminate cancerous cells, cytotoxic chemicals affect healthy proliferative cells as well (such as blood, bone marrow, and hair). Patients treated with chemotherapy have subsequently been noted to exhibit an impaired immune response [5].

Model Design

Overview

By constructing a model of the interactions between a three–layered tumor population and cytotoxic T lymphocytes (CTLs), and their subsequent disruption by chemotherapy, we should be able to observe chemotherapy–mediated transitions. The model can be constructed in three stages:

1. Three–layered Tumor: The central element of the model is the three–layered tumor that must arise independently as a result of variations in the concentrations of the nutrient fields. When unchecked by the immune system, the tumor should occupy all the space available to it.

2. Immunoediting Phases: The next desired observations are simulation outputs exhibiting characteristics of the three phases of immunoediting (reduction, oscillations/stability, uncontrolled growth) by varying the effectiveness of the immune reaction as a proxy for evolutionary resistance developed due to genetic instability.

3. Chemotherapeutic Disruption: Finally, by randomly choosing approximately 90% of the cells in the simulation to be killed and by downregulating the immune response, we can simulate a cytotoxic event. By varying the time of the event and the immunoediting context, we may be able to observe a deviation from the events exhibited in the unimpaired responses from the second stage.

Literature Survey

A cellular automata (CA) model published by de Pillis et al. served as the inspiration for the model in this study [6]. However, the published model had the inherent implication of cells making spontaneous appearances, immediate transitions between cell types, and the complete disappearance of tumor debris. Therefore, the premise was adapted to a Glazier-Graner-Hogeweg (GGH) model–based simulation to better represent the cellular dynamics involved [7, 8].

Methods

The model was implemented in the CompuCell3D environment (ver. 3.7.9) [9]. The CC3DML and Python scripts were customized to account for the model elements discussed in this section.

The central objects in the model are five cell types that represent the three nutrient–dependent layers of the tumor as well as the CTLs. An additional cell type that serves as the extracellular medium is included as well. Each cell exhibits the ability to change its position and regulate its volume. For the purpose of visualization, the cells are assigned a color as well. The 2D simulation environment functions as an object, with periodic boundaries. Three fields — two nutrients consumed by tumor cells and an attractant secreted by all tumor cells — diffuse through the environment.

Each cell exhibits the ability to adhere to other cells with varying magnitudes. All tumor cells release a secretion that attracts CTLs. Proliferating and quiescent tumor cells take up the fields representing nutrients (growth and survival). Cells switch between their proliferating and quiescent states depending on the amount of growth nutrient within, and turn necrotic if the survival nutrient falls below a certain threshold within the cell. Cells regulate their volume by growing and dividing (forming two quiescent cells) when in a proliferating state, staying constant when quiescent, and shrinking once necrotic. Additionally, CTL cells chemotax towards the source of the attractant. When in contact with a proliferating or quiescent tumor cell, the CTL kills the tumor cell with a certain probability.

Finally, at the boundary, growth and survival fields diffuse into the microenvironment. A summary of the model elements is provided in Appendix A.

CTL Behavior

CTLs are recruited by sampling the amount of attractant (A) at the borders of the microenvironment. This function is modulated by the strength of the immune response (I) chosen to simulate the effectiveness of the immune system depending on the stage of immunoediting being modeled. This results in an arrival rate $r_{arrival}$ which is later rounded to generate the number of CTLs arriving in each wave. In this project, constant a was chosen to have a value of 2.

$$r_{arrival} = \frac{I \times A}{a + A}$$

Once recruited to the microenvironment, CTL behavior is driven by two probabilities: the probability of dying and the probability of killing. The former determines the likelihood that a CTL will die after killing a tumor cell. This is accomplished by allowing each CTL to track the number of tumor cells it has already killed (K). In this simulation, constant b was given a value of 10.

$$P_{death} = \frac{K}{b + K}$$

The probability of the CTL killing a tumor cell in its neighborhood is correlated with the probability of its death.

$$P_{kill} = 1 - P_{death}$$

To simulate the reduced efficacy of CTL killing due to chemotherapeutic cytotoxins, P_{kill} is halved after the point of introduction.

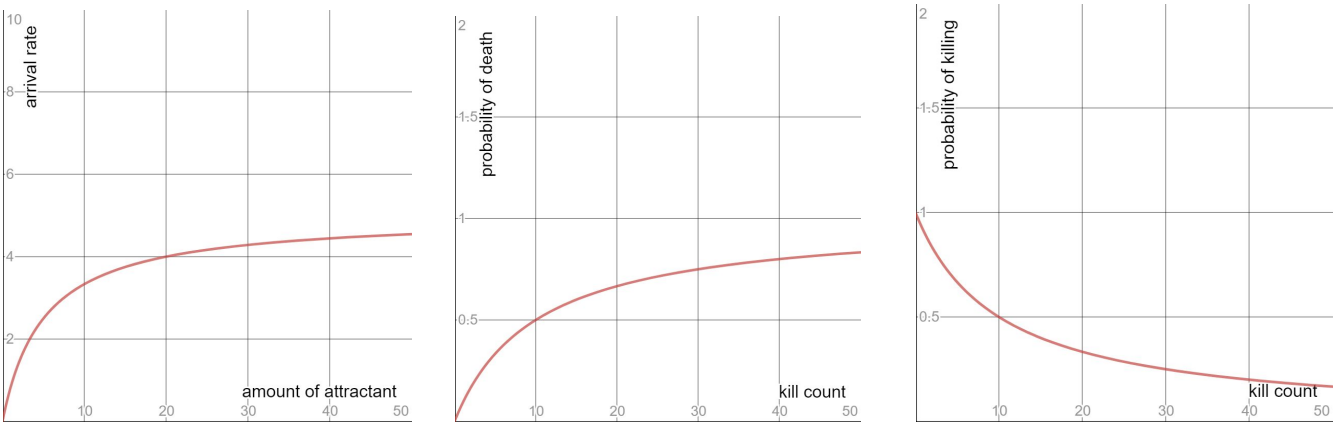

Figure 3. $r_{arrival}$, P_{death}, and P_{kill} saturate based on the input (I or K) and the chosen constants

Results

Simulations begin with a blob of proliferating cells positioned centrally. Time is simulated in units of Monte Carlo Steps. Due to the significant computing resources required, simulations were stopped once new behavior ceased.

Unchecked Tumor Growth

When the immune response is excluded from the simulation, the tumor consumes all the nutrients available to it and gradually fills up the entire simulation domain. The three layers develop due to the diffusion of the survival and growth nutrient fields. The tumor population eventually stabilizes at a high number once the boundary is reached.

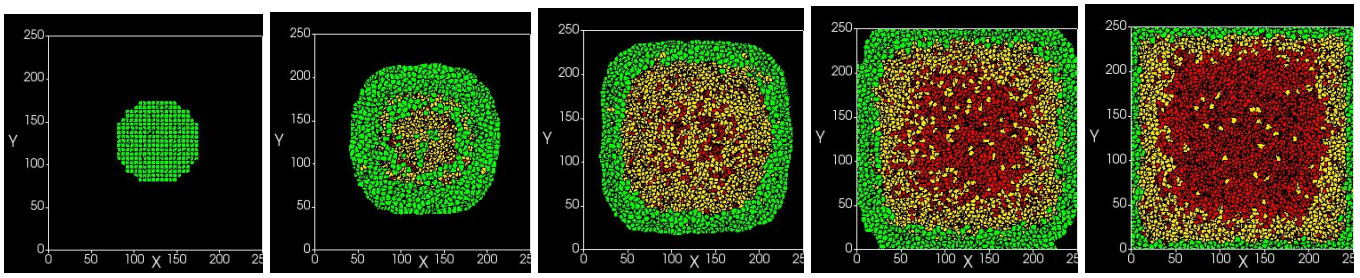

Figure 4. Cells at MCS = 0, 500, 1000, 1500, 2000

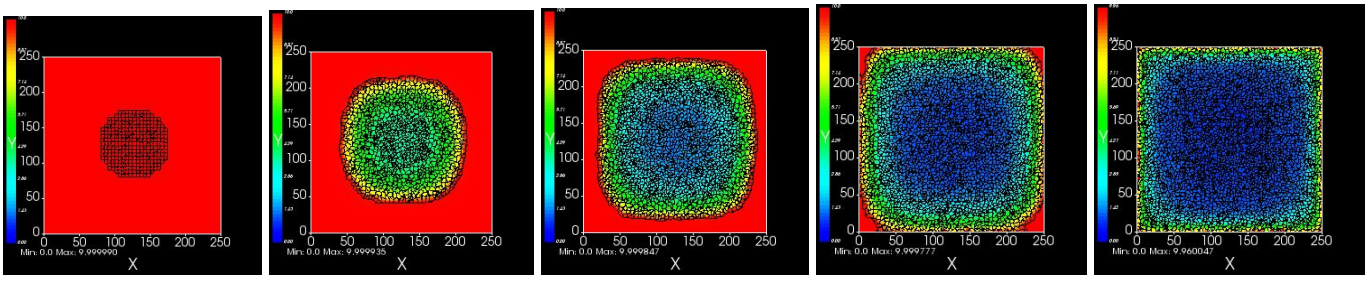

Figure 5. Growth Field at MCS = 0, 500, 1000, 1500, 2000

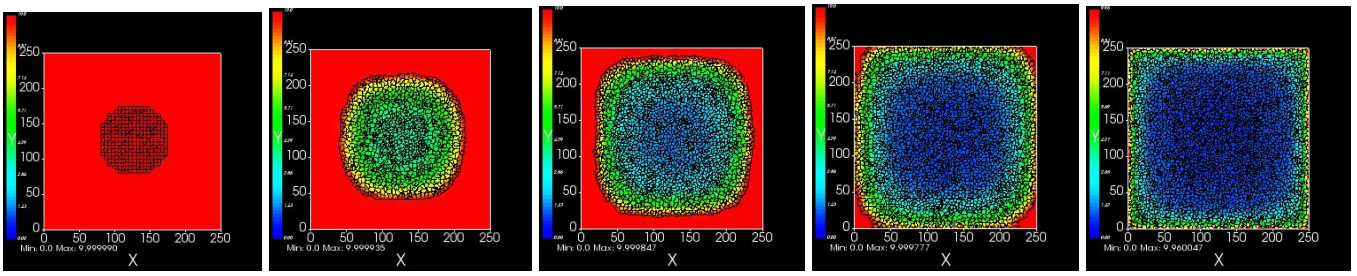

Figure 6. Survival Field at MCS = 0, 500, 1000, 1500, 2000

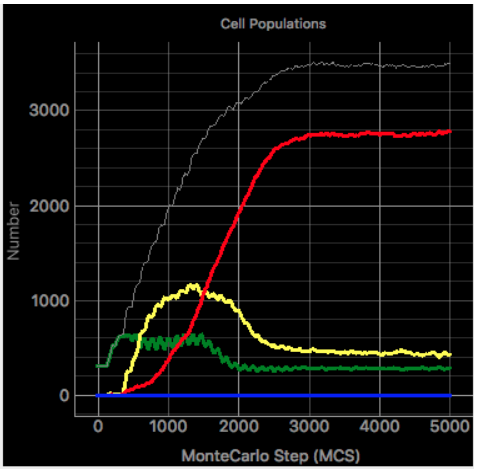

Figure 7. Proliferating cells (green) drive overall tumor growth (grey) but are later replaced by quiescent (yellow) and necrotic (red) cells

Immunoediting: Elimination

When the immune system is stimulated at a high magnitude ($I = 10$), the immune cells are attracted by the tumor cells at a high rate which results in complete tumor elimination before the tumor is allowed to stabilize. Destroying the proliferating and quiescent layers is sufficient to ensure complete elimination, since necrotic cells cannot switch states.

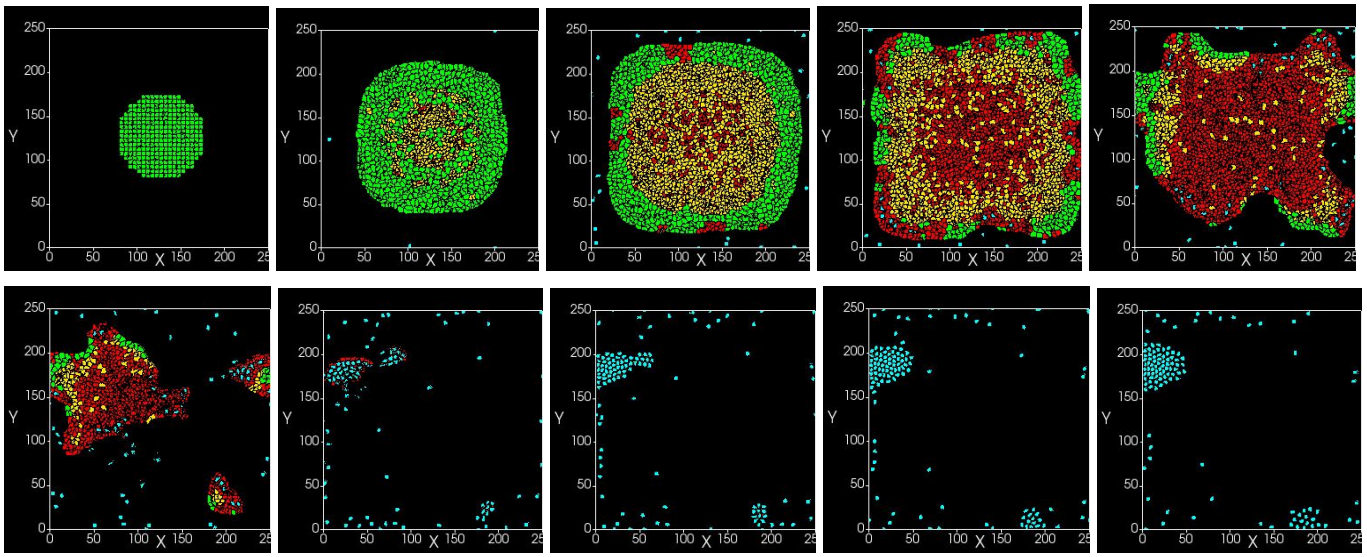

Figure 8. CTLs (blue) arrive at a rate sufficient to eliminate the proliferating and quiescent tumor layers, resulting in complete elimination (MCS = 0, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500)

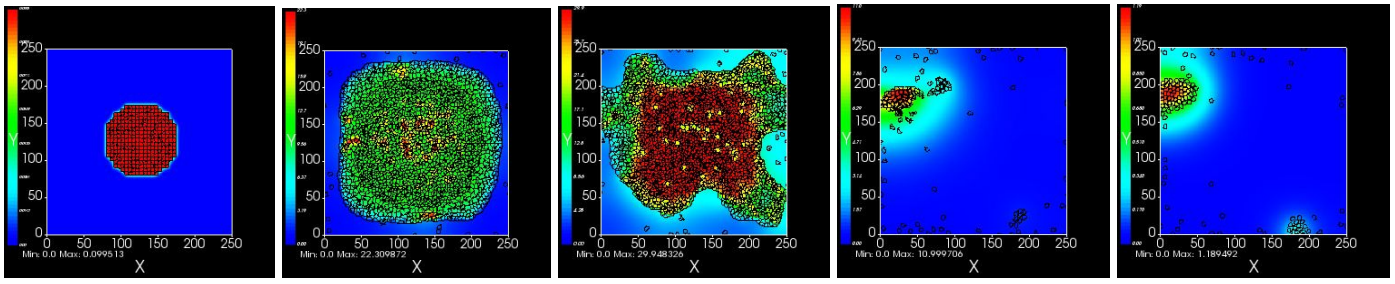

Figure 9. Attractant Field at MCS = 0, 1000, 2000, 3000, 4000

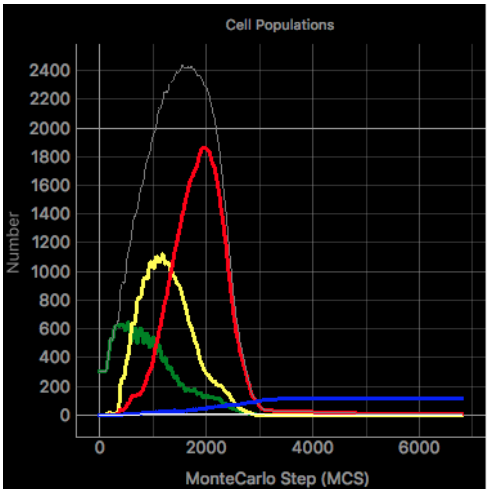

Figure 10. An increase in the CTL population (blue) drives tumor necrosis which eventually results in tumor elimination

Immunoediting: Equilibrium

When the immune system is stimulated at a mid–level magnitude ($I = 5$), the immune cells are attracted by the tumor cells at a rate which isn't high enough to eliminate the tumor, yet not low enough to allow tumor dominance. As the tumor grows in size, it attracts an increasing number of CTLs. However, when it reduces in size, the number of CTLs recruited begins to fall, allowing the proliferating cells to take advantage of the lower immune response until the cycle repeats. The surviving tumor populations tend to remain near the edges, i.e., the source of nutrients.

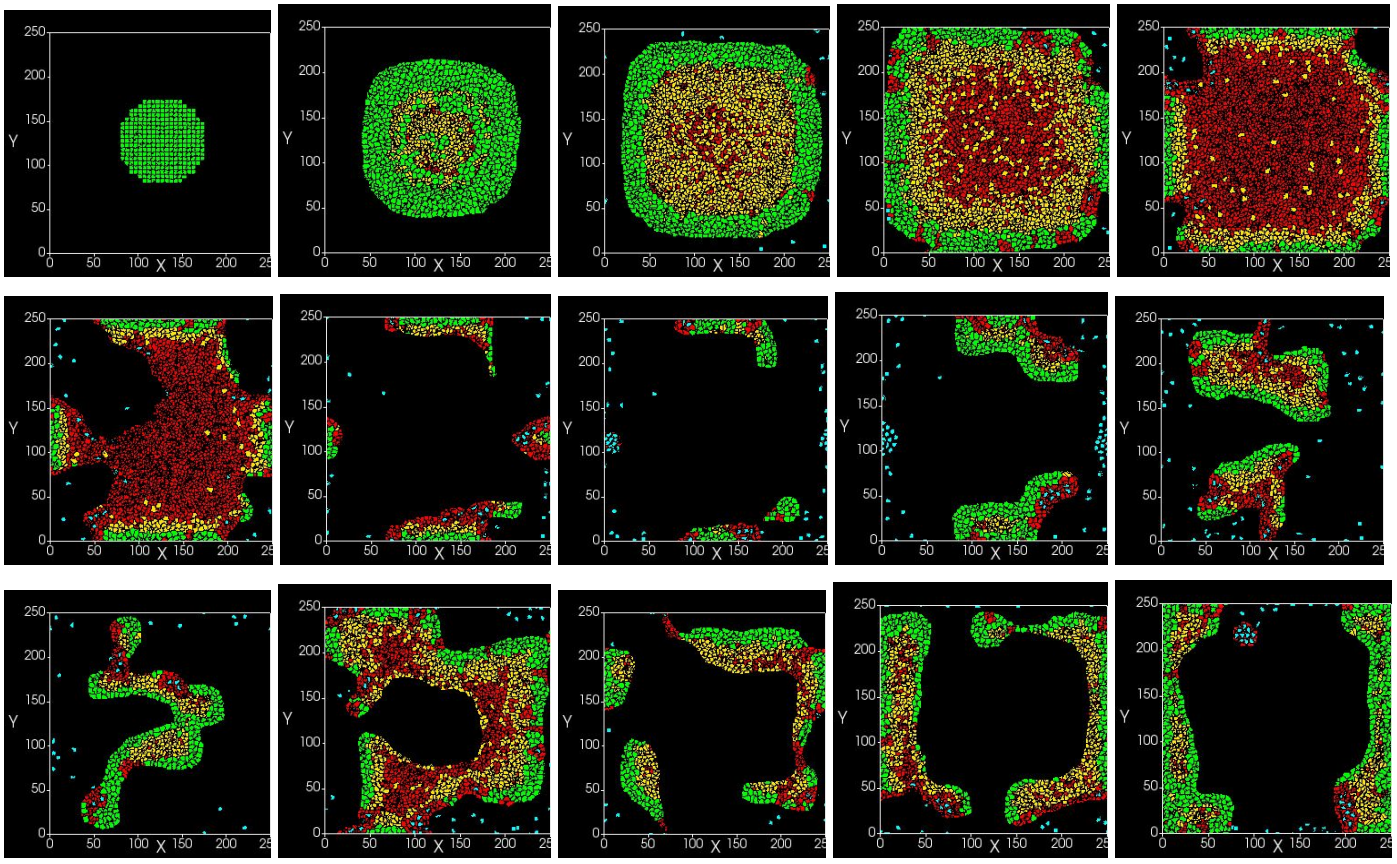

Figure 11. A sizable portion of the tumor is eliminated but secondary tumors prevent complete elimination (MCS = 0, 500, 1000, 1500, 2500, 3000, 3500, 4000, 5000, 7000, 8000, 10000, 11000, 13000, 15000)

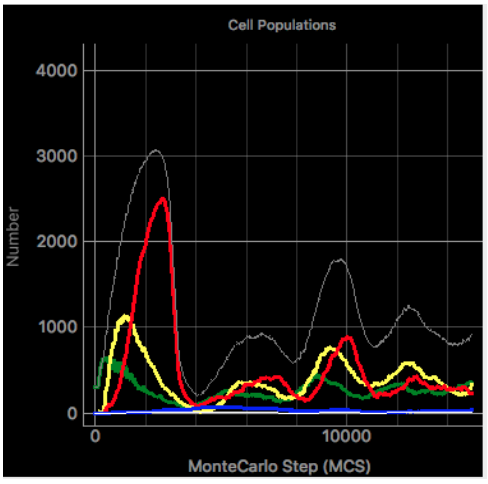

Figure 12. The equilibrium phase exhibits oscillations in cell populations as CTLs are recruited based on the size of the tumor

Immunoediting: Escape

When the immune system is stimulated at a low magnitude ($I = 2$), the immune cells are attracted by the tumor cells at a rate which isn't high enough to significantly affect tumor growth. Notably, the immune response continues even once the tumor reaches the boundaries but does not significantly affect the outcome.

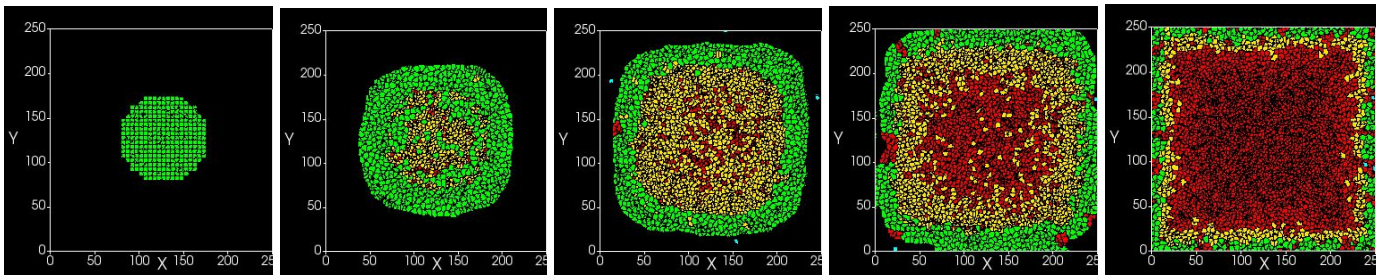

Figure 13. The stable tumor is attacked by CTL cells but receives enough nutrients at the boundary to remain dominant (MCS = 0, 500, 1000, 1500, 6500)

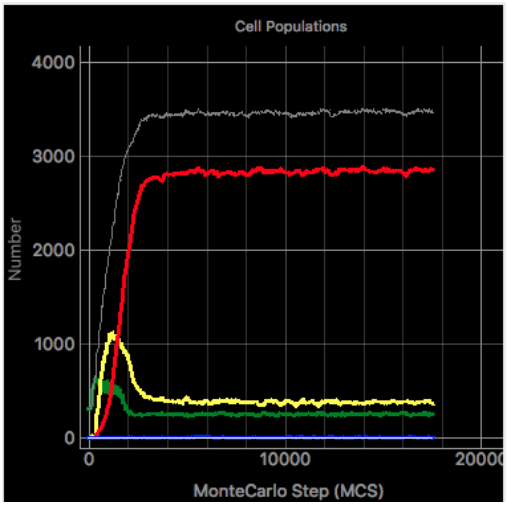

Figure 14. When compared to uninhibited tumor growth, there appear to be more perturbations in the stable tumor due to CTL action

Chemotherapeutic Intervention

In order to explore the effects of a cytotoxic event, we used models from the previous simulation. Since the outcome of each immunoediting phase was previously generated, any deviation due to chemotherapy could be isolated. The resultant simulations produced three cases of interest.

In the first case, equilibrium is disrupted when chemotherapy is administered at MCS = 3500. This nearly eliminates the tumor but also hampers the immune response, resulting in the reversion of the system to equilibrium.

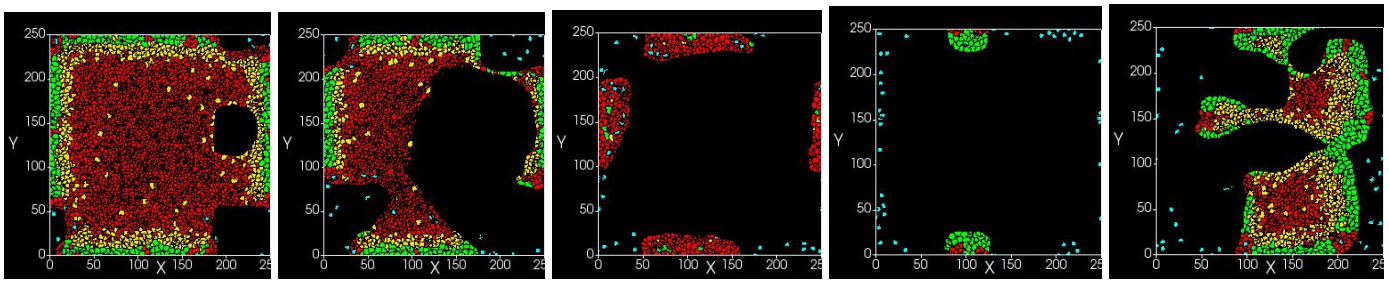

Figure 15. Equilibrium is disrupted without significant effect on the outcome
(MCS = 2000, 2500, 3500, 4500, 8000)

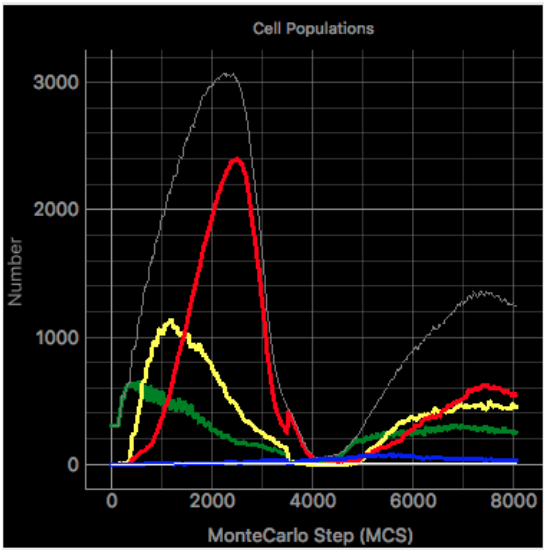

Figure 16. Chemotherapy is administered at MCS = 3500 (i.e., while the tumor is being brought under control by the immune system)

In the second case, equilibrium is disrupted at MCS = 2000. Since the chemotherapy was administered during a period of tumor stability, it appears to have assisted the immunoediting process transition from an otherwise equilibrating state to one in which the CTLs can destroy the remaining tumor cells.

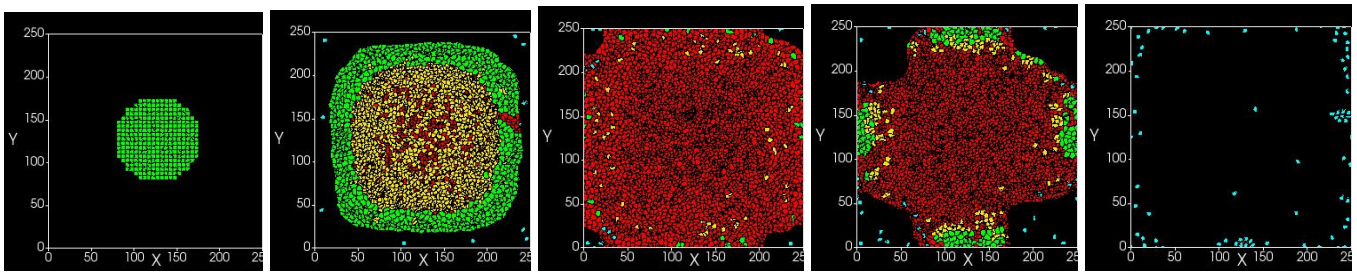

Figure 17. Equilibrium is disrupted, allowing the CTLs to eliminate the remaining tumor population
(MCS = 0, 1000, 2000, 2500, 4000)

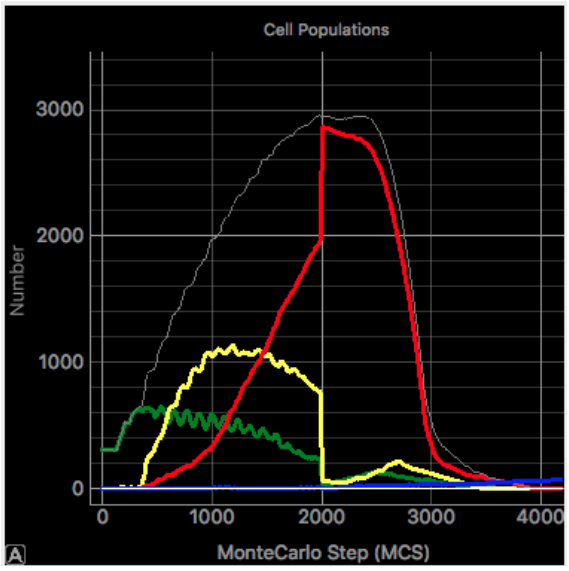

Figure 18. Chemotherapy is administered at MCS = 2000 (i.e., while the tumor is stable in size)

In the third and final case, chemotherapy is introduced at MCS = 2000 into an immunoediting context that would otherwise eliminate the tumor. In this case, rather than speeding up the elimination process, the cytotoxicity instead hampers the immune response by reducing the efficacy of the CTL cells resulting in a surviving population of tumor cells driving the simulation into an equilibration phase.

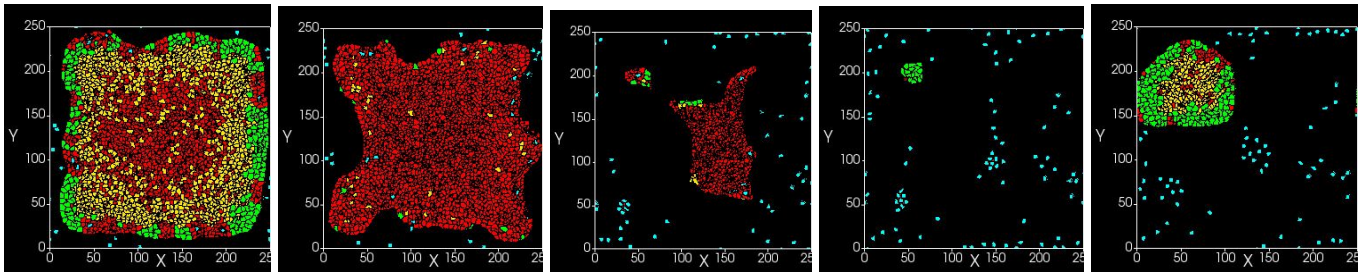

Figure 19. Elimination is disrupted, preventing the CTLs from completely eliminating the tumor (MCS =1500, 2000, 2500, 3000, 4500)

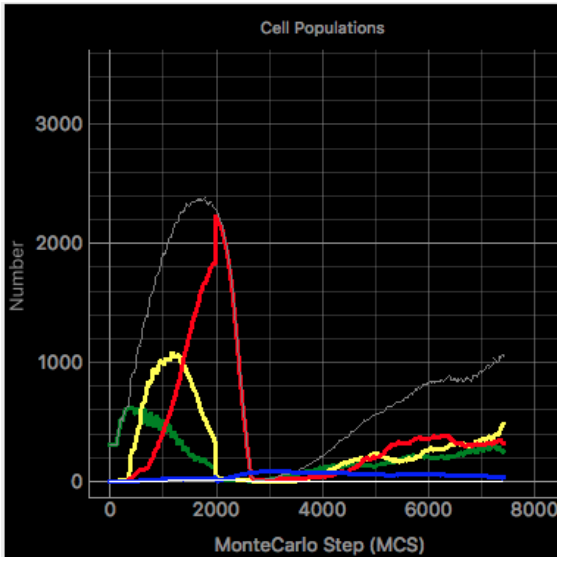

Figure 20. Chemotherapy is administered at MCS = 2000 (i.e., while the tumor is decreasing in size)

Discussion

The results of the computational simulations support the hypothesis that chemotherapeutic intervention can disrupt the outcome of immunoediting. Simulation results indicated that the conceptual model exhibits unrestrained tumor growth, characteristic immunoediting phases, and both favorable and unfavorable disruptions due to chemotherapy.

Oscillatory dynamics play important roles in gene expression, neuronal communication, and signaling in biotic communities [10]. Therefore, it is interesting to note the recurrence of oscillations across tumor–immune models [2].

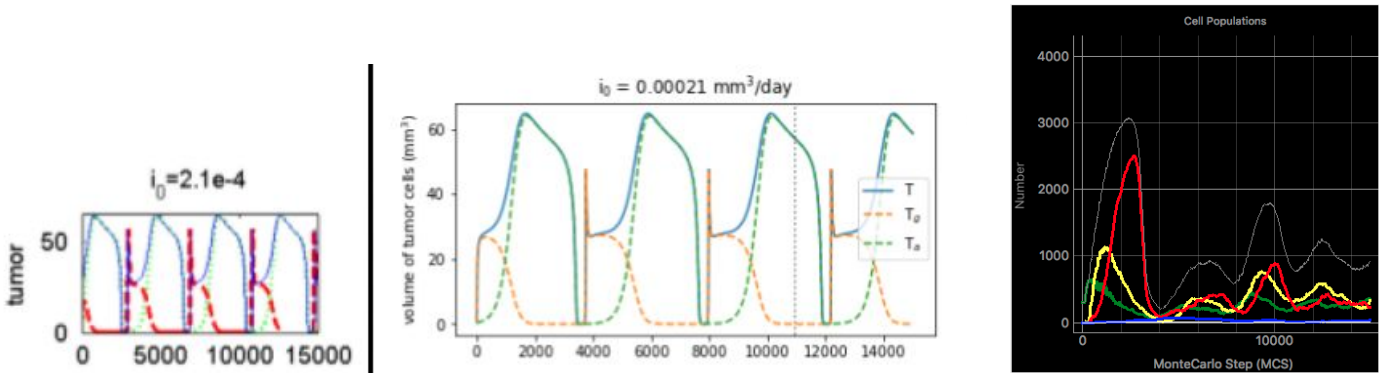

Figure 21. Oscillations persist across deterministic and stochastic models of tumor–immune equilibrium [2]

If clinically validated, oscillations in tumor size could have significant implications for the decisions involved when treating tumors in a clinical setting. For example, if an impending trough (low tumor size) in an oscillation caused by immunoediting is representative of a decade, it may be more prudent to avoid chemotherapeutic intervention in elderly patients. Additionally, determining the phase of the immunoediting process that will be disrupted by administering a cytotoxic compound may prove to be useful when determining the dose strength necessary to eliminate the tumor rather than assisting its development.

Naturally, the example above assumes an ideal realization of the model in a clinical setting. However, several improvements must be made before this conceptual model can be validated experimentally. Perhaps most significantly, experimental data must be used to calibrate the model rather than the abstract values used in this project, particularly the relative thickness of the three tumor layers, the growth rates, and the factors governing CTL behavior. The model can then be expanded to account for evolutionary changes in the tumor population. For example, what happens when a CTL can no longer bind to the surface of a tumor cell, or if tumor cells fail to generate an attractant?

It is important to note that the current model simplifies the immune response by focusing on one type of cell. Incorporating more types of cells involved in the response (such as natural killer cells and CD4 cells) would allow for a more complete picture of interactions within the immune system, and between its cells and those of the tumor.

Finally, nutrient competition in the microenvironment between the CTLs and the tumor cells can affect immunoediting outcomes [2]. Glucose and oxygen, for example, are consumed by both cell populations, and gluttony could locally drive one species to extinction.

In conclusion, the results of this project indicate that chemotherapeutic outcomes may depend on the immunoediting context in which cytotoxicity is introduced. By incorporating the processes discussed above, the model could be adapted for the experimental validation of its significance.

Appendix A: Model Elements

Object		Properties	Property Type
Cells	Proliferating, Quiescent, Necrotic, CTL, Medium	Position	Variable
		Volume	Variable
		Color	Parameter
Environment		X-extent	Parameter
		Y-extent	Parameter
		Boundary Condition	Parameter
Fields	Growth, Survival, Attractant	Diffusion Constant	Parameter
		Decay Rate	Parameter
		Boundary Condition	Parameter
		Initial Value	Parameter

Table 1. Objects in the model are assigned properties

Behaviors & Interactions	Participating Objects
Adhesion	Cells
Diffusion	Fields
Secretion	Proliferating Cells, Quiescent Cells, Necrotic Cells, Attractant Field
Uptake	Proliferating Cells, Quiescent Cells, Growth Field, Survival Field
Volume Regulation	Cells
Chemotaxis	CTL Cells, Attractant Field
Nutrient–modulated Proliferation	Proliferating Cells, Quiescent Cells, Growth Field
Nutrient–modulated Necrosis	Proliferating Cells, Quiescent Cells, Necrotic Cells, Survival Field
CTL–induced Death	CTL Cells, Proliferating Cells, Quiescent Cells

Table 2. Objects perform behaviors that drive the model

Object		Property	Parameter
Environment		Size	X-extent, Y-extent, 1
		Boundary Condition	Periodic
Fields	Growth Survival	Initial Value	Concentration
		Boundary Type	Constant Value
	Attractant	Initial Value	Concentration
		Boundary Type	Constant Derivative
Cells		Shape	Radius
		Position	Center Coordinates
		Composition	Cell Type

Table 3. Initial conditions set the state of the model at t_0 and boundary conditions determine behavior at simulation limits

Appendix B: Definitions

Term	Annotation on Terminology (Code)
Chemotaxis	C16422
Chemotherapy	C15632
Cytotoxic	C37893
CTL	C12543
Environment	C16551
Field	C70773
Malignant	C14143
Morphology	C17943
Necrotic	C25286

Table 4. All definitions were taken from the National Cancer Institute Thesaurus

Appendix C: CC3DML Script

```
<CompuCell3D Revision="20181205" Version="3.7.9">

   <Potts>

      <!-- Basic properties of CPM (GGH) algorithm -->
      <Dimensions x="250" y="250" z="1"/>
      <Steps>10000000</Steps>
      <Temperature>25.0</Temperature>
      <NeighborOrder>3</NeighborOrder>
      <Boundary_x>Periodic</Boundary_x>
      <Boundary_y>Periodic</Boundary_y>
   </Potts>

   <Plugin Name="CellType">

      <!-- Listing all cell types in the simulation -->
      <CellType TypeId="0" TypeName="Medium"/>
      <CellType TypeId="1" TypeName="Proliferating"/>
      <CellType TypeId="2" TypeName="Quiescent"/>
      <CellType TypeId="3" TypeName="Necrotic"/>
      <CellType TypeId="4" TypeName="CTL"/>
   </Plugin>

   <Plugin Name="Volume"/>

   <Plugin Name="CenterOfMass">

      <!-- Module tracking center of mass of each cell -->
   </Plugin>

   <Plugin Name="NeighborTracker">

      <!-- Module tracking neighboring cells of each cell -->
   </Plugin>

   <Plugin Name="BoundaryPixelTracker">

      <!-- Module tracking boundary pixels of each cell -->
      <NeighborOrder>1</NeighborOrder>
   </Plugin>

   <Plugin Name="Contact">

      <!-- Specification of adhesion energies -->
      <Energy Type1="Medium" Type2="Medium">10.0</Energy>
      <Energy Type1="Medium" Type2="Proliferating">20.0</Energy>
      <Energy Type1="Medium" Type2="Quiescent">20.0</Energy>
      <Energy Type1="Medium" Type2="Necrotic">20.0</Energy>
      <Energy Type1="Medium" Type2="CTL">20.0</Energy>
      <Energy Type1="Proliferating" Type2="Proliferating">8.0</Energy>
      <Energy Type1="Proliferating" Type2="Quiescent">11.0</Energy>
      <Energy Type1="Proliferating" Type2="Necrotic">10.0</Energy>
      <Energy Type1="Proliferating" Type2="CTL">13.0</Energy>
      <Energy Type1="Quiescent" Type2="Quiescent">8.0</Energy>
      <Energy Type1="Quiescent" Type2="Necrotic">9.0</Energy>
      <Energy Type1="Quiescent" Type2="CTL">13.0</Energy>
```

```
      <Energy Type1="Necrotic" Type2="Necrotic">8.0</Energy>
      <Energy Type1="Necrotic" Type2="CTL">13.0</Energy>
      <Energy Type1="CTL" Type2="CTL">100.0</Energy>
      <NeighborOrder>3</NeighborOrder>
   </Plugin>

    <Plugin Name="Chemotaxis">

       <!-- Specification of chemotaxis properties of select cell types. -->
       <ChemicalField Name="Attractant" Source="DiffusionSolverFE">
            <ChemotaxisByType ChemotactTowards="" Lambda="25000.0" SaturationCoef="10.0"
Type="CTL"/>
       </ChemicalField>
    </Plugin>

    <Plugin Name="Secretion">

       <!-- Specification of secretion properties of select cell types. -->
    </Plugin>

   <Steppable Type="DiffusionSolverFE">

      <!-- Specification of PDE solvers -->
      <DiffusionField Name="Survival">
         <DiffusionData>
            <FieldName>Survival</FieldName>
            <GlobalDiffusionConstant>10</GlobalDiffusionConstant>
            <GlobalDecayConstant>1e-05</GlobalDecayConstant>
            <InitialConcentrationExpression>10.0</InitialConcentrationExpression>
            <DiffusionCoefficient CellType="Proliferating">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Quiescent">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Necrotic">0.1</DiffusionCoefficient>
            <DecayCoefficient CellType="Proliferating">0.0001</DecayCoefficient>
            <DecayCoefficient CellType="Quiescent">0.0001</DecayCoefficient>
            <DecayCoefficient CellType="Necrotic">0.0001</DecayCoefficient>
         </DiffusionData>
         <SecretionData>
         </SecretionData>
         <BoundaryConditions>
            <Plane Axis="X">
               <ConstantValue PlanePosition="Min" Value="10.0"/>
               <ConstantValue PlanePosition="Max" Value="10.0"/>
            </Plane>
            <Plane Axis="Y">
               <ConstantValue PlanePosition="Min" Value="10.0"/>
               <ConstantValue PlanePosition="Max" Value="10.0"/>
            </Plane>
         </BoundaryConditions>
      </DiffusionField>
      <DiffusionField Name="Growth">
         <DiffusionData>
            <FieldName>Growth</FieldName>
            <GlobalDiffusionConstant>10</GlobalDiffusionConstant>
            <GlobalDecayConstant>1e-05</GlobalDecayConstant>
            <InitialConcentrationExpression>10.0</InitialConcentrationExpression>
            <DiffusionCoefficient CellType="Proliferating">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Quiescent">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Necrotic">0.1</DiffusionCoefficient>
            <DecayCoefficient CellType="Proliferating">0.0001</DecayCoefficient>
            <DecayCoefficient CellType="Quiescent">0.0001</DecayCoefficient>
```

```
            <DecayCoefficient CellType="Necrotic">0.0001</DecayCoefficient>
         </DiffusionData>
         <SecretionData>
         </SecretionData>
        <BoundaryConditions>
            <Plane Axis="X">
               <ConstantValue PlanePosition="Min" Value="10.0"/>
               <ConstantValue PlanePosition="Max" Value="10.0"/>
            </Plane>
            <Plane Axis="Y">
               <ConstantValue PlanePosition="Min" Value="10.0"/>
               <ConstantValue PlanePosition="Max" Value="10.0"/>
            </Plane>
         </BoundaryConditions>
      </DiffusionField>
      <DiffusionField Name="Attractant">
         <DiffusionData>
            <FieldName>Attractant</FieldName>
            <GlobalDiffusionConstant>10.0</GlobalDiffusionConstant>
            <GlobalDecayConstant>0.01</GlobalDecayConstant>
            <DiffusionCoefficient CellType="Proliferating">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Quiescent">0.1</DiffusionCoefficient>
            <DiffusionCoefficient CellType="Necrotic">0.1</DiffusionCoefficient>
         </DiffusionData>
         <SecretionData>
            <Secretion Type="Proliferating">0.1</Secretion>
            <Secretion Type="Quiescent">0.1</Secretion>
            <Secretion Type="Necrotic">0.3</Secretion>
         </SecretionData>
         <BoundaryConditions>
            <Plane Axis="X">
               <ConstantDerivative PlanePosition="Min" Value="0.0"/>
               <ConstantDerivative PlanePosition="Max" Value="0.0"/>
            </Plane>
            <Plane Axis="Y">
               <ConstantDerivative PlanePosition="Min" Value="0.0"/>
               <ConstantDerivative PlanePosition="Max" Value="0.0"/>
            </Plane>
         </BoundaryConditions>
      </DiffusionField>
   </Steppable>

   <Steppable Type="BlobInitializer">

      <!-- Initial layout of cells in the form of spherical (circular in 2D) blob -->
      <Region>
         <Center x="125" y="125" z="0"/>
         <Radius>50</Radius>
         <Gap>0</Gap>
         <Width>5</Width>
         <Types>Proliferating</Types>
      </Region>
   </Steppable>
</CompuCell3D>
```

Appendix D: Python Script

```
# This simulation code is compatible with CompuCell3D ver. 3.7.9
# Vedang Narain, 29 April 2019

from PySteppables import *
import CompuCell
import sys
import numpy as np

from PySteppablesExamples import MitosisSteppableBase

# set starting target volume and lambda volume
class ConstraintInitializerSteppable(SteppableBasePy):
    def __init__(self,_simulator,_frequency=1):
        SteppableBasePy.__init__(self,_simulator,_frequency)

    def start(self):
        for cell in self.cellList:
            cell.targetVolume=25
            cell.lambdaVolume=2.0

class GrowthSteppable(SteppableBasePy):
    def __init__(self,_simulator,_frequency=1):
        SteppableBasePy.__init__(self,_simulator,_frequency)

    # define plots
    def start(self):
        self.pW = self.addNewPlotWindow(_title='Cell Populations',_xAxisTitle='MonteCarlo Step (MCS)',_yAxisTitle='Number', _xScaleType='linear',_yScaleType='linear')
        self.pW.addPlot('population_PROLIFERATING',_style='Lines',_color='green',_size=3)
        self.pW.addPlot('population_QUIESCENT',_style='Lines',_color='yellow',_size=3)
        self.pW.addPlot('population_NECROTIC',_style='Lines',_color='red',_size=3)
        self.pW.addPlot('population_total',_style='Lines',_color='grey',_size=1)
        self.pW.addPlot('population_CTL',_style='Lines',_color='blue',_size=3)

    # initialize field variables
    def step(self,mcs):

        # Make sure Secretion plugin is loaded
        # make sure this field is defined in one of the PDE solvers
        field_survival = self.getConcentrationField('Survival')
        secretor_survival = self.getFieldSecretor('Survival')
        field_growth = self.getConcentrationField('Growth')
        secretor_growth = self.getFieldSecretor('Growth')
        field_attractant = self.getConcentrationField('Attractant')
        secretor_attractant = self.getFieldSecretor('Attractant')

        # track cell populations
        if not mcs%10: # every 10 mcs steps
            total_population_PROLIFERATING = 0; total_population_QUIESCENT = 0; total_population_NECROTIC = 0; total_population_CTL = 0; total_population_tumor = 0
            for cell in self.cellListByType(self.PROLIFERATING):
                total_population_PROLIFERATING+=1
            self.pW.addDataPoint('population_PROLIFERATING', mcs, total_population_PROLIFERATING)
            for cell in self.cellListByType(self.QUIESCENT):
                total_population_QUIESCENT+=1
            self.pW.addDataPoint('population_QUIESCENT', mcs, total_population_QUIESCENT)
            for cell in self.cellListByType(self.NECROTIC):
                total_population_NECROTIC+=1
            self.pW.addDataPoint('population_NECROTIC', mcs, total_population_NECROTIC)
            for cell in self.cellListByType(self.CTL):
                total_population_CTL+=1
            self.pW.addDataPoint('population_CTL', mcs, total_population_CTL)
            total_population_tumor = total_population_PROLIFERATING + total_population_QUIESCENT + total_population_NECROTIC
            self.pW.addDataPoint('population_total', mcs, total_population_tumor)

        # outline nutrient-dependent behavior
        for cell in self.cellListByType(self.PROLIFERATING,self.QUIESCENT): # iterate over PROLIFERATING and QUIESCENT cells
            res_growth = secretor_growth.uptakeInsideCellTotalCount(cell, 1000.0, 0.002) # arguments are: cell, max uptake, relative uptake
            res_survival = secretor_survival.uptakeInsideCellTotalCount(cell, 1000.0, 0.002) # arguments are: cell, max uptake, relative uptake
            growth_concentration_within = -(res_growth.tot_amount) # since amount is negative by default
            survival_concentration_within = -(res_survival.tot_amount) # since amount is negative by default
            if cell.type == 1 and growth_concentration_within >=0.2: # if cell is PROLIFERATING with sufficient growth factor
                cell.targetVolume+=(1*growth_concentration_within)/(2+growth_concentration_within) # increase the target volume
            if cell.type == 2 and growth_concentration_within >= 0.2: # if cell is QUIESCENT with sufficient growth factor
                cell.type = 1 # QUIESCENT cell becomes PROLIFERATING
                cell.targetVolume+=(1*growth_concentration_within)/(2+growth_concentration_within) # increase the target volume
```

```
            if cell.type == 1 and growth_concentration_within < 0.2: # if cell is PROLIFERATING with insufficient growth factor
                cell.type = 2 # PROLIFERATING cell becomes QUIESCENT
                cell.targetVolume = cell.volume # freeze the current volume
            if survival_concentration_within < 0.05: # if there is insufficient survival factor
                cell.type = 3 # cell becomes NECROTIC
                cell.targetVolume *= 0.98 # make the cell shrink

        # iterate over NECROTIC cells and make them shrink
        for cell in self.cellListByType(self.NECROTIC):
            cell.targetVolume*=0.98 # update the shrinking volume

        # seed CTL cells
        immune_strength = 5.0                   # enter desired strength of immune response, i.e., number of CTLs arriving per wave
        CTL_size = 4                            # enter desired CTL size
        lattice_size = 250                      # enter square lattice size here
        edge = lattice_size - CTL_size          # used in cell seeding
        if not mcs%50:                          # if there is no remainder, i.e., every x mcs steps
            border_attractant_raw = field_attractant[0, 0, 0] + field_attractant[125, 0, 0] + field_attractant[250, 0, 0] + field_attractant[250, 125, 0] \
                        + field_attractant[250, 250, 0] + field_attractant[125, 250, 0] + field_attractant[0, 250, 0] + field_attractant[0, 125, 0]
            border_attractant_float = (2.0*border_attractant_raw)/(5.0+border_attractant_raw)
            border_attractant = int(round(border_attractant_float)) # get rough estimate of border values
            if border_attractant >= 1:                  # once there's a sufficient amount of attractant at border
                for count in range(border_attractant):          # seed a proportional number of CTLs
                    while True:                         # infinite loop
                        x = np.random.random() * lattice_size       # random_factor * lattice_size
                        y = np.random.random() * lattice_size       # random_factor * lattice_size
                        cell_0 = self.cellField[x, y, 0]        # attempt to choose random cell field
                        if not cell_0 and not ((CTL_size < x < edge) and (CTL_size < y < edge)): # if there is no cell there and the coordinates are not away from the edge
                            break                       # exit loop
                    cell = self.newCell(self.CTL)               # create a new cell of type PROLIFERATING
                    self.cellField[x:x + CTL_size, y:y + CTL_size, 0] = cell     # dimensions of cell will be size x size x 1
                    cell.targetVolume = 25.0                    # set target volume
                    cell.lambdaVolume = 40.0                    # set lambda volume
                    cell.dict['kills'] = 0                      # give CTL new attribute 'kills' to track number of tumor cells killed

class MitosisSteppable(MitosisSteppableBase):
    def __init__(self,_simulator,_frequency=1):
        MitosisSteppableBase.__init__(self,_simulator, _frequency)
    def step(self,mcs):
        cells_to_divide = []

        # divide cell once volume > 50
        for cell in self.cellList:
            if cell.volume > 50:
                cells_to_divide.append(cell)
        for cell in cells_to_divide:
            self.divideCellRandomOrientation(cell)

    def updateAttributes(self):
        self.parentCell.targetVolume /= 2.0 # reduce parent target volume
        self.cloneParent2Child()

        # when PROLIFERATING cells divide, they form two QUIESCENT cells
        if self.parentCell.type==1:
            self.childCell.type=2
            self.parentCell.type=2

class DeathSteppable(SteppableBasePy):
    def __init__(self,_simulator,_frequency=1):
        SteppableBasePy.__init__(self,_simulator,_frequency)
    def step(self,mcs):

        chemo_mcs = 2000 # enter point of chemotherapy administration

        # CTL cells make tumor cells necrotic
        for cell in self.cellListByType(self.CTL):
            kills = cell.dict['kills']
            prob_death = kills/(10.0+kills)     # probability of death is directly proportional to kill count
            if mcs < chemo_mcs:                 # if chemotherapy not yet administered
                prob_kill = 1.0 - prob_death    # probability of killing is inversely proportional to kill count
            elif mcs >= chemo_mcs:              # if chemotherapy administered
                prob_kill = 0.5 * (1.0 - prob_death) # probability of killing is halved
                natural = 0.2
            for neighbor, commonSurfaceArea in self.getCellNeighborDataList(cell): # iterate over cell's neighbors
                if (neighbor and neighbor.type < 3) and (np.random.random() < prob_kill): # if neighbor is present and type PROLIFERATING or QUIESCENT, roll the killing dice
                    neighbor.type = 3 # make cell NECROTIC
                    cell.dict['kills'] += 1 # increment CTL kill count by 1
                    if np.random.random() < prob_death: # roll the death dice + 5% to account for natural death
```

```
                cell.targetVolume=0
                cell.lambdaVolume=100
            break # move onto the next cell

# chemotherapy
if mcs==chemo_mcs: # simulate chemotherapeutic perturbation
    for cell in self.cellListByType(self.CTL):
        if np.random.random() < 0.9: # kill all cells with 90% probability
            cell.targetVolume=0
            cell.lambdaVolume=100
    for cell in self.cellListByType(self.PROLIFERATING, self.QUIESCENT):
        if np.random.random() < 0.9: # kill all cells with 90% probability
            cell.type = 3 # cell becomes NECROTIC
            cell.targetVolume *= 0.98
```

References

[1]García, Alvaro & Iarosz, Kelly & Batista, Antonio & Seoane, Jesús & Viana, Ricardo & Sanjuán, Miguel. (2019). The role of dose-density in combination cancer chemotherapy.

[2]I. Kareva and F. Berezovskaya, "Cancer immunoediting: A process driven by metabolic competition as a predator–prey–shared resource type model", Journal of Theoretical Biology, vol. 380, pp. 463-472, 2015.

[3]G. Dunn, A. Bruce, H. Ikeda, L. Old and R. Schreiber, "Cancer immunoediting: from immunosurveillance to tumor escape", Nature Immunology, vol. 3, no. 11, pp. 991-998, 2002. Available: 10.1038/ni1102-991.

[4]D. Mittal, M. Gubin, R. Schreiber and M. Smyth, "New insights into cancer immunoediting and its three component phases—elimination, equilibrium and escape", Current Opinion in Immunology, vol. 27, pp. 16-25, 2014.

[5]Kang, Duck-Hee et al. “Significant impairment in immune recovery after cancer treatment.” Nursing research vol. 58,2 (2009): 105-14. doi:10.1097/NNR.0b013e31818fcecd

[6]L. G. de Pillis, D. G. Mallet, and A. E. Radunskaya, “Spatial Tumor-Immune Modeling,” Computational and Mathematical Methods in Medicine, vol. 7, no. 2-3, pp. 159-176, 2006.

[7]Graner F, Glazier JA. Simulation of biological cell sorting using a two-dimensional extended Potts model. Phys Rev Lett. 1992;69(13):785–790.46.

[8]Glazier JA, Graner F. Simulation of the differential adhesion driven rearrangement of biological cells. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics. 1993;47:2128–54.

[9]Swat M, Thomas G, Belmonte J, Shirinifard A, Hmeljak D, Glazier J. Multi-Scale Modeling of Tissues Using CompuCell3D. Methods in Cell Biology. 2012;110:325–366. Pmid:22482955

[10]Shibata, K., Itoh, Y., Itoh, K., Watanabe, M. and Yamaguchi, T. (2017). Primordial oscillations in life: Direct observation of glycolytic oscillations in individual HeLa cervical cancer cells. Chaos: An Interdisciplinary Journal of Nonlinear Science, 27(10), p.104602.