

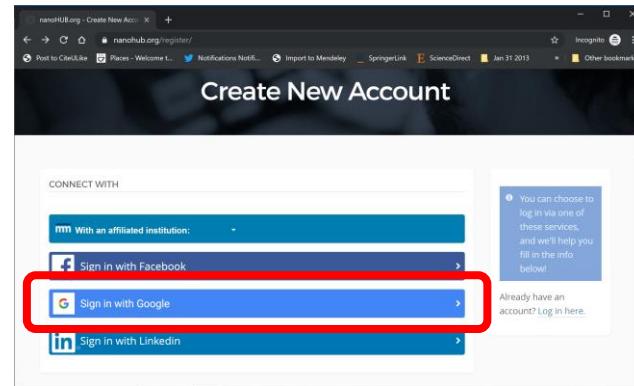
nanoHUB Account

- This talk's online PhysiCell models are cloud-hosted on nanoHUB.org.
- nanoHUB is **free**, but it requires a one-time registration.

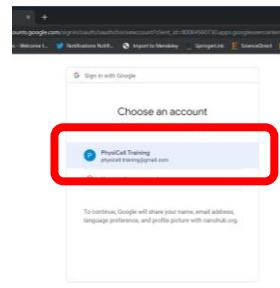
Steps:

1. Visit <https://nanohub.org/register>
2. Choose "Sign in with Google"
3. Choose a Google account
4. Click "No" (so it doesn't try to associate with some other nanoHUB account)
5. Finish filling in details, and you're done!
6. Use your google account to sign in in the future.

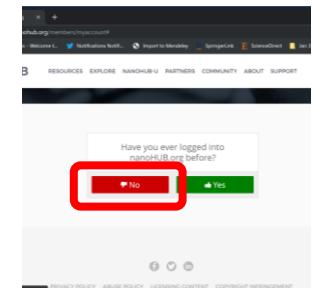
2



3



4



Introduction to Agent-Based Modeling in Cancer (extended version)



Paul Macklin, Ph.D.

Intelligent Systems Engineering
Indiana University

Source code, extended details and more:

GitHub.com/PhysiCell-Training/UCI-sysbio-2022

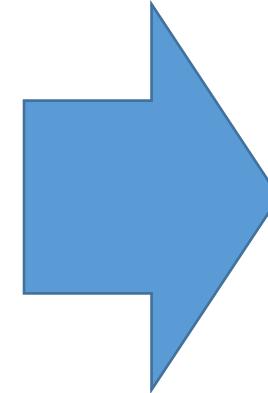
June 13, 2022

Part I:

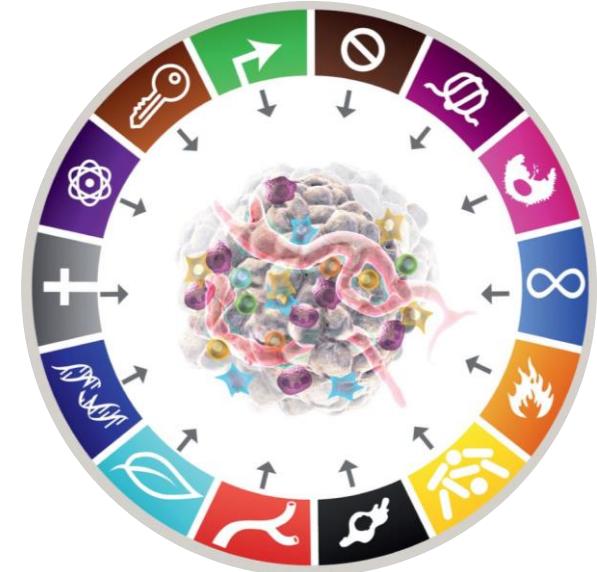
Introduction to ABMs

From single cells to cancer ecosystems ...

- Single-cell behaviors:
 - Growth
 - Division
 - Differentiation
 - Death
 - Consumption
 - Metabolism
 - Secretion
 - Signaling
 - Mutations
 - Motility
- Cell-cell interactions:
 - Adhesion
 - Mechanics
 - Predation
 - Contact communication
- Physical constraints:
 - Diffusion limits
 - Mechanical barriers



Multicellular cancer ecosystem



Multicellular systems biology seeks to *understand* these systems.
Multicellular systems engineering seeks to *control* them.

Source: Hanahan (2022)
DOI: [10.1158/2159-8290.CD-21-1059](https://doi.org/10.1158/2159-8290.CD-21-1059)

Scientists use [models*] to detangle complex systems.

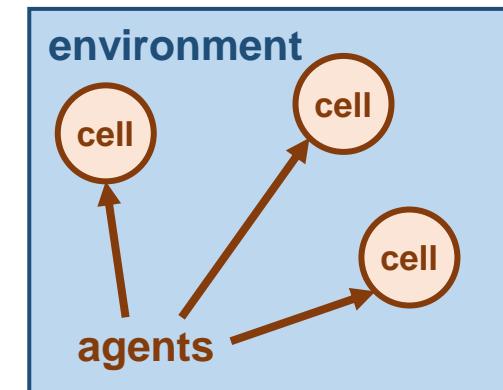
* animal, *in vitro*, engineered, mathematical, ...

Key parts of a multicellular virtual laboratory

- **Model multiple diffusing chemical factors**
 - Growth substrates and metabolites
 - Signaling factors
 - Drugs and therapeutic compounds
- **Model many cells in these chemical environments**
 - Environment-dependent behavior (including molecular-scale "logic")
 - Mechanical interactions
 - Heterogeneity:
 - ◆ individual states
 - ◆ individual parameter values
 - ◆ individual model rules
- **Run many copies of the model in high throughput**
 - Discover the rules that best match observations.
 - Identify and exploit weaknesses that can restore control

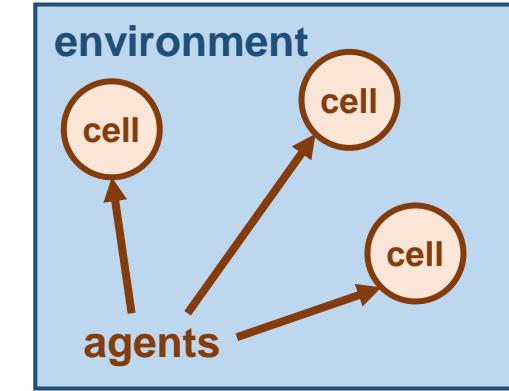
What is an agent-based model?

- Individual cells are modeled as discrete (and stochastic) software **agents**.
 - Individual state variables
 - Individual rules
 - ◆ often written as systems of ordinary differential equations (ODEs) or other constitutive laws
 - **The cell agents encode our biological knowledge and hypotheses as rules**
- Cell agents live in a **virtual tissue microenvironment**.
 - Diffusible substrates
 - ◆ modeled as partial differential equations (PDEs)
 - Mechanical barriers
 - **The microenvironment encodes physical constraints**
- Cells agents **interact** with each other and the environment.
 - Mechanics and contact interactions
 - Secretion, uptake and export of diffusible substrates



What's the connection to biology and physics?

- The **cell agents encode our biological knowledge and hypotheses**:
 - Cell variables (member data) are selected to record important biological quantities
 - ◆ Volume, cell cycle state, energy, ...
 - Cell rules (methods) encode biological hypotheses
 - ◆ Increase motility in low oxygen, down-regulate cycling under compression, ...
 - Cell rules are often written at mathematical models
 - ◆ Potential functions for mechanics, systems of ODEs for metabolism, ...
- The **microenvironment encodes physical constraints**:
 - *Chemical transport*: diffusion and advection equations (PDEs)
 - *Tissue mechanics*: viscoelastic, plastoelastic or other solid mechanics
- Most agent-based models combine **discrete** cell agents and **continuum** microenvironment processes. This is a **hybrid continuum-discrete approach**.



Types of cell-based models

- **lattice-bound**

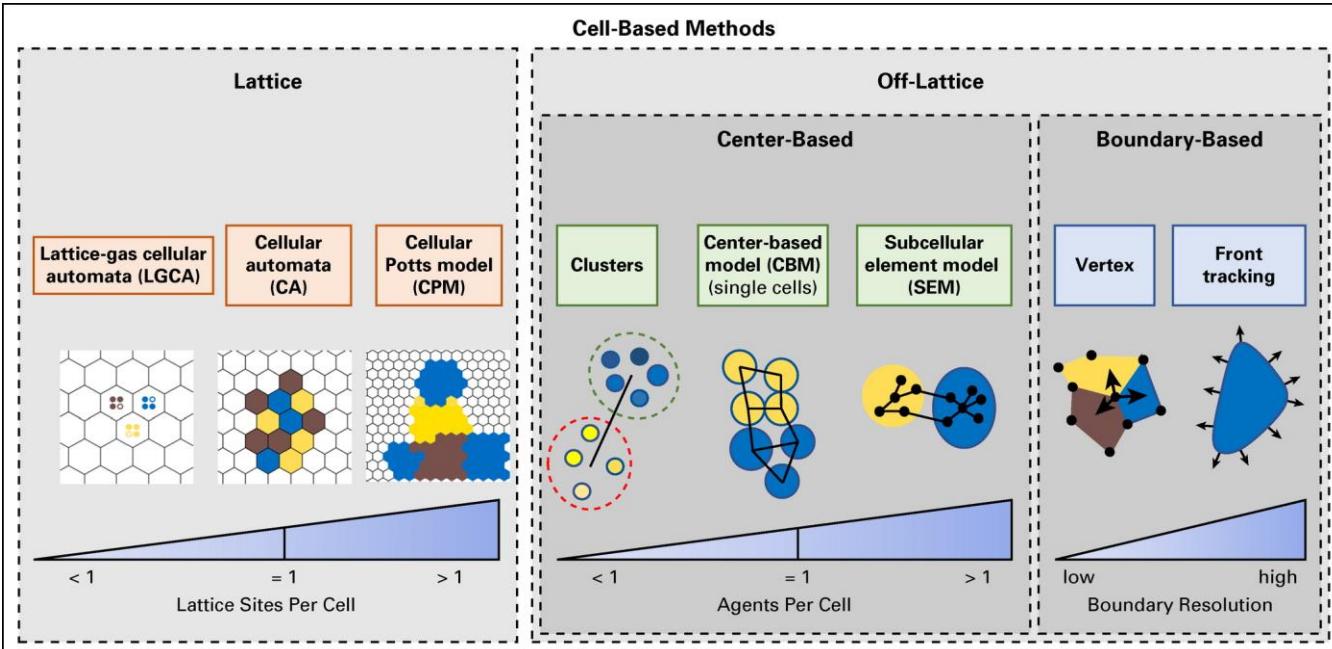
- resolution:

- ◆ < 1 site / cell:
 - » lattice gas
 - ◆ 1 site / cell
 - » cellular automaton
 - ◆ many sites / cell
 - » cellular Potts

- **off-lattice**

- **center-based**

- **boundary-based**



J. Metzcar, Y. Wang, R. Heiland, and P. Macklin. A review of cell-based computational modeling in cancer biology. *JCO Clinical Cancer Informatics* 3:1-13, 2019 (invited review). DOI: [10.1200/CCI.18.00069](https://doi.org/10.1200/CCI.18.00069).

BioFVM: Simulating 3-D biotransport

Design goal: Simulate multiple diffusing substrates in 3D with desktops or single HTC/HPC nodes

Typical use: pO_2 , glucose, metabolic waste, signaling factors, and a drug, on 10 mm^3 at $20 \mu\text{m}$ resolution

Features:

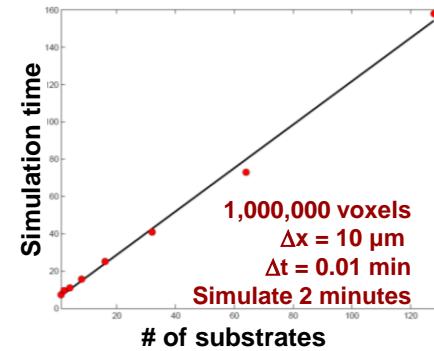
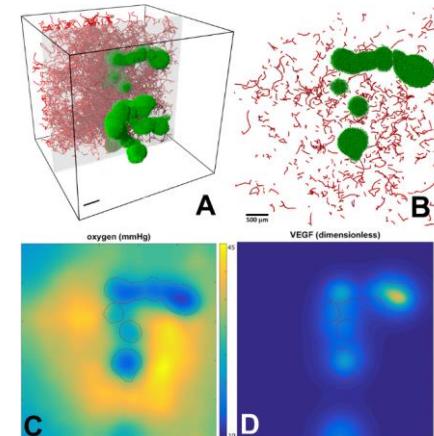
- Off-lattice cell secretion and uptake
- 2nd-order accurate (space), 1st-order accurate (time), numerically stable

Method:

- Operator splitting, LOD, customized Thomas solvers, etc.
- Standard C++11, cross-platform
- OpenMP parallelization
- $O(n)$ cost scaling in # substrates, # voxels
- Easy to simulate 5-10 substrates on 10^6 voxels

Reference: Ghaffarizadeh et al., *Bioinformatics* (2016)

DOI: [10.1093/bioinformatics/btv730](https://doi.org/10.1093/bioinformatics/btv730)



PhysiCell: A multicellular framework

Design goal: Simulate 10^6 or more cells in 2D or 3D on desktops or single HPC nodes

Features:

- Off-lattice cell positions
- Mechanics-based cell movement
- Cell processes (cycling, motility, ...)
- Signal-dependent phenotype
- Can dynamically attach custom data and functions on a cell-by-cell basis

Method:

- Standard C++11, cross-platform
- OpenMP parallelization
- $O(n)$ cost scaling in # cells

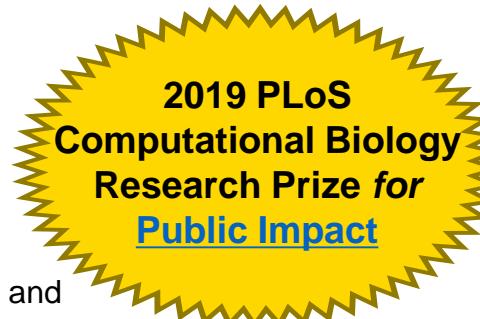
Reference: Ghaffarizadeh et al.,
PLoS Comput. Biol. (2018)

DOI: [10.1371/journal.pcbi.1005991](https://doi.org/10.1371/journal.pcbi.1005991)

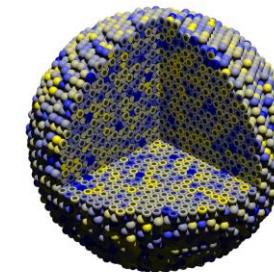


Try this model yourself!

nanohub.org/tools/pc4heterogen



Current time: 0 days, 0 hours, and 0.00 minutes
18317 cells



Competition in a 3-D tumor
[View on YouTube](#) (8K)

Some brief mathematical notes (1)

- Diffusion equations take the following form:
 - Diffusion and decay throughout the domain
 - Secretion, uptake, and net export are centered at cell positions
 - ◆ Rate parameters can change dynamically over time for *each individual cell*
 - ◆ Secretion term mimics many typical forms. Use export if you need a particular form.

$$\frac{\partial \rho}{\partial t} = D \nabla^2 \rho - \lambda \rho + \sum_i \overbrace{\delta(\mathbf{x} - \mathbf{x}_i)}^{\text{Dirac delta}} \left\{ V_i \left(\overbrace{\frac{\text{secretion}}{S_i(\rho_i^* - \rho)}} - \overbrace{\widetilde{U}_i \rho}^{\text{uptake}} \right) + \overbrace{\widetilde{E}_i}^{\substack{\text{generalized} \\ \text{net export}}} \right\}$$

Some brief mathematical notes (2)

- Cell state transitions are stochastic:
 - If a transition rate between states i and j is r_{ij} then:
 - ◆ The probability of a transition event during time interval $[t, t + \Delta t]$ is:
$$\text{Prob}(i \rightarrow j) \approx r_{ij}\Delta t$$
(Tacit assumption: at most one event in the time interval)
 - ◆ This has close links to Poisson process theory
 - Algorithmic implementation:
 - ◆ Pick a uniformly random number on the real line $u \in U(0,1)$
 - ◆ If $u \leq r_{ij}\Delta t$, then the event happens. Otherwise not.
- Coarse-graining to ODEs can be helpful for calibrating cycling and death

Some brief mathematical notes (3)

- Motility is a biased random walk:
 - \mathbf{d}_{bias} : preferred migration direction
 - $0 \leq b \leq 1$: migration bias
 - τ_{persist} : persistence time (time until choosing a new direction)
 - s : migration speed
 - In $[t, t + \Delta t]$, the probability of choosing a new direction is $\frac{\Delta t}{\tau_{\text{persist}}}$
 - New direction:
- $$\mathbf{d}_{\text{mot}} = b \mathbf{d}_{\text{bias}} + (1 - b) \xi$$
- ξ is a random unit vector
 - Set \mathbf{d}_{bias} based on biological hypotheses (e.g., a chemical gradient)
 - Normalize this direction, and multiply by migration speed
- **Calibration:** Set s based on observations. Bias parameters by ABC

Some brief mathematical notes (4)

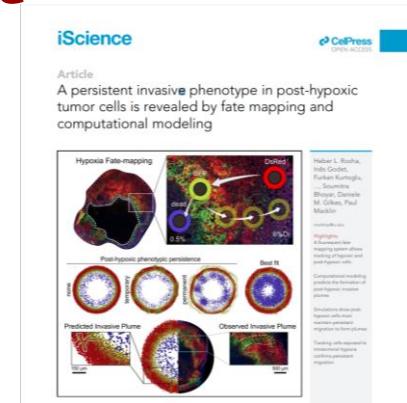
- Intracellular models can take a variety of forms:
 - Boolean network models
 - ◆ Solve with the *PhysiBoSS* package
 - Reaction kinetics (or systems of ODEs)
 - ◆ Solve with the *libRoadrunner* package
 - Flux-balance laws (ideal for metabolism)
 - ◆ Solve with *PhysiFBA* package (in progress)
 - Pre-trained deep neural networks (DNNs)
 - ◆ Run a complex model in high throughput on a cluster. Generate training data.
 - » Features = model inputs. Labels = model outputs.
 - ◆ Train a DNN in *keras* (e.g., in *scikit-learn*) as a faster ***surrogate model***
 - ◆ Import the DNN into PhysiCell (in progress)
- All of these require (for every single cell agent for every time step):
 - Send microenvironment and agent data to intracellular model as inputs
 - Evaluate intracellular model
 - Use intracellular model outputs to write cell phenotype variables.

Phenotype-centric modeling

- We have the key biological processes implemented out-of-the-box:
 - Cycle progression (variety of models available)
 - Apoptotic and necrotic death
 - Mechanics and motility
 - Secretion / Uptake / Export
 - Phagocytosis, fusion, "attack", etc.
 - Mutations
 - ...
- Each process is regulated by corresponding biophysical parameters
 - Generally they are process time scales!
- Modelers write functions to control these process parameters
 - Vary with microenvironmental conditions
 - Vary with cell state
 - Use this to write your biological hypotheses

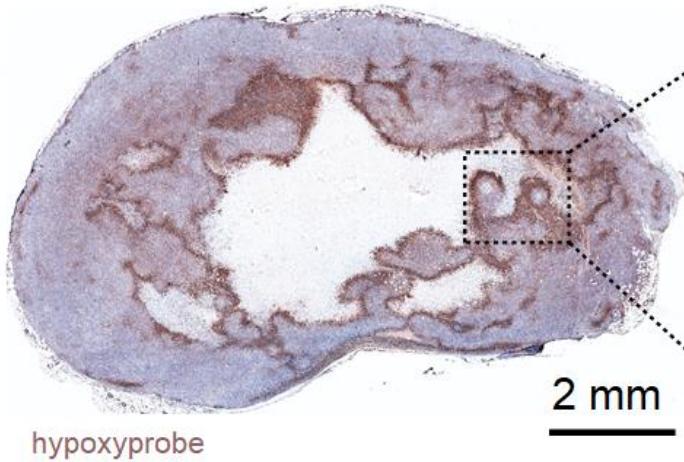
Example: hypoxia-driven breast cancer invasion

Rocha et al., *iScience* (2021)
[DOI: 10.1016/j.isci.2021.102935](https://doi.org/10.1016/j.isci.2021.102935)



Intratumoral hypoxia

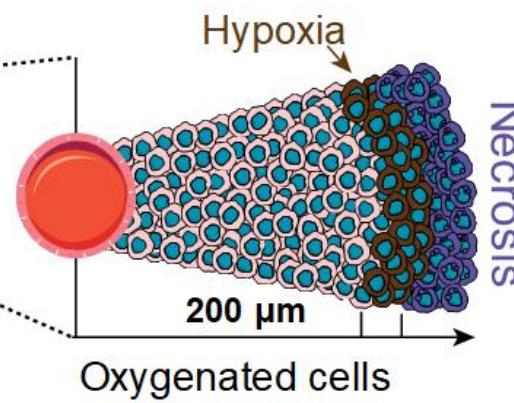
Mouse tumor



Blood vessels



Hypoxia



hypoxyprobe

2 mm

Oxygenated cells

200 μ m

Necrosis

Fate-mapping intratumoral hypoxia

nature communications

View All Nature Research Journals

Search 

Login 

Explore Content ▾ Journal Information ▾ Publish With Us ▾

Sign Up For Alerts 

RSS Feed

nature > nature communications > articles > article

Article | Open Access | Published: 24 October 2019

Fate-mapping post-hypoxic tumor cells reveals a ROS-resistant phenotype that promotes metastasis

Inés Godet, Yu Jung Shin, Julia A. Ju, I Chae Ye, Guannan Wang & Daniele M. Gilkes 

Nature Communications 10, Article number: 4862 (2019) | Cite this article

8904 Accesses | 17 Citations | 167 Altmetric | Metrics

Abstract

Hypoxia is known to be detrimental in cancer and contributes to its development. In this work, we present an approach to fate-map hypoxic cells *in vivo* in order to determine their cellular response to physiological O₂ gradients as well as to quantify their contribution to metastatic spread. We demonstrate the ability of the system to fate-map hypoxic cells in 2D, and in 3D spheroids and organoids. We identify distinct gene expression patterns in cells that experienced intratumoral hypoxia *in vivo* compared to cells exposed to hypoxia *in vitro*. The intratumoral hypoxia gene-signature is a better prognostic indicator for distant metastasis-free survival. Post-hypoxic tumor cells have an ROS-resistant phenotype that provides a survival advantage in the bloodstream and promotes their ability to establish overt metastasis. Post-hypoxic cells retain an increase in the expression of a subset of hypoxia-inducible genes at the metastatic site, suggesting the possibility of a 'hypoxic memory.'

Download PDF 

Sections  Figures  References 

Abstract

Introduction

Results

Discussion

Methods

Data availability

References

Acknowledgements

Author information

Ethics declarations

Additional information

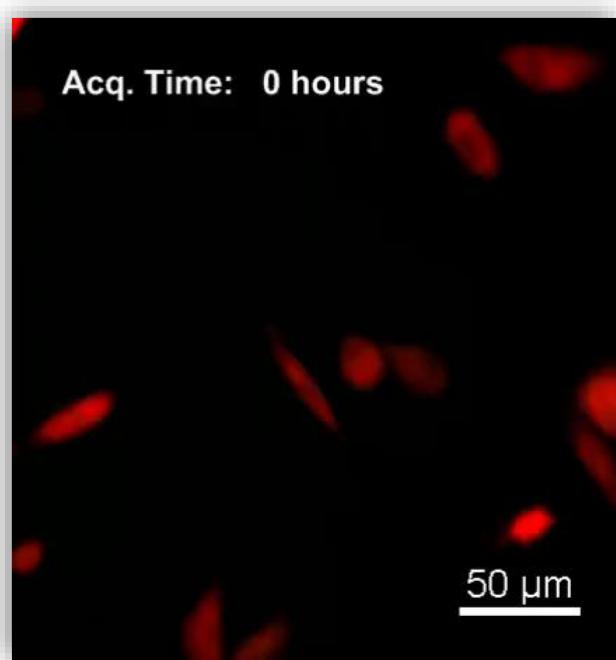
Supplementary information

Rights and permissions

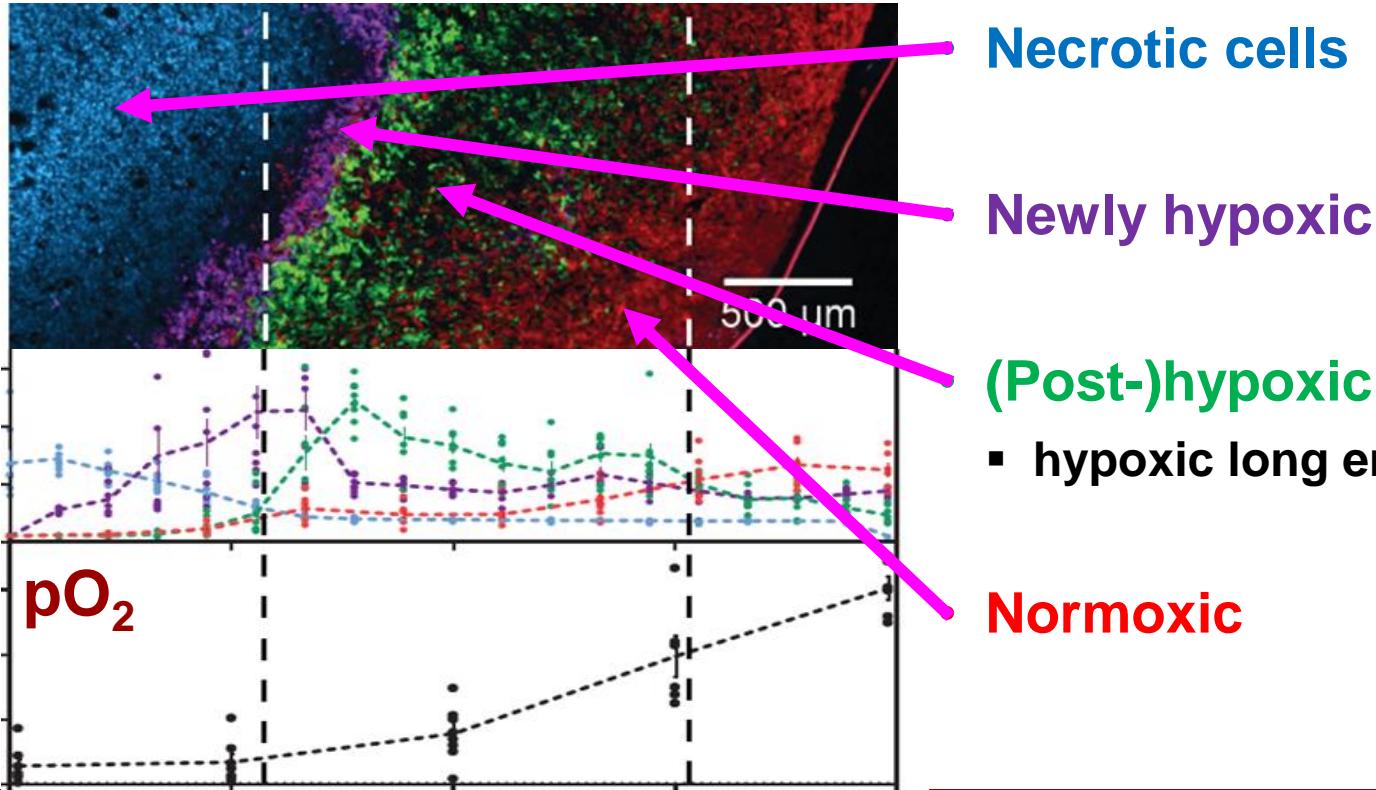
About this article

Further reading

Comments



Hypoxic and post-hypoxic cells



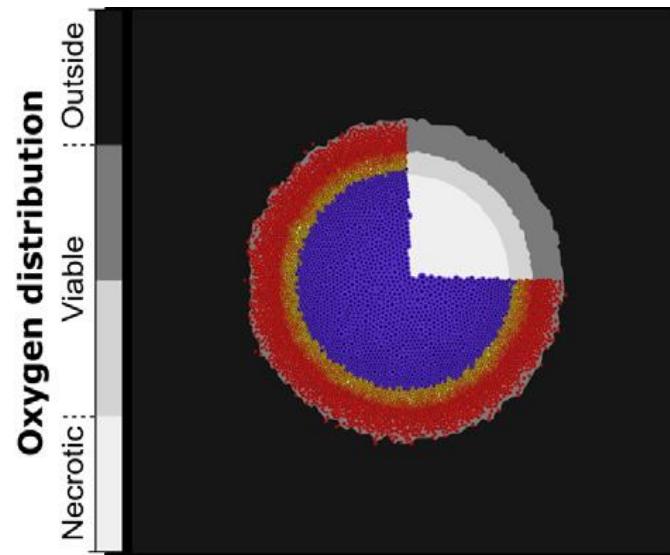
Questions

What are the **rules** of hypoxic cell motility?

How persistent is their response to hypoxic stress?

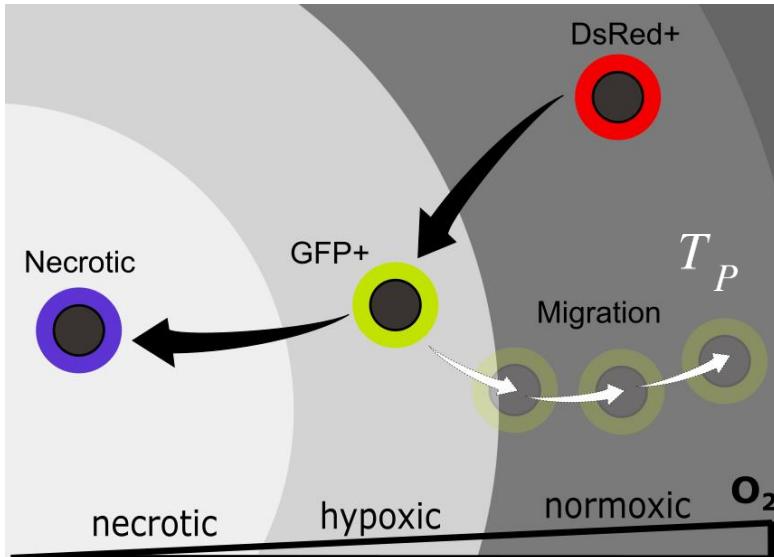
Model overview

- Simulate oxygen diffusion and uptake
- Live cells are **normoxic (RFP)** or **hypoxic (GFP)**.
- Proliferation and necrosis vary with pO_2 and pressure
- Model transition from **RFP** to **GFP** via ODEs
- **GFP** cell motility is based on current pO_2 and its phenotypic persistence.



Phenotypic transitions

Eventually, cells may undergo phenotypic transitions due to their **motility** or **changes in the microenvironment**.



Simple ODE model for protein expression based on the "genes"

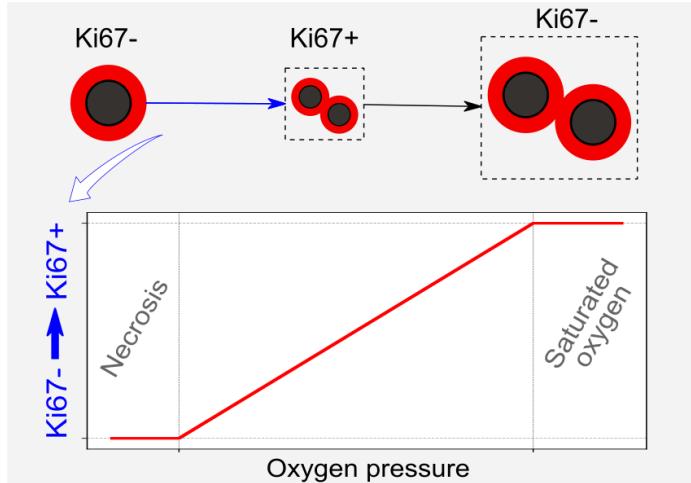
$$G = (G_0, G_1)$$

$$\frac{d[DsRed]}{dt} = G_0 \alpha_0 (1 - [DsRed]) + \beta_0 (G_0 - [DsRed])$$

$$\frac{d[GFP]}{dt} = G_1 \alpha_1 (1 - [GFP]) + \beta_1 (G_1 - [GFP])$$

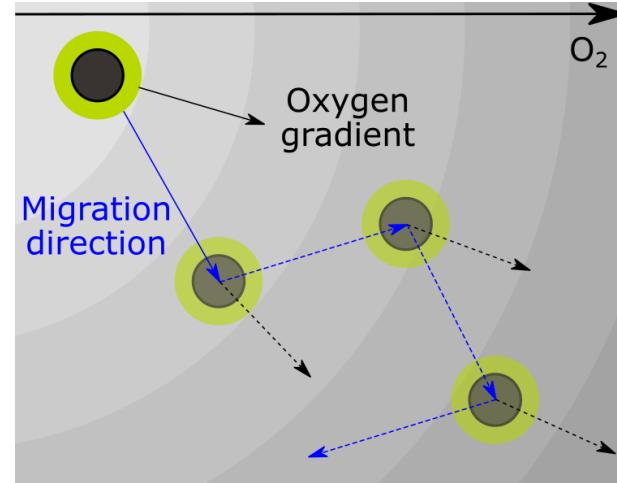
Proliferation and migration

Cell proliferation



Basic Ki-67 model

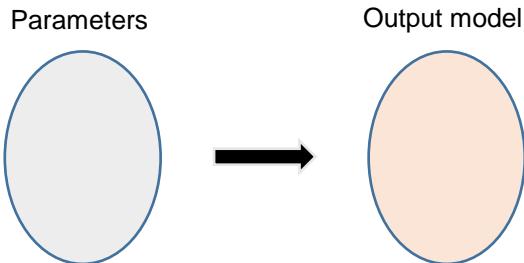
Cell migration



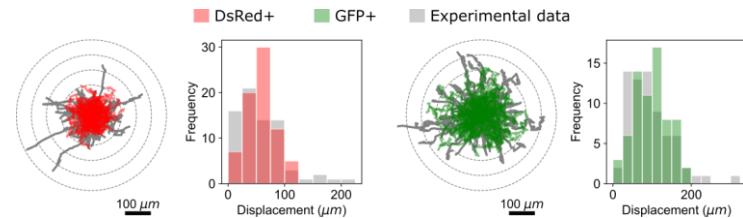
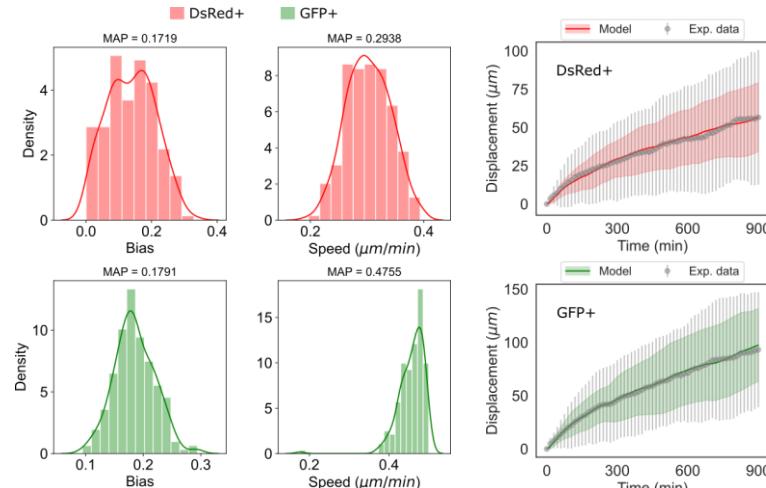
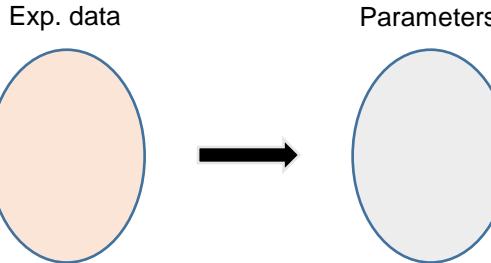
- Speed (s)
- Bias (b)
- Persistence time (t)

Biological observations calibrate cellular motility in hypoxia computational model

Forward problem



Inverse problem



Phenotypic persistence drives invasion

Phenotypic Persistence:

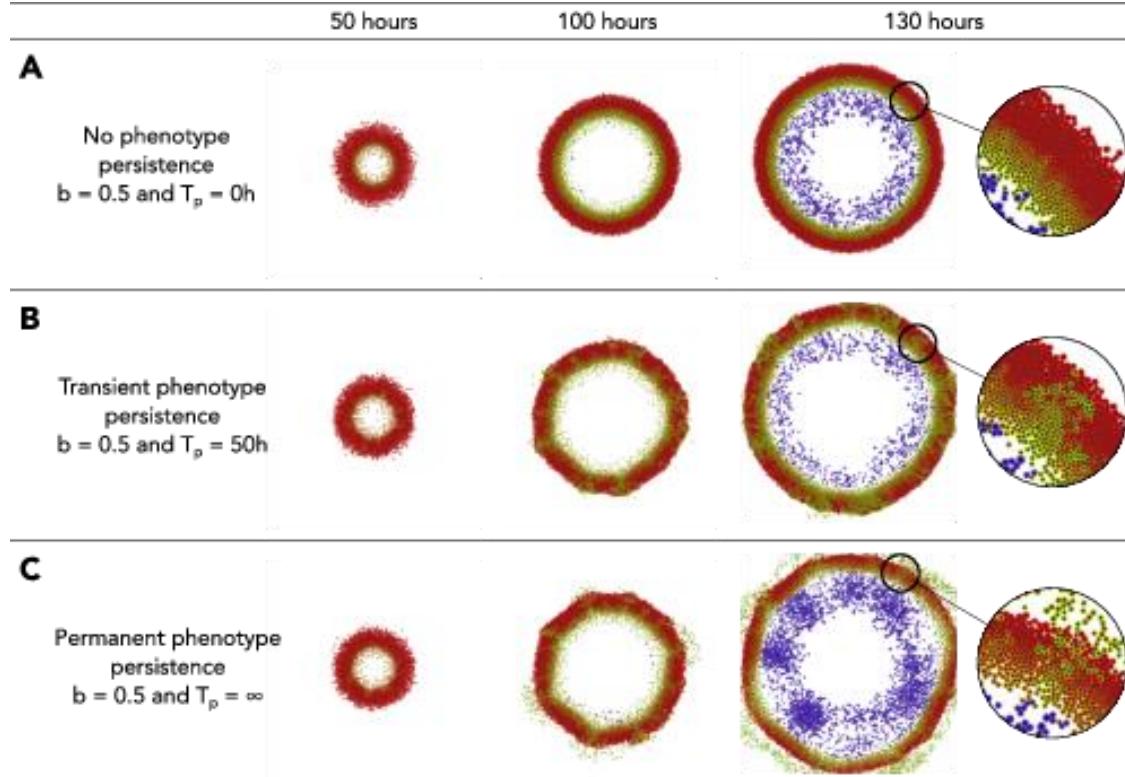
- T_p : duration of hypoxic response

Without persistence ($T_p = 0$) – Row A

- Migration halts at perinecrotic boundary
- Tumors maintain a concentric structure:
 - Oxygenated viable rim (red)
 - hypoxic (or formerly hypoxic) annulus (green)
 - Necrotic core (purple)

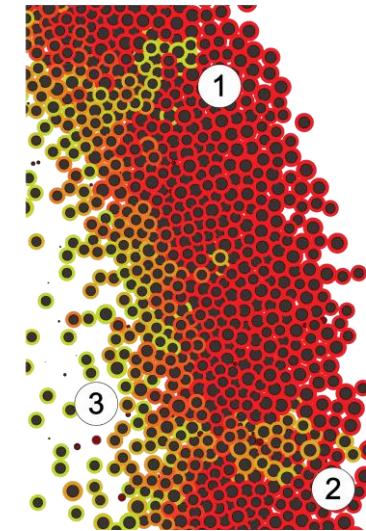
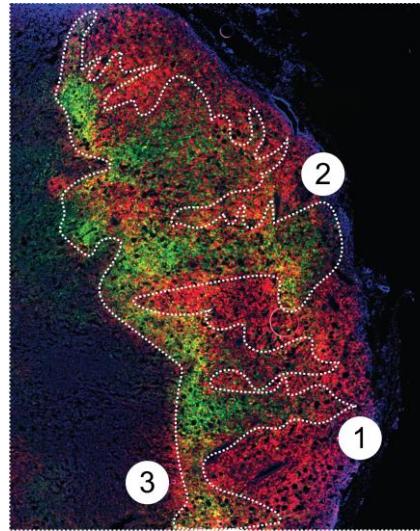
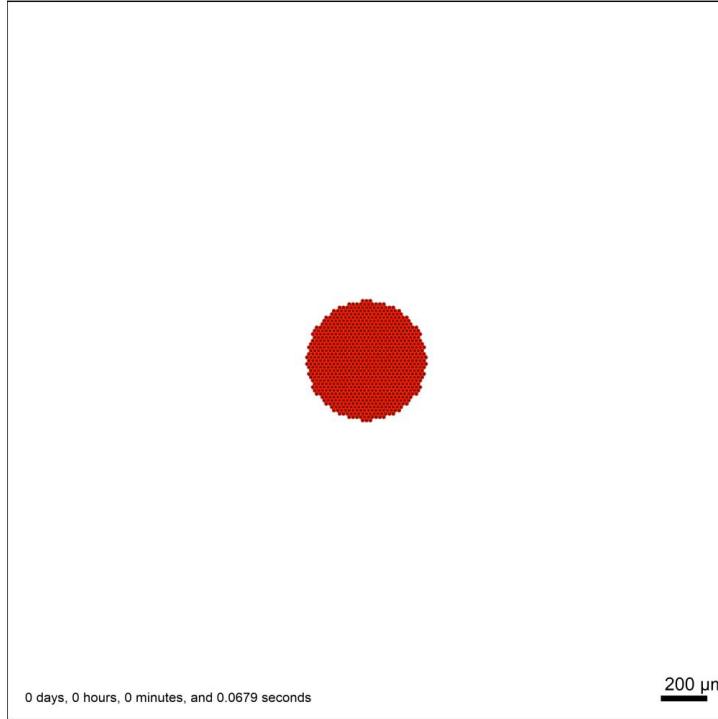
With persistence ($T_p > 0$) – Rows B & C

- Hypoxic cells can continue migrating
- Hypoxic cells "punch through" the oxygenated tumor region

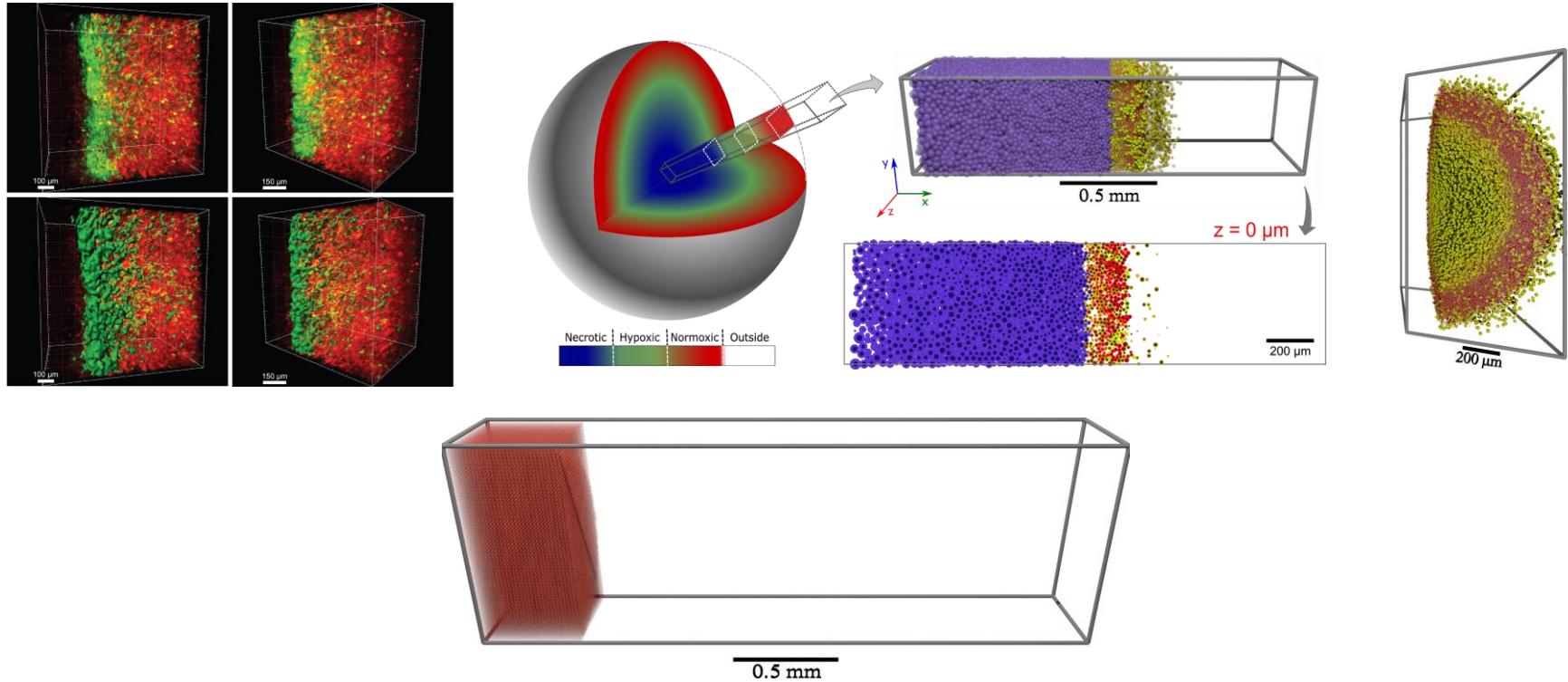


Mathematical model explains biological observations

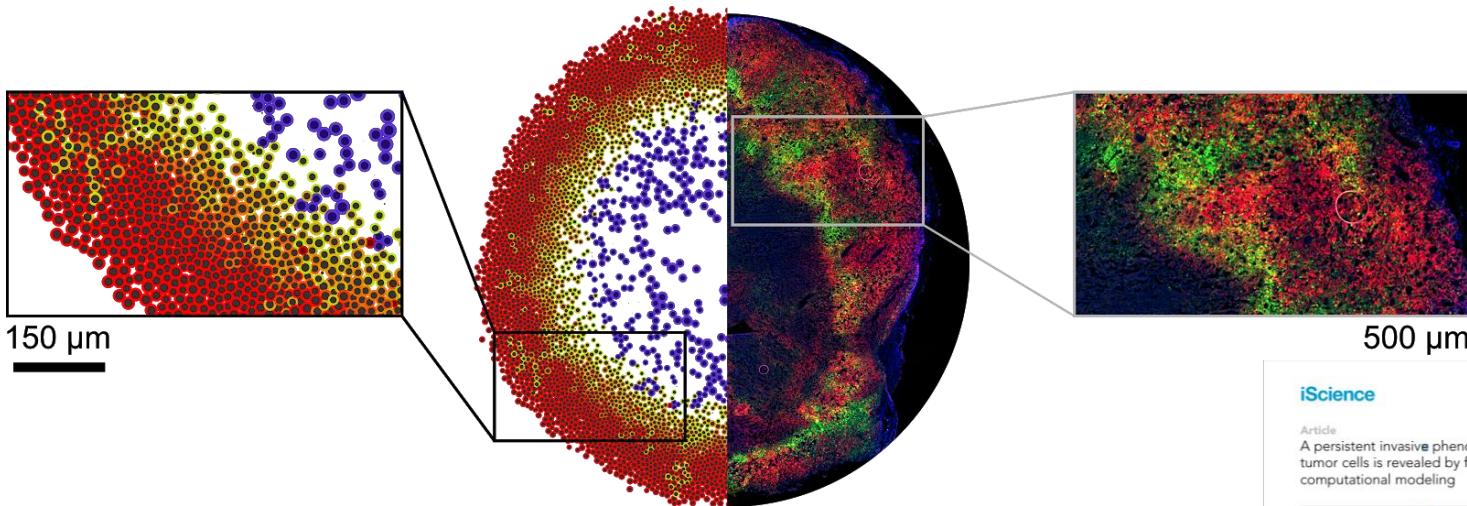
Current time: 0 days, 0 hours, and 0.00 minutes, z = 0.00 μm
889 agents



It also works in 3D



Explore this model



Try this model yourself!
nanohub.org/tools/pc4tumorhypoxia

Rocha et al., *iScience* (2021)
[DOI: 10.1016/j.isci.2021.102935](https://doi.org/10.1016/j.isci.2021.102935)

iScience CellPress OPEN ACCESS

Article
A persistent invasive phenotype in post-hypoxic tumor cells is revealed by fate mapping and computational modeling

Nelson L. Rocha,
Iñaki Goñi,
Jesús M. Martínez,
Sumantra Bhattacharya,
Danielle D. Slatnik,
Paul Macklin

Abstract
Hypoxia fate-mapping reveals that post-hypoxic stem-like tumor cells exhibit a persistent invasive phenotype. This phenotype is driven by a combination of increased migration and reduced apoptosis rates.

Hypoxia Fate-mapping
Post-hypoxic stem-like cells are labeled with DsRed and tracked over time. A zoomed-in view shows a single cell's migration path and its fate.

Post-hypoxic stem-like persistence
Circular plots show the percentage of surviving stem-like cells over time for different hypoxia conditions.

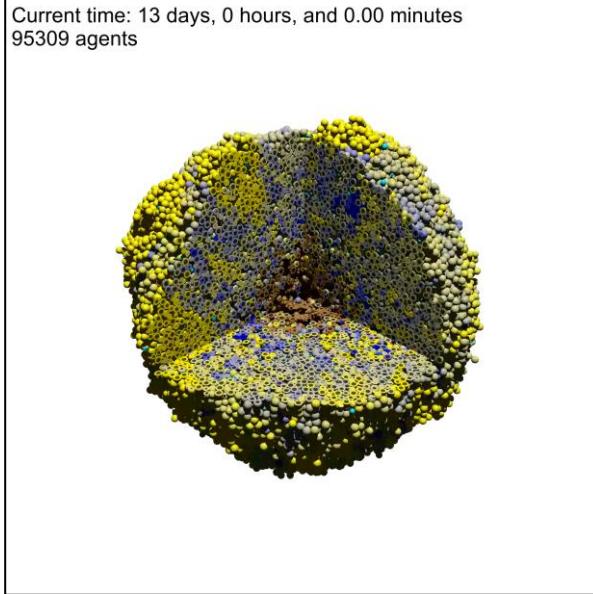
Predicted invasive plume
A heatmap shows the predicted migration of post-hypoxic stem-like cells from a central source.

Observed invasive plume
A heatmap shows the observed migration of post-hypoxic stem-like cells from a central source.

Contributions
Authorship
Funding
Acknowledgments
Competing interests
References
Supplemental information
Article history
Citation
CrossMark

Other examples

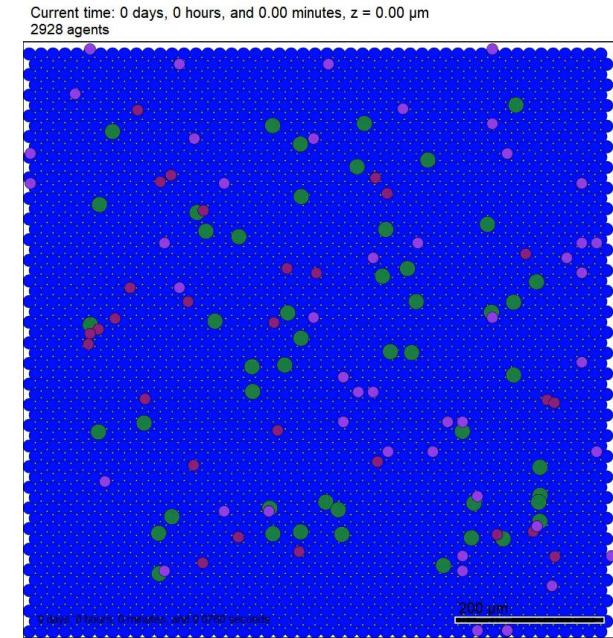
Immune cells attacking a heterogeneous tumor



Micrometastasis growth in liver parenchyma



SARS-CoV-2



Part II: Building a tumor-immune model online

Goal: Build a simple tumor-immune model

- **Tumor:**
 - Wild Type tumor cells:
 - ◆ Proliferate and apoptose at a constant rate
 - » Oxygen increases proliferation*
 - » Pressure reduces proliferation*
 - » Low oxygen increases necrosis*
 - ◆ Secrete a tumor factor
 - ◆ Dead cells release debris
 - ◆ Randomly mutate
 - ◆ Damage increases apoptosis*
 - Mutant tumor cells:
 - ◆ Random migration, slower proliferation
 - ◆ Less secretion of tumor factor
- **Immune:**
 - Macrophages
 - ◆ Attracted to tumor factor and debris
 - ◆ Secrete inflammatory factor
 - ◆ Phagocytose (consume) dead cells
 - CD8+ T cells
 - ◆ Attracted to inflammatory factor
 - ◆ Attack (cause damage to) tumor cells
 - » Faster damage rate to WT tumor cells
 - Monocytes
 - ◆ Attracted to inflammatory factor
 - ◆ Phagocytose (live) tumor cells
 - » Faster rate for WT tumor cells

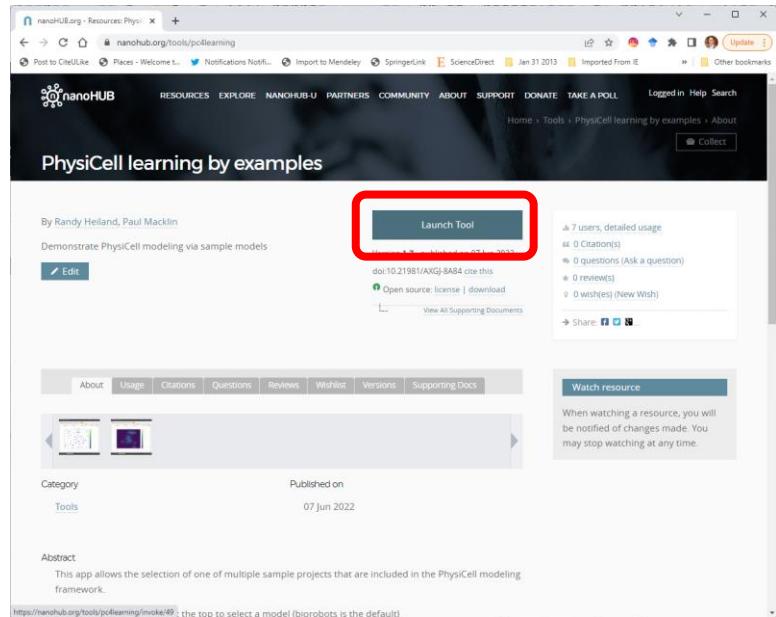
* requires additional C++

Approach

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

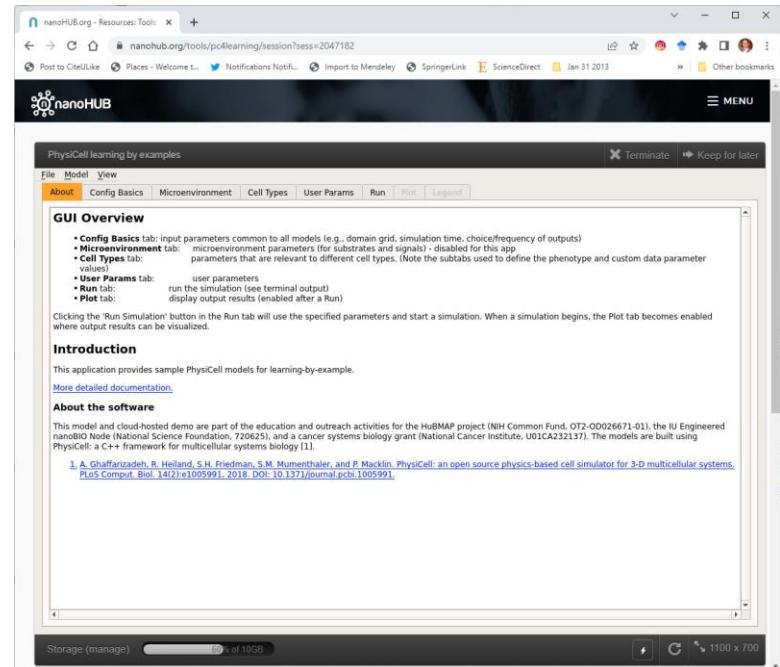
Start the online simulator

- Go to the tool on nanoHUB:
 - <https://nanohub.org/tools/pc4learning>
- Make sure you're logged on.
- Click the “launch tool” button



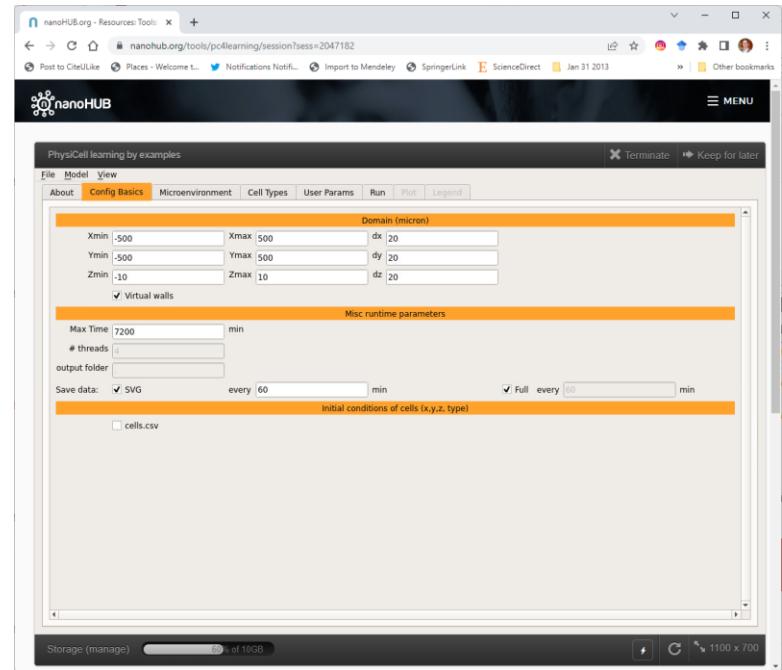
App navigation

- **Overview:**
 - basic overview
- **Config basics:**
 - Domain size, Simulation duration, Data output
- **Microenvironment:**
 - Define diffusing substrates and boundary conditions
- **Cell types:**
 - Define cell types, including their base phenotypes (behaviors)
- **User params:**
 - Model-specific parameters
- **Run:**
 - Start running in the cloud and view (virtual) console output
- **Plot:**
 - Plot the cells and diffusing substrates
- **Legend:**
 - Define the coloring of the plotted cell types



Set up the domain

- Go to the **config basics** tab
- Choose domain settings (in μm)
 - leave $Z_{\min} = -\frac{1}{2}\Delta z$ and $Z_{\max} = \frac{1}{2}\Delta z$ for 2D models
 - Use “virtual walls” to apply a force to keep cells from leaving the domain
- Choose time settings (in min):
 - Max time is the simulation end time
- Choose save settings:
 - SVG: required for plotting cell positions
 - Full: required for plotting diffusing substrates



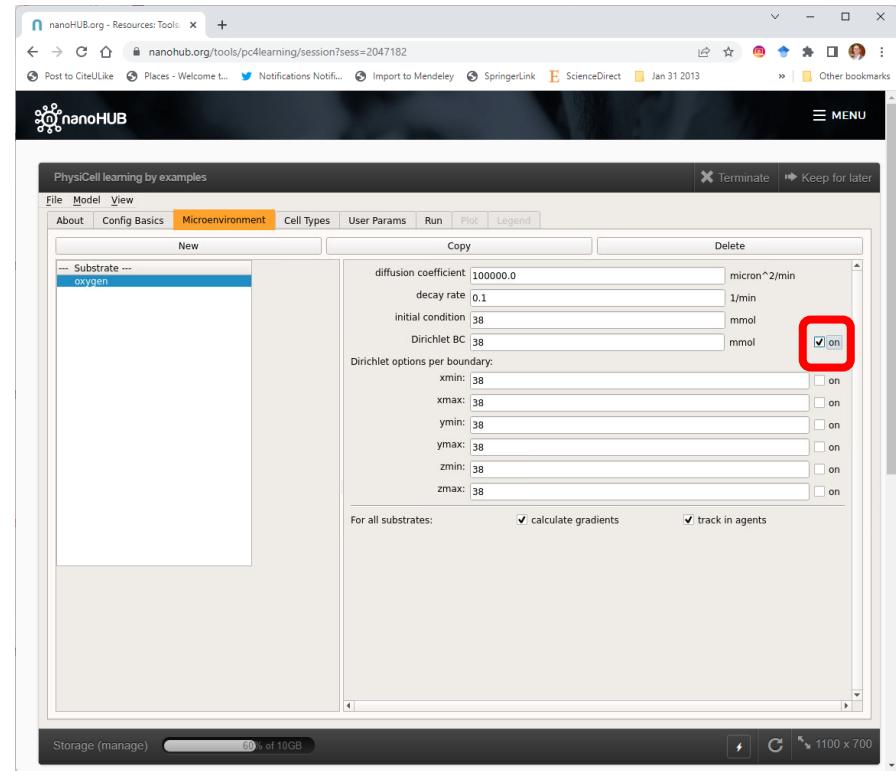
Leave these settings at default values today

Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

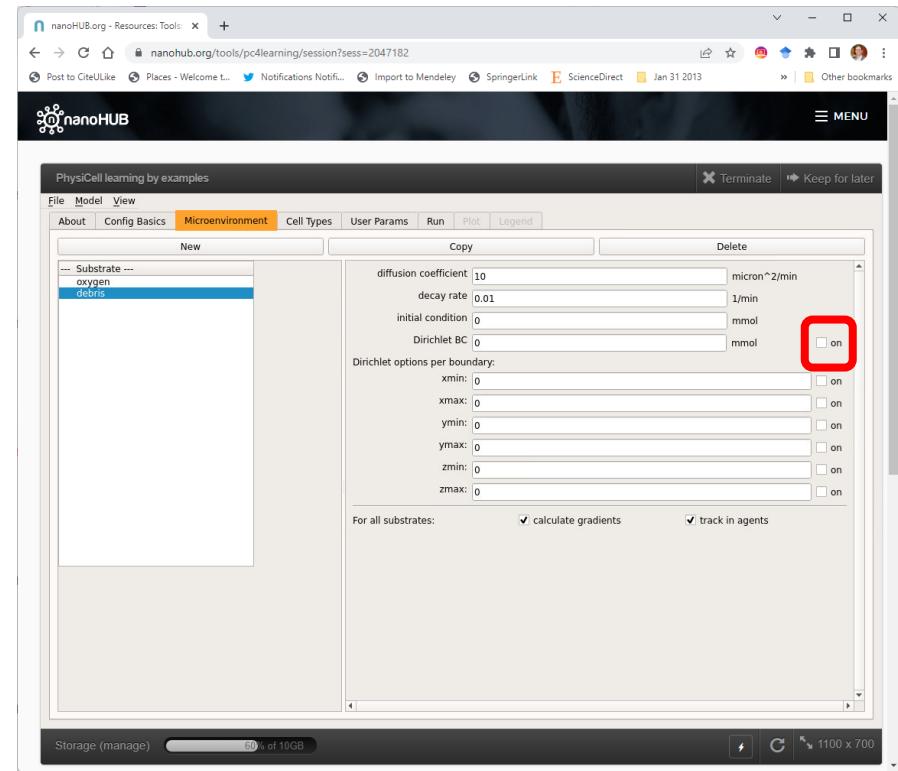
Define substrates (1)

- Go to the **microenvironment** tab
- Double-click **substrate**
 - Rename to **oxygen**
 - Keep diffusion at $100000 \mu\text{m}^2/\text{min}$
 - Set decay at 0.1 min^{-1}
 - ◆ 1 mm diffusion length scale
 - (1 mm length scale)
 - Set Dirichlet boundary conditions to 38 mmHg (5% O₂: physioxic conditions)
 - Set initial condition to 38 mmHg



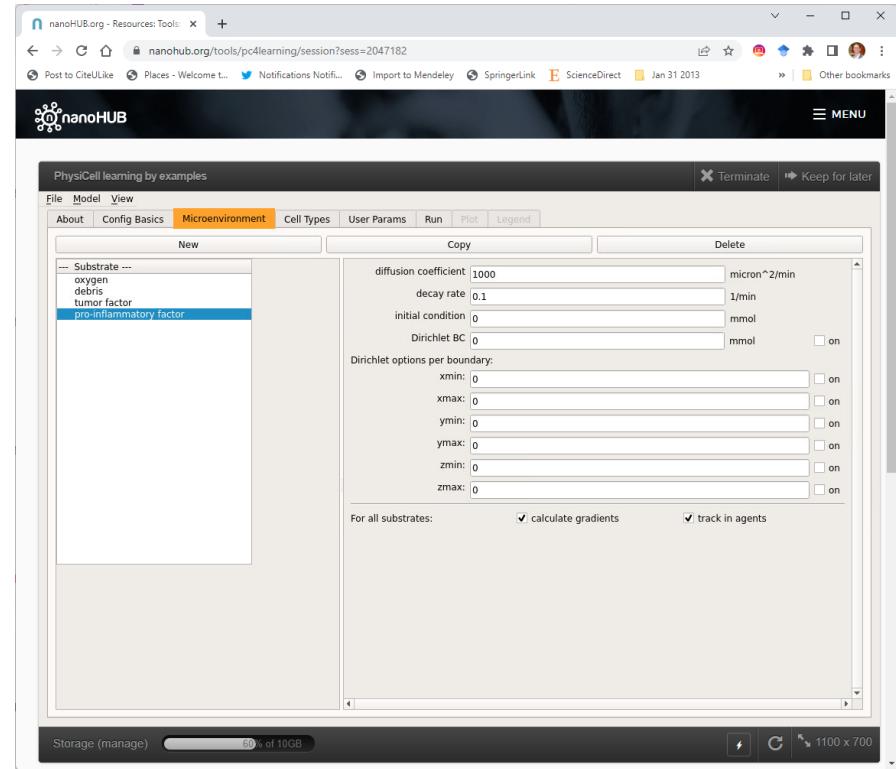
Define substrates (2)

- Make sure **oxygen** is selected
- Click **copy**
- Rename to **debris**
 - Set diffusion at $10 \mu\text{m}^2/\text{min}$
 - Set decay at 0.01 min^{-1}
 - ◆ $100 \mu\text{m}$ diffusion length scale
 - Disable Dirichlet boundary condition
 - Set initial condition to 0



Define substrates (3)

- Make sure **debris** is selected
- Click **copy**
- Rename to **tumor factor**
 - Set diffusion at $1000 \mu\text{m}^2/\text{min}$
 - Set decay at 0.1 min^{-1}
 - ◆ $100 \mu\text{m}$ diffusion length scale
 - Disable Dirichlet boundary condition
 - Set initial condition to 0
- Copy **tumor factor** and repeat to create **pro-inflammatory factor**

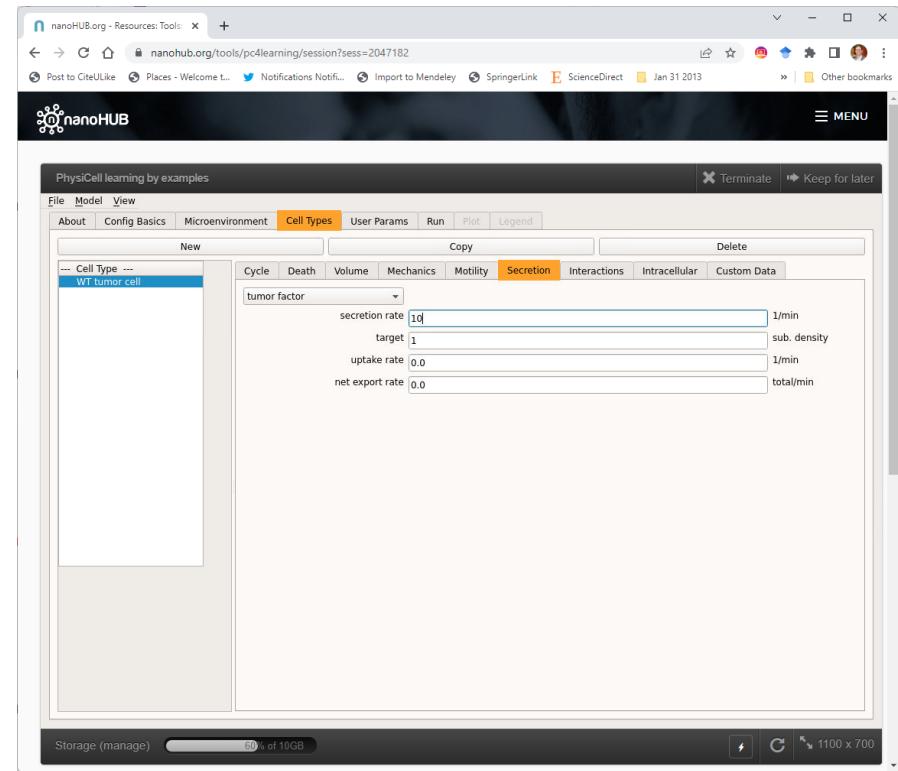


Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

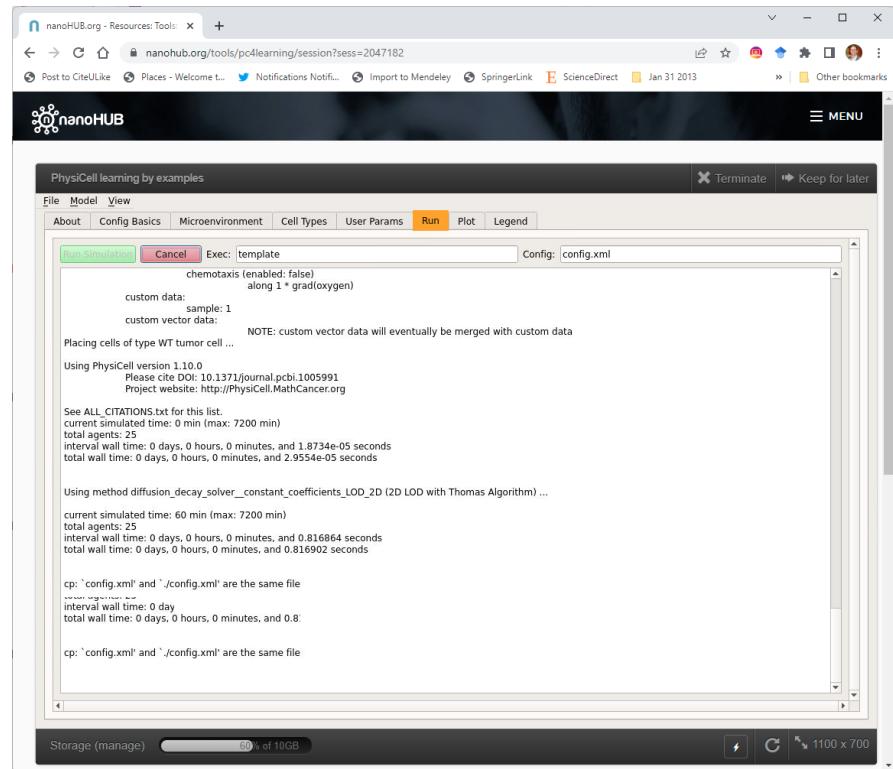
Define WT tumor cells

- Go to the **cell types** tab
- Double-click on **default**
 - Rename to **WT tumor cell**
- Go to **Secretion**
 - Choose **oxygen** from drop-down
 - ◆ Set uptake rate to 10
 - » 100 μm length scale in packed tissue
 - Choose **tumor factor**
 - ◆ Set secretion rate to 10
 - ◆ Set target to 1



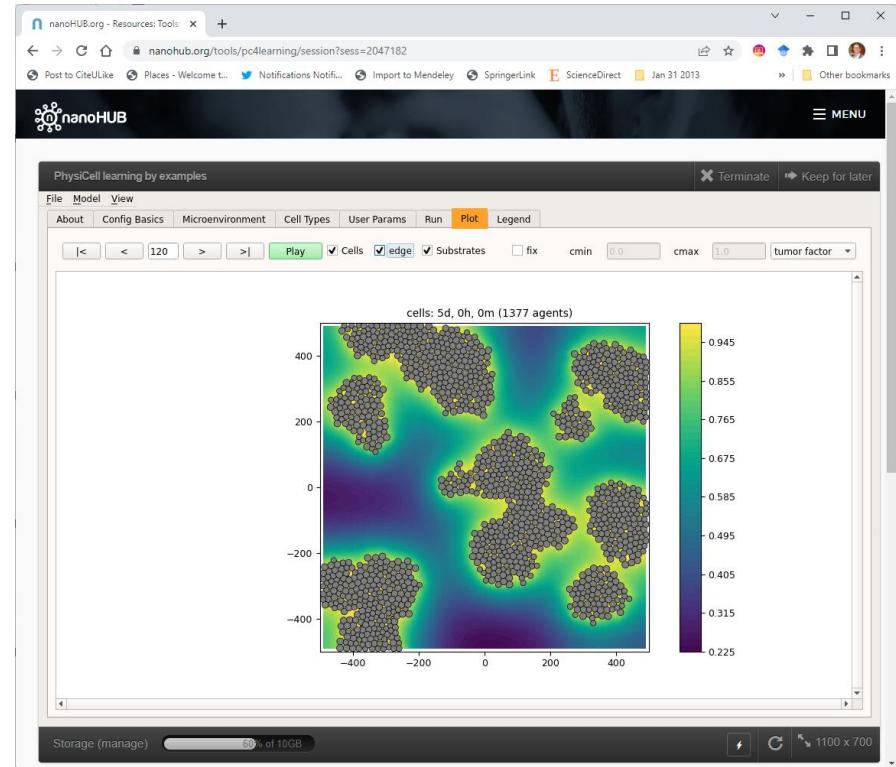
Test the model!

- Go to the **user params** tab
 - Set **number of cells** to 25
- Go to the **run** tab
 - Click **run simulation**



View the results

- Go to the **plot** tab
 - Click **play** to automatically animate
 - Click **pause** to stop playback
 - Click **<** or **>** to advance by 1 frame
 - Click **|< or >|** to go to the start or end
- Check **substrates** to plot substrates below the cells
 - Use the drop-down to choose the substrate

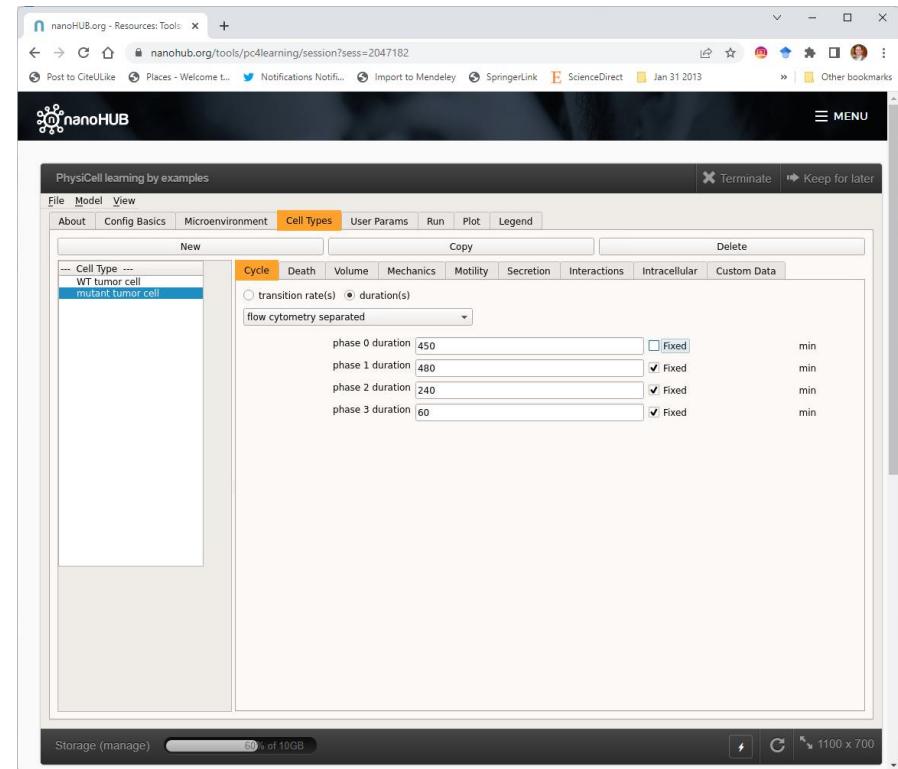


Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

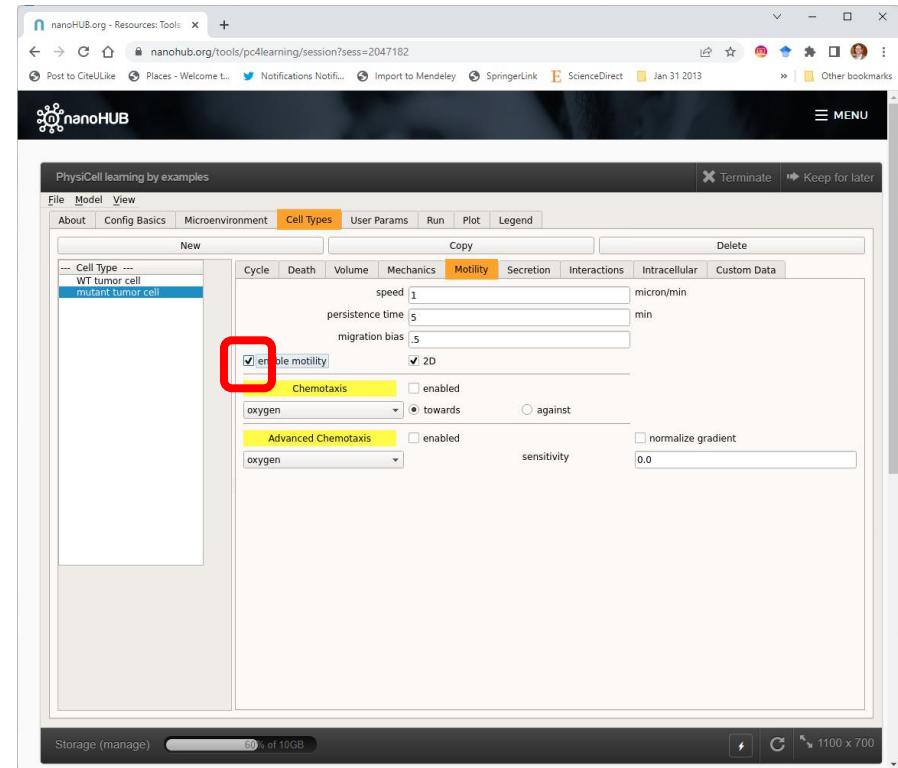
Define mutant tumor cells (1)

- Go to the **cell types** tab
- Click on **WT tumor cell**
- Copy the cell type
 - Rename it to **mutant tumor cell**
- Reduce cycling
 - Click on the **cycle** tab
 - Increase time in G0/G1 to **450** min.



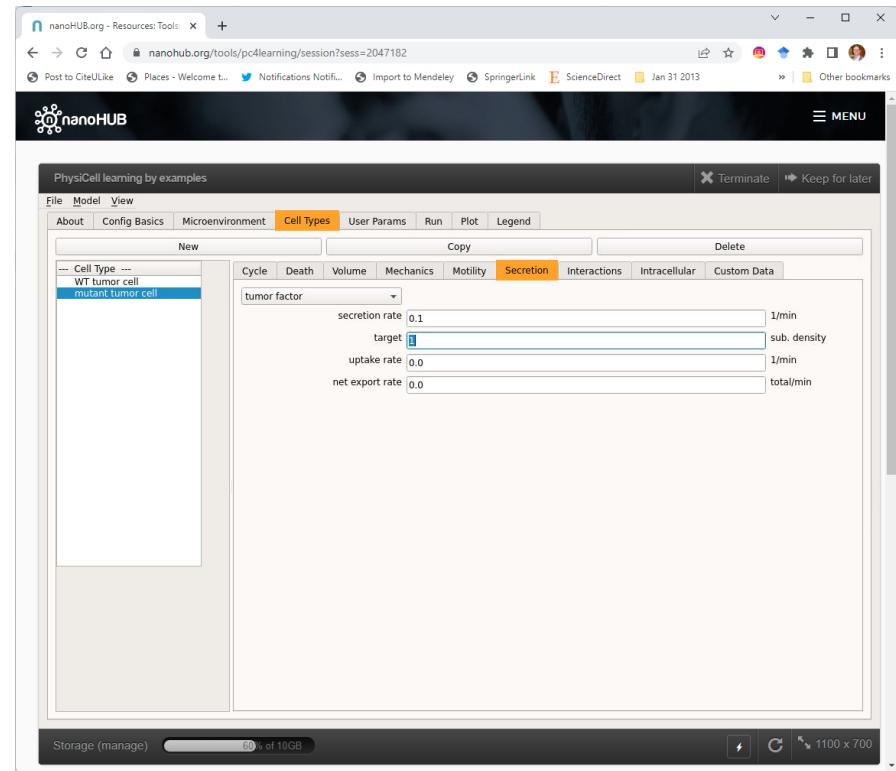
Define mutant tumor cells (2)

- Turn on random motility
 - Click on the **motility** tab
 - Leave mean speed at **1 $\mu\text{m}/\text{min}$**
 - Set persistence time to **5 min**
 - Leave bias at **0.5**
 - ◆ (doesn't affect this result)
 - Make sure to click the checkbox to enable motility!



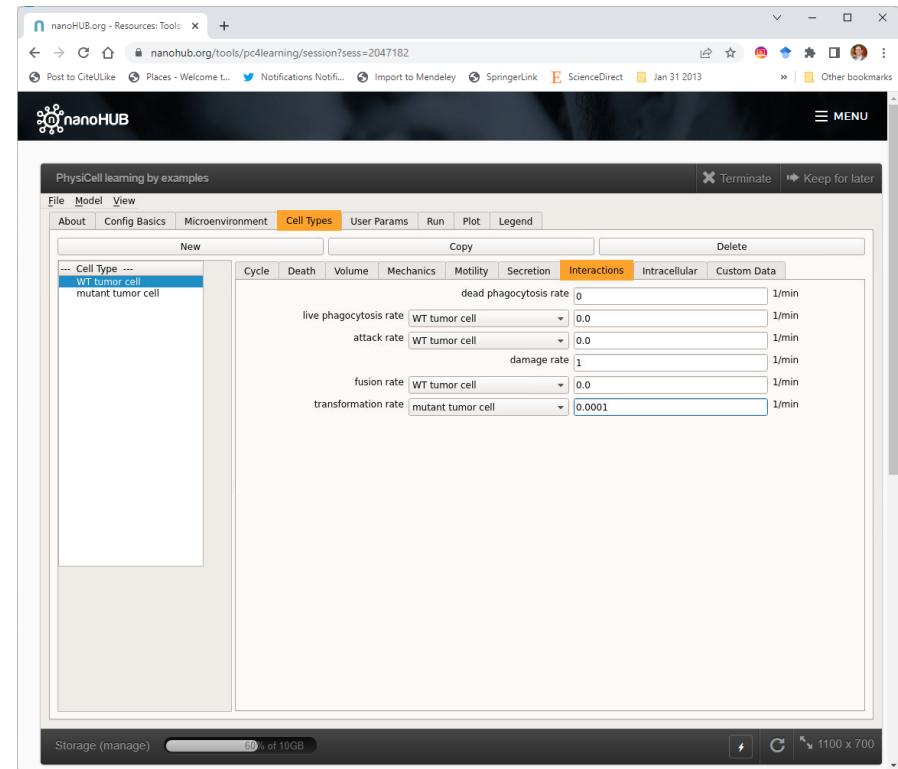
Define mutant tumor cells (3)

- Reduce tumor factor secretion
 - Click on the **secretion** tab
 - Choose **tumor factor** from the drop-down menu
 - Set the value to **0.1**



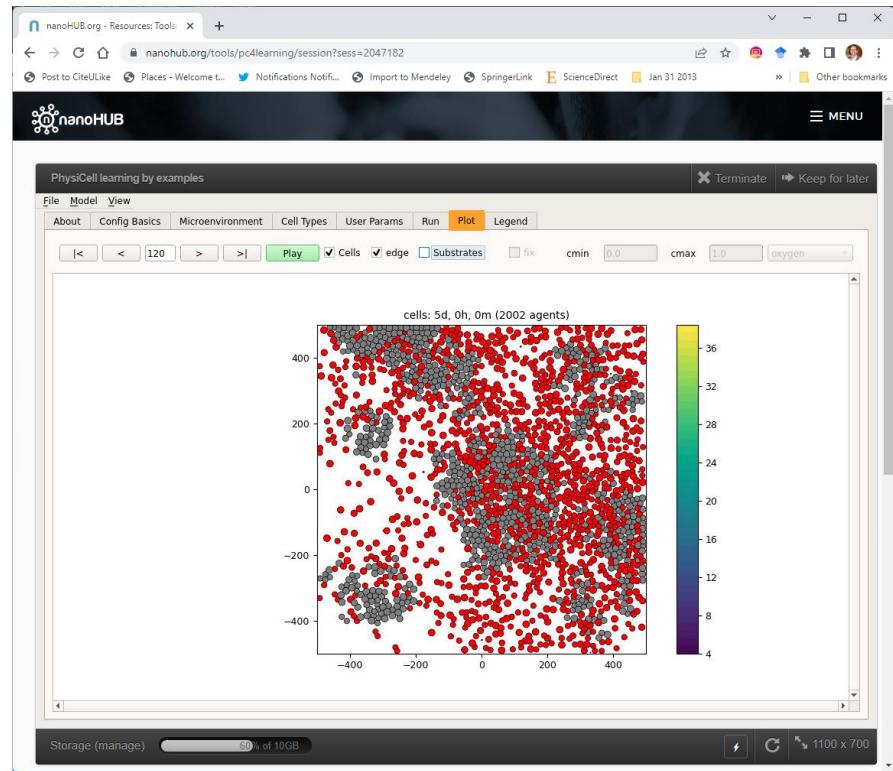
Enable mutations

- Select the **WT tumor cell** type
- Go to the **interactions** tab
- Go to **transformation rate**
- Choose **mutant tumor cell** from the drop-down
- Set the rate to **0.0001**



Test the model!

- Go to the **run** tab
 - Click **run simulation**
- Visualize the results in the **plot** tab
 - Grey: WT tumor cells
 - Red: mutant tumor cells
 - See the **legend** tab to verify colors
- Notice:
 - Red cells are motile
 - Red cells proliferate more slowly
 - Grey cells turn to red cells

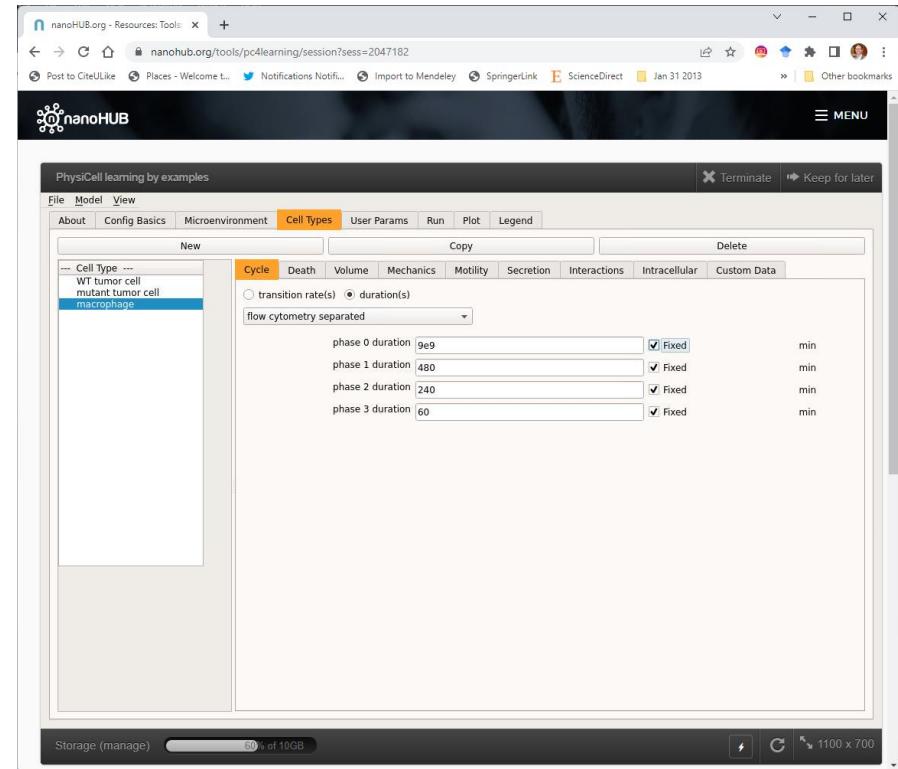


Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

Define macrophages (1)

- Go to the **cell types** tab
- Click on **mutant tumor cell**
- Copy the cell type
 - Rename it to **macrophage**
- Disable cycling
 - Click on the **cycle** tab
 - Increase time in G0/G1 to **9e9** min with **fixed duration**



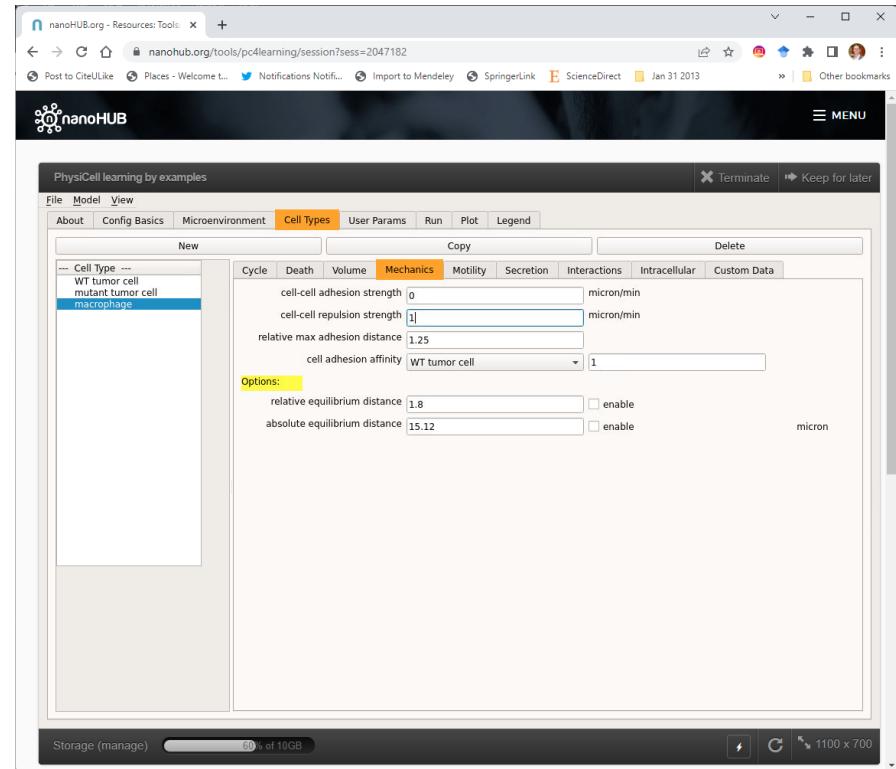
Define macrophages (2)

- Disable apoptotic death
 - Click on the **death** tab
 - Go to the section on **apoptosis**
 - Set the **death rate** to 0

The screenshot shows the PhysiCell learning by examples software interface on nanoHUB.org. The 'Death' tab is selected under the 'Cell Types' tab. The 'Apoptosis' section is active, showing parameters for unlysed fluid change rate (0.05), lysed fluid change rate (0), cytoplasmic biomass change rate (1.66667e-02), nuclear biomass change rate (5.83333e-03), calcification rate (0), and relative rupture volume (2.0). The 'Necrosis' section is also visible. The 'macrophage' cell type is selected in the sidebar.

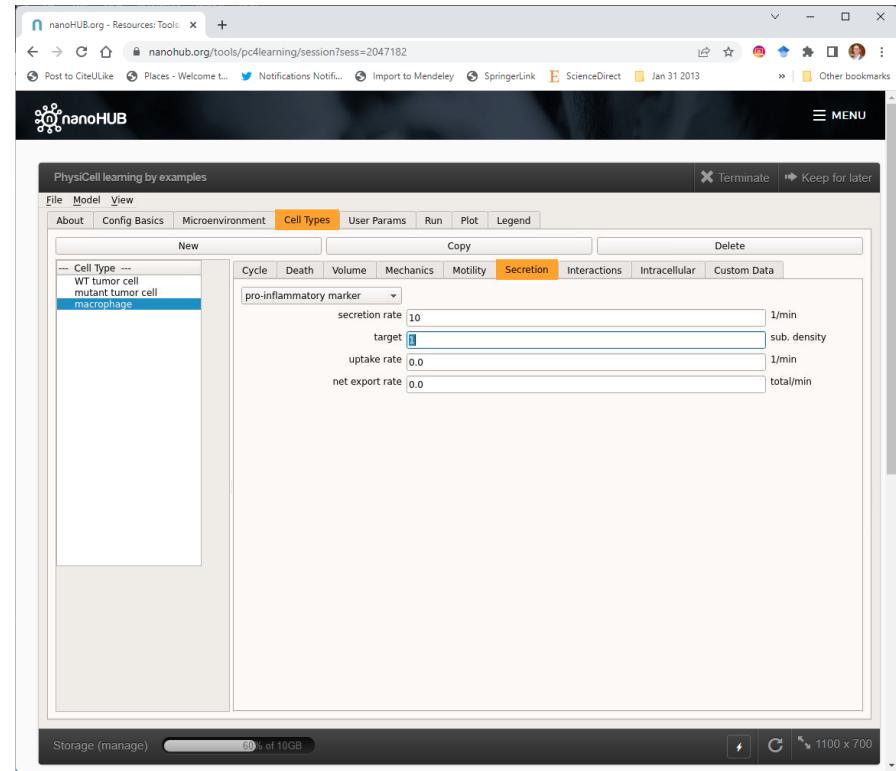
Define macrophages (3)

- Disable adhesion
 - Click on the **mechanics** tab
 - Set **cell-cell adhesion** to **0**
- Make the cells more deformable (allow more overlap)
 - Click on the **mechanics** tab
 - Set **cell-cell repulsion** to **1**



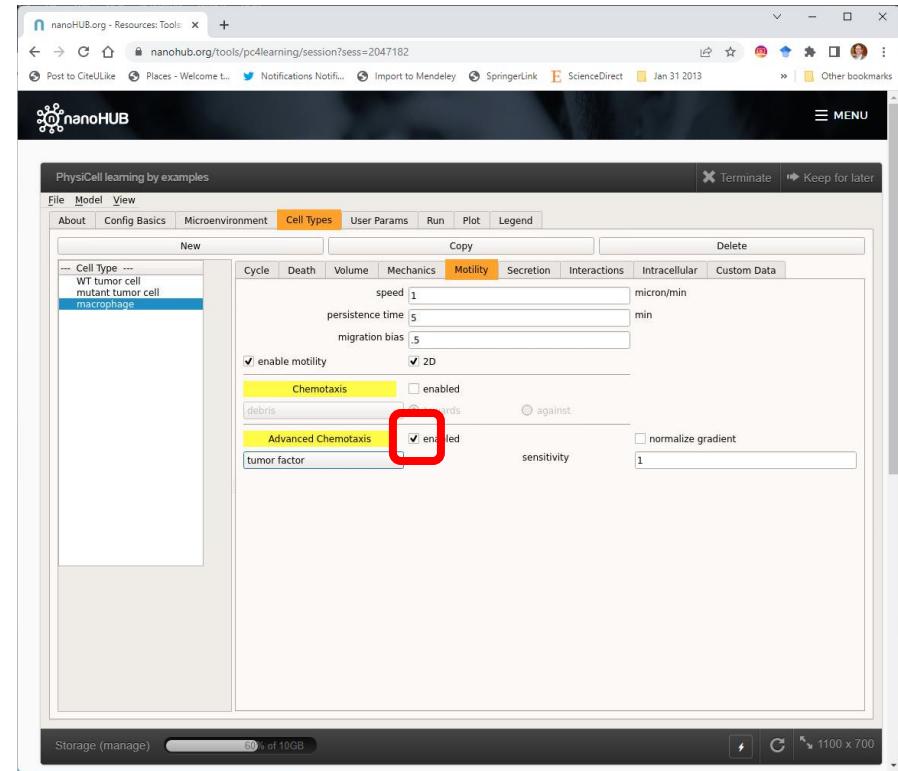
Define macrophages (3)

- Disable secretion of tumor factor
 - Go to the **secretion** tab
 - Choose **tumor factor** from the drop-down
 - Set the **secretion rate** to **0**
- Enable secretion of pro-inflammatory factor
 - Go to the **secretion** tab
 - Choose **pro-inflammatory factor** from the drop-down
 - Set the **secretion rate** to **10**
 - Set the **target** to **1**



Define macrophages (4)

- Turn on chemotaxis up both debris and tumor factor
 - Click on **motility**
 - Set **migration bias** at **0.5**
 - Make sure motility is enabled with the checkbox
 - Go to **advanced chemotaxis**
 - Choose **debris** in the drop-down and set its sensitivity to **0.1**
 - Choose **tumor factor** in the drop-down and set its sensitivity to **1**
 - Make sure you click the checkbox to enable advanced motility



LUDDY

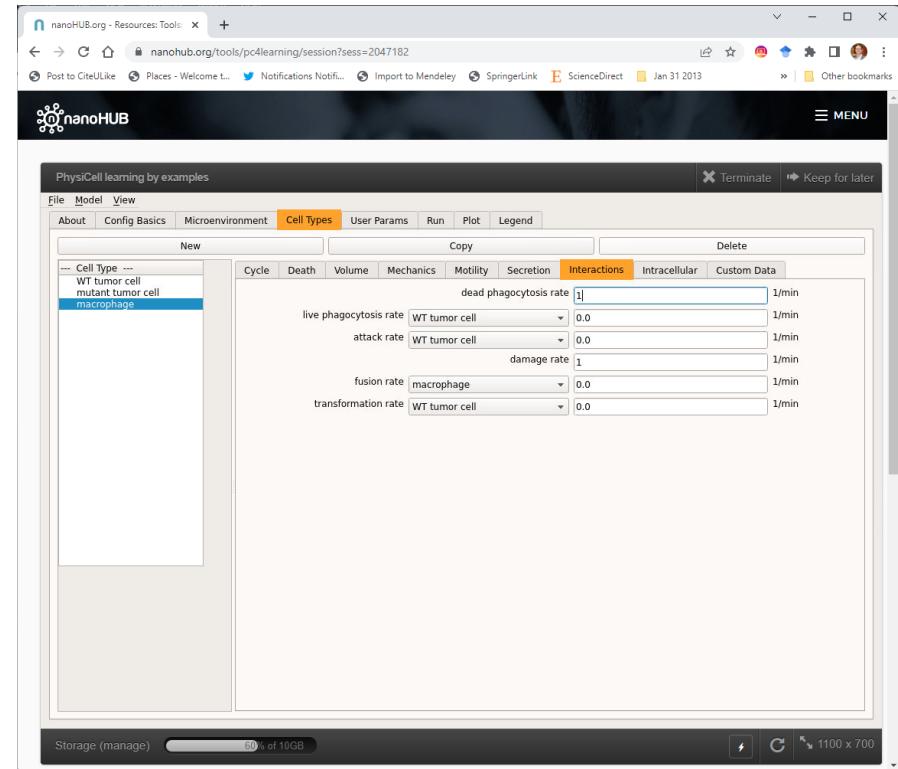
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Macklin Lab
@MathCancer
MathCancer.org

Define macrophages (5)

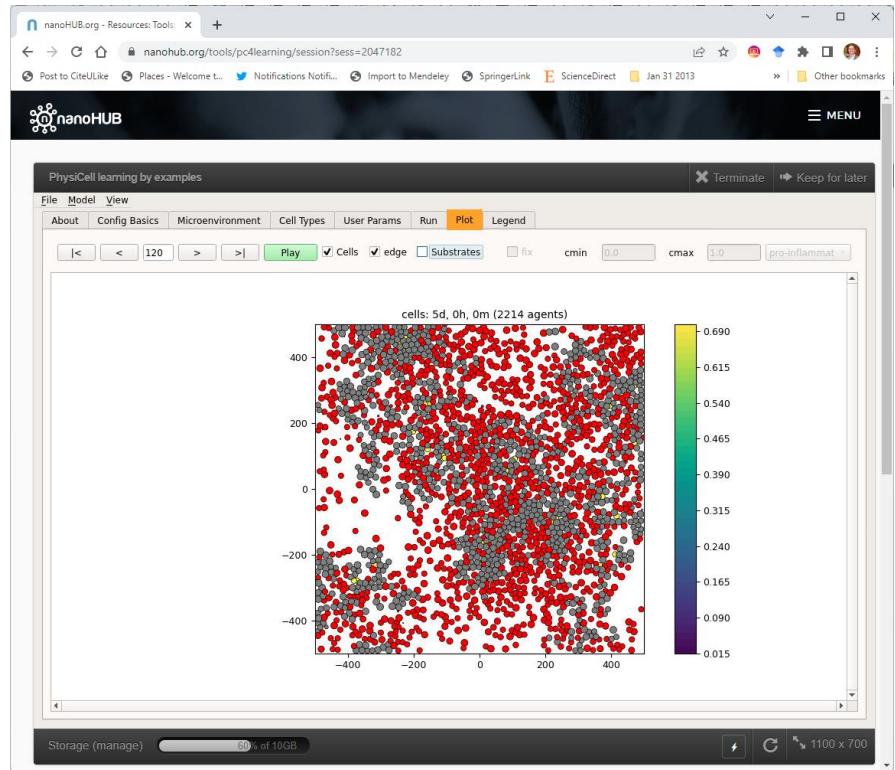
- Enable phagocytosis of dead cells
 - Click on **interactions**
 - Set the **dead phagocytosis rate** to 1

Note: If the dead cell phagocytosis rate is r , then the mean time the macrophage spends in contact with a dead cell before phagocytosing it is $\frac{1}{r}$.



Test the model!

- Go to the **run** tab
 - Click **run simulation**
- Visualize the results in the **plot** tab
 - Grey: WT tumor cells
 - Red: mutant tumor cells
 - Yellow: macrophages
- Notice:
 - Yellow cells will co-localize with tumor cells, with a preference for grey.

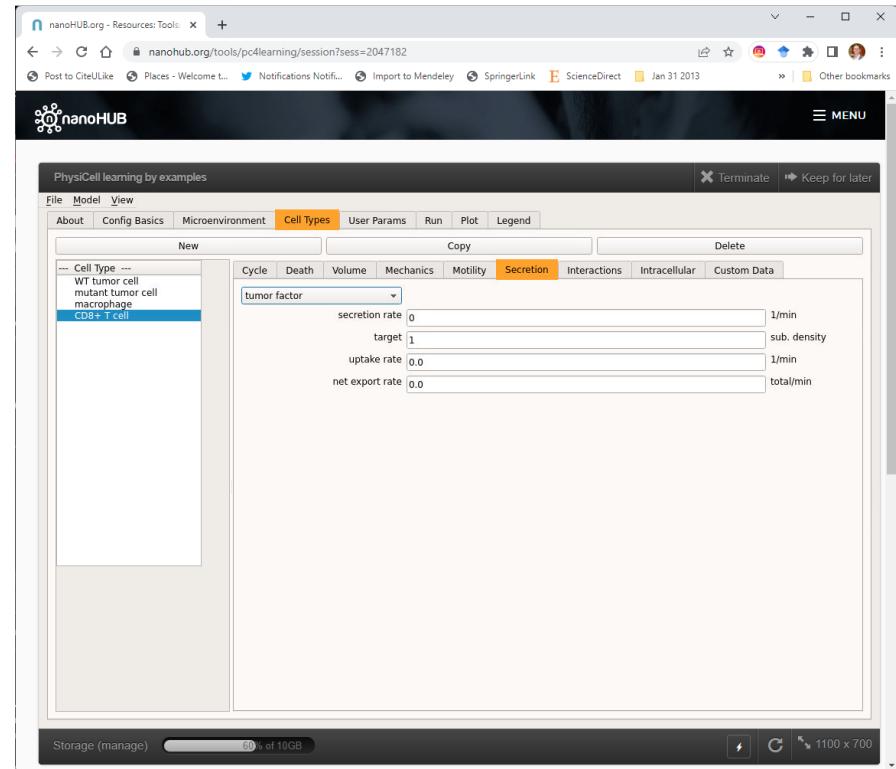


Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

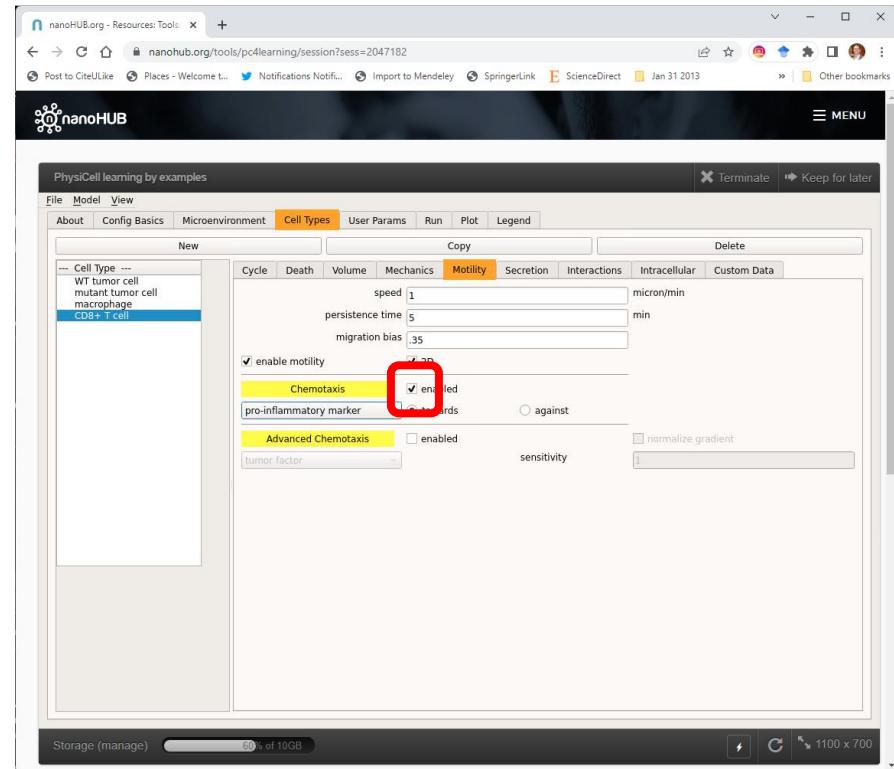
Define CD8+ T cells (1)

- Go to the **cell types** tab
- Click on **macrophage**
- Copy the cell type
 - Rename it to **CD8+ T cell**
- Disable secretion
 - Click on the **secretion** tab
 - Choose **pro-inflammatory factor** from the drop-down menu
 - Set the **secretion rate** to **0**



Define CD8+ T cells (2)

- Change to basic chemotaxis towards pro-inflammatory factor
 - Click on the **motility** tab
 - Uncheck to disable advanced chemotaxis
 - Check to enable chemotaxis
 - Set **migration bias** at **0.35**
 - Select **pro-inflammatory factor** from the drop-down menu
 - Make sure **towards** is selected for migration up the gradient



Define CD8+ T cells (3)

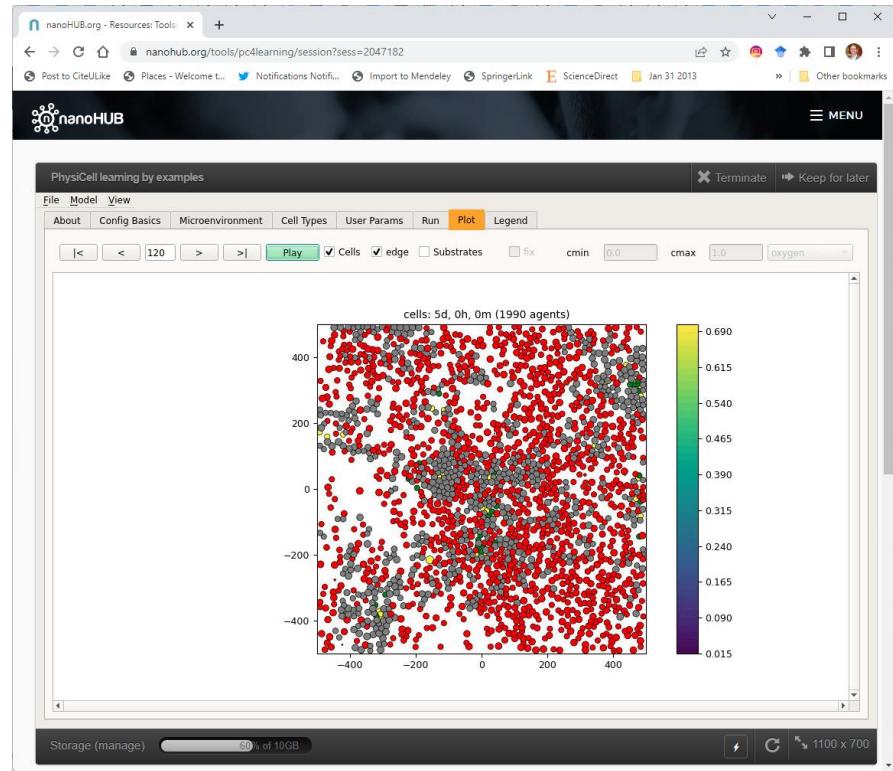
- Disable dead cell phagocytosis
 - Click on the **interactions** tab
 - Set **dead cell phagocytosis** to **0**
- Enable attack of tumor cells
 - Click on the **interactions** tab
 - Go to **attack rate**
 - ◆ Choose **WT tumor cell**
 - ◆ Set **attack rate to 10**
 - Use a reduced attack rate for mutants
 - ◆ Choose **mutant tumor cell**
 - ◆ Set **attack rate to 0.1**

The screenshot shows the PhysiCell learning by examples interface on nanoHUB.org. The 'Interactions' tab is active. The 'Cell Types' panel lists 'WT tumor cell', 'mutant tumor cell', 'macrophage', and 'CD8+ T cell', with 'CD8+ T cell' currently selected. The main panel displays interaction parameters:

Parameter	Type	Value	Unit
dead phagocytosis rate	WT tumor cell	0	1/min
live phagocytosis rate	WT tumor cell	0.0	1/min
attack rate	WT tumor cell	10	1/min
attack rate	mutant tumor cell	0.1	1/min
damage rate		1	1/min
fusion rate	macrophage	0.0	1/min
transformation rate	WT tumor cell	0.0	1/min

Test the model!

- Go to the **run** tab
 - Click **run simulation**
- Visualize the results in the **plot** tab
 - Grey: WT tumor cells
 - Red: mutant tumor cells
 - Yellow: macrophages
 - Green: CD8+ T cells
- Notice:
 - Green cells will co-localize with macrophages
 - No extra cell death because we need a C++ function to tie death to damage.



Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

Define monocytes (1)

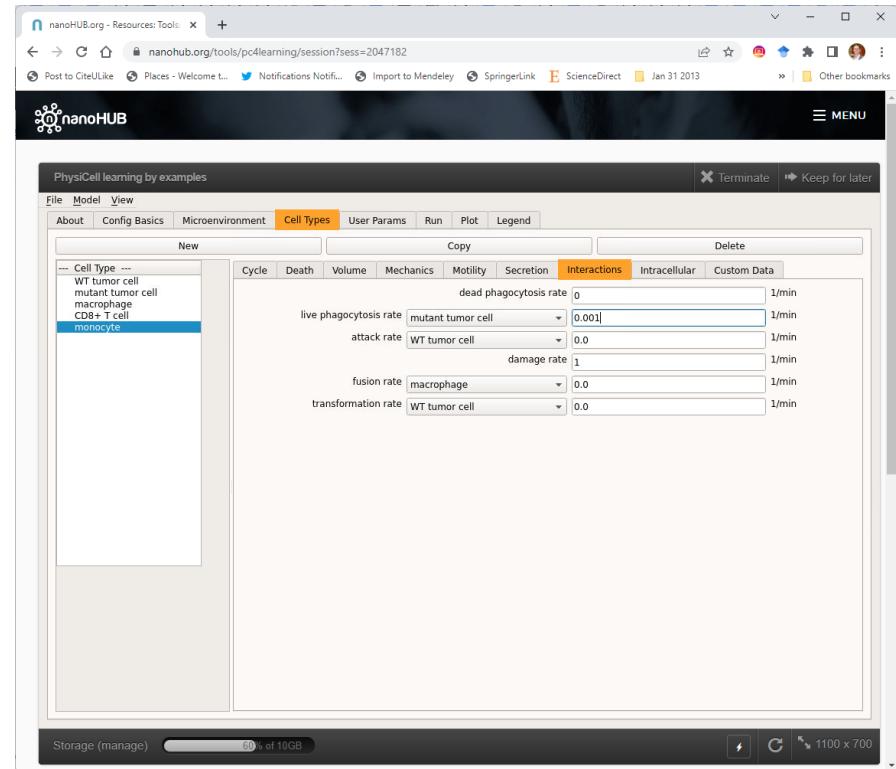
- Go to the **cell types** tab
- Click on **CD8+ T cell**
- Copy the cell type
 - Rename it to **monocyte**
- Disable attack of tumor cells
 - Click on the **interactions** tab
 - Go to **attack rate**
 - ◆ Choose **WT tumor cell**
 - ◆ Set **attack rate** to **0**
 - Use a reduced attack rate for mutants
 - ◆ Choose **mutant tumor cell**
 - ◆ Set **attack rate** to **0**

The screenshot shows the PhysiCell learning by examples interface on nanoHUB.org. The 'Cell Types' tab is selected. In the sidebar, under 'Cell Type', 'monocyte' is highlighted. The 'Interactions' tab is active, displaying various interaction parameters:

Parameter	Value	Unit	
dead phagocytosis rate	0	1/min	
live phagocytosis rate	WT tumor cell	0.0	1/min
attack rate	WT tumor cell	0.0	1/min
damage rate	1	1/min	
fusion rate	macrophage	0.0	1/min
transformation rate	WT tumor cell	0.0	1/min

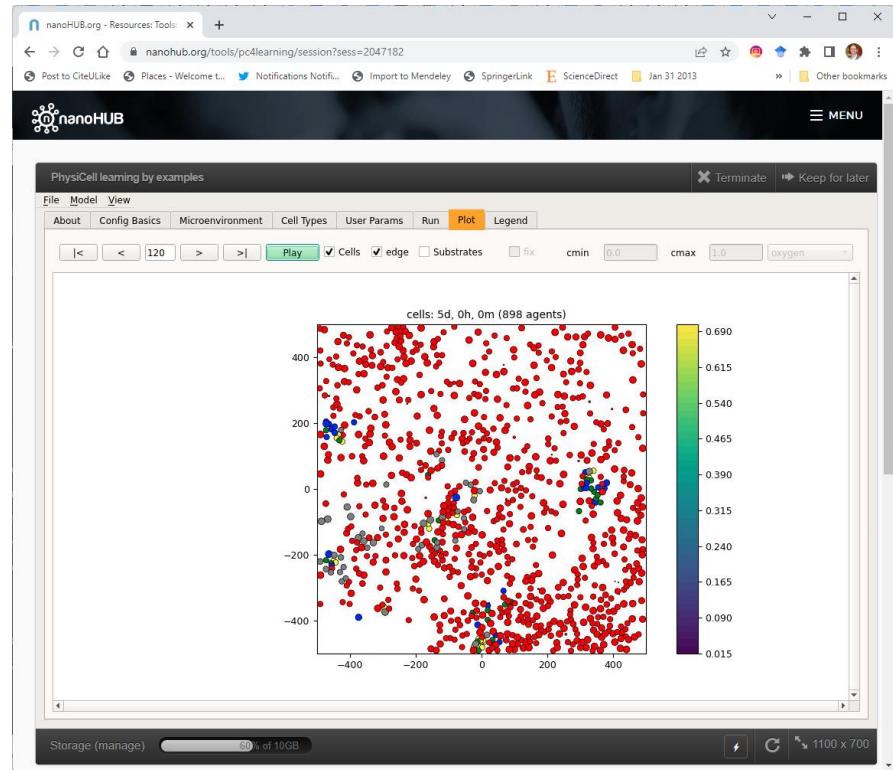
Define monocytes (2)

- Enable phagocytosis of tumor cells
 - Click on the **interactions** tab
 - Go to **live phagocytosis rate**
 - ◆ Choose **WT tumor cell**
 - ◆ Set **live phagocytosis rate to 0.005**
 - Use a reduced rate for mutants
 - ◆ Choose **mutant tumor cell**
 - ◆ Set **live phagocytosis rate to 0.001**



Test the model!

- Go to the **run** tab
 - Click **run simulation**
- Visualize the results in the **plot** tab
 - Grey: WT tumor cells
 - Red: mutant tumor cells
 - Yellow: macrophages
 - Green: CD8+ T cells
 - Blue: monocytes
- Notice:
 - Blue cells preferentially eat grey cells
 - Blue cells get larger as they absorb the cells until they digest them.



Checklist

- Use online simulator: <http://nanohub.org/tools/pc4learning>
- Build iteratively:
 - Set the domain
 - Add diffusing substrates
 - Add WT tumor cells and test
 - Add mutant tumor cells and test
 - Add macrophages and test
 - Add CD8+ T cells and test
 - Add monocytes and test

Part III:

Adding dynamical phenotypes with C++

Adding more advanced dynamics

- We are missing a few elements of our model:
 - Tumor cells:
 - ◆ Oxygen promotes cycling
 - ◆ High pressure reduces cycling
 - ◆ Damage increases apoptosis
 - ◆ Low oxygen induces necrosis
 - ◆ Dead cells release debris
- We'll use a C++ phenotype function to accomplish this

Note: We won't have time for this today, but these slides give the step-by-step details. See the full working code in the GitHub repository:

<https://GitHub.com/PhysiCell-Training/UCI-sysbio-2022>

Approach

Create a template project in PhysiCell

- See tutorial:
 - ◆ <https://github.com/physicell-training/ws2021#session-1-working-with-projects-in-physicell>
- make template && make

Save the config file from our online version

Declare a custom `tumor_phenotype` function in `./custom_modules/custom.h`

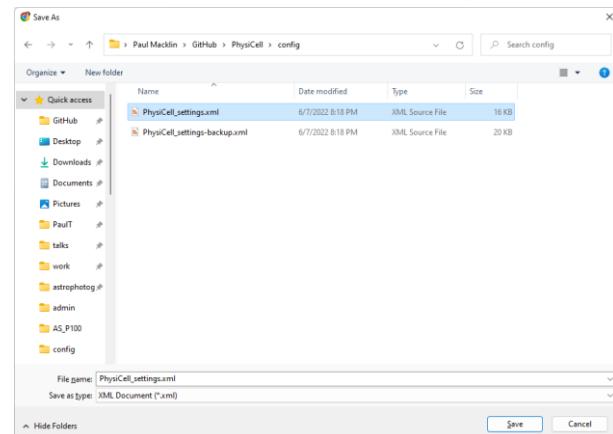
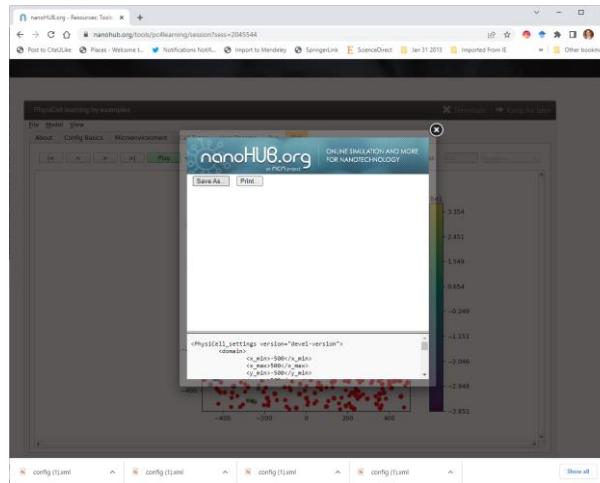
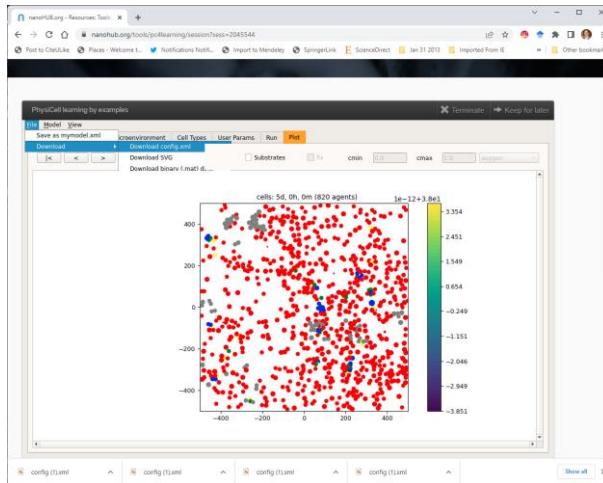
Implement this custom model in `./custom_modules/custom.cpp`

Assign the custom phenotype function to WT tumor cells and mutant tumor cells

Compile and run

Save your config file

- File → Download → Download config.xml
- Choose Save as ... in the nanoHUB dialog.
- Overwrite PhysiCell_settings.xml in ./config in your PhysiCell repository



Checklist

Create a template project in PhysiCell

- See tutorial:
 - ◆ <https://github.com/physicell-training/ws2021#session-1-working-with-projects-in-physicell>
- make template && make

Save the config file from our online version

Declare a custom `tumor_phenotype` function in `./custom_modules/custom.h`

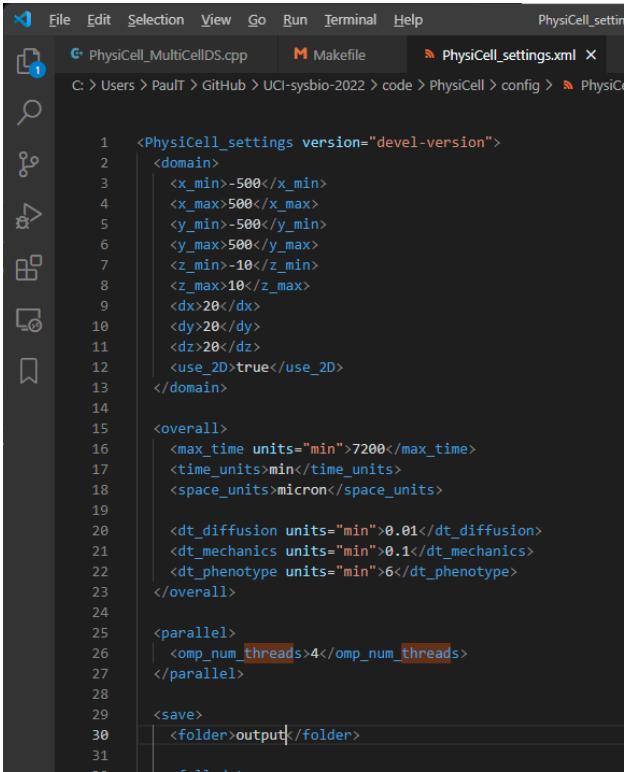
Implement this custom model in `./custom_modules/custom.cpp`

Assign the custom phenotype function to WT tumor cells and mutant tumor cells

Compile and run

Choose a better output directory

- Open **PhysiCell_settings.xml** in **./config**
- Search for **folder**
- Set the value to **output**
 - (This is between the **<output>** starting tag and the **</output>** ending tag.)

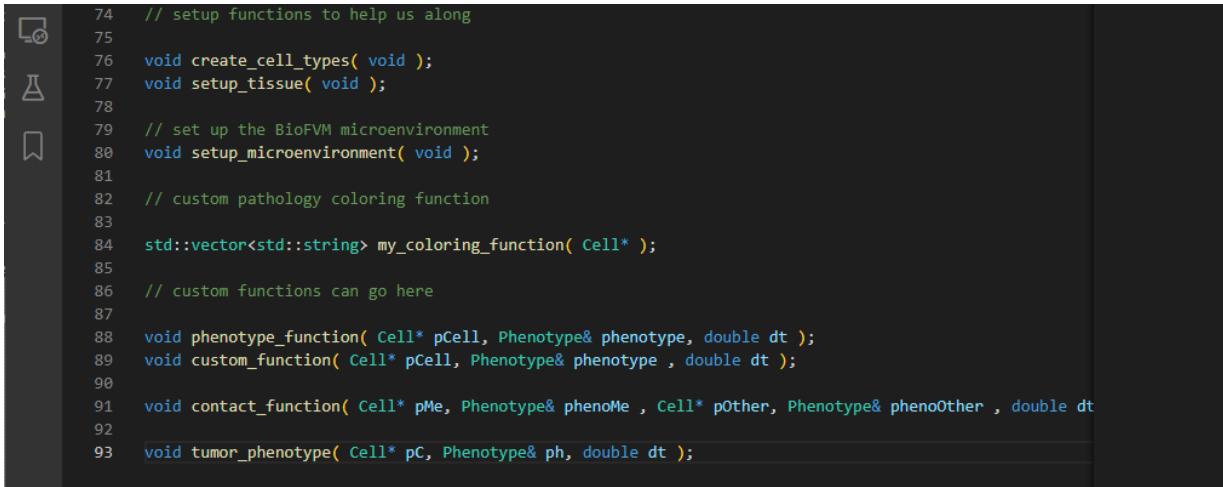


```
1  <PhysiCell_settings version="devel-version">
2    <domain>
3      <x_min>-500</x_min>
4      <x_max>500</x_max>
5      <y_min>-500</y_min>
6      <y_max>500</y_max>
7      <z_min>-10</z_min>
8      <z_max>10</z_max>
9      <dx>20</dx>
10     <dy>20</dy>
11     <dz>20</dz>
12     <use_2D>true</use_2D>
13   </domain>
14
15   <overall>
16     <max_time units="min">7200</max_time>
17     <time_units>min</time_units>
18     <space_units>micron</space_units>
19
20     <dt_diffusion units="min">0.01</dt_diffusion>
21     <dt_mechanics units="min">0.1</dt_mechanics>
22     <dt_phenotype units="min">6</dt_phenotype>
23   </overall>
24
25   <parallel>
26     <omp_num_threads>4</omp_num_threads>
27   </parallel>
28
29   <save>
30     <folder>output</folder>
31   </save>
32
33 </PhysiCell_settings>
```

Declare a tumor phenotype function

- Open **./custom_modules/custom.h**
- Declare a function (at the bottom)

```
void tumor_phenotype( Cell* pC, Phenotype& ph, double dt );
```



```
74 // setup functions to help us along
75
76 void create_cell_types( void );
77 void setup_tissue( void );
78
79 // set up the BioFVM microenvironment
80 void setup_microenvironment( void );
81
82 // custom pathology coloring function
83
84 std::vector<std::string> my_coloring_function( Cell* );
85
86 // custom functions can go here
87
88 void phenotype_function( Cell* pCell, Phenotype& phenotype, double dt );
89 void custom_function( Cell* pCell, Phenotype& phenotype , double dt );
90
91 void contact_function( Cell* pMe, Phenotype& phenoMe , Cell* pOther, Phenotype& phenoOther , double dt
92
93 void tumor_phenotype( Cell* pC, Phenotype& ph, double dt );
```

Checklist

Create a template project in PhysiCell

- See tutorial:
 - ◆ <https://github.com/physicell-training/ws2021#session-1-working-with-projects-in-physicell>
- make template && make

Save the config file from our online version

Declare a custom `tumor_phenotype` function in `./custom_modules/custom.h`

Implement this custom model in `./custom_modules/custom.cpp`

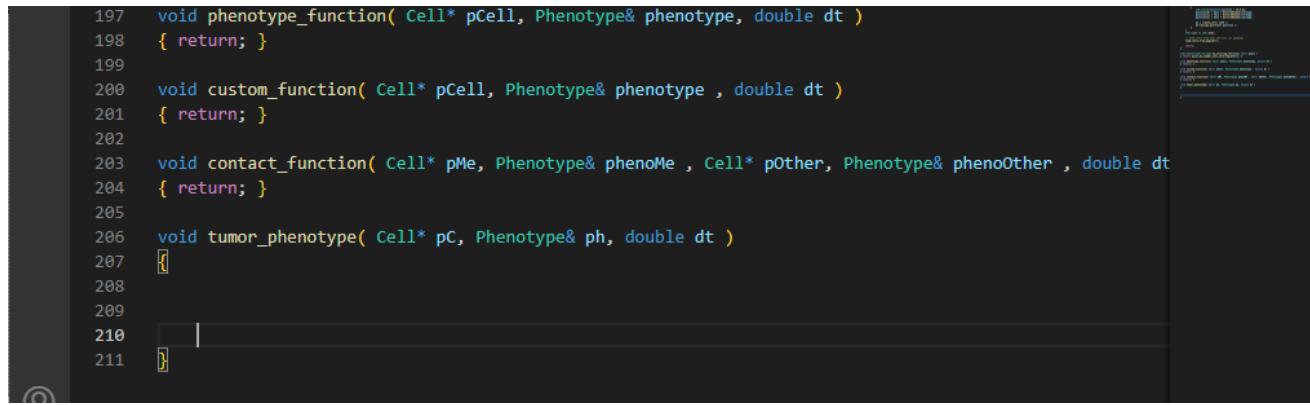
Assign the custom phenotype function to WT tumor cells and mutant tumor cells

Compile and run

Implement a tumor phenotype function (1)

- Open `./custom_modules/custom.cpp`
- Start the function:

```
void tumor_phenotype( Cell* pC, Phenotype& ph, double dt )
{ return; }
```



A screenshot of a code editor showing a C++ file. The code defines several functions: phenotype_function, custom_function, contact_function, and tumor_phenotype. The tumor_phenotype function is partially implemented, with the opening brace and the start of the function body visible. The code editor has a dark theme with syntax highlighting.

```
197 void phenotype_function( Cell* pCell, Phenotype& phenotype, double dt )
198 { return; }
199
200 void custom_function( Cell* pCell, Phenotype& phenotype , double dt )
201 { return; }
202
203 void contact_function( Cell* pMe, Phenotype& phenoMe , Cell* pOther, Phenotype& phenoOther , double dt
204 { return; }
205
206 void tumor_phenotype( Cell* pC, Phenotype& ph, double dt )
207 [
208
209
210 ]
211 ]
```

Implement a tumor phenotype function (2)

- Outline:
 - Get the current oxygen (σ), mechanical pressure (p), damage (D), and death status (dead)
 - Set the cycle entry rate (cap the linear factor between 0 and 1):

$$r_{01} = r_{01}^* \left(\frac{\sigma - 5}{38 - 5} \right) (1 - p)$$

- Set the necrosis rate (cap the linear factor between 0 and 1):

$$d_N = d_N^* \left(\frac{5 - \sigma}{5 - 2.5} \right)$$

- Set the apoptosis rate

$$d_A = d_A^* + (100 \cdot d_A^* - d_A^*)H(D)$$

- ◆ H is a Hill response function with half-max 120 and Hill power 1

- Set debris export proportional to cell volume if dead

Implement a tumor phenotype function (3)

```
// get the current oxygen (o2), mechanical pressure (p), damage (D), death status (dead)

double o2 = get_single_signal( pC , "oxygen" );
double p = get_single_signal( pC , "pressure" );
double D = get_single_signal( pC, "damage" );
bool dead = (bool) get_single_signal( pC, "dead" );

// if dead, set debris release and exit
if( dead )
{
    // release rate proportional to volume
    double export_rate = get_single_signal( pC, "volume");
    // set behavioral parameter
    set_single_behavior( pC, "debris export" , export_rate );
    return;
}
```

Implement a tumor phenotype function (4)

```
// set the cycle entry rate
    // get base cycle rate from cell definition
double rate = get_single_base_behavior(pC, "cycle entry" );
    // scale by O2
rate *= linear_response_function( o2 , 5 , 38 );
    // scale by pressure
rate *= decreasing_linear_response_function( p , 0 , 1 );
    // set the behavioral parameter
set_single_behavior(pC, "cycle entry" , rate ) ;

// set the necrosis rate
rate = 2.8e-3; // cells survive 6 hours in no O2
    // scale by decreasing O2
rate *= decreasing_linear_response_function( o2 , 2.5 , 5 );
    // set the behavioral parameter
set_single_behavior(pC, "necrosis" , rate ) ;
```

Implement a tumor phenotype function (5)

```
// set the apoptosis rate
    // get the base rate from cell definition
double rate0 = get_single_base_behavior(pC, "apoptosis");
    // increase with damage based on Hill function
double rateMax = 100 * rate0;
rate = rate0 + (rateMax-rate0)*Hill_response_function(D,120,1);
    // set the behavioral parameter
set_single_behavior( pC, "apoptosis" , rate );

return;
```

Checklist

Create a template project in PhysiCell

- See tutorial:
 - ◆ <https://github.com/physicell-training/ws2021#session-1-working-with-projects-in-physicell>
- make template && make

Save the config file from our online version

Declare a custom `tumor_phenotype` function in `./custom_modules/custom.h`

Implement this custom model in `./custom_modules/custom.cpp`

Assign the custom phenotype function to WT tumor cells and mutant tumor cells

Compile and run

Assign phenotype function

```
// find create_cell_types

/*
   Put any modifications to individual cell definitions here.

   This is a good place to set custom functions.
 */

cell_defaults.functions.update_phenotype = phenotype_function;
cell_defaults.functions.custom_cell_rule = custom_function;
cell_defaults.functions.contact_function = contact_function;

Cell_Definition* pCD = find_cell_definition( "WT tumor cell");
pCD->functions.update_phenotype = tumor_phenotype;

pCD = find_cell_definition( "mutant tumor cell");
pCD->functions.update_phenotype = tumor_phenotype;

/*
   This builds the map of cell definitions and summarizes the setup.
 */

display_cell_definitions( std::cout );

return;
}
```

Checklist

Create a template project in PhysiCell

- See tutorial:
 - ◆ <https://github.com/physicell-training/ws2021#session-1-working-with-projects-in-physicell>
- make template && make

Save the config file from our online version

Declare a custom `tumor_phenotype` function in `./custom_modules/custom.h`

Implement this custom model in `./custom_modules/custom.cpp`

Assign the custom phenotype function to WT tumor cells and mutant tumor cells

Compile and run

Compile and run

```
// compile and run  
make && ./project
```

Note for Windows users: make && project

```
// create jpegs  
make jpeg
```

```
// create gif and movie  
make gif
```

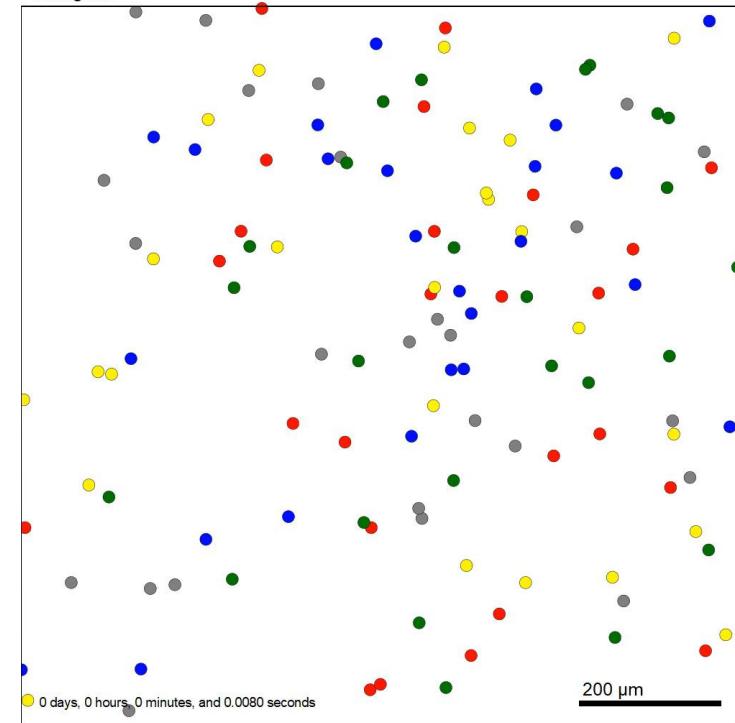
```
make movie
```



- WT tumor cell
- mutant tumor cell
- macrophage
- CD8+ T cell
- monocyte

Source code, extended details and more:
GitHub.com/PhysiCell-Training/UCI-sysbio-2022

Current time: 0 days, 0 hours, and 0.00 minutes, z = 0.00 μm
125 agents



Part IV:

Ongoing work and Opportunities

Continuous refinements for usability

- July 2017 (Version 1.2.0)
 - Restructured code around **cell phenotype**
- February 2018 (Version 1.3.0)
 - Introduced **XML-based configuration files**
- September 2018 (1.4.0)
 - **XML-based domain and user parameters**
- June 2019 (1.5.0)
 - Tracking of **internalized substrates**
 - Introduced standardized **predation** (ingest)
- August 2019 (1.6.0)
 - **XML-based setup** of the diffusion equations
- May 2020 (1.7.0)
 - Add **standardized chemotaxis** (with XML)
 - **XML-based cell definitions**
- March 2021 (1.8.0)
- July 2021 (1.9.0)
 - Formally released **XML-based cell definitions**
 - Introduced **cell contact functions**
 - Automatic tracking of **neighbor cells**
 - Auto-generated **legend**
 - Introduced **virtual walls**
 - Read a list of **initial cell locations** from CSV
- May 2022 (1.10.0)
 - Formalize **standardized intracellular modeling**
 - First support for **Model Builder GUI**

New work: reproducible model language

- As (hand-coded) model complexity grows:
 - Harder to understand the full model
 - Harder to clearly communicate the current biological hypotheses
 - Harder for domain experts to participate in real time
- **Goal:** Create a formalism for agent rules that:
 - Can be written in human-readable "plain English"
 - ◆ Facilitates tools for easy model construction
 - ◆ *Turns model building into knowledge mapping*
 - Can readily be "translated" to a standard mathematical form
 - ◆ Model can parse the rules without hand-coding
 - ◆ More reusable, maintainable model

Example: chemokine-driven cycling

- Biological hypothesis statement: INFG promotes cell cycling

- Rule: INFG increases cell cycle entry

- Mathematical translation:

$$r_{\text{cycle}} = r_0 + (10r_0 - r_0) \frac{[\text{INFG}]^{1.5}}{0.1^{1.5} + [\text{INFG}]^{1.5}}$$

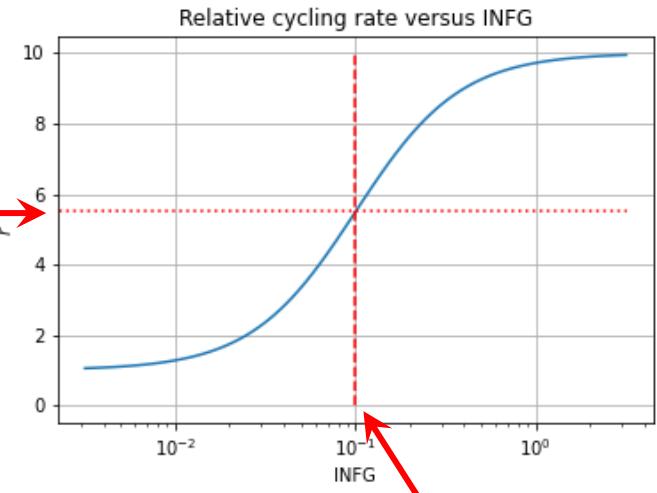
- Refined rule (with parameters):

▪ Hill response function	
▪ Hill power:	1.5
▪ half-max:	0.1
▪ base value:	r_0
▪ tenfold max response:	$r_U = 10 r_0$

- XML markup:

```
<rule>
    <signal name="INFG"/>
    <behavior name="cycle entry"/>
    <response type="increase" form="hill">
        <max_response type="relative">10</max_response>
        <hill_power>1.5</hill_power>
        <half_max>0.1</half_max>
    </response>
</rule>
```

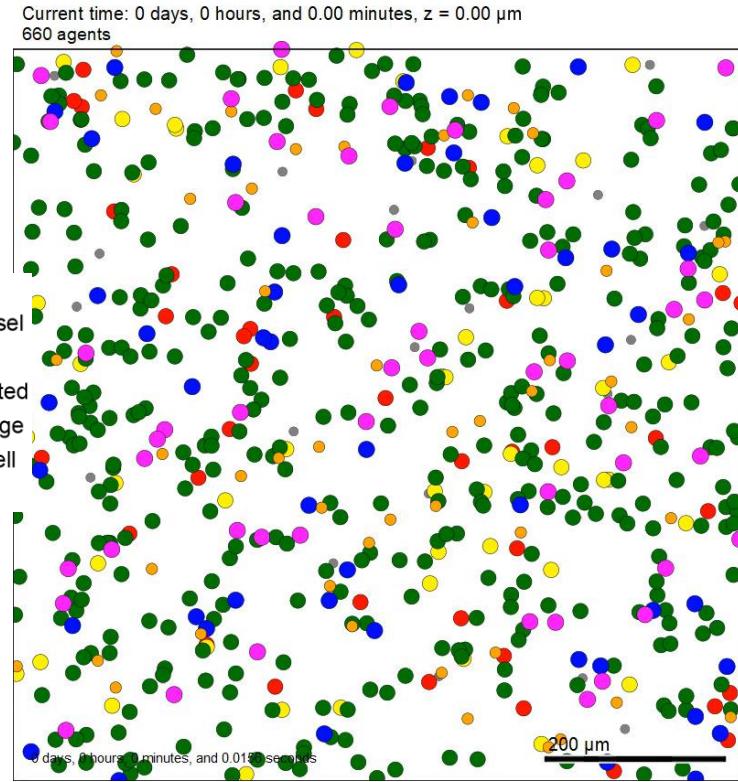
Half of max response



half-max level

Example: Tissue versus virulent bacteria

- Stem cells
 - Divide, differentiate
 - Killed by toxin
- Differentiated cells
 - Divide
 - Killed by toxin
- Blood vessels
 - Release resource
- Bacteria
 - Colonize near resources (via quorum)
 - Release toxin
 - Killed by damage
- Macrophages
 - Phagocytose dead cells
 - Release pro-inflammatory factor
- CD8+ T cells
 - Attracted to pro-inflammatory factor
 - Damage bacteria
- Neutrophils
 - Attracted to pro-inflammatory factor
 - Phagocytose bacteria



LUDDY

SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

The future is modeling in *real time* with knowledge mapping.

1. Meet with domain experts to formulate behavioral hypotheses
 - Oxygen increases cycle entry in tumor cells
 - Oxygen decreases necrotic death in tumor cells
 - Oxygen decreases secretion of VEGF in tumor cells
 - VEGF increases motility in endothelial cells
 - Contact with endothelial cells inhibits proliferation in endothelial cells
 - Cell debris increases secretion of IL-6 by macrophages
 - ...
2. Immediately import the rules and simulate behavior
3. Get domain expert feedback in real time to improve the hypotheses

Over time, we'll create community-curated library of reusable behavioral hypotheses.

Last Year's Virtual Training Course

15 Virtual Sessions:

- PhysiCell Essentials and Modeling Workflows
- Graphical Model Editor
- Phenotype
- Microenvironment
- Functions
- Chemical Communication / Interactions
- Contact Communication / Interactions
- Intracellular Modeling with ODEs / SBML
- Extensions for high performance computing (HPC)
- Cloud-hosted Model Sharing
- ... and more!

Sessions include:

- Slides (PDF format)
- YouTube recordings
- Source code

github.com/PhysiCell-Training/ws2021



Apply Today! PhysiCell ws2022

- July 24-30, 2022
- Fully virtual
- Tutorial sessions option to general public
- Competitive selection for mentored hackathon



2022 Virtual PhysiCell Workshop and Hackathon
July 24-30, 2022



- Build and explore multicellular agent-based simulations of cancer and other systems
- Learn to share your models online
- Meet other modelers in the CSBC / PS-ON community
- Compete in an exclusive mentored hackathon
- PhysiCell swag for accepted participants
- Application and full agenda at QR code or:
<https://github.com/PhysiCell-Training/ws2022>

