

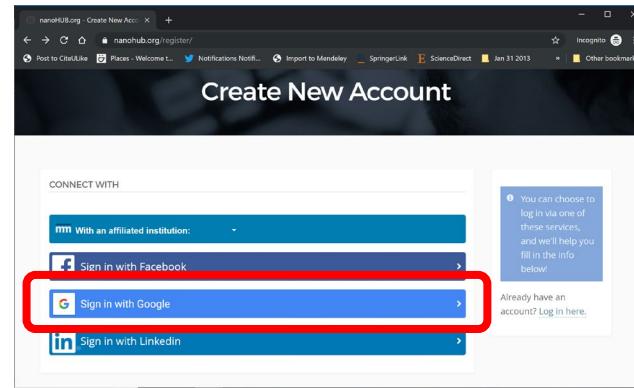
nanoHUB Account

- These tutorials use cloud-hosted PhysiCell models on nanoHUB.org.
- nanoHUB is **free**, but it requires a one-time registration.

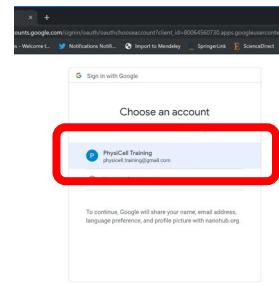
• Steps:

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 nanoHIB account)
5. Finish filling in details, and you're done!
6. Use your google account to sign in in the future.

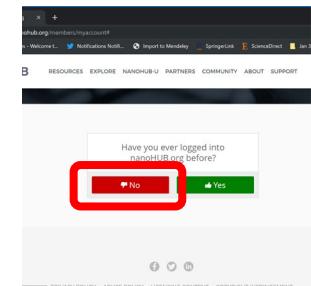
2



3



4



Introduction to multicellular modeling with PhysiCell

Get lectures and
materials here!



Paul Macklin, Ph.D.

Intelligent Systems Engineering
Indiana University

January 25, 2020

Thanks: Partners

- **Colon cancer metabolic cross-talk in organoids:**
 - Stacey **Finley** (USC, U01 Contact PI)
 - Shannon **Mumenthaler** (USC, U01 PI)
- **Hypoxia in breast cancer invasion:**
 - Daniele **Gilkes** (JHU, JKTGF Contact PI)
- **ECM and leader-follower interactions:**
 - Andy **Ewald** (JHU, BCRF & JKTGF Contact PI)
 - Newton (USC), Peyton (Umass), Bader (JHU), Hass
- **PhysiCell core team:**
 - Randy Heiland (IU)
 - **alumni:** S.H. Friedman, A. Ghaffarizadeh (USC)
- **High-throughput on HPC:**
 - Jonathan **Ozik**, Nicholson **Collier**, Justin **Wozniak**, Charles **Macal** (Argonne National Lab)
 - Chase Cockrell, Gary **An** (University of Vermont)
- **Cloud-deployed PhysiCell models:**
 - Gerhard **Klimeck**, Zentner, others (NanoHUB Cyberplatform at Purdue)
 - Geoffrey **Fox** (IU PI, nanoBIO Node)
- **PhysiCell software extensions & refinements**
 - MPI / HPC extensions: Barcelona Supercomputing Center (Montagud, Valencia, others ...)
 - Boolean network extensions: Institut Curie (Letort, Montagud, Stoll, Barillot, Zinov'yev, Calzone)
 - Flux balance extensions: Miguel Ponce de Leon (Barcelona Supercomputing Center)
 - GPU computing prototypes: Sunita Chandrasekaran (Delaware)
- **IU PhD students:**
 - John Metzcar (hypoxia, invasion),
 - Yafei Wang (liver mets, nanotherapy)
 - Furkan Kurtoglu (multicellular metabolism)
 - Aneequa Sundus (cyanobacteria, synthetic multicellular systems, machine learning)
- **IU Undergraduate students:**
 - **ECM and invasion:** D. Murphy, B. Duggan
 - **PhysiCell community:** K. Konstantinopoulos, D. Willis, B. Yu, M. Chen
 - **Python, Jupyter & fun:** D. Taylor, B. Anderson, D. Mishler
 - **Alumni:** T. Mahajan, B. Fisher, E. Bower, T. Zhang, E. Connor

Thanks: Funders

- NIH (current)
 - NIH CSBC U01 (1U01CA232137), **PIs** Finley* / Macklin / Mumenthaler
 - ◆ 2019-2020 U01 supplement for PhysiCell training and software ecosystem
 - High-End instrumentation grant (1S10OD018495-01), **PI** Foster
- NIH (past support that helped PhysiCell)
 - Provocative Questions grant (1R01CA180149), **PIs** Agus / Atala / Soker
 - NIH PS-OC center grant (5U54CA143907), **PIs** Agus / Hillis
 - HuBMAP CCF contract (OT2 OD026671), **PI** Börner
- NSF:
 - Engineered nanoBIO Hub (1720625), **PI** Fox. **Co-PIs** Douglas, Glazier, Macklin, Jadhao
 - Cyanobacteria / Synthetic Biology (1818187), **PI** Kehoe, **Co-PI** Macklin
- Breast Cancer Research Foundation & JKTGF, PI Macklin
 - projects with **PIs** Agus, Gilkes, Peyton, Ewald, Newton, Bader



JAYNE KOSKINAS
TED GIOVANIS

Foundation for
Health and Policy



NATIONAL
CANCER
INSTITUTE



BCRF
BREAST
CANCER
RESEARCH
FOUNDATION

Cancer is a systems problem

Interconnected systems and processes:

- Single-cell behaviors
- Cell-cell communication
- Physics-imposed constraints (e.g., diffusion)
- Systems of systems (e.g., immune system)

In cancer, these systems become dysregulated.

Treatments target parts of these systems.

Cancer is a **complex system**:

changing one part can have **surprising effects!**

Modeling can help **understand** this system.

This is **multicellular systems biology**.

If we can **control** these systems, we've arrived at
multicellular systems engineering.



Source: Hanahan & Weinberg (2011)
DOI: [10.1016/j.cell.2011.02.013](https://doi.org/10.1016/j.cell.2011.02.013)

**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

- **Model many cells in these chemical environments**

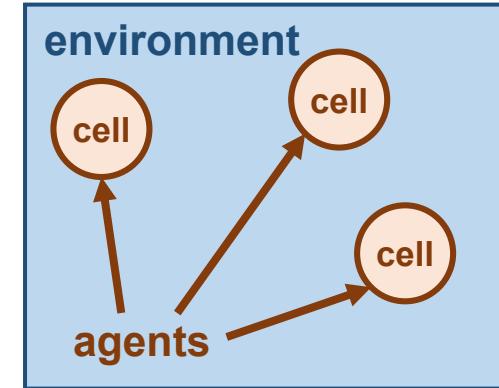
- Environment-dependent behavior (including molecular-scale "logic")
- Mechanical interactions
- Heterogeneity:
 - ◆ individual states
 - ◆ individual parameter values
 - ◆ individual model rules

- **Run many copies of the model in high throughput**

- Discover the rules that best match observations.
- Identify and exploit weaknesses that can restore control

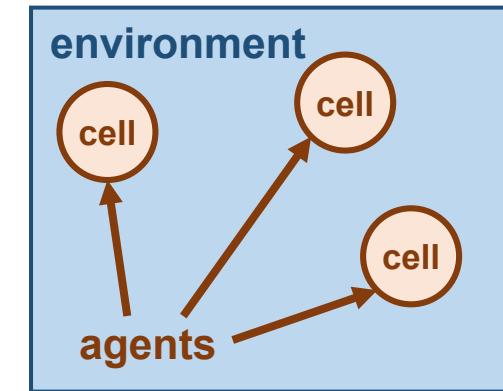
What is an agent-based model?

- Each cell is modeled as a separate software object (an **agent**) with:
 - **member data:** internal state variables
 - ◆ Position, Size, Cycle State, molecular variables,
 - **methods:** cellular processes
 - ◆ Cycling, Death, Motility, Growth, Adhesion, ...
- Virtual cells move a virtual (**micro**)environment
 - Usually liquid (e.g., water or interstitial fluid)
 - Chemical movement (oxygen, glucose, signaling factors)
 - ◆ Typically diffusion: solve partial differential equations (PDEs)
 - ◆ May also require advection for environments with flow
 - May include mechanical components like extracellular matrix (ECM)
 - ◆ Finite element methods or related methods



What's the connect 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**.



Introducing PhysiCell

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 $20\text{ }\mu\text{m}$ resolution

Features:

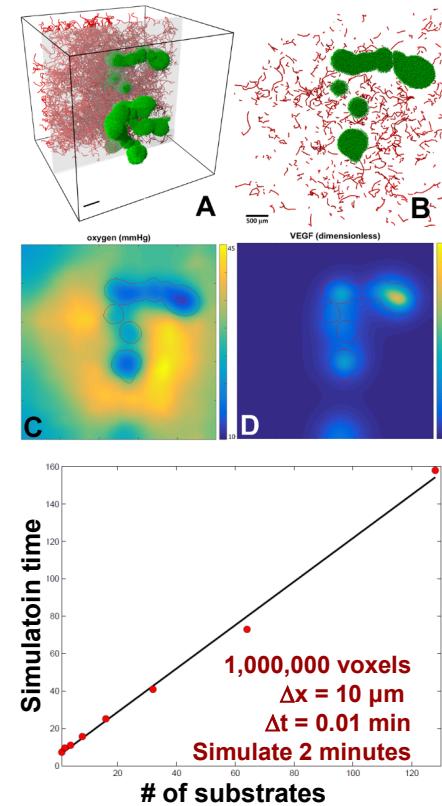
- Off-lattice cell secretion and uptake
- 2nd-order accurate (space), 1st-order accurate (time), numerically stable

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
- **Deployed from Raspberry Pi to Crays**

Method:

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

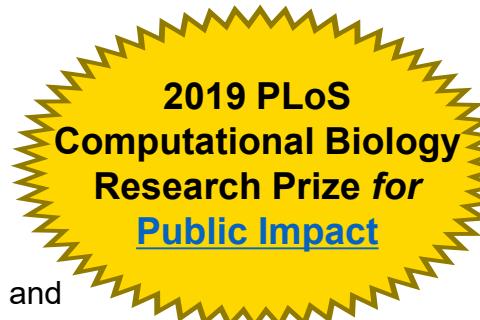
Reference: Ghaffarizadeh et al.,
PLoS Comput. Biol. (2018)

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

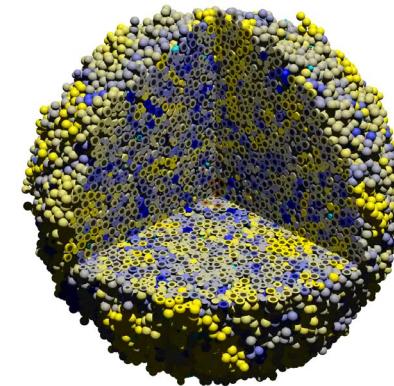


Try this model yourself!

nanohub.org/tools/pc4heterogen



Current time: 7 days, 0 hours, and 0.00 minutes
53916 cells



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

Let's try an example!

- **cancer biorobots:**

- **green:**

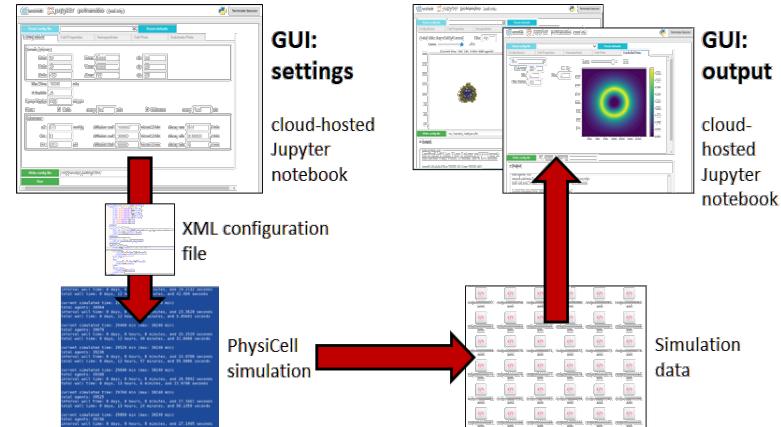
- ◆ cycle entry scales with O₂
 - ◆ O₂ depletion causes necrosis
 - ◆ cumulative drug exposure causes apoptosis

- **blue:**

- ◆ drug-loaded "cargo"

- **red:**

- ◆ worker cells that seek and haul cargo towards hypoxic zones



XML+Jupyter architecture



Try this model yourself!

<https://nanohub.org/tools/pc4cancerbots>

A first look under the hood

PhysiCell agents

- Each PhysiCell agent has:
 - functions to **sample the microenvironment**
 - **State** elements (things that are measured from an external frame of reference)
 - ◆ ID, type, position, velocity, base-to-apex orientation, ...
 - **Phenotype**
 - ◆ hierarchically-ordered behavioral state and parameters (much more detail soon)
 - user-defined **functions**
 - ◆ functions that act on the cell's phenotype (or state)
 - ◆ can replace the built-in mechanics and other behaviors
 - user-defined **custom data**
 - ◆ used to track additional details specific to user models, if they aren't already

Documentation: User Guide, Sec. 9

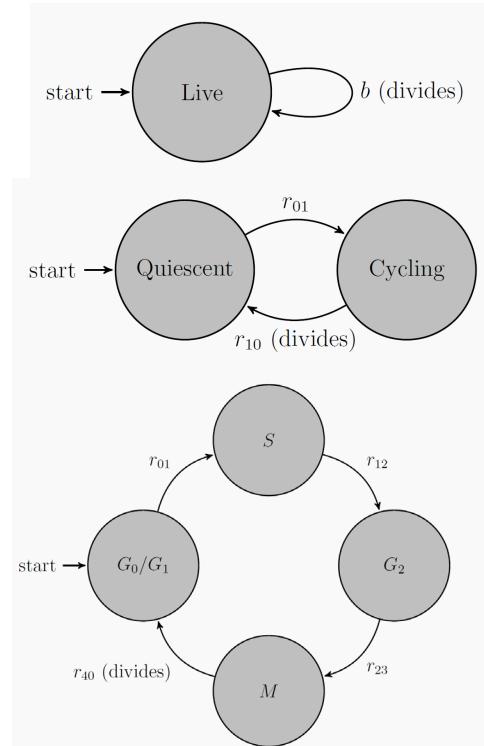
Cell phenotype

- One of the most critical data elements in a PhysiCell Cell is ***phenotype***
- Hierarchically organize key behavioral elements:
 - Phenotype
 - ◆ **cycle**: advancement through a cell cycle model
 - ◆ **death**: one or more types of cell death
 - ◆ **volume**: cell's volume regulation
 - ◆ **geometry**: cell's radius and surface area
 - ◆ **mechanics**: adhesion and resistance to deformation ("repulsion")
 - ◆ **motility**: active motion (other than "passive" mechanics)
 - ◆ **secretion**: both release and uptake of chemical substrates. Interfaces with BioFVM
 - ◆ **molecular**: a place to store internalized

Documentation: User Guide, Sec. 10

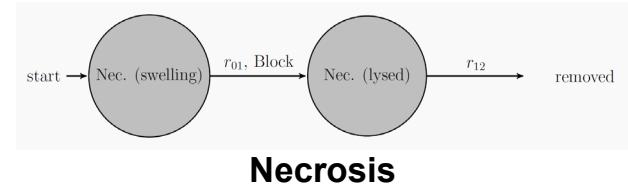
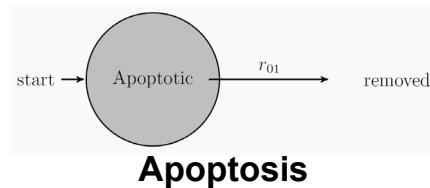
Phenotype: Cycle

- Each agent's **phenotype** had a **cycle** with:
 - **Cycle model**
 - ◆ A directional graph: **nodes** are cycle **phases** $\{P_i\}$ and **edges** are **transition rates** $\{r_{ij}\}$
 - ◆ r_{ij} is the transition rate from phase P_i to phase P_j
 - ◆ One of the transitions must be marked as a *division transition*
 - ◆ Users can attach **arrest condition** functions to these transitions (e.g., size checks)
 - **Cycle data**
 - ◆ stores the cell's current transition rates
- **Documentation:** User Guide, Sec. 11.1



Phenotype: Death

- Death has one or more death models:
 - A specialized cycle model with a *removal* transition rate
 - Extra parameters to help govern cell volume
 - Each death model has an associate death rate
 - Also stores an easy Boolean **dead** to easily check if the cell is alive.
- PhysiCell has built-in apoptosis and necrosis death models



Documentation: User Guide, Sec. 11.2

Phenotype: Volume

- **volume** records the cell's sub-volumes:
 - nuclear and cytoplasmic
 - solid vs. fluid
 - calcified fraction
 - key parameters
- a very simple **default model** to regulate volume based on ODEs
 - Change the parameters, target values based on environment and cell state

$$\frac{dV_F}{dt} = r_F(V_F^* - V_F)$$

$$\frac{dV_{NS}}{dt} = r_N(V_{NS}^* - V_{NS})$$

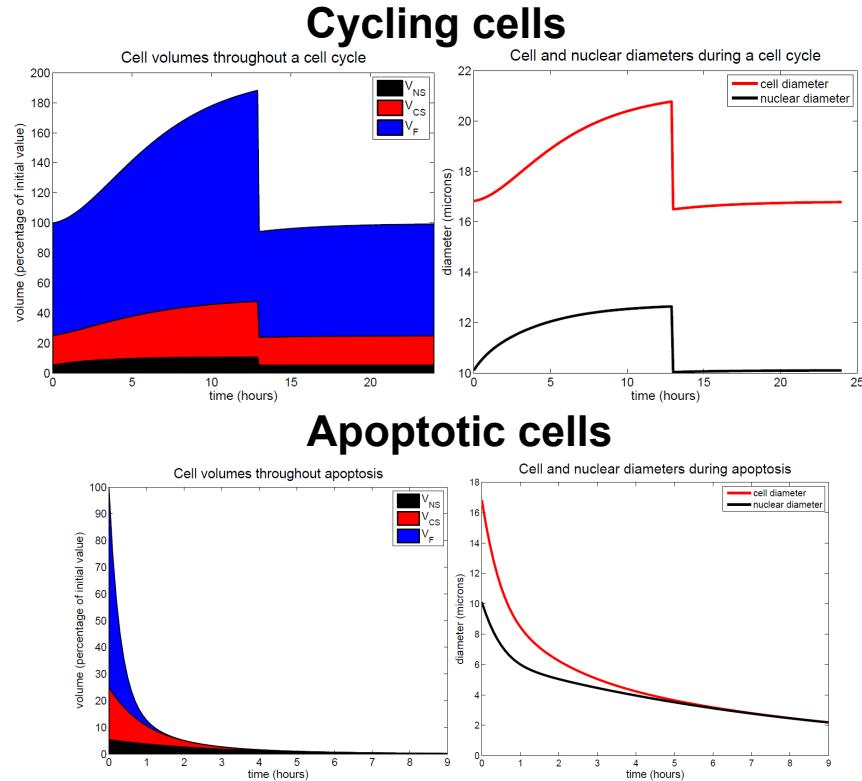
$$\frac{dV_{CS}}{dt} = r_C(V_{CS}^* - V_{CS})$$

Documentation: User Guide 11.3

Phenotype: Geometry

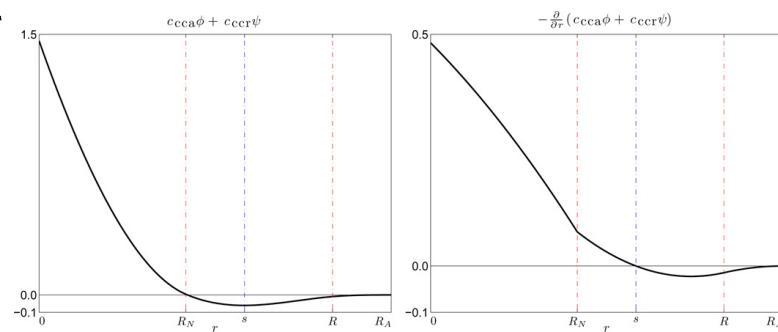
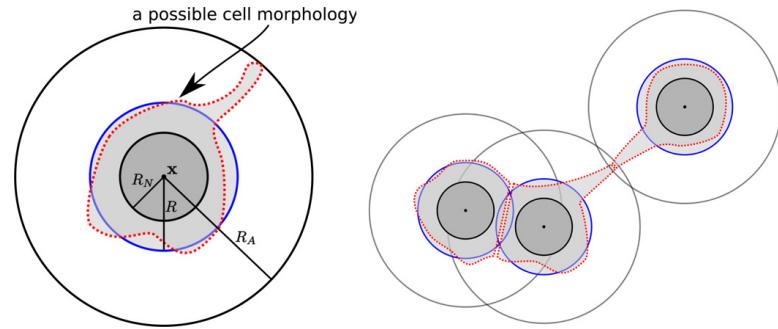
- **geometry records:**
 - surface area (not actively tracked)
 - equivalent spherical radius
 - equivalent nuclear radius

Documentation: User Guide 11.4



Phenotype: Mechanics

- **Mechanics** keeps parameters for adhesion and "repulsion"
 - Key parameter: maximum adhesion distance
 - ◆ a multiple of the cell's radius
 - (as a multiple of the cell's radius)
- Default model uses potential functions, but this can be supplemented or replaced.



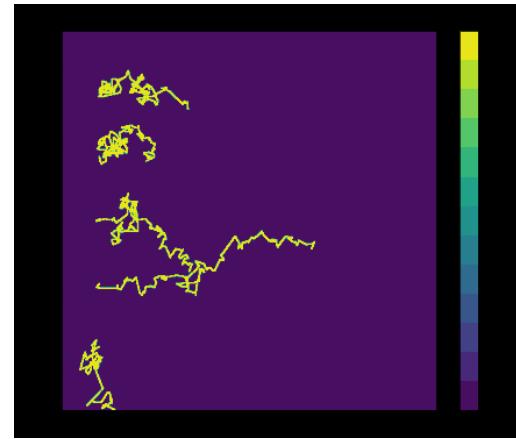
Documentation: User Guide 11.5

Phenotype: Motility

- **Motility** controls biased random migration

- Migration speed s
- Bias direction \mathbf{d}_{bias}
- Migration bias $0 \leq b \leq 1$
 - ◆ If $b = 1$, deterministic motion
 - ◆ If $b = 0$, purely Brownian motion
- Persistence time T_{per}

$$\mathbf{v}_{\text{mot}} \sim s(b\mathbf{d}_{\text{bias}} + (1 - s)\mathbf{d}_{\text{rand}})$$



Documentation: User Guide Sec. 11.6



Try this model yourself! (2D)

<https://nanohub.org/tools/motility>

Phenotype: Secretion

- **Secretion** stores parameters for secretion and uptake of diffusing substrates

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (D \nabla \rho) - \lambda \cdot \rho + \sum_i \delta(x - x_i) V_i (S_i \cdot (\rho_i^* - \rho) - U_i \cdot \rho)$$

Future: Will add an additional "Export" term since the **S** form isn't right for all types of measurements and assays.

Documentation: User Guide Sec. 11.7

Phenotype: Molecular

- **Molecular** is where total internalized substrates are tracked. (optional)
 - A fraction (or all) of substrates can be released at cell death
 - A fraction (or all) of substrates can be transferred when a cell is ingested
 - Internalized substrates are divided among daughter cells
- Eventual support for molecular-scale models will be attached here.

Documentation: User Guide Sec. 11.8

Phenotype-centric programming

- The core cell behaviors are implemented:
 - Cell cycling (with user-selectable models)
 - Cell death
 - Cell adhesion / repulsion
 - Cell motility
 - Cell secretion / uptake
- Modelers can focus on writing functions that control these behaviors.
- This is **phenotype-centric programming**.

A note about time steps

- PhysiCell is designed to account for the multiple time scales inherent to these problems, and has 3 time scales:
 - $\Delta t_{\text{diffusion}}$ diffusion, secretion, and uptake (default: 0.01 min)
 - $\Delta t_{\text{mechanics}}$ cell movement (default: 0.1 min)
 - Δt_{cell} phenotype and volume changes (default: 6 min)
- This allows some efficiency improvements: not all functions need to be evaluated at each time step.
- See the PhysiCell method paper. (Oddly, not in the User Guide (yet).)

Examples

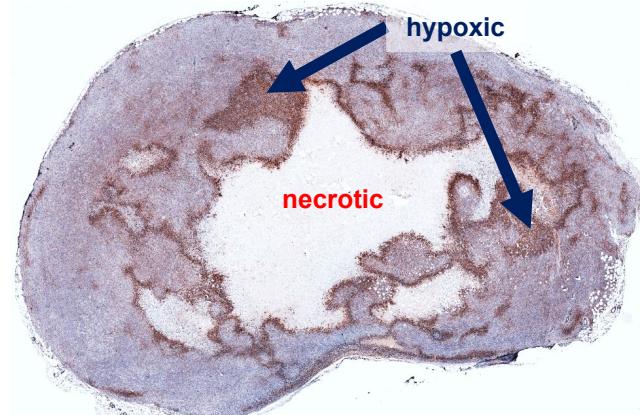
Example 1:

Cancer cell response to hypoxic stress

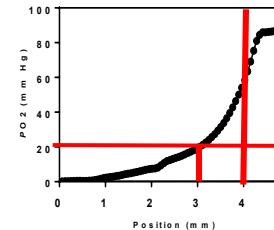
with Daniele Gilkes (Johns Hopkins)

Hypoxia in breast cancer

- Most breast cancers are hypoxic
 - normal breast: $pO_2 \sim 65$ mmHg
 - breast cancer: $pO_2 \sim 10$ mmHg
 - (Tatum *et al.* *Int. J. Radiat. Biol.* 2006)
- Hypoxia drives phenotype changes
 - Hypoxic responses at $\sim 8\text{-}10$ mmHg
 - transformation into stem-like cells
 - Increased motility
 - Increased ECM remodeling
 - Increased glycolysis
 - Increased acidosis
 - (maybe) decreased adhesion
 - VEGF secretion (+angiogenesis)



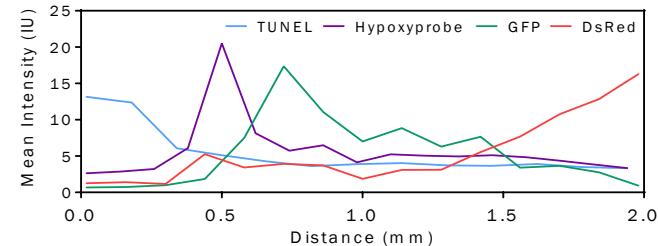
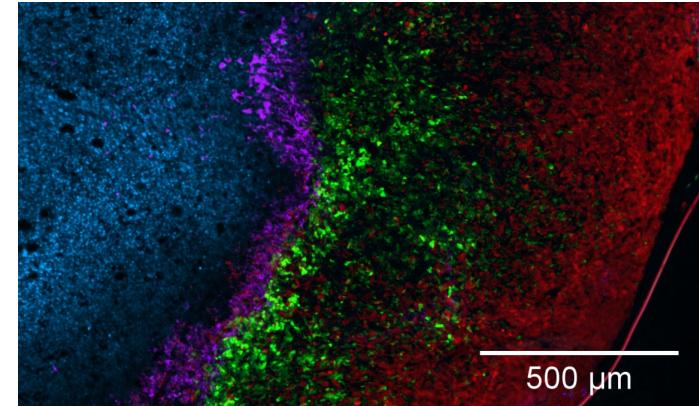
Hypoxic breast tumor (via hypoxyprobe)



Radial pO_2 profile (optical measurement)
Source: Gilkes lab, Johns Hopkins

A novel hypoxia reporter

- Gilkes developed a permanent color change:
 - Cells express red fluorescent protein (RFP)
 - Cells express green fluorescent protein (GFP) after exposure to hypoxia < 10 mmHg
 - This change is **permanent and inherited by daughter cells**
- This lets us determine hypoxic origins of metastases!



Labeled *in vivo* tumor cross-section
Source: Gilkes lab, Johns Hopkins

Observational data and questions

- **Observational data**
 - Both green and red lung metastases are detected.
 - ➔ Both cell populations can leave the tumor to metastasize
 - Green cells are first observed at the perinecrotic boundary. (Not a surprise!)
 - Green cells are later found near the outer edge of tumors
 - ➔ Green cells must be motile for at least some time
 - In similar conditions, red and green cells have similar Ki67 fractions
 - ➔ Hypoxia does not permanently change proliferation characteristics
 - Necrotic cores are observed (see TUNEL)
 - ➔ Not all cells escape!
- **Some Questions**
 - Are red cells ever motile?
 - Are all green cells motile, or just some of them?
 - How long do cells keep their motile phenotype? (**phenotypic persistence**)
 - ◆ Only when pO_2 is low? For a short time? Permanently?
 - Do cells act independently, or do they coordinate?

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

**How persistent is their
response to hypoxic stress?**

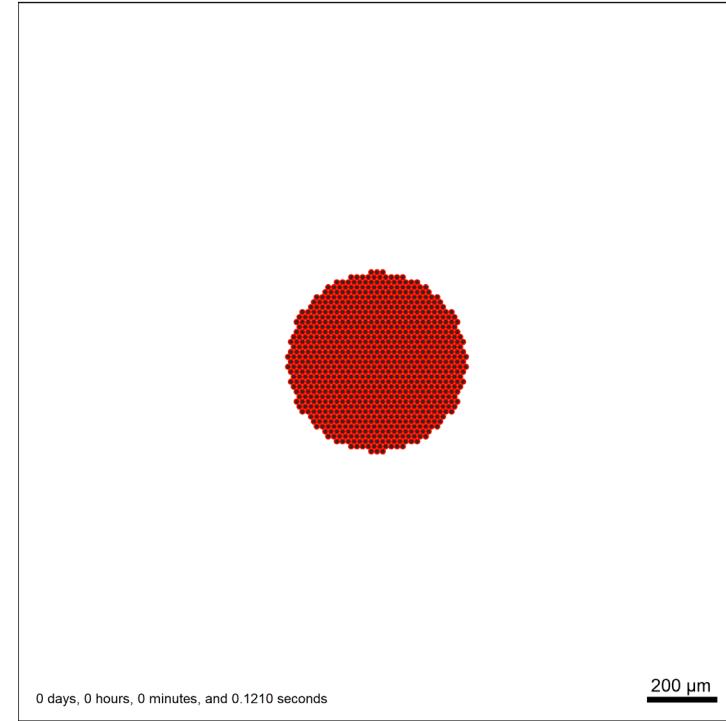
No phenotypic persistence

- GFP+ cells:
 - If $pO_2 < 10$ mmHg:
 - ◆ Same division rates
 - ◆ Speed: $0.25 \mu\text{m} / \text{min}$
 - ◆ 50% bias along ∇pO_2
 - If $pO_2 > 10$ mmHg
 - ◆ Set speed = 0.0

Matching observations:

- [] GFP+ cells reach edge
- [x] Necrotic core (a bit)
- [] GFP+ microcolonies

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



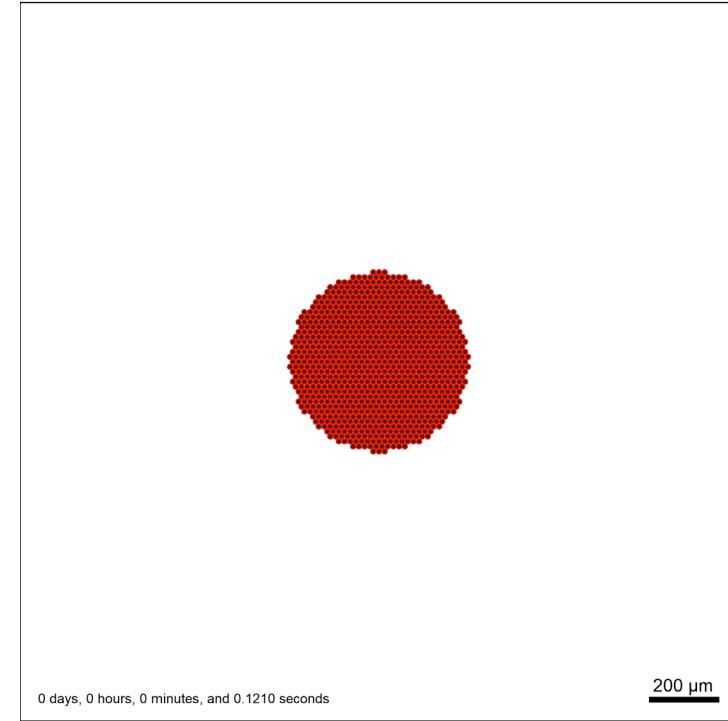
Phenotypic permanence

- GFP+ cells:
 - If $pO_2 < 10$ mmHg:
 - ◆ Same division rates
 - ◆ Speed: $0.25 \mu\text{m} / \text{min}$
 - ◆ 50% bias along ∇pO_2
 - If $pO_2 > 10$ mmHg
 - ◆ No change

Matching observations:

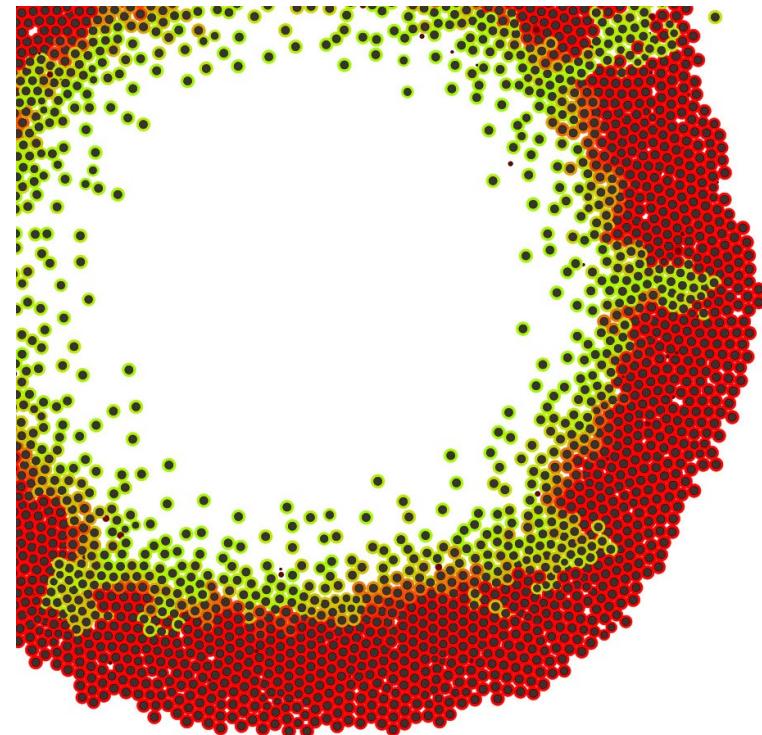
- [x] GFP+ cells reach edge
- [] Necrotic core
- [] GFP+ microcolonies

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



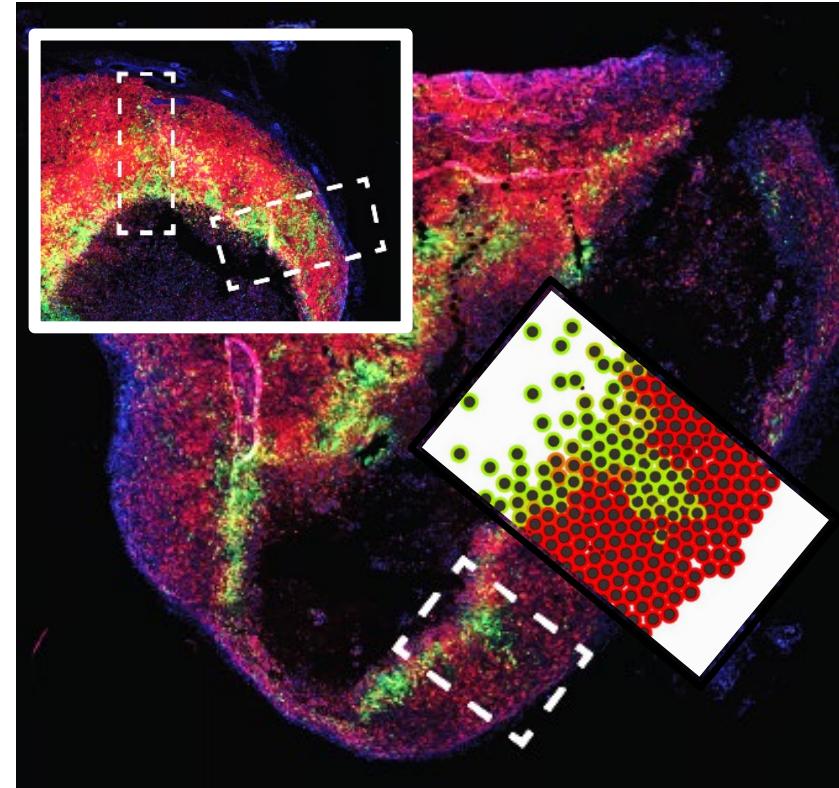
Novel prediction: hypoxic plumes

- It **looks like** collective motion, but it's purely mechanics
 - Cells are motile
 - If one motile cell finds a gap, it's easier for others to exploit it
 - A "plume" of hypoxic cells grows
- Model suggests a **therapeutic strategy**:
 - Make hypoxic response less persistent to reduce escape.
- **They're observed *in vivo***
 - MDA-MB-231 in mice at ~20 days
 - **Source:** Gilkes lab (JHU)



Novel prediction: hypoxic plumes

- It *looks like* collective motion, but it's purely mechanics
 - Cells are motile
 - If one motile cell finds a gap, it's easier for others to exploit it
 - A "plume" of hypoxic cells grows
- Model suggests a **therapeutic strategy**:
 - Make hypoxic response less persistent to reduce escape.
- **They're observed *in vivo***
 - MDA-MB-231 in mice at ~20 days
 - **Source:** Gilkes lab (JHU)



**Novel imaging reveals a
hidden structure.**

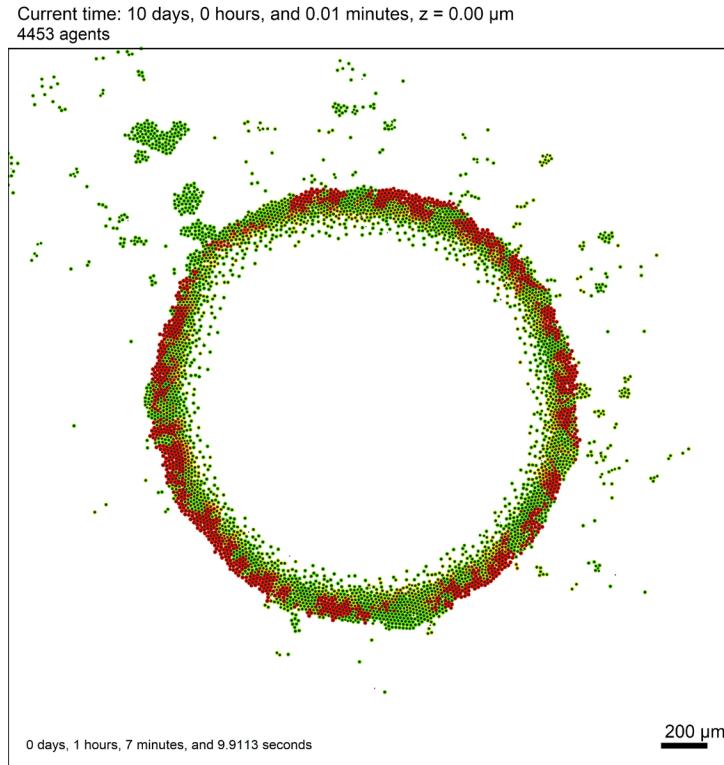
**Computational modeling
helps explain it.**

Phenotypic persistence

- GFP+ cells:
 - If $pO_2 < 10 \text{ mmHg}$:
 - ◆ Same division rates
 - ◆ Speed: $0.25 \mu\text{m} / \text{min}$
 - ◆ 50% bias along ∇pO_2
 - If $pO_2 > 10 \text{ mmHg}$
 - ◆ Motile phenotype has mean persistence time T_{persist}
 - ◆ In $[t, t + \Delta t]$, stop migration with probability $P = \Delta t / T_{\text{persist}}$

Matching observations:

- [x] GFP+ cells reach edge
- [] Necrotic core
- [x] GFP+ microcolonies



Example 2: **Cancer-immune contact interactions**

with Argonne National Lab and Gary An

Simple model of cancer-immune interactions

Heterogeneous tumor cells (blue to yellow):

- Cycle entry rate scales with O₂
- Cells necrose in very low O₂
- Yellow cells are most proliferative;
 - blue are least proliferative
- Yellow cells are most immunogenic
 - simplified model of MHC

Immune cells (red):

- Biased random walk towards tumor
- Test for contact with cells
- Form adhesion
- Attempt to induce apoptosis
 - (e.g., FAS receptor)
 - success depends on immunogenicity
- Eventually detach from cell, continue search

Movie: [[View on YouTube](#) (4K)]

References:

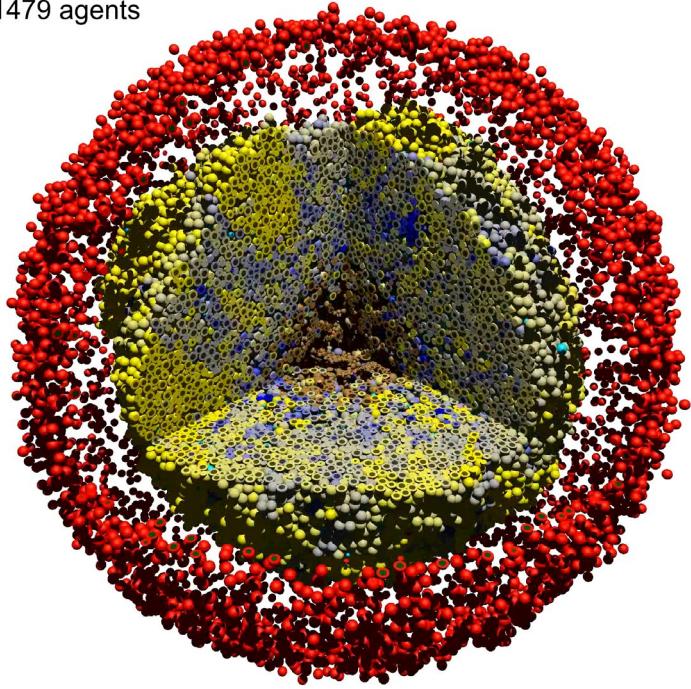
- [Ghaffarizadeh et al. \(2018\)](#)
- [Ozik et al. \(2018\)](#)
- [Ozik et al. \(2019\)](#)



Try this model yourself! (2D)

nanohub.org/tools/pc4cancerimmune

Current time: 14 days, 0 hours, and 3.00 minutes
111479 agents



Exploring high-dimensional design spaces

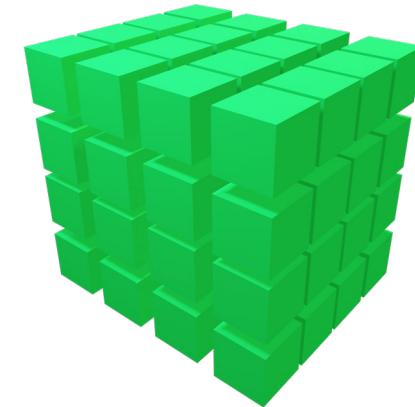
- We missed a lot of parameters. Let's increase to a 6-D design space.

1. Immune cell apoptosis rate (related to total killing capacity)
2. Oncoprotein threshold p_T (cancer cells are invisible if $p < p_T$)
3. Immune kill rate (rate attached immune cells can induce apoptosis)
4. Immune cell attachment rate
5. Immune cell attachment lifetime
6. Immune cell migration bias

} original
parameters

- Design space is a **constrained** hypercube:

- **Biological** constraints
 - ◆ Cells can only move so fast
 - ◆ Limits of receptor dynamics ...
- **Clinical** constraints
 - ◆ Can't use infinitely many immune cells
 - ◆ Sensitivity limits (otherwise overactive immune system, cytokine storms, etc.) ...



Scenarios to explore

Cancer control

1) Number of tumor cells at end (N_{final}) doesn't exceed initial count (N_{initial})

Cancer remission

2) Can we reduce cancer cells by 90% ($N_{\text{final}} \leq 0.1 N_{\text{initial}}$)?

3) Can we reduce cancer cells by 99% ($N_{\text{final}} \leq 0.01 N_{\text{initial}}$)?

Treatment optimization:

4) Can we minimize N_{final} ?

Approach:

Problem 4 is fairly traditional:

 Use genetic algorithm (*)

Problems 1-3 are harder:

 Can't densely sample 6-D design space! (Even on HTC!)

 531,441 discrete points in design space

Use active learning to find the shape of the "good design" subspace.

Using active learning

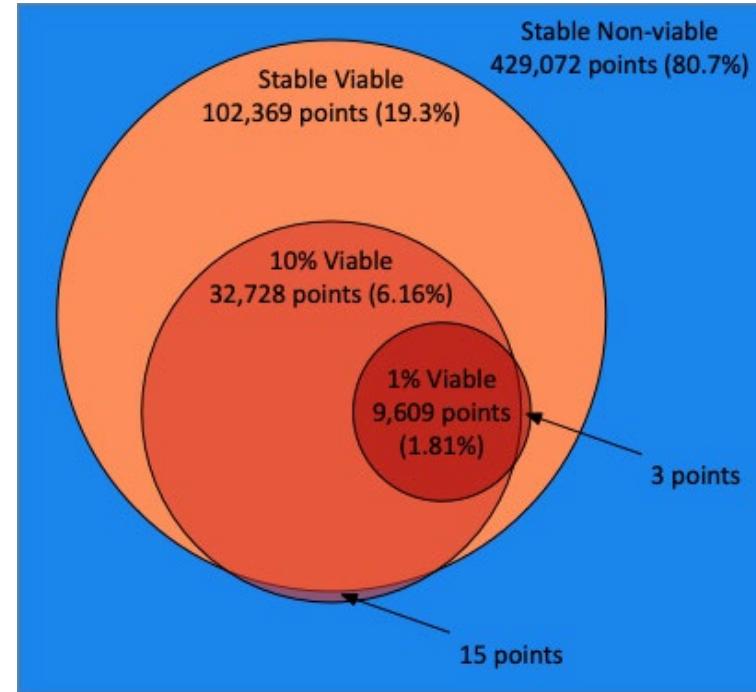
- For each design scenario (e.g., 10% scenario), build a binary DT classifier:
 - **True**: points that meet the design goal (e.g., $N_{\text{final}} \leq 0.1 N_{\text{start}}$)
 - **False**: points that don't meet the design goal (e.g., $N_{\text{final}} > 0.1 N_{\text{start}}$)
- **Run 1000 simulations at a time to build the classifier:**
 - 50 points in the 6-parameter space
 - 20 replicates per sample
 - Classify samples as true/false
- **Active learning** helps us choose samples that refine the decision boundary

Bonus: Use the Gini coefficients to **rank** the parameters

Reference:
[Ozik et al. \(2019\)](#)

How did HPC+ML enable new science?

- HPC gives the ***topology*** of the design space:
 - Each design scenario is an isosurface.
 - Finding multiple surfaces gives the topology.
 - More aggressive treatment goals drastically **shrink the viable design space**
- HPC+ML makes it **feasible to** find several design surfaces to **see the topology**
 - ~ 30,000 to 40,000 simulations per contour
 - **Active learning:** Reduced from 10^7 to 10^4 simulations
 - ~ 48,000 core hours for each surface
 - ~ 250 days (nonstop) on high-end workstation
 - ~ 2 weeks (nonstop) on a smallish cluster
 - ~ 12 hours on a Cray at ANL



Reference: [Ozik et al. \(2019\)](#)

Human learning from ABM + ML

Top two parameters

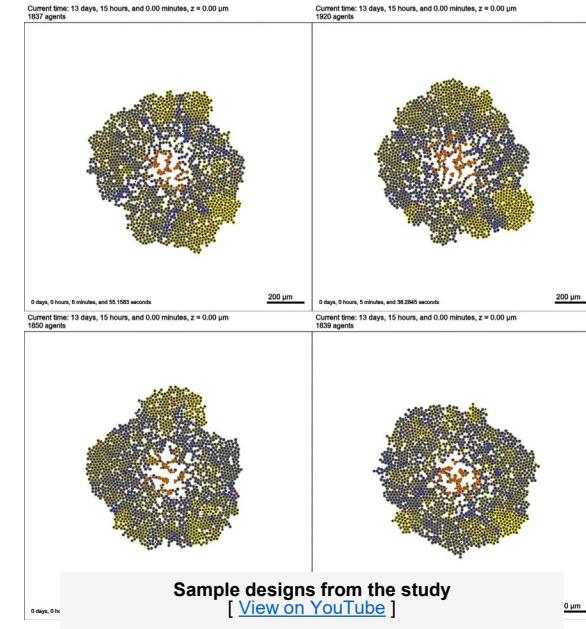
1. Immune cell apoptosis rate (d_1)

- ◆ Minimizing d_1 is analogous to maximizing functional lifetime of immune cells.
 - » $1/d_1$ is the mean lifetime of an immune cell
 - » increases the max number of cell kills for each immune cell
 - » analogous to effects of T cell exhaustion
 - » largely a biological constraint

2. Oncoprotein threshold (d_2)

- ◆ Decreasing d_2 corresponds to increasing immune cell sensitivity
- ◆ Increasing sensitivity without selectivity would have toxicity effects
- ◆ Both a biological and a clinical constraint

Machine learning helped us interpret the agent-based model results.



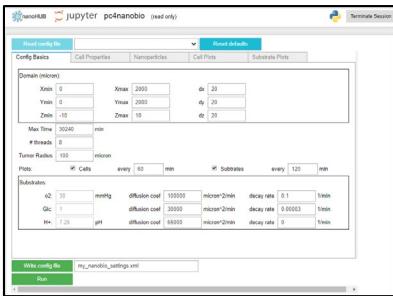
Try this model yourself!
nanohub.org/tools/pc4cancerimmune

Reference:
[Ozik et al. \(2019\)](#)

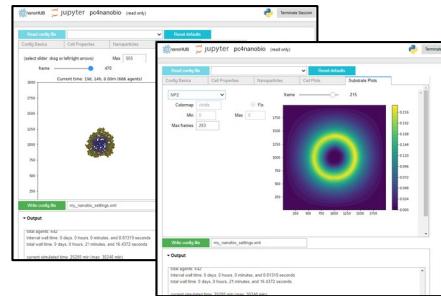
cloud-hosted models

Jupyter-based GUIs

Goal: Make PhysiCell-powered simulators user-friendly, shareable, and available without installing / compiling.



GUI:
settings
Jupyter
notebook



GUI:
output
Jupyter
notebook



XML
config file

```
interval cell time: 0 days, 8 hours, 4 minutes, and 42.481 seconds
current simulated time: 0 days, 8 hours, 4 minutes, and 42.481 seconds
total wait time: 0 days, 8 hours, 4 minutes, and 23.3629 seconds
total cell times: 0 days, 12 hours, 41 minutes, and 0.050911 seconds
current wait time: 0 days, 12 hours, 41 minutes, and 0.050911 seconds
total agents: 39679
current agents: 39679
interval cell time: 0 days, 8 hours, 4 minutes, and 25.3529 seconds
total wait time: 0 days, 12 hours, 49 minutes, and 51.6898 seconds
current simulated time: 0 days, 12 hours, 49 minutes, and 51.6898 seconds
total agents: 39679
current agents: 39679
interval cell time: 0 days, 8 hours, 4 minutes, and 24.9798 seconds
total wait time: 0 days, 12 hours, 57 minutes, and 56.9986 seconds
current simulated time: 0 days, 12 hours, 57 minutes, and 56.9986 seconds
total agents: 39679
current agents: 39679
interval cell time: 0 days, 8 hours, 4 minutes, and 26.9792 seconds
total wait time: 0 days, 13 hours, 8 minutes, and 27.9792 seconds
current simulated time: 0 days, 13 hours, 8 minutes, and 27.9792 seconds
total agents: 39679
current agents: 39679
interval cell time: 0 days, 8 hours, 4 minutes, and 27.1461 seconds
total wait time: 0 days, 13 hours, 14 minutes, and 27.1459 seconds
current simulated time: 0 days, 13 hours, 14 minutes, and 27.1459 seconds
total agents: 39679
current agents: 39679
interval cell time: 0 days, 8 hours, 4 minutes, and 27.3195 seconds
```

PhysiCell
simulation



Simulation
data

The Jupyter notebook and executable can be **cloud-hosted as an app**. This allows **model sharing** for broad audiences.

Use case: "Try this model yourself!"

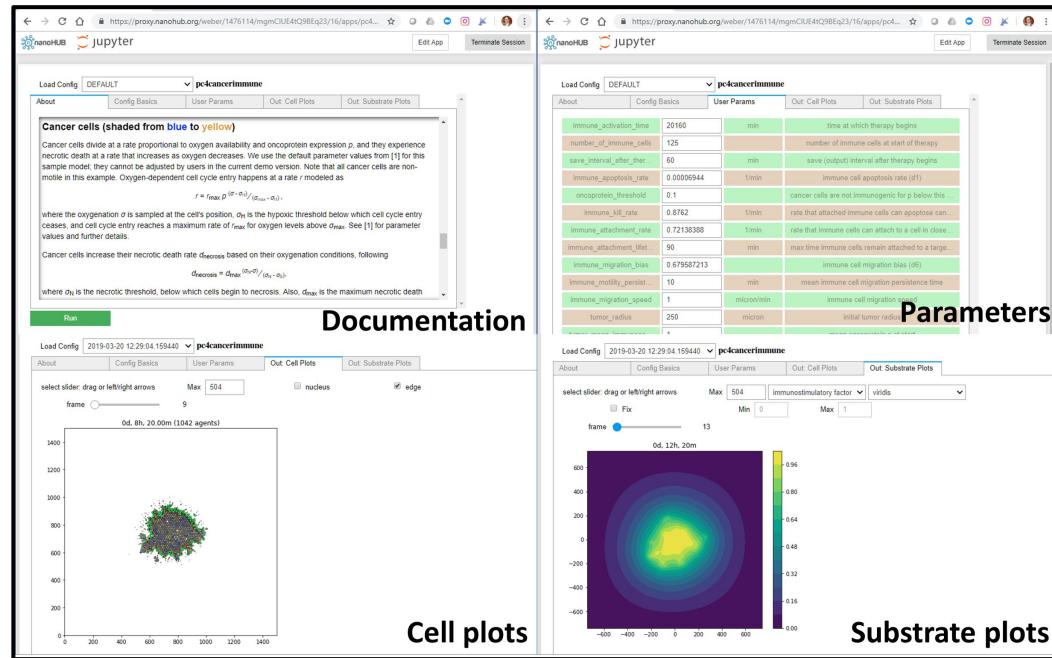
- With `xml2jupyter`, we can automatically create a Jupyter-based GUI for a PhysiCell model, and host it on nanoHUB as an interactive model.
- The apps can easily be **included in talks, posters, and presentations.**
- We include ***publication companion apps*** in every paper to help readers explore and understand the method.

This should be standard practice.



Try this model yourself! (2D)

nanohub.org/tools/pc4cancerimmune

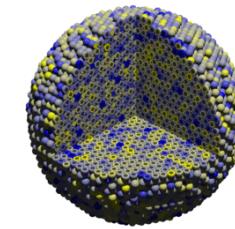
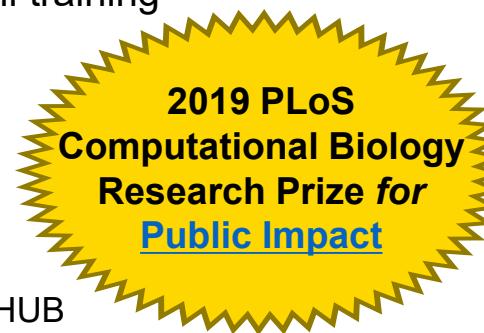


A Year of PhysiCell: training and open source with the NCI

Thanks to the NCI, we are ramping up PhysiCell training across the CSBC / PSON network

- **Winter 2019:** All-new training materials
 - A series of ~10-15 minute training modules
 - ◆ Embedded webapps to illustrate key concepts
 - New website and wiki at PhysiCell.org
 - Basic online development environment on nanoHUB
- **Early Summer 2020:** funded hackathon at IU
 - Contribute to the PhysiCell ecosystem
- **Summer 2020:** 2 funded 6-week visitors to IU
 - Likely recruited from the hackathon participants
 - Build a model or new PhysiCell capability
 - Learn to create PhysiCell-powered nanoHUB apps

PhysiCell immunotherapy model



Try this model online!
nanohub.org/tools/pc4cancerimmune

PhysiCell open source framework:

- 5+ diffusing chemical factors
- Highly customizable cell agents
- Works on Linux, OSX, Windows, ...
- Built for high-throughput studies on HPC
- Share models online via nanoHUB

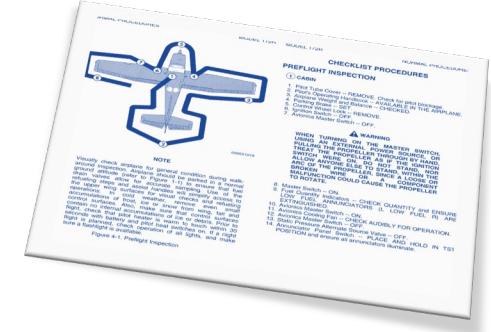
Laboratory 5: Learning Objectives

- Modify, build and run pre-bundled PhysiCell models
- Load, process, and visualize data in Python
- Create animations and movies from simulation data
- Create new models "from scratch"
- Create Jupyter notebooks as graphical user interfaces (GUIs)
- Share interactive simulations online as cloud-hosted models
- Help shape the future of PhysiCell in development priorities
- Survey future community opportunities, including NCI-funded hackathon

Preflight Checklist

- We created a "preflight checklist" of software to install before the lab.
 - This includes detailed tutorials on how to set up the C++ compiler.

[Link:](https://github.com/physicell-training/UCI-sysbio-2020) github.com/physicell-training/UCI-sysbio-2020



Further reading (1)

- **BioFVM method paper (3-D diffusion)**

A. Ghaffarizadeh, S.H. Friedman, and P. Macklin. BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations. *Bioinformatics* 32(8):1256-8, 2016. DOI: [10.1093/bioinformatics/btv730](https://doi.org/10.1093/bioinformatics/btv730).

- **PhysiCell method paper (agent-based model)**

A. Ghaffarizadeh, R. Heiland, S.H. Friedman, S.M. Mumenthaler, and P. Macklin. PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems. *PLoS Comput. Biol.* 14(2):e1005991, 2018. DOI: [10.1371/journal.pcbi.1005991](https://doi.org/10.1371/journal.pcbi.1005991).

- **PhysiBoSS (PhysiCell + MaBoSS for Boolean networks)**

G. Letort, A. Montagud, G. Stoll, R. Heiland, E. Barillot, P. Macklin, A. Zinov'yev, and L. Calzone. PhysiBoSS: a multi-scale agent based modelling framework integrating physical dimension and cell signalling. *Bioinformatics* 35(7):1188-96, 2019. DOI: [10.1093/bioinformatics/bty766](https://doi.org/10.1093/bioinformatics/bty766).

- **xml2jupyter paper (create GUIs for cloud-hosted models)**

R. Heiland, D. Mishler, T. Zhang, E. Bower, and P. Macklin. xml2jupyter: Mapping parameters between XML and Jupyter widgets. *Journal of Open Source Software* 4(39):1408, 2019. DOI: [10.21105/joss.01408](https://doi.org/10.21105/joss.01408).

- **PhysiCell+EMEWS (high-throughput 3D PhysiCell investigation)**

J. Ozik, N. Collier, J. Wozniak, C. Macal, C. Cockrell, S.H. Friedman, A. Ghaffarizadeh, R. Heiland, G. An, and P. Macklin. High-throughput cancer hypothesis testing with an integrated PhysiCell-EMEWS workflow. *BMC Bioinformatics* 19:483, 2018. DOI: [10.1186/s12859-018-2510-x](https://doi.org/10.1186/s12859-018-2510-x).

- **PhysiCell+EMEWS 2 (HPC accelerated by machine learning)**

J. Ozik, N. Collier, R. Heiland, G. An, and P. Macklin. Learning-accelerated Discovery of Immune-Tumour Interactions. *Molec. Syst. Design Eng.* 4:747-60, 2019. DOI: [10.1039/c9me00036d](https://doi.org/10.1039/c9me00036d).

Further reading (2)

- **A review of cell-based modeling (in cancer):**

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).

- **Progress on multicellular systems biology:**

P. Macklin, H.B. Frieboes, J.L. Sparks, A. Ghaffarizadeh, S.H. Friedman, E.F. Juarez, E. Jockheere, and S.M. Mumenthaler. "Progress Towards Computational 3-D Multicellular Systems Biology". In: . Rejniak (ed.), *Systems Biology of Tumor Microenvironment*, chap. 12, pp. 225-46, Springer, 2016. ISBN: 978-3-319-42021-9. (invited author: P. Macklin). DOI: [10.1007/978-3-319-42023-3_12](https://doi.org/10.1007/978-3-319-42023-3_12).

- **Challenges for data-driven multicellular systems biology**

P. Macklin. Key challenges facing data-driven multicellular systems biology. *GigaScience* 8(10):giz127, 2019. DOI: [10.1093/gigascience/giz127](https://doi.org/10.1093/gigascience/giz127)