PhysiCell Essentials Short Course:

Introduction to Agent-Based Modeling and PhysiCell

Paul Macklin, Ph.D.

Intelligent Systems Engineering
Indiana University

Updated: 2025.02.16

PhysiCell Curriculum

- **PhysiCell Essentials Short Course (this short course)**
  - **Prerequisites:**
    - Basic knowledge of cell biology, concepts of mathematical functions
  - **Software requirements:**
    - Web browser access, OR installation of PhysiCell Studio
  - **Curriculum:**
    - **Introduction (this session)**
    - *Optional: Desktop Installation of PhysiCell Studio*
    - Hands-on work Part 1: Getting Started, and Villager/Zombie Model
    - Hands-on work Part 2: Cancer Chemotherapy & Immunology Models
    - *Optional: Notes and Tips on Parameter Estimates*
- **Integration of Boolean Networks with PhysiBoSS**
  - Learn how to integrate Boolean signaling networks into PhysiCell Models
- **Advanced PhysiCell Modeling**
  - Learn about creating non-standard model components and visualization in C++
  - Learn about C++ extensions for ODE models, ECM fibers, and more.
- **PhysiCell for Developers**
  - Learn about PhysiCell core C++ structure, and contributing to PhysiCell core and extensions

Session Goals

- Introduce multicellular systems biology
- Introduce concepts of **signal-response** cell interactions
- Introduce agent-based models (ABMs)
- Introduce key components of the PhysiCell modeling framework
- Show examples of agent-based modeling
- Introduce next-generation concepts for agent-based (bio)modeling

From single cells to tissue ecosystems

- **Single-cell behaviors:**
  - Growth
  - Division
  - Differentiation
  - Death
  - Consumption
  - Metabolism
  - Secretion
  - Signaling
  - Mutations
  - Variability
  - Damage response
  - Motility
- **Cell-cell interactions:**
  - Adhesion
  - Mechanics
  - Predation / Phagocytosis
  - Effector attack
  - Fusion
  - Contact communication
- **Physical constraints:**
  - Diffusion limits
  - Mechanical barriers

Multicellular tissue 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

Scientists use [models*] to detangle complex systems.

*** animal, *in vitro*, engineered, mathematical, conceptual …**

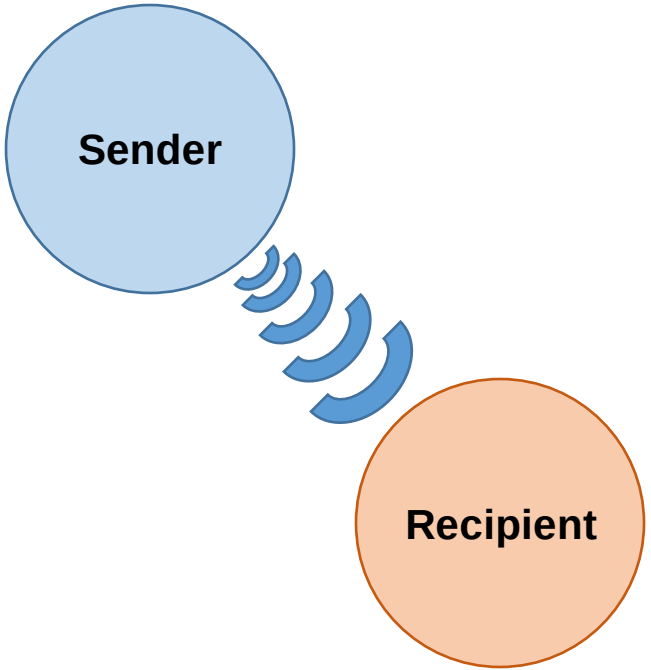
We use agent-based models as our virtual laboratory.

First, a conceptual model

Signal-Response as a Conceptual Framing

- Much of the complexity of this system can be decomposed into pairwise interactions between a ***sender*** and a ***recipient***

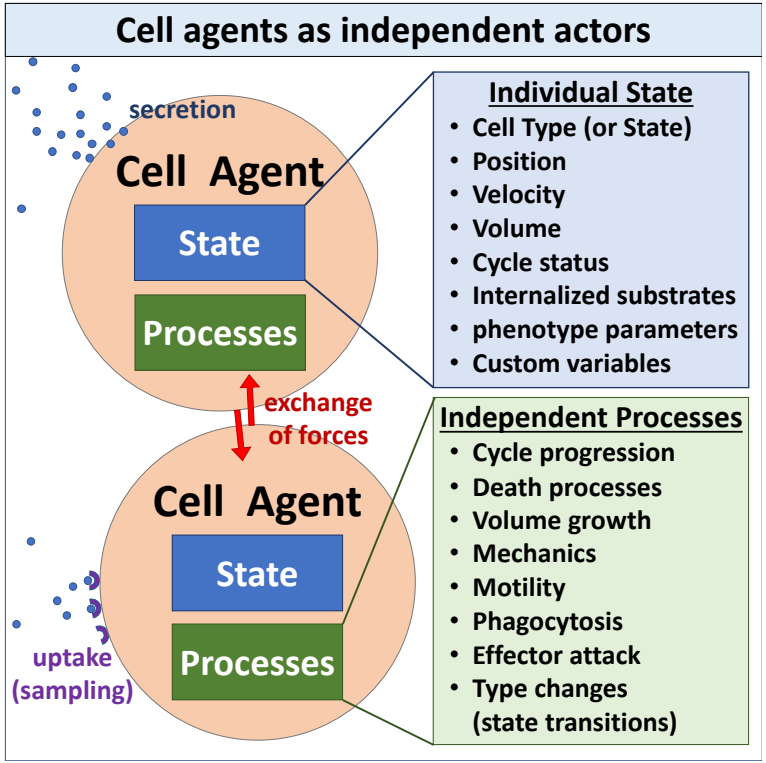
- A **signal** is a stimulus that can elicit a behavioral **response** in the recipient**:**
  - A macrophage (**sender**) secretes IL-6 (**signal**) that drives chemotaxis (**response**) in a CD8 T cell (**recipient**)
  - An epithelial cell (**sender**) exerts pressure (**signal**) that decreases cycle entry (**response**) in another epithelial cell (**recipient**)



Agent-based models are well-suited to this framing

Agent-based models: overview

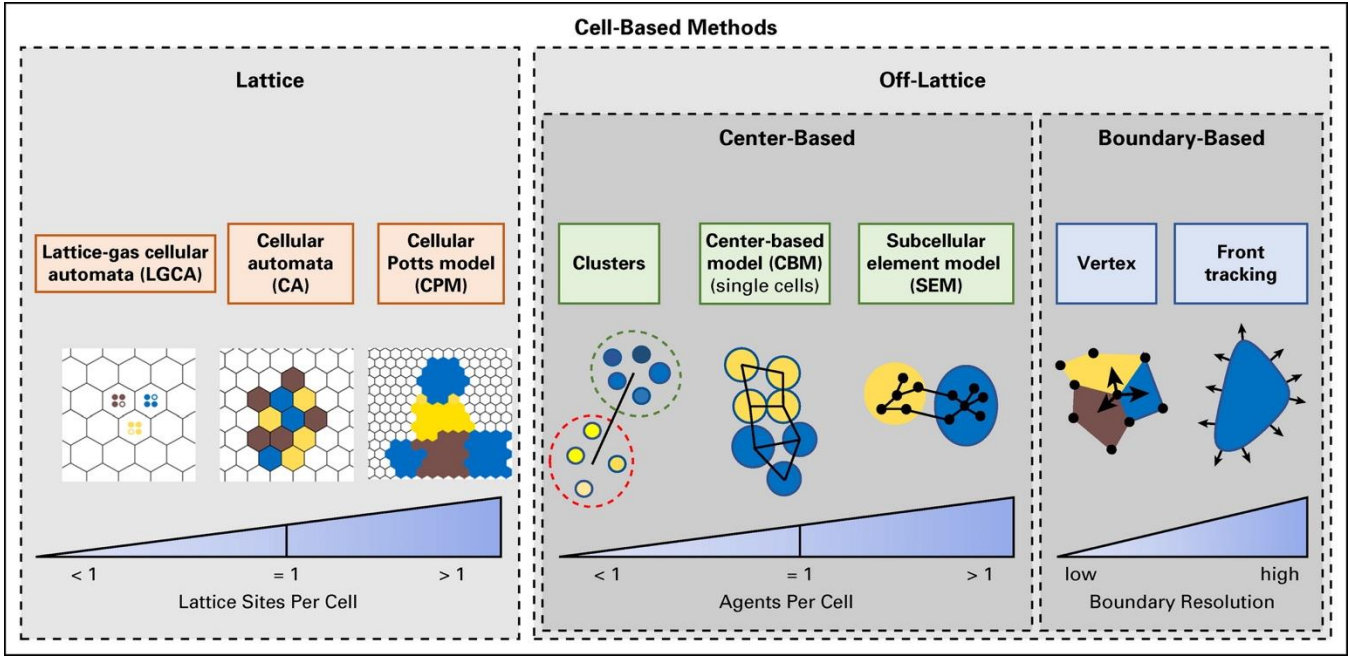
- Each cell is an **independent agent** with:
  - **Individual state**
    - Type
    - Position
    - Velocity
    - Phenotype parameters
    - Custom variables
  - **Independent processes**
    - Cycle and death processes
    - Volume growth
    - Mechanics and motility
    - Secretion and uptake / sampling
    - Phagocytosis, effector attack
    - State transitions (change of type)
    - Custom processes



Types of cell-based models

- **lattice-bound:**
  cell position and sizes align with a regular lattice
  - resolution:
    - many cells / site:
      - » lattice gas
    - 1 site / cell
      - » cellular automaton
    - many sites / cell
      - » cellular Potts

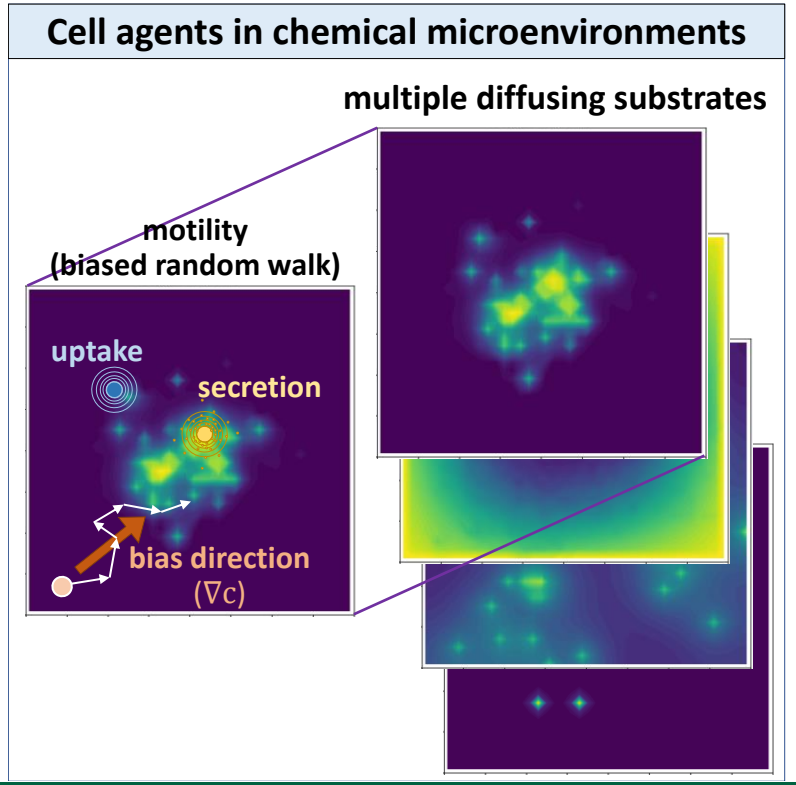
- **off-lattice:**
  cells can take any position and size
  - **center-based**
  - boundary-based



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.

Cell agents live in a virtual environment

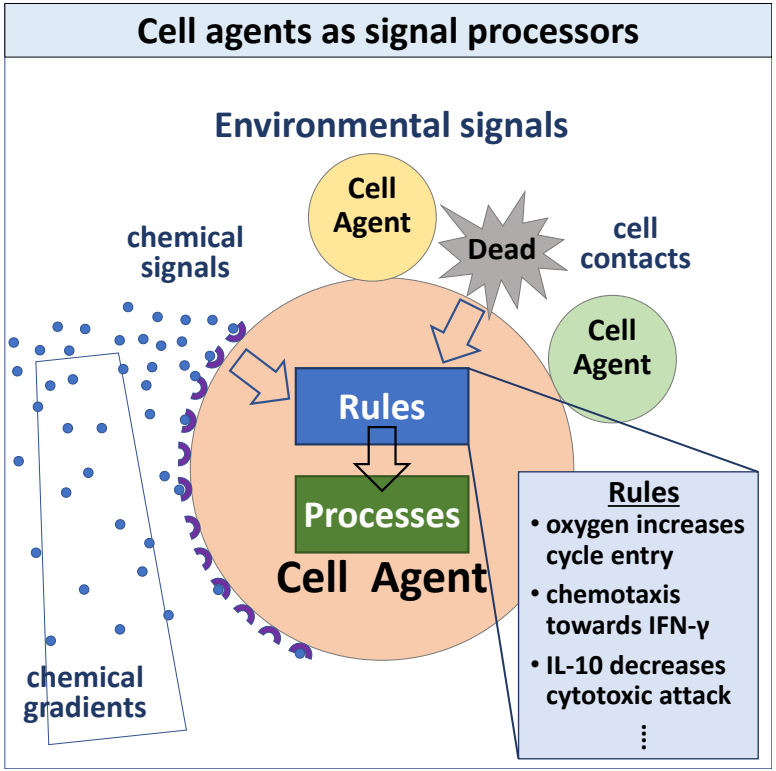
- Cells can secrete or consume chemical substrates
- Substrates diffuse and decay
- Cells can sample substrates
- Cells can perform biased random walks that may align with gradients (e.g., chemotaxis)
- Can also add additional mechanical detail (e.g., viscoelastic ECM models)



Cell agents are signal processors

- Cells interact through chemical and physical **signals** (or stimuli)
  - Secreted chemical signals & gradients
  - ECM properties
  - Contact with a live or dead cell

    …
- Signals drive changes in **behavior**
  - Increased or decreased rates of cycling or death
  - Changes in motility, secretion, phagocytosis, …
- Signal-behavior relationships are **agent rules**



Key modeling step: b = f(s)

- A *key modeling step* for ABMs is defining the functional relationship ***f*** between a set of signals **s** and a set of behaviors **b:**

$$\mathbf{b} = \boldsymbol{f}(\mathbf{s})$$

- Traditionally, ABMs write ***f*** as custom code.

- Boolean networks, ODEs, FBA, and NN models are sophisticated forms of ***f.***

- All these functional relationships require:
  - Mapping quantities in the ABM framework to inputs of ***f***
  - Computing ***f***
  - Mapping outputs of ***f*** to parameters in the ABM

- More recently, we defined an intermediate level ***f*** via a grammar.
  - **Most models can be written in this grammar, making modeling faster, reusable, and reproducible.**

Parallel Notions to Continuum Models

- **Continuum models:**
  - A general form applies to a broad class of problems.
    - ***Example:*** Conservation of mass, momentum, and energy
  - Constitutive laws (extra hypotheses) adapts the general form to *specific* problems.
    - ***Example:*** Darcy's law for pressure-driven flows

- **Agent-based models:**
  - Agents have general forms for cycling, death, secretion, migration, ….
  - Extra (time & space-dependent) rules trigger and regulate these core (sub)-models
    - These are the **constitutive laws** for ABMs.

Our virtual laboratory

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 μm resolution

Features:

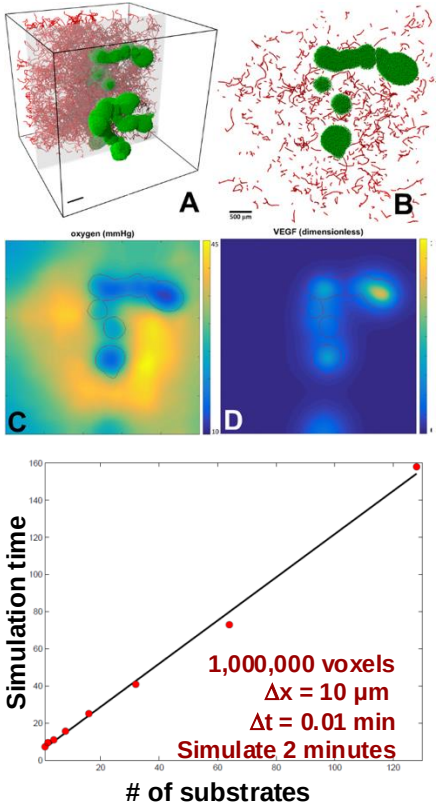
- Off-lattice cell secretion and uptake
- 2^{nd}-order accurate (space), 1^{st}-order accurate (time), numerically stable

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 10+ diffusing substrates on 10^6 voxels

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

DOI: 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:

- Fully coupled diffusion solvers
- Mechanics-based cell movement
- Cell processes (cycling, motility, …)
- Signal-dependent phenotype
- Can dynamically attach custom data and functions on a cell-by-cell basis

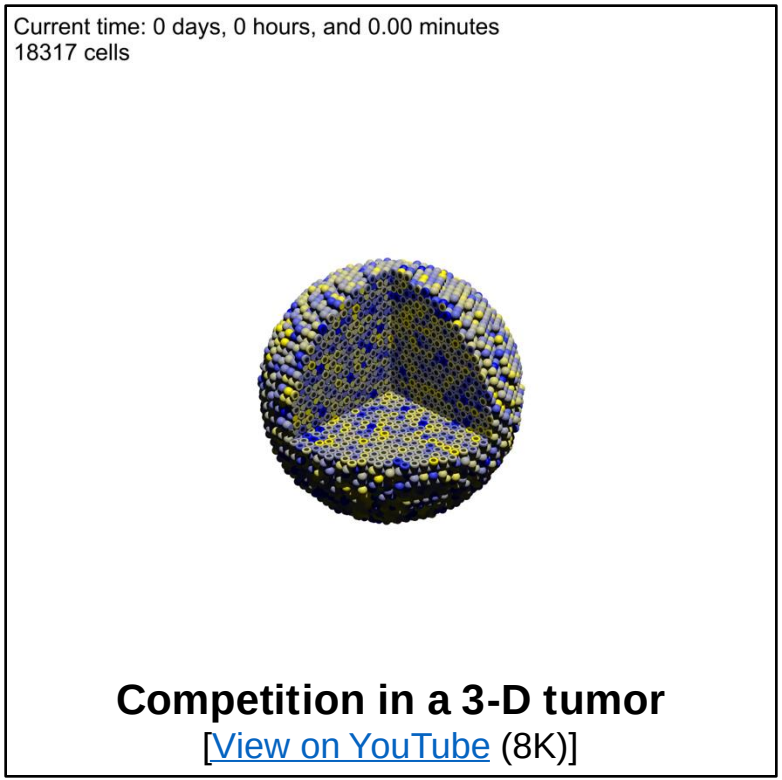
2019 PLoS Computational Biology Research Prize *for* Public Impact

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

Try this model yourself!
nanohub.org/tools/pc4heterogen



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

PhysiBoSS: PhysiCell + MaBoSS

Design goal: Directly integrate Boolean signaling networks in each cell agent

MaBoSS (from Institut Curie):

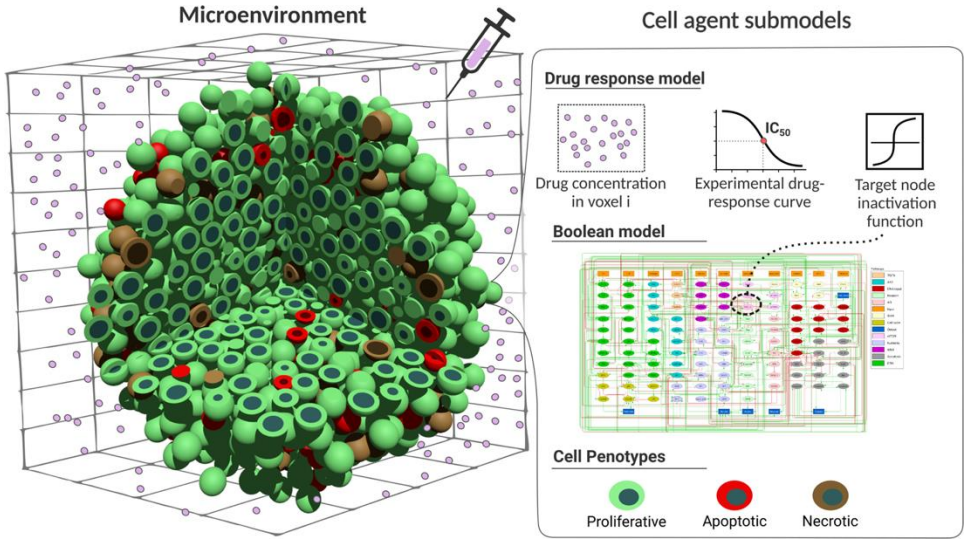
- Continuous-time Markovian simulator for Boolean models
- Describe the cell's intracellular signaling and regulatory networks.

Method:

- Each PhysiCell agent has a MaBoSS model and data
- PhysiCell sends cell and tissue data to MaBoSS as inputs
- MaBoSS advances solution a fixed time
- MaBoSS sends outputs to key PhysiCell agent parameters

Reference 1: Letort et al., Bioinformatics (2019)
DOI: 10.1093/bioinformatics/bty766

Reference 2: Ponce-de-Leon et al., npj Sys. Biol. Appl. (2023)
DOI: 10.1038/s41540-023-00314-4



PhysiBoSS simulation of combination therapies in LNCaP

PhysiCell as a virtual laboratory

- Choose important chemical signals
  - These become diffusible fields
- Choose important cell types
  - These become our cell definitions
- Clearly state our biological hypotheses as signal-response statements
  - These become our agent rules
- Perform virtual experiments to ask ***what if*** questions
  - What hypotheses does it take to match reality?
    - Which rules are the most important?
  - Which rules (and parameters) can be tuned to steer the system?

Rocha et al., *iScience* (2021)
DOI: 10.1016/j.isci.2021.102935

Example:

Exploring phenotypic persistence in hypoxic breast cancer

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

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

Download PDF

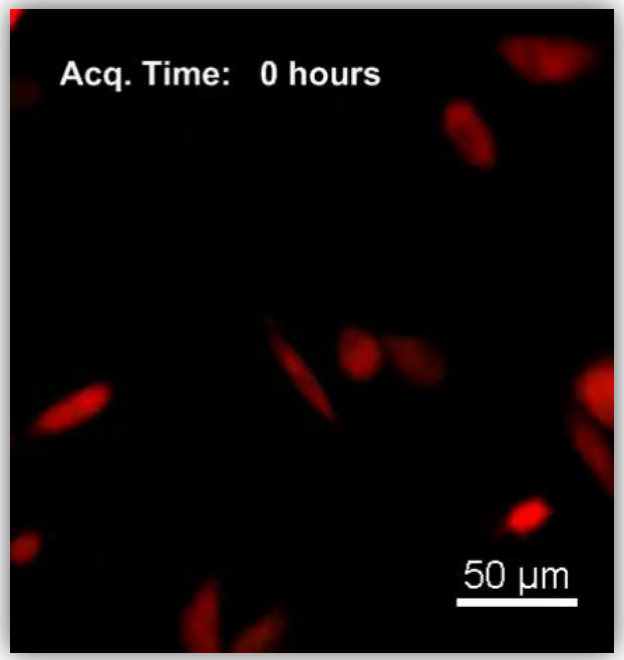
Sections | Figures | References

Abstract

What are the rules of hypoxic cancer cells after they escape hypoxia?

Do they resume their old program?

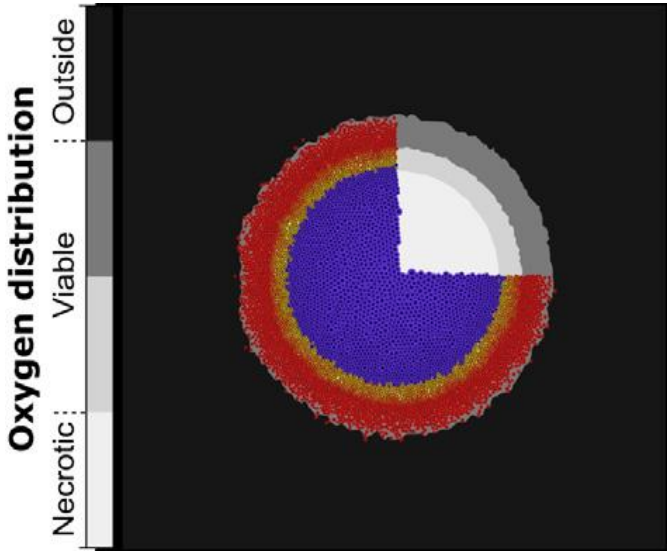
How soon?



Daniele Gilkes Lab, Johns Hopkins

Model overview

- Simulate oxygen diffusion and uptake
- Proliferation and necrosis vary with pO_2 and mechanical pressure
- Live cells are **normoxic (RFP)** or **hypoxic (GFP)**.
- Model transition from **RFP** to **GFP** via ODEs
- **GFP** cells migrate up pO_2 gradients
  - **Phenotypic persistence:** How long do **GFP** cells keep their migratory behavior after leaving hypoxic regions



Phenotypic persistence drives invasion

Phenotypic Persistence:

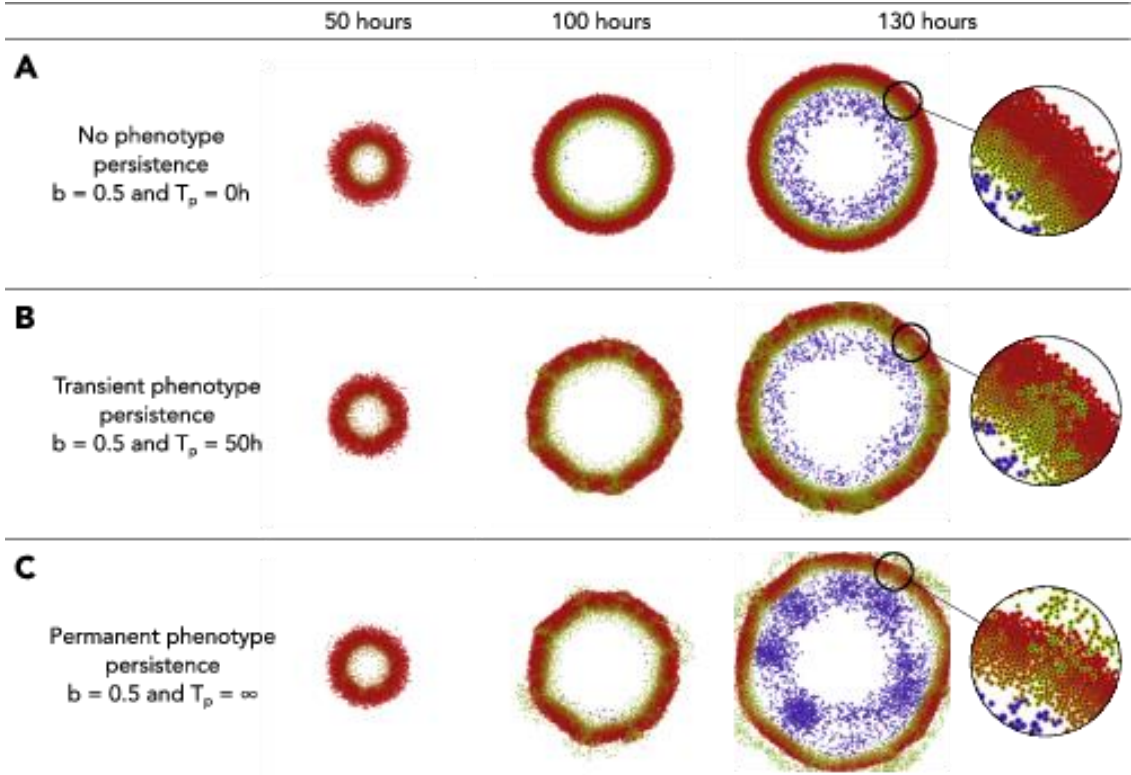
- T_p : duration of hypoxic response

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

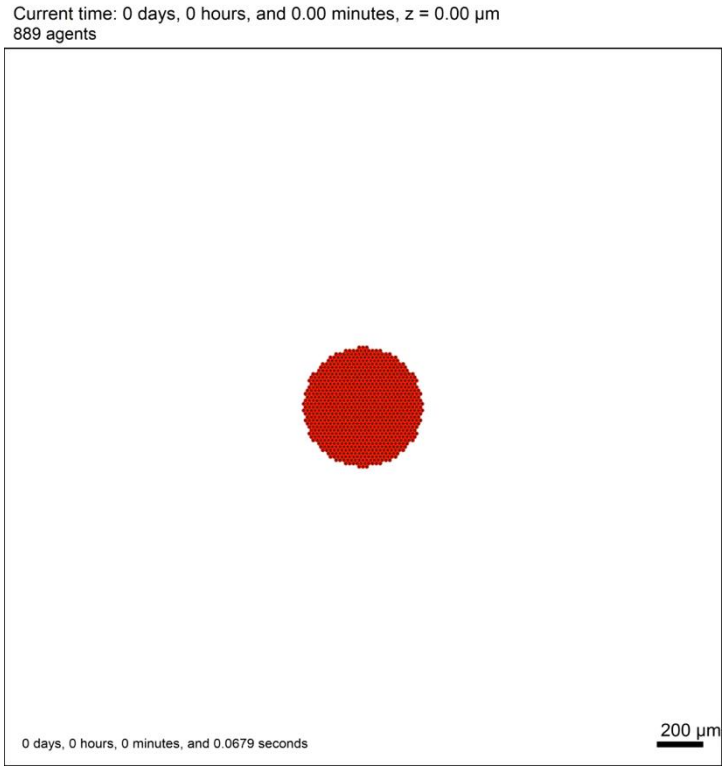
- Migration halts at perinecrotic boundary
- Tumors maintain a concentric structure:
  - Oxygenated viable rim (red)
  - hypoxic (or formerly hypoxic) annulus (green)
  - Necrotic core (purple)

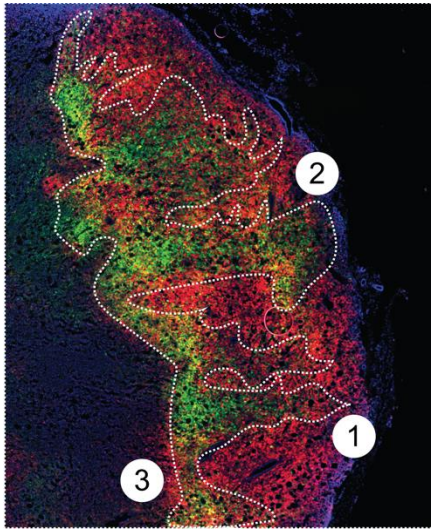
With persistence ($T_p > 0$) – Rows B & C

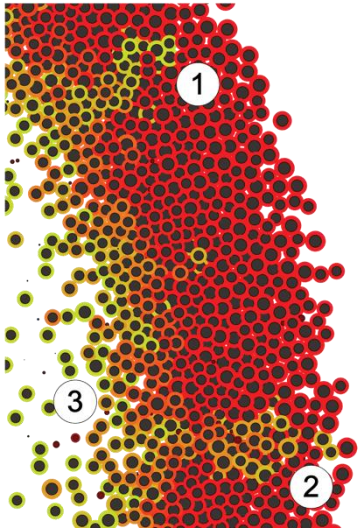
- Hypoxic cells can continue migrating
- Hypoxic cells "punch through" the oxygenated tumor region
- Cell act individually, but it *looks* like collective behavior.
  - Risk of over-interpreting single snapshots!



Mathematical model explains biological observations







Try this model yourself!

nanohub.org/tools/pc4tumorhypoxia

Takeaway:

Simulation models + novel imaging can explain biology better together than separately

Rethinking modeling

Key computational modeling steps

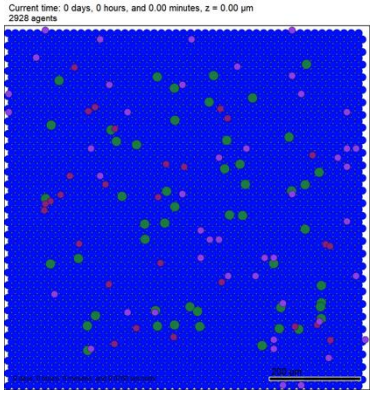
1. Formulate hypotheses:
   - How do biophysical signals drive cell behaviors?
   - Requires a conversation between biologists and mathematicians

2. Transform hypotheses into mathematics
   - Typically performed manually for each hypothesis

3. Implement mathematical statements as code
   - C++, Python, Java, …
   - Typically **hand-written code**

Sample: COVID-19 macrophage model

- Macrophage hypotheses
  - 5.MPhi.1 Resident (unactivated) and newly recruited macrophages move along debris gradients.
  - 5.MPhi.2 Macrophages phagocytose dead cells. Time taken for material phagocytosis is proportional to the size of the debris
  - 5.MPhi.3 Macrophages break down phagocytosed materials
  - 5.MPhi.4 After phagocytosing dead cells, macrophages activate and secrete pro-inflammatory cytokines
  - 5.MPhi.5 Activated macrophages can decrease migration speed
  - 5.MPhi.6 Activated macrophages have a higher apoptosis rate
  - 5.MPhi.7 Activated macrophages migrate along chemokine and debris gradients
  - 5.MPhi.8 Macrophages are recruited into tissue by pro-inflammatory cytokines
  - 5.MPhi.9 Macrophages can die and become dead cells only if they are in an exhausted state
  - 5.MPhi.10 Macrophages become exhausted (stop phagocytosing) if internalised debris is above a threshold
  - 5.MPhi.11 $CD8^+$ T cell contact stops activated macrophage secretion of pro-inflammatory cytokine and switches to M2 phase, secreting anti-inflammatory cytokine.
  - 5.MPhi.12 $CD4^+$ T cell contact induces activated macrophage phagocytosis of live infected cell



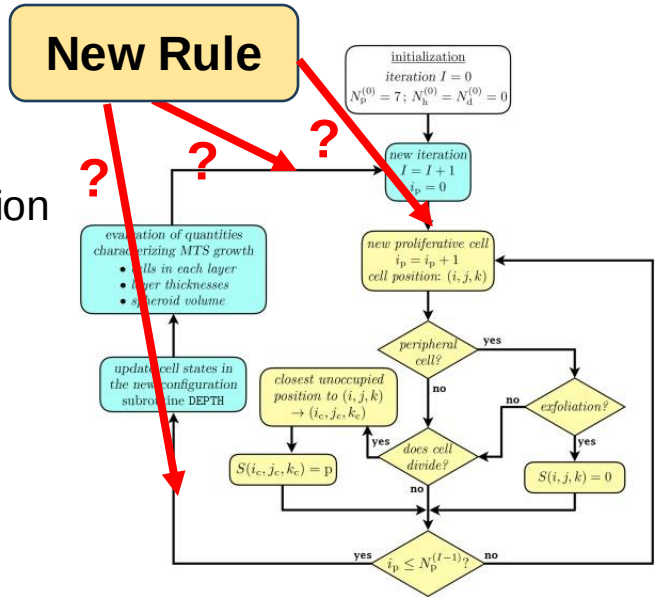
PhysiCell model of COVID19

These hypotheses become hand-coded functions in C++.

Problems with hand-written models

- Many models re-implement recurring elements
  - Does not leverage prior modeling
  - Increases likelihood of errors
  - **Large coding effort** discourages multidisciplinary participation
  - Variations in implementation add complexity to interpretation

- **Perhaps most importantly, as complexity grows:**
  - Hard to understand the full model
  - Hard to communicate the current biological hypotheses
  - Hard to integrate new biological hypotheses
  - Hard for domain experts to participate in real time



DOI: 10.1016/j.ejmp.2020.07.026

Code profiling: identify bottlenecks

- Software analysis: code profiling:
  - For a simulation run, where do we spend the most time?
  - Use this to focus optimization

- Profiling by Sunita Chandrasakaran's group (U. Delaware):
  - 65% of computation time is spent on diffusion
  - If we can accelerate diffusion 10x:

$$\text{Time}_{\text{new}} = \left(\frac{1}{10} \cdot 0.65 + 0.35\right) \text{Time}_{\text{old}} \approx 0.42\ \text{Time}_{\text{old}}$$

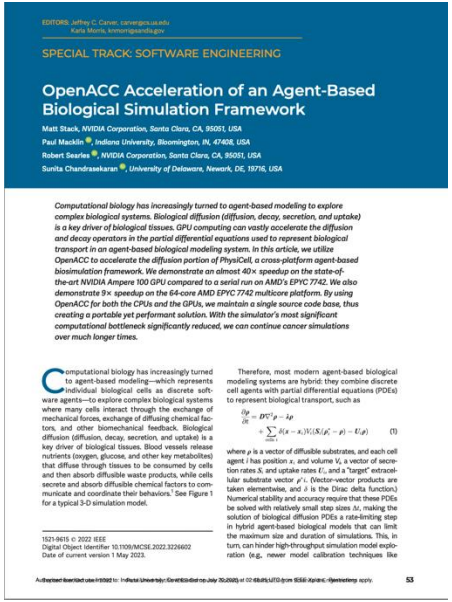
  - If we can accelerate diffusion 100x:

$$\text{Time}_{\text{new}} = \left(\frac{1}{100} \cdot 0.65 + 0.35\right) \text{Time}_{\text{old}} \approx 0.36\ \text{Time}_{\text{old}}$$

  - If we can accelerate diffusion 1000x:

$$\text{Time}_{\text{new}} = \left(\frac{1}{1000} \cdot 0.65 + 0.35\right) \text{Time}_{\text{old}} \approx 0.35\ \text{Time}_{\text{old}}$$

- Notice the **rapidly diminishing returns!** **Key lessons:**
  - Once the bottleneck is gone, move on to the next one!
  - This is the economics of code optimization! (Decreasing marginal utility)



EDITORS: Jeffrey C. Carver, carver@cs.ua.edu
Karla Morris, kmorris@sandia.gov

SPECIAL TRACK: SOFTWARE ENGINEERING

OpenACC Acceleration of an Agent-Based Biological Simulation Framework

Matt Stack, *NVIDIA Corporation, Santa Clara, CA, 95051, USA*
Paul Macklin, *Indiana University, Bloomington, IN, 47408, USA*
Robert Searles, *NVIDIA Corporation, Santa Clara, CA, 95051, USA*
Sunita Chandrasekaran, *University of Delaware, Newark, DE, 19716, USA*

Computational biology has increasingly turned to agent-based modeling to explore complex biological systems. Biological diffusion (diffusion, decay, secretion, and uptake) is a key driver of biological tissues. GPU computing can vastly accelerate the diffusion and decay operators in the partial differential equations used to represent biological transport in an agent-based biological modeling system. In this article, we utilize OpenACC to accelerate the diffusion portion of PhysiCell, a cross-platform agent-based biosimulation framework. We demonstrate an almost 40× speedup on the state-of-the-art NVIDIA Ampere 100 GPU compared to a serial run on AMD's EPYC 7742. We also demonstrate 9× speedup on the 64-core AMD EPYC 7742 multicore platform. By using OpenACC for both the CPUs and the GPUs, we maintain a single source code base, thus creating a portable yet performant solution. With the simulator's most significant computational bottleneck significantly reduced, we can continue cancer simulations over much longer times.

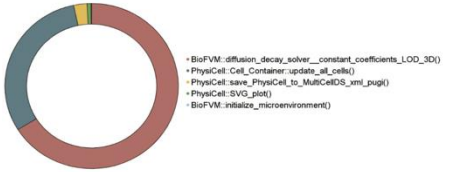
Computational biology has increasingly turned to agent-based modeling—which represents individual biological cells as discrete software agents—to explore complex biological systems where many cells interact through the exchange of mechanical forces, exchange of diffusing chemical factors, and other biomechanical feedback. Biological diffusion (diffusion, decay, secretion, and uptake) is a key driver of biological tissues. Blood vessels release nutrients (oxygen, glucose, and other key metabolites) that diffuse through tissues to be consumed by cells and then absorb diffusible waste products, while cells secrete and absorb diffusible chemical factors to communicate and coordinate their behaviors.[1] See Figure 1 for a typical 3-D simulation model.

Therefore, most modern agent-based biological modeling systems are hybrid: they combine discrete cell agents with partial differential equations (PDEs) to represent biological transport, such as

$$\frac{\partial \rho}{\partial t} = D\nabla^2\rho - \lambda\rho + \sum_{\text{cells } i} \delta(x - x_i)V_i(S_i(\rho_i^* - \rho) - U_i\rho) \quad (1)$$

where ρ is a vector of diffusible substrates, and each cell agent i has position x_i and volume V_i, a vector of secretion rates S_i and uptake rates U_i, and a "target" extracellular substrate vector ρ_i^*. (Vector–vector products are taken elementwise, and δ is the Dirac delta function.) Numerical stability and accuracy require that these PDEs be solved with relatively small step sizes Δt, making the solution of biological diffusion PDEs a rate-limiting step in hybrid agent-based biological models that can limit the maximum size and duration of simulations. This, in turn, can hinder high-throughput simulation model exploration (e.g., newer model calibration techniques like

1521-9615 © 2022 IEEE
Digital Object Identifier 10.1109/MCSE.2022.3226602
Date of current version 1 May 2023.



Let's generalize this analysis

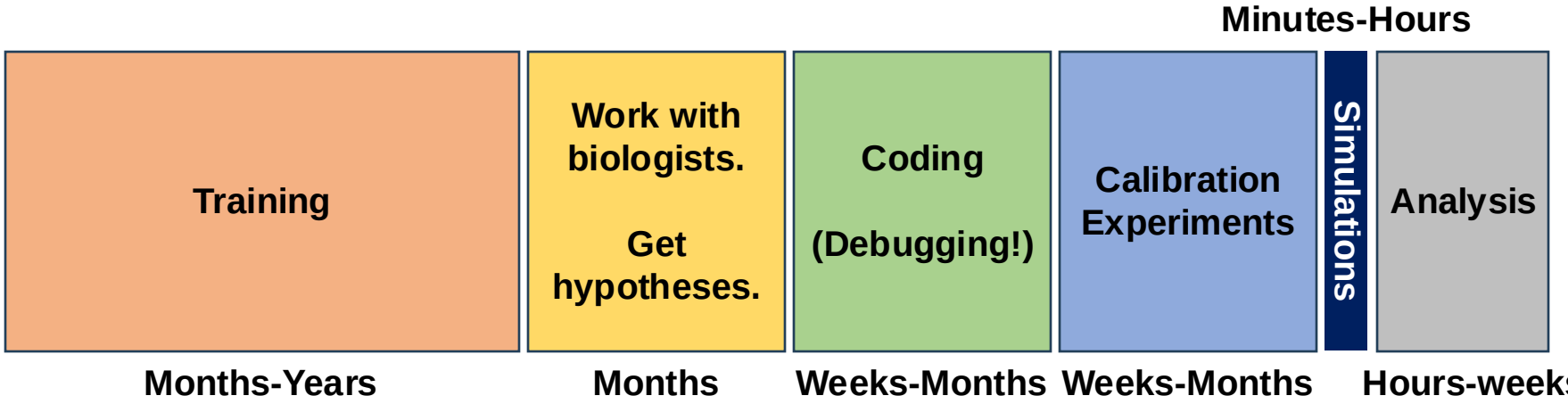
1. Identify the biggest bottleneck

2. Improve that speed by 1-2 orders of magnitude, but no more!

3. After that, move on to the next bottleneck.

"Code profiling" for scientific projects

- Let's apply this to scientific projects.



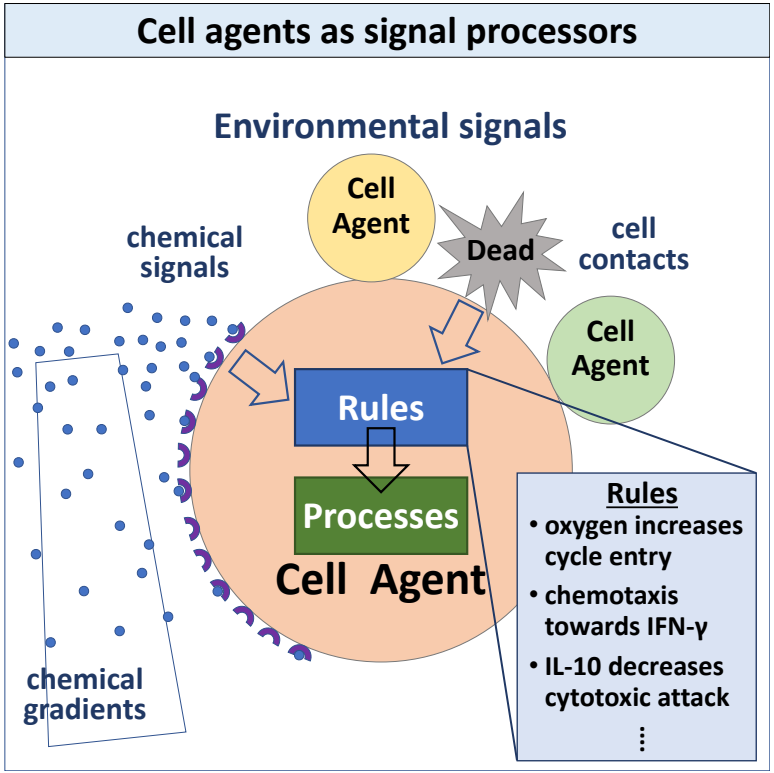
- Improving **simulation speed** speeds up investigations.
- We also need to speed up the **learning and development bottlenecks**!

Creating a computable model grammar

- **Goal:** Create a language for cell 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
  - Can easily integrate new knowledge with prior knowledge
  - Can combine data-driven and knowledge-driven workflows

Key elements for a computable model grammar

- A "dictionary" of signals (stimuli)
- A "dictionary" of reference behaviors
- A grammar to connect signals to behavioral responses
- Map grammar statements onto mathematics and code



A dictionary of signals

- A dictionary of signals that can be used as inputs for hypothesis statements

- **Diffusible chemical substrates**
  - extracellular and intracellular concentrations
  - extracellular gradients

- **Cell mechanics / physics**
  - Cell pressure
  - Cell volume

- **Contact**
  - Number of contacts with each cell type
  - # of contacts with live and dead cells
  - Contact with basement membrane

- **Attack and damage interactions**
  - Accumulated damage
  - Cumulative attack time
  - Attack status
    - Am I attacking?
    - How much damage have I delivered?

- **Live / dead status**
  - Dead, apoptotic, necrotic

- **Global information**
  - Current simulation time (for event timing)

- **Custom symbols**

Each **symbol uniquely maps to a mathematical quantity** at a cell's position

A dictionary of behaviors

- Based on years of modeling, we created a "dictionary" of standardized behaviors ***and well-defined reference models***

- **Cycling**
  - Exit rates from each cycle phase
  - Asymmetric division
- **Death**
  - Apoptotic and necrotic death rates
- **Transport**
  - Secretion, uptake, and export rates
- **Migration and chemotaxis**
  - Migration speed, bias, persistence time
  - Chemotactic sensitivities (to each diffusible factor)
- **Mechanics and Adhesion**
  - Adhesion and repulsion potential coefficients
  - Adhesion affinities (to each cell type)
  - Elastic adhesion constant, maximum number of adhesions
  - Rate of forming and breaking elastic adhesions
- **Transition / Type change / Transformation**
  - Rate of transforming (to each cell type)
- **Fusion**
  - Rate of fusing (combining with) each cell type
- **Phagocytosis (or ingestion / predation)**
  - Rate of ingesting dead cells
    - Apoptotic & Necrotic rates can differ!
  - Rate of ingesting live cells (one rate for each type)
- **Effector Attack**
  - Rate of initiating attacks on live cells (one for each type),
  - Immunogenicity (one for each cell type)
  - Duration of attacks, rate of causing damage during attack
- **Damage and Repair**
  - Damage rate (e.g., via doxorubicin)
  - Repair rate
- **Custom symbols**

Each **symbol uniquely maps to a mathematical parameter** in a reference process model.

Example Reference Behavior Models

- **Migration**
  - Choose migration bias $\mathbf{d}_{\text{bias}}$:
    - Uses chemotactic sensitives (s_i) to each chem gradient (∇c_i):

$$\boldsymbol{d}_{\text{bias}} = \frac{\sum_i s_i \, \nabla c_i}{\left| \sum_i s_i \, \nabla c_i \right|}$$

  - Choose migration direction:
    - Uses speed s, bias b, and bias direction $\mathbf{d}_{\text{bias}}$:

$$\boldsymbol{v}_{\text{migration}} = s \frac{(1-b)\boldsymbol{\xi} + b \, \boldsymbol{v}_{\text{bias}}}{\left| (1-b)\boldsymbol{\xi} + b \, \boldsymbol{v}_{\text{bias}} \right|}$$

  - Continue with persistence time T_{persist}:

$$\text{P}(\text{choose new dirction in } [t, t+\Delta t]) = \frac{\Delta t}{T_{\text{persist}}}$$

- **(Effector) Attack**
  - If attacker cell *i* is not attacking a cell
    - Determine whether to attack a neighbor *j:*
      - » Uses rate of cell *i* attacking cell of type *j* ($r_{A,ij}$)
      - » Uses immunogenicity of *j* to cell *i* (I_{ji})

$$\text{Prob}(i \text{ attacks } j \text{ in } [t, t+\Delta t]) = r_{A,ij} I_{ji} \, \text{dt}$$

  - While attacking:
    - Form mechanical adhesion (spring link)
    - Cause damage in target cell
      - » rate of causing attack damage ($r_{D,i}$)

$$\frac{dD_j}{dt} = r_{Di}$$

    - Determine whether to end attack:

$$\text{P}(\text{end attack in } [t, t+\Delta t]) = \frac{\Delta t}{T_{\text{attack}}}$$

  - In the target cell:
    - Requires a damage response rule ("Damage increases apoptosis")

Hypothesis statements

- For [cell type T], [S] increases / decreases [B] [optional arguments]
  - **Cell type T** is as cell type defined in the simulation model
  - **S** is a signal in our signal dictionary
  - **B** is a behavioral parameter in our behavior dictionary
- **Examples:**
  - For M0 macrophages, necrotic cell debris increases transformation to M1 macrophages
  - For malignant epithelial cells, doxorubicin increases apoptosis
  - Radioisotope edition:
    - For malignant epithelial cells, internalized radioisotope increases alpha particle secretion (release)
    - For malignant epithelial cells, alpha particles increase damage rate
    - For malignant epithelial cells, damage increases apoptosis

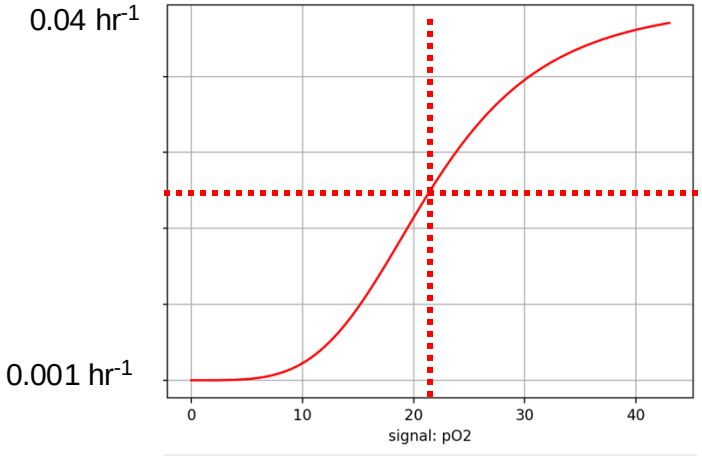
Mathematical Mapping

- If signal *S* increases / decreases behavior *B*
  - Vary behavioral parameter *p* with base value p_0 and maximal response p_{M}

$$p(s) = p_0 + (p_{\mathrm{M}} - \mathrm{p}_0)R(s) = \big(1 - R(s)\big) \cdot p_0 + R(s) \cdot p_M$$

  - We generally use **Hill response functions:**

$$R(s) = \frac{s^h}{s_{\mathrm{half}}^h + s^h} = \frac{\left(\frac{s}{s_{\mathrm{half}}}\right)^h}{1 + \left(\frac{s}{s_{\mathrm{half}}}\right)^h} \qquad \text{if } s \geq 0.$$



- **Example:** Oxygen increases cycle entry

$$r_{01} = 0.001 + (0.042 - 0.001)\frac{(\mathrm{pO}_2)^4}{21.5^4 + (\mathrm{pO}_2)^4}$$

Integrating many hypotheses

- **Multivariate Hill response functions**
  - Can integrate multiple signals with independent half-maxes and Hill powers
  - Reduce back down to original Hill function if all but one input is zero

- **Total up response:**

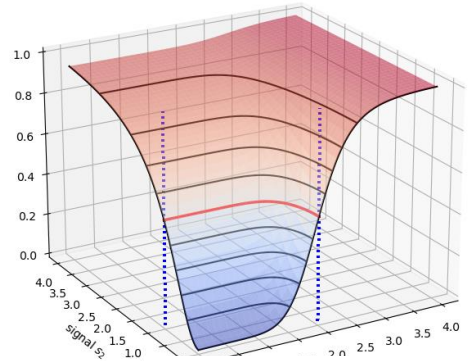
$$U = H_M(\boldsymbol{u}; \boldsymbol{u}_{\text{half}}, \boldsymbol{p}) = \frac{\left(\frac{u_1}{u_1^*}\right)^{p_1} + \left(\frac{u_2}{u_2^*}\right)^{p_2} + \cdots + \left(\frac{u_m}{u_m^*}\right)^{p_m}}{1 + \left(\frac{u_1}{u_1^*}\right)^{p_1} + \left(\frac{u_2}{u_2^*}\right)^{p_2} + \cdots + \left(\frac{u_m}{u_m^*}\right)^{p_m}}$$

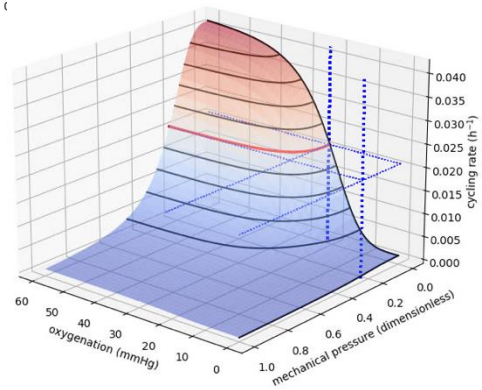
- **Total down response:**

$$D = H_M(\boldsymbol{d}; \boldsymbol{d}_{\text{half}}, \boldsymbol{q}) = \frac{\left(\frac{d_1}{d_1^*}\right)^{q_1} + \left(\frac{d_2}{d_2^*}\right)^{q_2} + \cdots + \left(\frac{d_n}{d_n^*}\right)^{q_n}}{1 + \left(\frac{d_1}{d_1^*}\right)^{q_1} + \left(\frac{d_2}{d_2^*}\right)^{q_2} + \cdots + \left(\frac{d_n}{d_n^*}\right)^{q_n}}.$$

- **Integrated response:**

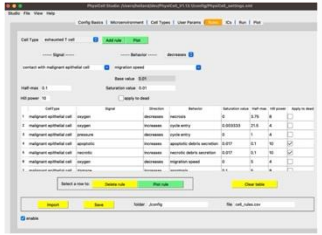
$$p(\boldsymbol{u}, \boldsymbol{d}) = (1 - D) \cdot [(1 - U) \cdot p_0 + U \cdot p_M] + D \cdot p_m$$



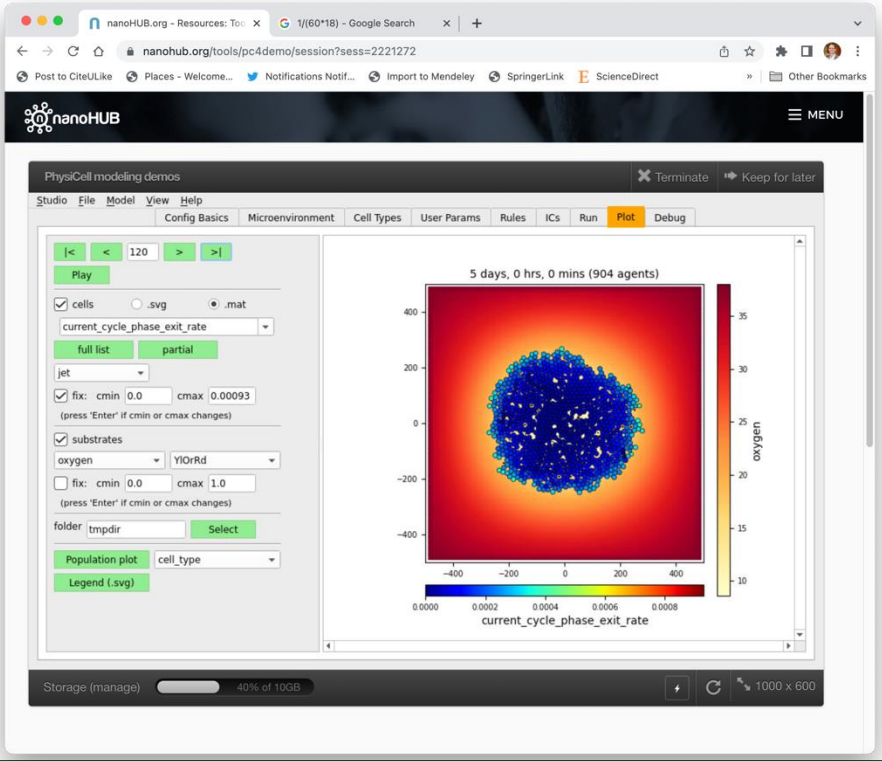


Building models *on-the-fly in the cloud*

- The language is constrained enough to create a data format.
- A fixed data format makes GUIs possible.
- We can bundle this as a cloud-hosted app.
  - http://nanohub.org/tools/pcstudio
- Now, **in real time:**
  - Choose cell types and diffusing factors
  - Write rules
  - Simulate and visualize
  - Ask biologist for feedback
  - Write more rules
  - Simulate, visualize, and repeat



The modeler-biologist feedback loop can be minutes instead of weeks or months.

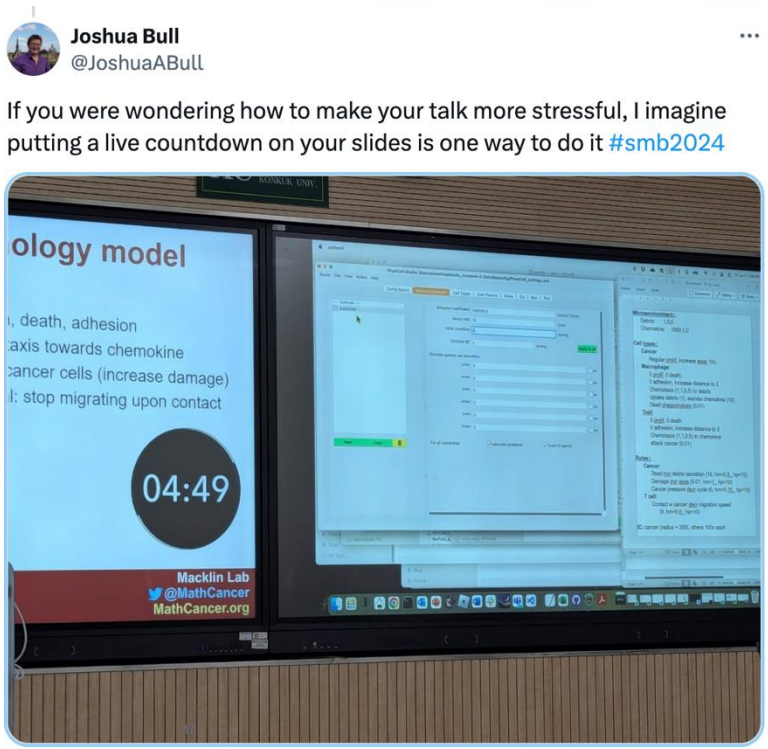


New Possibilities with PhysiCell Studio

- Interactive model editing & exploration

- Interdisciplinary instruction
  - No installation required
  - No coding required
  - **Afternoon short course**:
    - Get started with basic tumor-immune and chemotherapy models
  - **Weeklong hackathons:**
    - ***Day 1:*** Learn overall model framework
    - ***Days 2-5:*** Mentored hackathon, daily starter model demos

- Live modeling in talks
  - 5-minute demo at SMB 2024! ☺



Heiland et al., **PhysiCell Studio: a graphical tool to make agent-based modeling more accessible.** GigaByte (2024). DOI: 10.46471/gigabyte.128

Automated model annotation

- We auto-generate formatted tables as we parse the rules

- The underlying hypotheses are summarized for inclusion in the methods section for later papers.
  - Easier for future scientists to understand the model assumptions.
  - Easier to reuse in future models.
  - Also helpful for internal QC!

Cell Hypothesis Rules (detailed)

In tumor cells:

- oxygen increases cycle entry from 0 towards 0.00072 with a Hill response, with half-max 21.5 and Hill power 4.
- pressure decreases cycle entry from 0 towards 0 with a Hill response, with half-max 1 and Hill power 4.
- oxygen decreases necrosis from 0.0028 towards 0 with a Hill response, with half-max 3.75 and Hill power 8.
- damage increases apoptosis from 7.2e-05 towards 0.072 with a Hill response, with half-max 180 and Hill power 2.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.
- IFN-gamma decreases migration speed from 0.5 towards 0 with a Hill response, with half-max 0.25 and Hill power 2.

In M0 macrophage cells:

- contact with dead cell increases transform to M1 macrophage from 0 towards 0.05 with a Hill response, with half-max 0.1 and Hill power 10.
- contact with dead cell decreases migration speed from 1 towards 0.1 with a Hill response, with half-max 0.1 and Hill power 4.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

In M1 macrophage cells:

- contact with dead cell decreases migration speed from 1 towards 0.1 with a Hill response, with half-max 0.1 and Hill power 4.
- oxygen decreases transform to M2 macrophage from 0.01 towards 0 with a Hill response, with half-max 5 and Hill power 4.
- IFN-gamma increases cycle entry from 7.2e-05 towards 0.00036 with a Hill response, with half-max 0.25 and Hill power 2.
- IFN-gamma increases phagocytose dead cell from 0.01 towards 0.05 with a Hill response, with half-max 0.25 and Hill power 2.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

In M2 macrophage cells:

- contact with dead cell decreases migration speed from 1 towards 0.1 with a Hill response, with half-max 0.1 and Hill power 4.
- IFN-gamma decreases cycle entry from 7.2e-05 towards 0 with a Hill response, with half-max 0.25 and Hill power 2.
- IFN-gamma increases phagocytose dead cell from 0.01 towards 0.05 with a Hill response, with half-max 0.25 and Hill power 2.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

In naive T cell cells:

- IL-10 decreases transform to CD8 T cell from 0.001 towards 0 with a Hill response, with half-max 0.25 and Hill power 2.
- IFN-gamma increases transform to CD8 T cell from 0.001 towards 0.01 with a Hill response, with half-max 0.25 and Hill power 2.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

In CD8 T cell cells:

- IFN-gamma increases cycle entry from 7.2e-05 towards 0.00093 with a Hill response, with half-max 0.25 and Hill power 2.
- IL-10 decreases attack tumor from 0.01 towards 0 with a Hill response, with half-max 0.25 and Hill power 2.
- IL-10 decreases migration speed from 1 towards 0.1 with a Hill response, with half-max 0.25 and Hill power 2.
- contact with tumor decreases migration speed from 1 towards 0.1 with a Hill response, with half-max 0.1 and Hill power 2.
- IL-10 increases transform to exhausted T cell from 0 towards 0.005 with a Hill response, with half-max 0.25 and Hill power 4.
- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

In exhausted T cell cells:

- dead increases debris secretion from 0 towards 0.017 with a Hill response, with half-max 0.1 and Hill power 10. Rule applies to dead cells.

Example: tumor-immune

In tumor cells:
- oxygen increases cycle entry
- pressure decreases cycle entry
- oxygen decreases necrosis
- damage increases apoptosis
- dead increases debris secretion
- IFN-gamma decreases migration speed

In M0 macrophages:
- contact w/ dead cell increases transform to M1 mac
- contact w/ dead cell decreases migration speed
- dead increases debris secretion

In M1 macrophages:
- contact with dead cell decreases migration speed
- oxygen decreases transform to M2 macrophage
- IFN-gamma increases cycle entry
- IFN-gamma increases phagocytose dead cell
- dead increases debris secretion

In M2 macrophages:
- contact with dead cell decreases migration speed
- IFN-gamma decreases cycle entry
- IFN-gamma increases phagocytose dead cell
- dead increases debris secretion

In naive T cells:
- IL-10 decreases transform to CD8 T cell
- IFN-gamma increases transform to CD8 T cell
- dead increases debris secretion

In CD8 T cells:
- IFN-gamma increases cycle entry
- IL-10 decreases attack tumor
- IL-10 decreases migration speed
- contact w/ tumor decreases migration speed
- IL-10 increases transform to exhausted T cell
- dead increases debris secretion

In exhausted T cells:
- dead increases debris secretion

Joint work with:

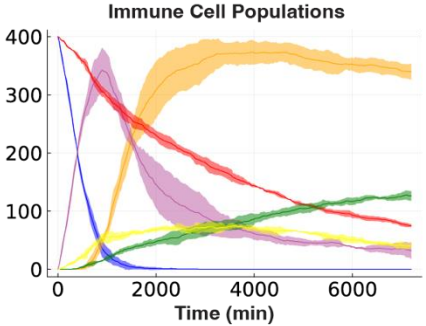
U. Maryland:
- **Elana Fertig**
- Daniel Bergman

Johns Hopkins:
- **Genevieve Stein-O'Brien**
- Jeanette Johnson

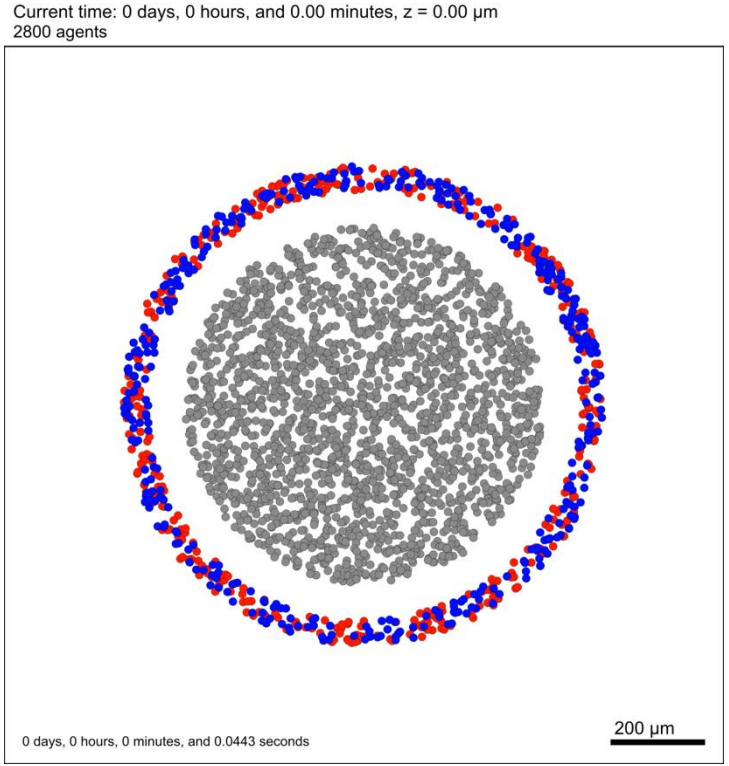
OHSU:
- **Laura Heiser**
- Lisa Coussens
- Joe Gray
- Young Hwan-Chang

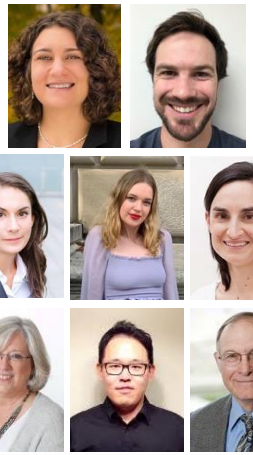
Example: tumor-immune

- tumor
- M0 macrophage
- M1 macrophage
- M2 macrophage
- naive T cell
- CD8 T cell
- exhausted T cell



Johnson et al, *BioRxiv*, 2023





Joint work with:

U. Maryland:
- **Elana Fertig**
- Daniel Bergman

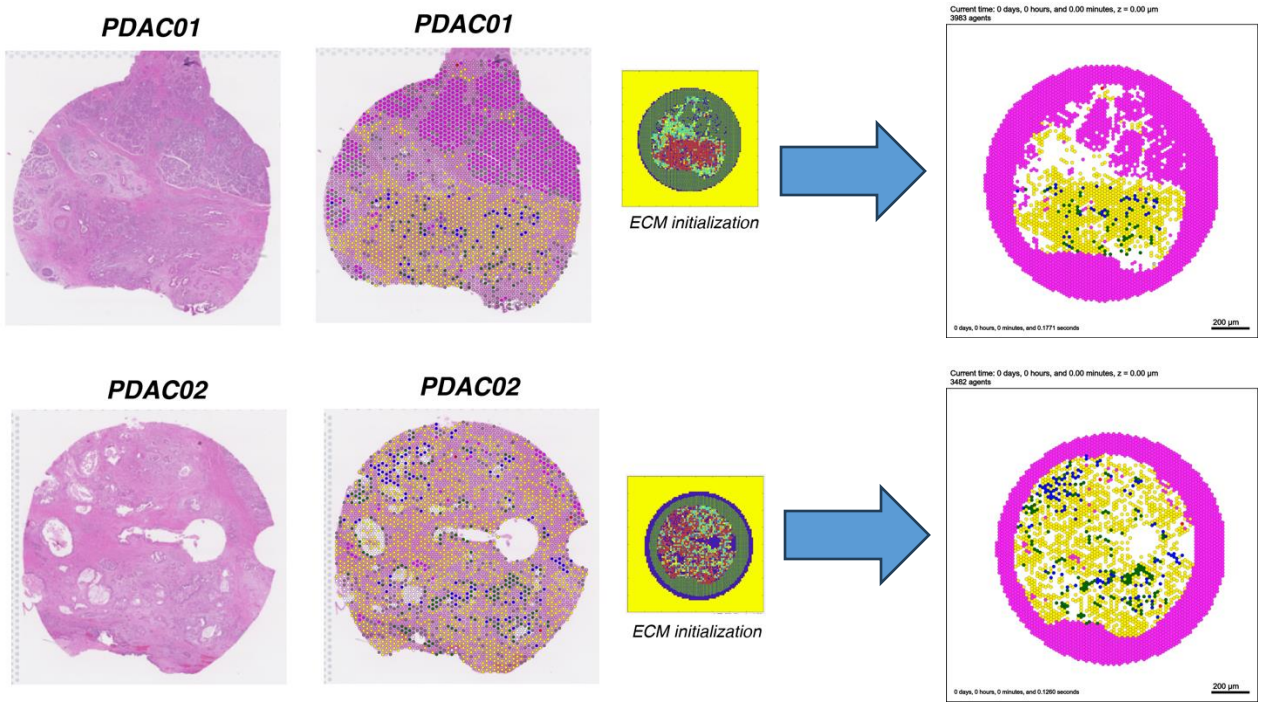
Johns Hopkins:
- **Genevieve Stein-O'Brien**
- Jeanette Johnson

OHSU:
- **Laura Heiser**
- Lisa Coussens
- Joe Gray
- Young Hwan-Chang

We can use the language to connect genomics with dynamical modeling

Creating models from spatial transcriptomic data

CONVERGENCE INSTITUTE



Johnson et al, *BioRxiv*, 2023

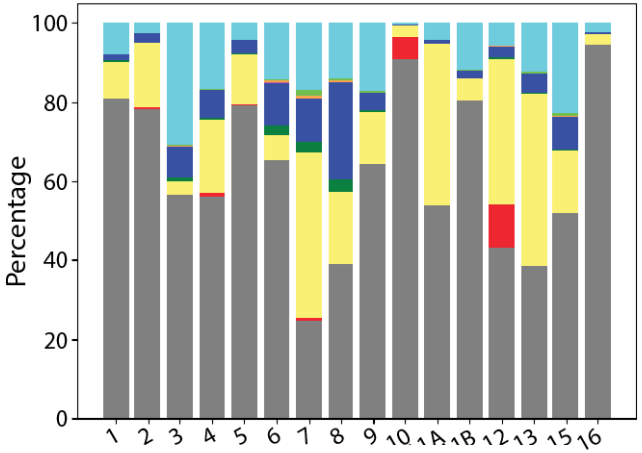
Elana Fertig | Jeanette Johnson | Daniel Bergman | Maxwell Booth

Model predictions:

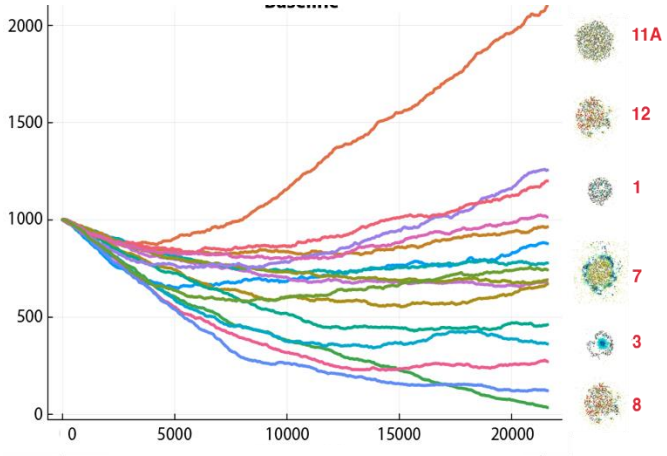
- Test hypothesis:
  - Epithelial-fibroblast interactions can transform epithelial cells
- Composition & geometry of TME drive divergent trajectories
- Consistent with subtype switching in PDAC progression to invasion

Simulating immunotherapy trials (1)

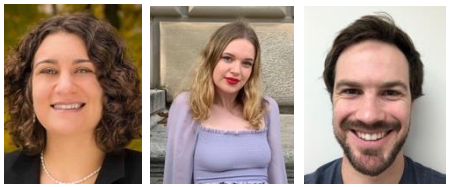
- Initialize to 16 patient tissues
  - Same immune cell rules and parameters for each tissue
  - Set initial immune cell composition to match tissue
  - Simulate immune response sans therapy



Initial immune cell compositions



Untreated growth curves



Elana Fertig | Jeanette Johnson | Daniel Bergman

Johnson et al, *BioRxiv*, 2023

Differences in immune composition lead to widely divergent growth curves, even without any other biological differences

Simulating immunotherapy trials (2)

- Simulate immunotherapies in the 16 patients
  - **immune checkpoint inhibitor:** shift T cell population to PDL1 low (more active killing)
  - **CD137 agonist:** shift to T cells that can kill independent of PD1/PDL1
  - **GVAX (vaccine):** attracts more T cells → higher initial T cell population

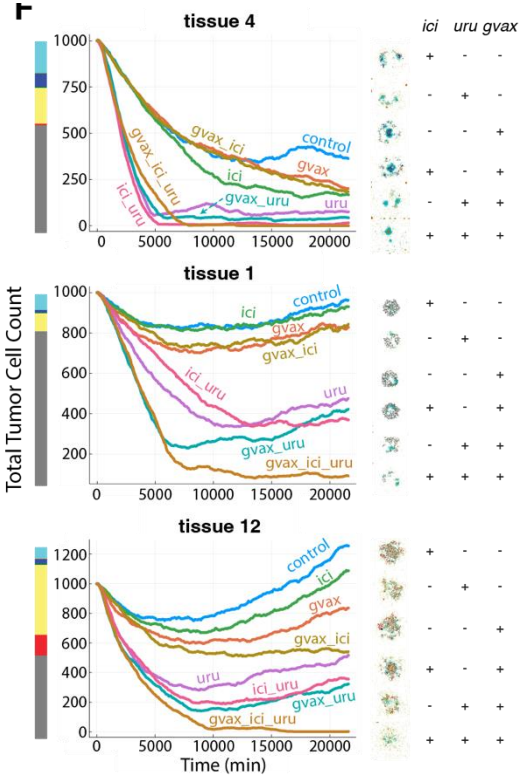
Combinations increase immune activity and infiltration.

Impact (and best therapy choice) highly dependent on initial immune composition



Elana Fertig | Jeanette Johnson | Daniel Bergman

Johnson et al, *BioRxiv*, 2023



Ultimately, we envision many paths

- Expert-driven
  - Tap centuries of learning by biologists and other experts

- Data-driven
  - Automated analysis of scRNAseq data
    - Who is the sender? What signal? (who expresses diffusible and other factors?)
    - Who is the recipient? (Who expresses receptors for the signal?)
    - What is the response? (Can receptor activation be correlated with functional changes?)

- AI-driven Literature Analysis
  - Mine PubMed with NLP, Chat-GPT, etc. to identify relationships
    - Constrained / structured prompts → grammar-formatted rules → human quality control

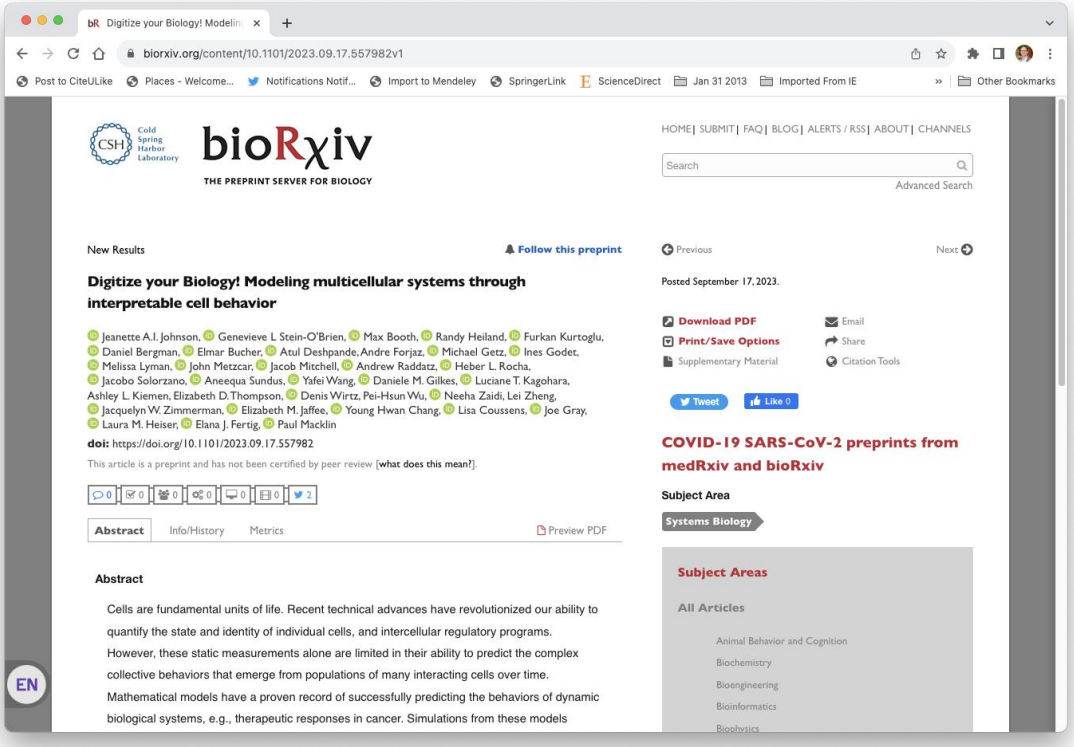
- All of these paths could be represented in this framing, integrating data-driven and knowledge-driven modeling paths

New preprint!

Preprint with the modeling language and spatial transcriptomics:

Under review at *Cell*

DOI: 10.1101/2023.09.17.557982



PhysiCell Curriculum: Next Steps

- **PhysiCell Essentials Short Course (this short course)**
  - **Prerequisites:**
    - Basic knowledge of cell biology, concepts of mathematical functions
  - **Software requirements:**
    - Web browser access, OR installation of PhysiC
  - **Curriculum:**
    - Introduction
    - *Optional: Desktop Installation of PhysiCell Studio*
    - **Hands-on work Part 1: Getting Started, and Villager/Zombie Model (next session!)**
    - Hands-on work Part 2: Cancer Chemotherapy & Immunology Models
    - *Optional: Notes and Tips on Parameter Estimates*
- **Integration of Boolean Networks with PhysiBoSS**
  - Learn how to integrate Boolean signaling networks into PhysiCell Models
- **Advanced PhysiCell Modeling**
  - Learn about creating non-standard model components and visualization in C++
  - Learn about C++ extensions for ODE models, ECM fibers, and more.
- **PhysiCell for Developers**
  - Learn about PhysiCell core C++ structure, and contributing to PhysiCell core and extensions