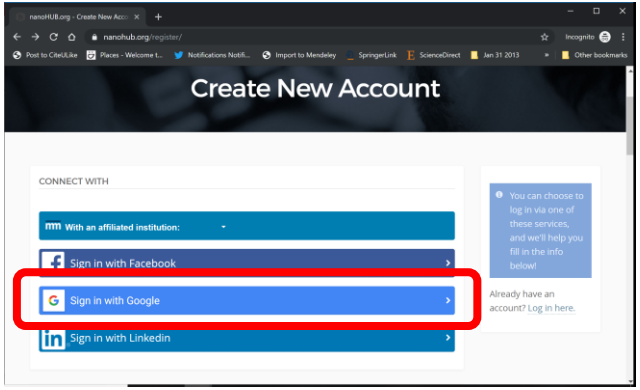
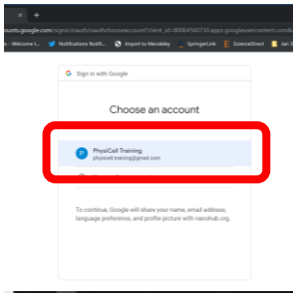
nanoHUB Account

- 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.

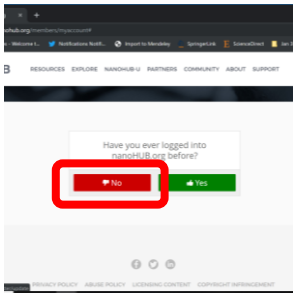
2



3



4



Agent-based modeling of multicellular systems and cancer in PhysiCell

Part 1: Introduction

Get lectures and materials here!

github.com/physicell-training/CAMBAM_2020

Paul Macklin, Ph.D.

Intelligent Systems Engineering
Indiana University

August 13, 2020

A big thank you

- This work is supported by:
  - **National Cancer Institute** & **Breast Cancer Research Foundation:**
    - Simulation methods were originally developed for cancer.
  - **National Science Foundation:**
    - Helped us automatically share complex simulation models on the cloud.
  - **Jayne Koskinas Ted Giovanis Foundation for Health and Policy**
    - A large emergency grant to jumpt-start a COVID-19 modeling coalition.
    - Funding for breast cancer research (jointly with Johns Hopkins and others).
  - Generous computing resources at Indiana University

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)

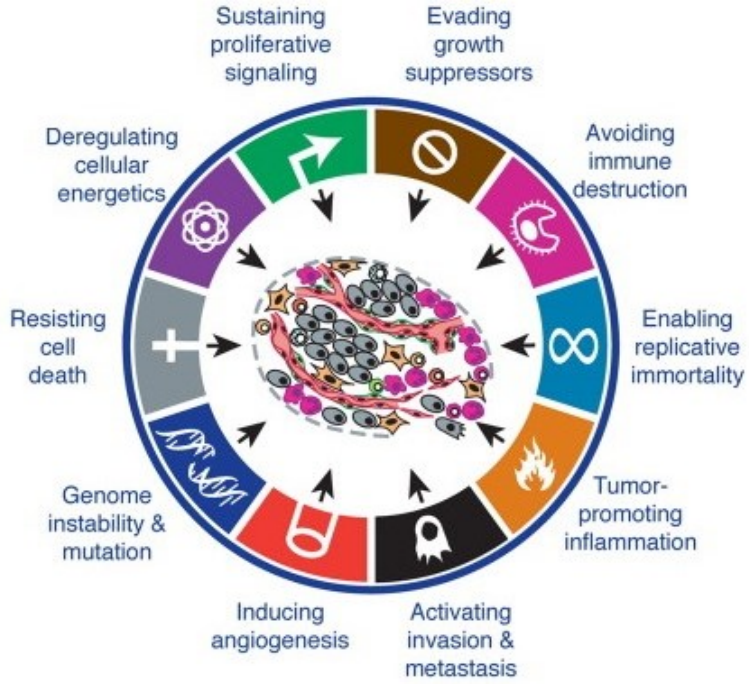
In cancer, these systems become dysregulated.

Treatments target *parts* of these systems.

Cancer is a **complex systems:**
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

Scientists use [models*] to detangle complex systems.

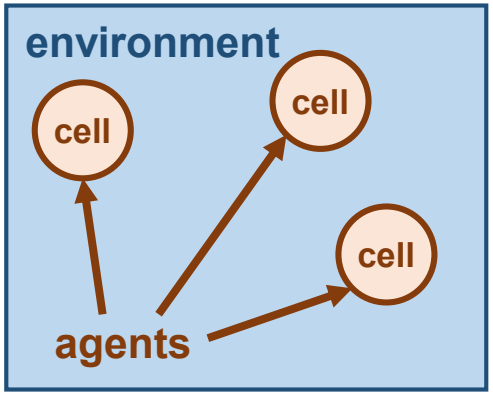
* animal, *in vitro*, engineered, mathematical, …

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?

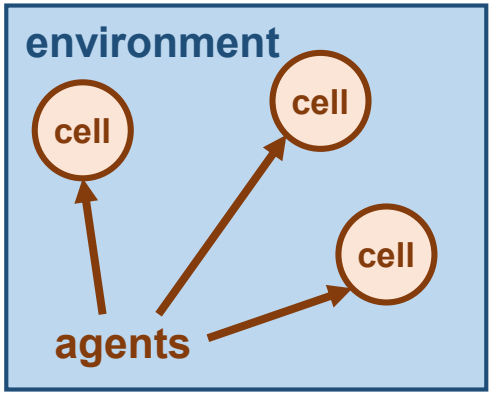
- Each cell is modeled as a separate software object (an **agent**) with:
  - **member data:** internal state variables
    - Position, Size, Cycle State, molecular variables, ….
  - **methods:** cellular processes
    - Cycling, Death, Motility, Growth, Adhesion, …



- 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 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.**

Key elements for an agent-based multicellular model

- **Stage (microenvironment):**
  - What the diffusing chemical species?
  - Do we need to model extracellular matrix (ECM) mechanics?
  - Do we need to model blood vessels?

- **Players (cell types):**
  - One or more cell types?
  - Fibroblasts?
  - One or more immune cell type?
  - Others?

- **Script (cell behaviors & parameters):**
  - Cycling, death, motility, uptake …
  - Interactions (e.g., mechanics, hunting)
  - Conversions among types (e.g., differentiation, mutation)
  - Custom data and functions?
  - Molecular-scale models?

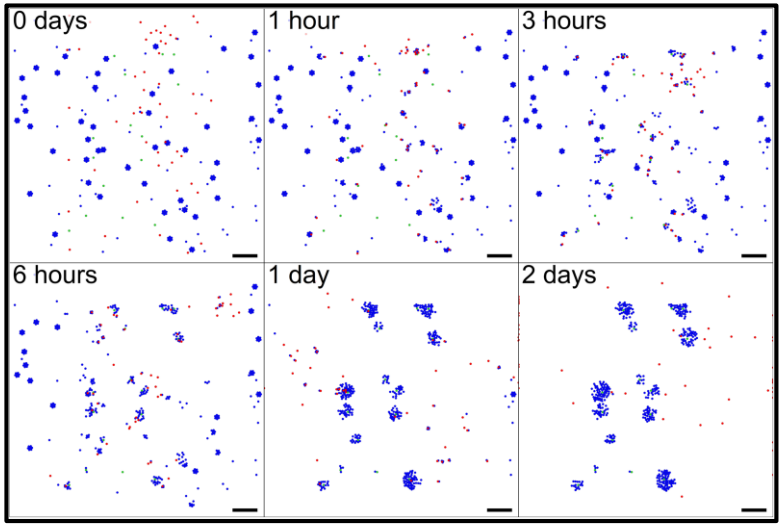
Essentially:

- Where do they live?
- Who's there?
- What do they do?

Example: biological cargo delivery system

- **The stage:**
  - two diffusing chemical signals

- **The players and rules:**
  - **directors (green):**
    - secrete director signal to attract workers
  - **cargo (blue):**
    - **undocked**: secrete cargo signal to attract workers
    - **docked**: turn off signal
  - **workers (red):**
    - **undocked**: seek cargo via chemotaxis
    - **docked:** seek directors via chemotaxis, release cargo in high signal areas



Try this model yourself!

https://nanohub.org/tools/pc4biorobots

pc4biorobots exercises for later

1. **Cargo and workers only**
   - Set # of directors to zero.
   - Set max time to 120 minutes.
   - Click run. What happens?
   - Plot the cargo signal. How does this explain the behavior?
2. **Full model**
   - Set # of directors to 15
   - Set max time to 1000 minutes.
   - Click run. What happens?
   - Plot the director signal. How does this explain the behavior?
3. **Modify workers (1)**
   - Set drop threshold to 0.1
   - Click run. What happens?
   - Plot the director signal. How does this explain the behavior?
4. **Modify workers (2)**
   - Set attached migration bias to 0.3.
   - Click run. What happens?

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 μm resolution

Features:

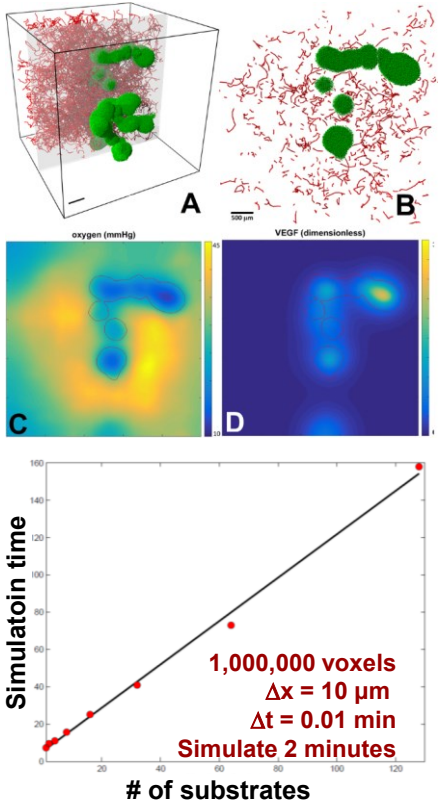
- Off-lattice cell secretion and uptake
- 2^{nd}-order accurate (space), 1^{st}-order accurate (time), numerically stable

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 5-10 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

2019 PLoS Computational Biology Research Prize *for* Public Impact

Features:

- Off-lattice cell positions
- Mechanics-based cell movement
- Cell processes (cycling, motility, …)
- Signal-dependent phenotype
- Can dynamically attach custom data & functions on a cell-by-cell basis
- **Deployed from Raspberry Pi to Crays**

Method:

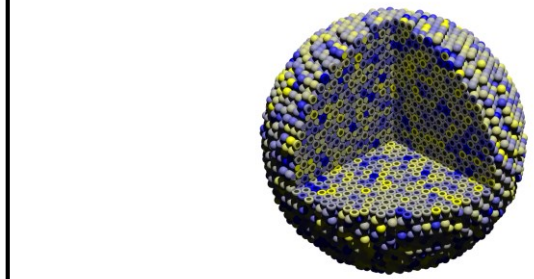
- Standard C++11, cross-platform
- OpenMP parallelization
- O(n) cost scaling in # cells

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

DOI: 10.1371/journal.pcbi.1005991

Try this model yourself!

nanohub.org/tools/pc4heterogen



Competition in a 3-D tumor

[View on YouTube (8K)]

LUDDY
SCHOOL OF INFORMATICS, COMPUTING, AND ENGINEERING

Macklin Lab
@MathCancer
MathCancer.org

Key parts of a PhysiCell model (1)

- **Microenvironment (stage):**
  - diffusing substrates
    - diffusion coefficient
    - decay rate
    - boundary conditions
    - Defined in XML configuration file

- **Cell Definitions (types of players):**
  - name
  - default phenotype (more on next page)
  - defined in XML configuration file

Key parts of a PhysiCell model (2)

- **Cell agents (individual players)**:
  - Which cell type? (the cell agent is initialized based on a cell definition)
  - State variables:
    - position
    - mechanical pressure
    - interaction list (optional)
  - Phenotype **(the script)**
    - Cell cycle
    - Volume
    - Death
    - Motility
    - Mechanics
    - Substrate uptake & release
  - Custom variables
  - Custom functions that act upon the phenotype, variables, and state **(script)**

More mathematical details (for reference)

Cell phenotype

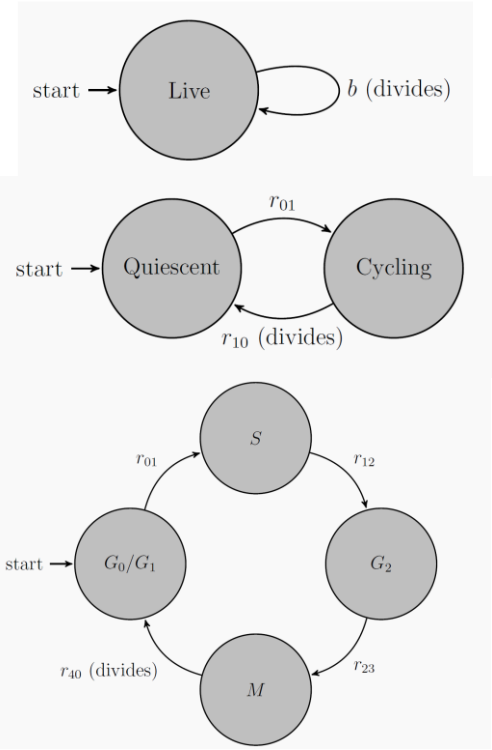
- One of the most critical data elements in a PhysiCell Cell is ***phenotype***
- Hierarchically organize key behavioral elements:
  - Phenotype
    - **cycle**: advancement through a cell cycle model
    - **death**: one or more types of cell death
    - **volume:** cell's volume regulation
    - **geometry:** 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

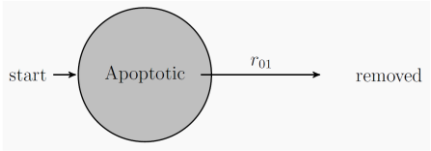
- Each agent's **phenotype** had a **cycle** with:
  - Cycle **model**
    - A directional graph: *nodes* are cycle **phases** $\{P_i\}$ and *edges* are **transition rates** $\{r_{ij}\}$
    - r_{ij} is the transition rate from phase P_i to phase P_j
    - One of the transitions must be marked as a *division 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



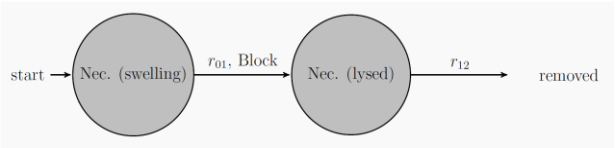
Phenotype: Death

- **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.



Apoptosis

- PhysiCell has built-in apoptosis and necrosis death models



Necrosis

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

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

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

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

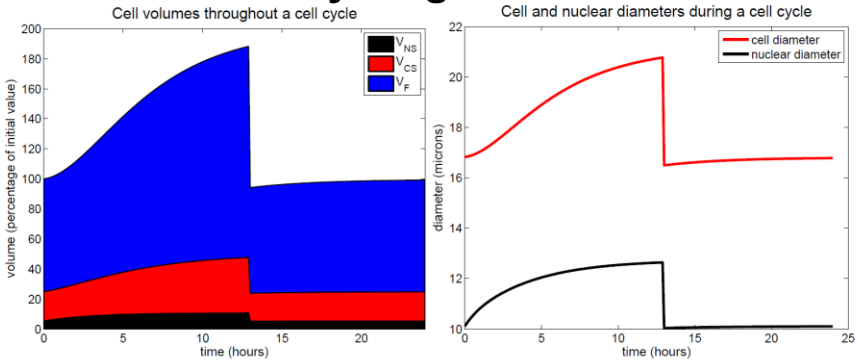
Documentation: User Guide 11.3

Phenotype: Geometry

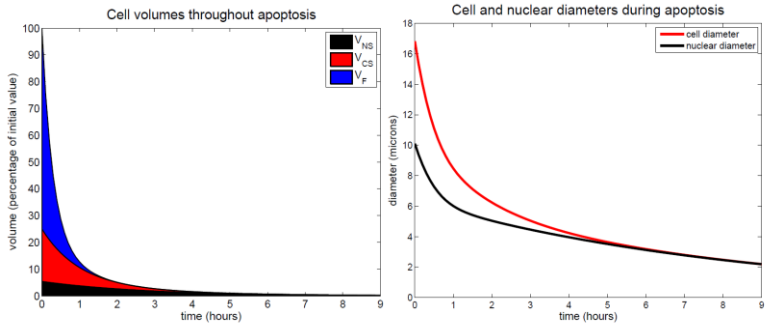
- **geometry** records:
  - surface area (not actively tracked)
  - equivalent spherical radius
  - equivalent nuclear radius

Documentation: User Guide 11.4

Cycling cells



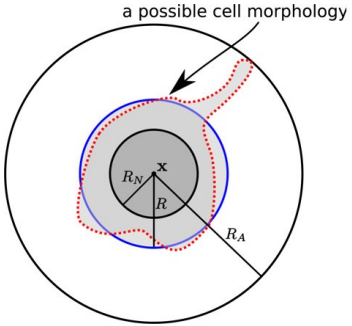
Apoptotic cells

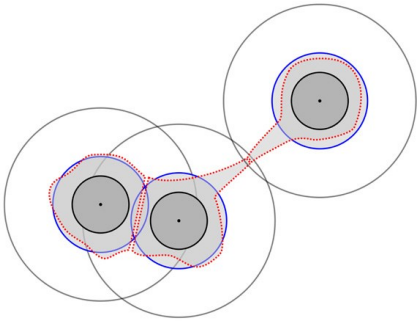


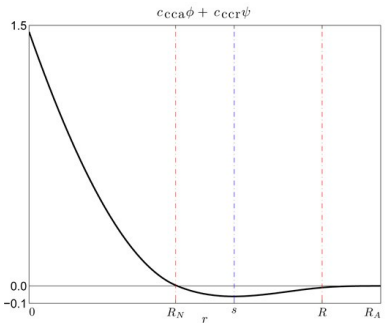
Phenotype: Mechanics

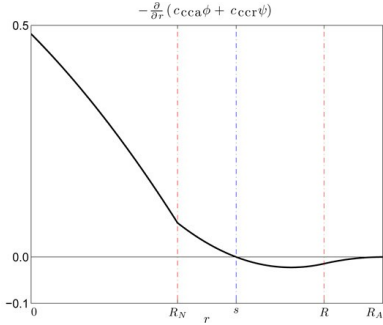
- **Mechanics** keeps parameters for adhesion and "repulsion"
  - Key parameter: maximum adhesion distance
    - a multiple of the cell's radius
  - (as a multiple of the cell's radius)
- Default model uses potential functions, but this can be supplemented or replaced.

Documentation: User Guide 11.5





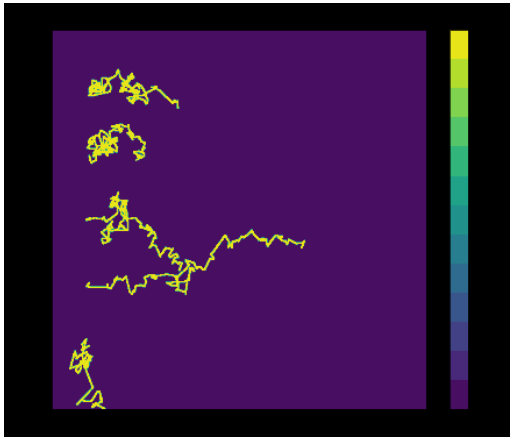




Phenotype: Motility

- **Motility** controls biased random migration
  - Migration speed s
  - Bias direction $\mathbf{d}_{\text{bias}}$
  - Migration bias $0 \leq b \leq 1$
    - If $b = 1$, deterministic motion
    - If $b = 0$, purely Brownian motion
  - Persistence time T_{per}

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



This is an example of an ***educational microapp***: a small, targeted app designed to illustrate a single biological (or biomathematical) cocept.

Try this model yourself! (2D)

https://nanohub.org/tools/trmotility

Phenotype: Secretion

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

$$\frac{\partial \boldsymbol{\rho}}{\partial t} = \nabla \cdot (\boldsymbol{D} \nabla \boldsymbol{\rho}) - \boldsymbol{\lambda} \cdot \boldsymbol{\rho} + \sum_i \delta(\boldsymbol{x} - \boldsymbol{x}_i) V_i (\boldsymbol{S}_i \cdot (\boldsymbol{\rho}_i^* - \boldsymbol{\rho}) - \boldsymbol{U}_i \cdot \boldsymbol{\rho} + \boldsymbol{E}_i)$$

PhysiCell automatically tracks the mass of substrates removed from the tissue (added to cells) or added to tissue (removed from cells).

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

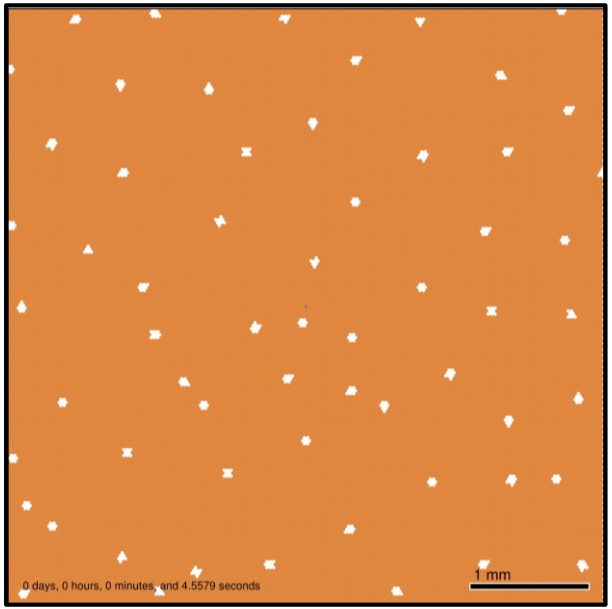
- PhysiCell is designed to account for the multiple time scales inherent to these problems, and has 3 time scales:
  - $\Delta t_{\text{diffusion}}$ diffusion, secretion, and uptake (default: 0.01 min)
  - $\Delta t_{\text{mechanics}}$ cell movement (default: 0.1 min)
  - Δt_{cell} phenotype and volume 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).)

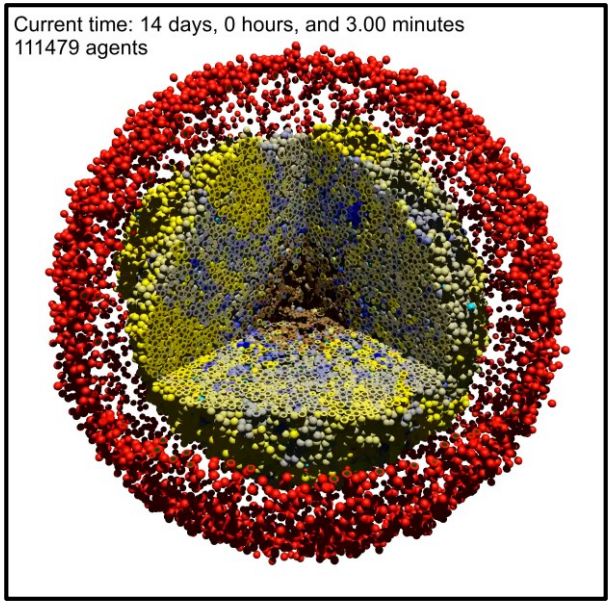
Some examples

Tumor-parenchyma mechanical feedbacks



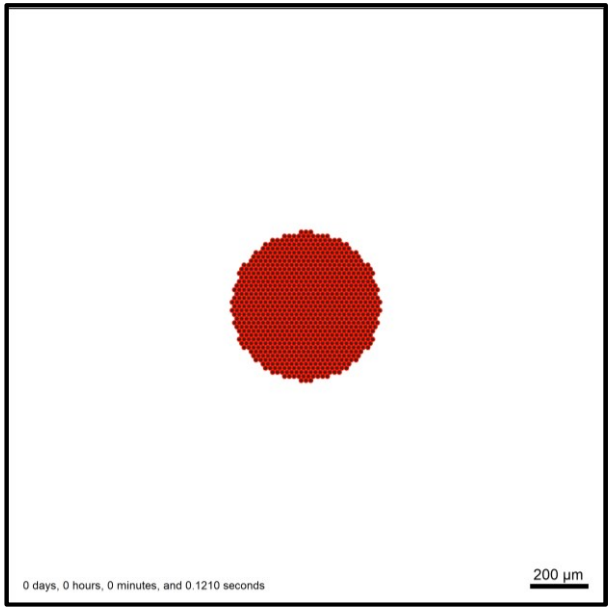
Y. Wang (IU), with Mumemthaler (USC), Sparks (Miami U), Frieboes (U Louisville)

Cancer immunotherapy



with G. An (U. Vermont), Ozik, Wozniak, and Collier (Argonne National Laboratory)

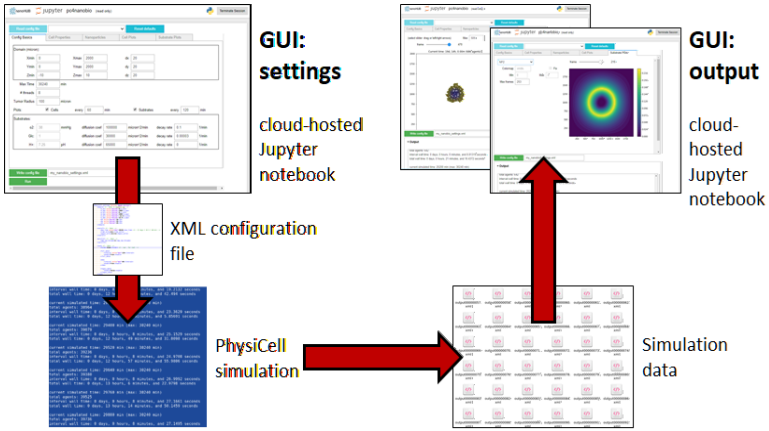
Phenotypic persistence in breast cancer



H. Rocha (IU), with D. Gilkes (Johns Hopkins U.)

PhysiCell ecosystem

- **xml2jupyter**: automatically build Jupyter notebook GUI for any PhysiCell model, then share them on the cloud via nanoHUB
- **PhysiBoSS**: Combine PhysiCell agents with Boolean signal networks (with Institut Curie & Barcelona Supercomputing center)
- **EMEWS**: large-scale model exploration on high performance computing (with Argonne Nat'l Lab)
- **Python loader**: load PhysiCell data into Python for analysis, visualization
- **More**: convert data for 3D raytracing, 3D model exploration with game engines, machine learning on PhysiCell data, SBML, …



XML+Jupyter architecture

A detailed cancer example

cancer-immune contact interactions

Heterogeneous tumor cells (blue to yellow):

- Cycle entry rate scales with O_2
- Cells necrose in very low O_2
- 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

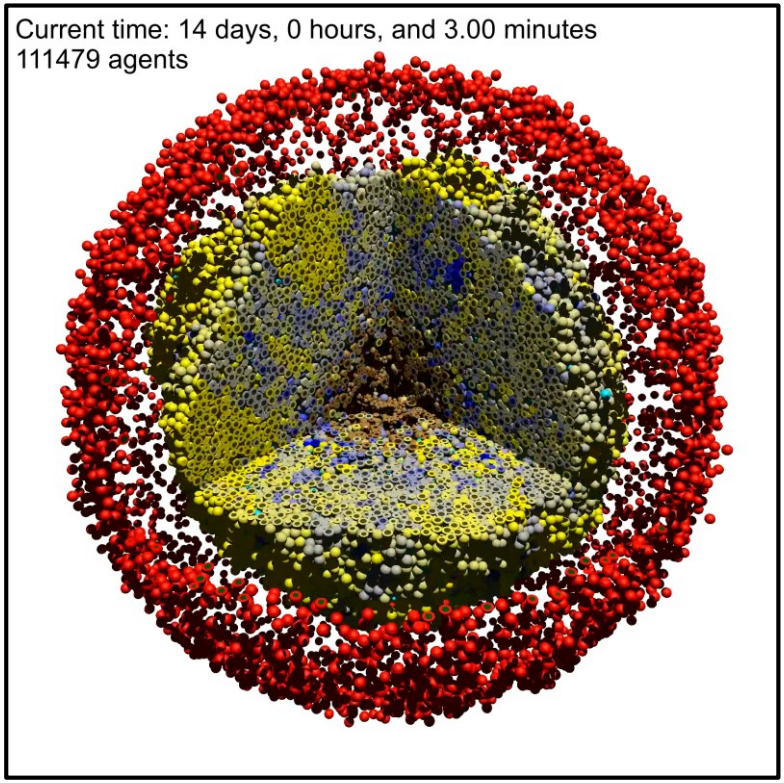
Movie: [View on YouTube (4K)]

References:

- Ghaffarizadeh et al. (2018)
- Ozik et al. (2018)
- Ozik et al. (2019)

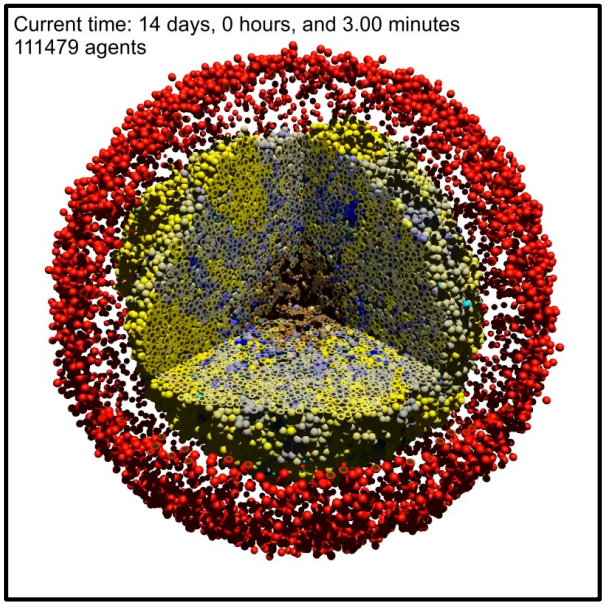
Try this model yourself! (2D)

nanohub.org/tools/pc4cancerimmune



High-throughput investigations on HPC

3-D tumor-immune model



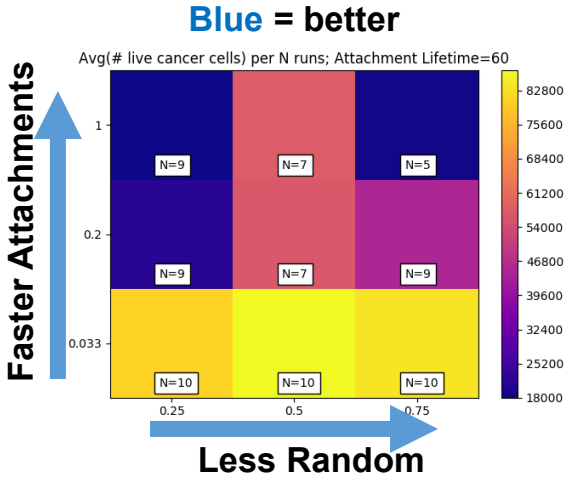
Explore 3 parameters:

- migration bias
- attachment rate
- attachment lifetime

- 27 parameter sets
- 10 replicates per set
- ~2 days per run
- ~1.5 years of computing

HTC is the only feasible path

ANL: Do all 270 runs over a weekend

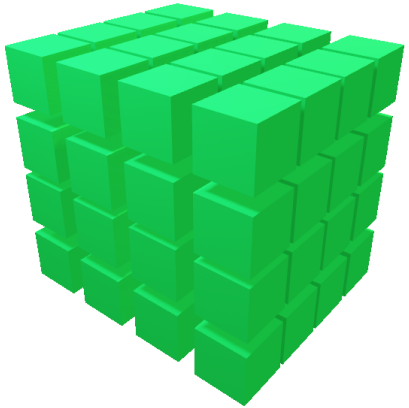


Reference:
Ozik et al. (2018)

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% ($N_{final} \leq 0.1\ N_{initial}$)?
3) Can we reduce cancer cells by 99% ($N_{final} \leq 0.01\ N_{initial}$)?

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., $N_{final} \leq 0.1\ N_{start}$)
  - **False**: points that don't meet the design goal (e.g., $N_{final} > 0.1\ N_{start}$)

- **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)

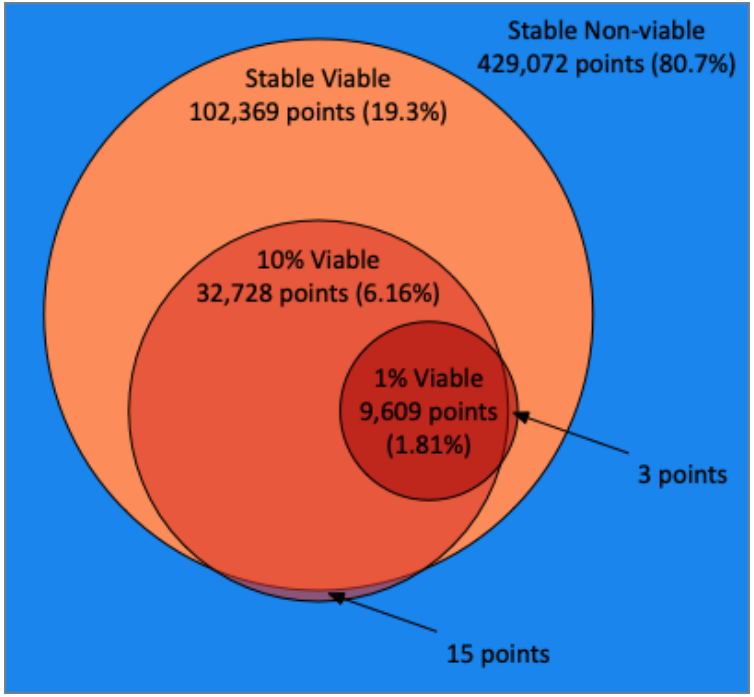
How did HPC+ML enable new science?

- 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

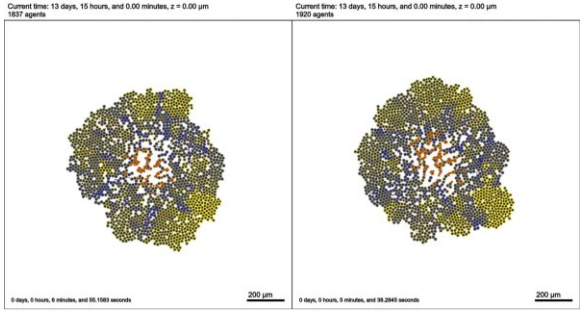


Reference: Ozik et al. (2019)

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



Sample designs from the study
[View on YouTube]

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

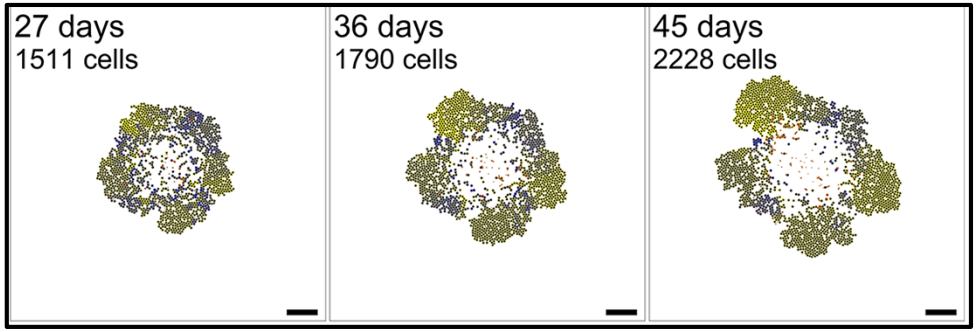
Try this model yourself!
nanohub.org/tools/pc4cancerimmune

Reference:
Ozik et al. (2019)

Let's try some examples

pc4heterogen

- **cancer heterogeneity:**
  - **cancer cells**
    - each has an "oncoprotein" *p*
    - cycle entry scales with O_2
    - cells with higher *p* cycle faster
    - O_2 depletion causes necrosis
  - **blue:**
    - lowest value of *p:* least able to use O_2 to cycle
  - **gold:**
    - highest value of *p*: most able to use O_2 to cycle
  - **orange:**
    - necrotic cell



Try this model yourself!

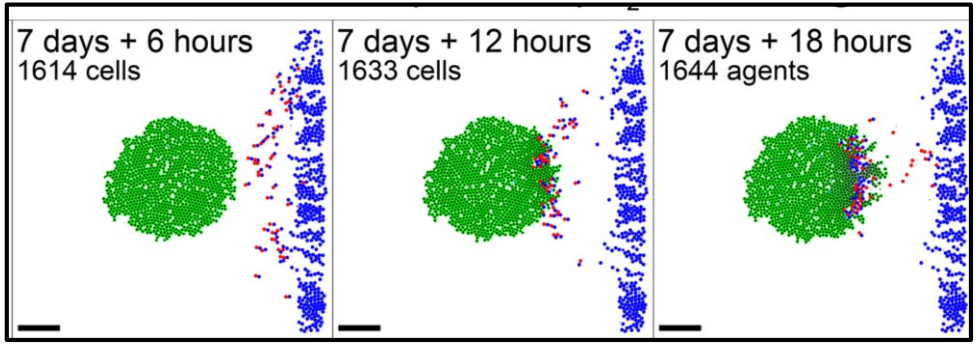
https://nanohub.org/tools/pc4heterogen

pc4heterogen exercises

1. **homogeneous cancer cells**
   - Set max time to 5760 minutes.
   - Set cell and substrate plot intervals to 120 minutes
   - Set oncoprotein standard deviation to 0.0
   - Click run. What happens?
2. **Add heterogeneity**
   - Set oncoprotein standard deviation to 0.5
   - Click run. What happens?
   - Plot the therapeutic. How does this explain the behavior?
3. **Increase oncoprotein heterogeneity**
   - Set oncoprotein standard deviation to 3
   - Set max oncoprotein value to 9
   - Click run. What happens?
4. **Set min oncoprotein (on your own)**
   - Set min oncoprotein value to 0.5
   - Increase max time to 14400 minutes
   - Click run. What happens?

pc4cancerbots

- **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



Try this model yourself!

https://nanohub.org/tools/pc4cancerbots

pc4cancerbots exercises

1. **Cancer cells only**
   - Set # injected cells to 0
   - Increase tumor radius to 400
   - Set max time to 2880 minutes.
   - Click run. What happens?
   - Plot the oxygen. How does this explain the behavior?
2. **Add therapy (full model)**
   - Set # of injected cells to 500
   - Set therapy activation time to 120
   - Increase max time to 4320 minutes
   - Click run. What happens?
   - Plot the therapeutic. How does this explain the behavior?
3. **Modify treatment**
   - Set attached worker migration bias to 0.2
   - Click run. What happens?
4. **Modify treatment (on your own)**
   - Set cargo release o2 threshold to 15
   - Increase max time to 14400 minutes
   - Click run. What happens?

Some models to explore

On nanoHUB:

- **pc4heterogen:** heterogeneous cancer growth (https://nanohub.org/tools/pc4heterogen)
- **pc4cancerbots:** use the "biorobots" as a cell-based cancer therapy (https://nanohub.org/tools/pc4cancerbots)
- **pc4livermedium:** tumor-stroma biomechanical feedbacks (https://nanohub.org/tools/pc4livermedium)
- **pc4cancerimmune:** basic cancer immunotherapy model (https://nanohub.org/tools/pc4cancerimmune)
- **pc4covid19:** COVID-19 simulation model (https://nanohub.org/tools/pc4covid19)
- **trmotility**: training on biased random cell migration (https://nanohub.org/tools/trmotility)
- **pc4thanos:** *Avengers Endgame* battle using cell rules (https://nanohub.org/tools/pc4thanos)

Bundled in PhysiCell:

- biorobots, cancer biorobots, heterogeneity, cancer immunotherapy (3D version), virus-macrophage sample, project templates

What's next? Part 2

- Download PhysiCell from GitHub
- Compile a bundled / standardized C++ model
- Configure and run the model
- Read and plot data in Jupyter
- Build a cancer model via XML

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.

- **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.

- **PhysiBoSS (PhysiCell + MaBoSS for Boolean networks)**
  G. Letort, A. Montagud, G. Stoll, R. Heiland, E. Barillot, P. Macklin, A. Zinovyev, 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.

- **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.

- **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.

- **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.

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.

- **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.

- **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

- **COVID-19 community preprint**

  Y. Wang et al., Rapid community-driven development of a SARS-CoV-2 tissue simulator. *bioRxiv* 2020.04.02.019075 (2020). DOI: 10.1101/2020.04.02.019075

Some links

- PhysiCell project & downloads: http://PhysiCell.org
- Twitter updates: https://twitter.com/PhysiCell
- Tutorials: http://www.mathcancer.org/blog/physicell-tutorials/
- Tools (in progress): https://github.com/PhysiCell-Tools
- Training (in progress): https://github.com/PhysiCell-Training
- Wiki (in progress): http://PhysiCell.org/wiki