Learning-accelerated HPC investigations of cancer immunotherapy designs

Paul Macklin, Ph.D.

Intelligent Systems Engineering
Indiana University

June 20, 2019

Acknowledgements: Partners

- **Colon cancer metabolism:**
  - Stacey **Finley** (USC, U01 Contact PI)
- **Colon cancer organoids:**
  - Shannon **Mumenthaler** (USC, U01 PI)
- **HPC via EMEWS:**
  - Jonathan **Ozik**, Nicholson **Collier**, Justin **Wozniak**, Charles **Macal** (Argonne National Lab), Chase Cockrell,
  - Gary **An** (University of Vermont)
- **Cloud-deployed PhysiCell models:**
  - Gerhard **Klimeck**, Lynn **Zentner**, Martin **Hunt**, Steve **Clark**, others (NanoHUB Cyberplatform at Purdue)
  - Geoffrey **Fox** (IU PI, nanoBIO Node)
- **PhysiCell:**
  - Randy Heiland (IU)
  - **alumni:** S.H. Friedman (OKSI), A. Ghaffarizadeh (USC)
- **PhysiCell performance refinements**
  - GPU computing: Sunita **Chandrasekaran** (U. Delware)
  - RepastHPC extensions: **Ozik**, **Collier** (ANL)
- **IU PhD students:**
  - John Metzcar (hypoxia, invasion),
  - Yafei Wang (liver mets, nanotherapy)
  - Furkan Kurtoglu (metabolism, hypoxia)
  - Aneequa Sundus (cyanobacteria, synthetic multicellular systems)
- **IU Undergraduates:**
  - **Metastasis:** B. Fischer, D. Murphy, K. Konstantinopoulos, B. Duggan
  - **Nanotherapy:** T. Mahjan
  - **Jupyter GUIs:** E. Bower, D. Mishler, T. Zhang
  - **PhysiCell technologies:** E. Freeman

Acknowledgements: Funders

- **NIH** (current)
  - NIH CSBC U01 (1U01CA232137), **PIs** Finley* / Macklin / Mumenthaler
  - 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
- **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

NIH NATIONAL CANCER INSTITUTE

NSF

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)

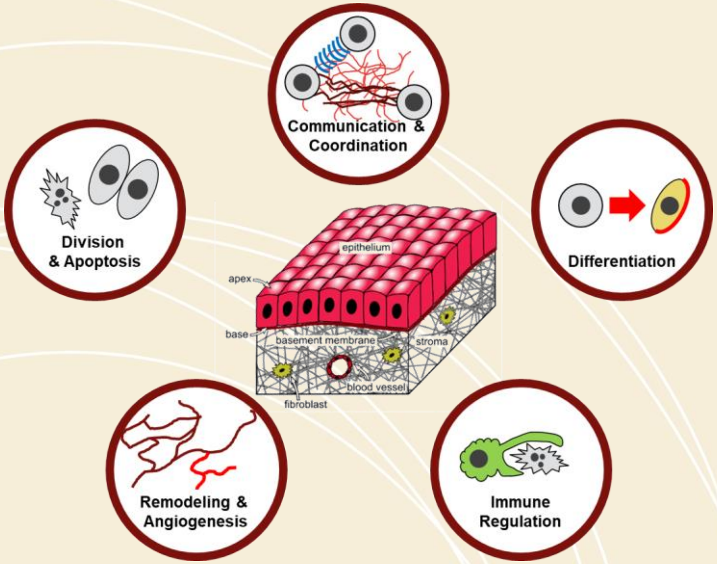
In cancer, these systems fall out of balance.

Treatments target parts of these systems.

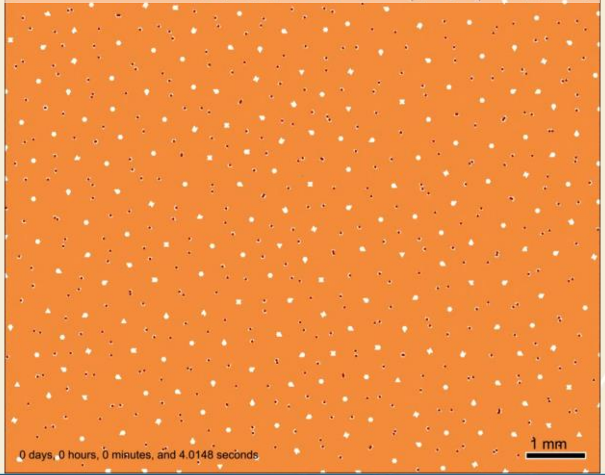
As with any complex system, changing one part can have surprising effects!

Modeling can help **understand** this system. This is **multicellular systems biology.**

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



Metastatic seeding in 1 cm^2 of liver parenchyma



Analogy: multicellular biology as a play

- The **microenvironment** is the **stage**.
- The **cells** are the **actors**.
- The **cells** follow their own **scripts**.

- ***BUT*:**
  - The scripts change based on the stage. (ME-dependent phenotype)
  - The actors' dialog is critical. (cell-cell communication)
  - The actors can tear up and remodel the stage. (tissue remodeling)
  - The actors can ignore their scripts and ad lib. (Mutations, evolution)

It's our job as scientists to figure out each actor's script by watching the play.

Clinicians and engineers want to rewrite the script.

Cancer immunotherapy (main issues)

- Immune system is altered to increase the response to cancerous cells
  - Immune cells can recognize tumor cells as a threat
  - Immune cells can either destroy or induce death in tumor cells
  - **Most common:** block PD1/LPD1 pathway stop immunosuppression
- It's been **game changer in metastatic melanoma**, with durable and even complete responses in a previously incurable cancer.
- **Only 20%** of patients have durable partial or complete response
- **Systems problem**: immune system can both help and hurt tumor cells

Can we understand and help the remaining 80%?

Simulation toolbox

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:

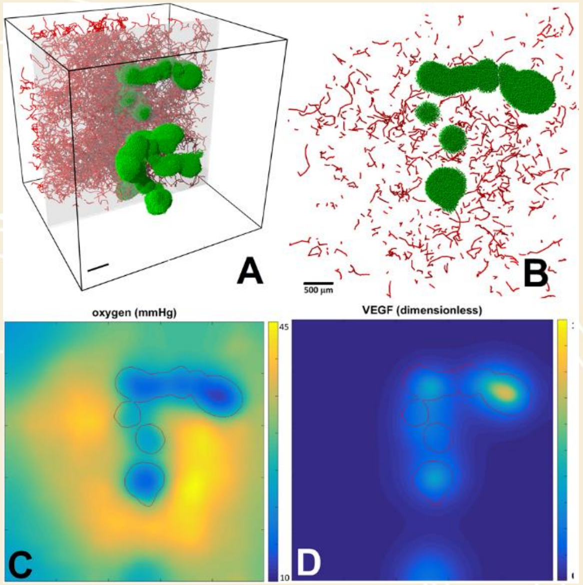
- Off-lattice cell secretion and uptake
- 2nd-order accurate (space), 1st-order accurate (time), numerically stable

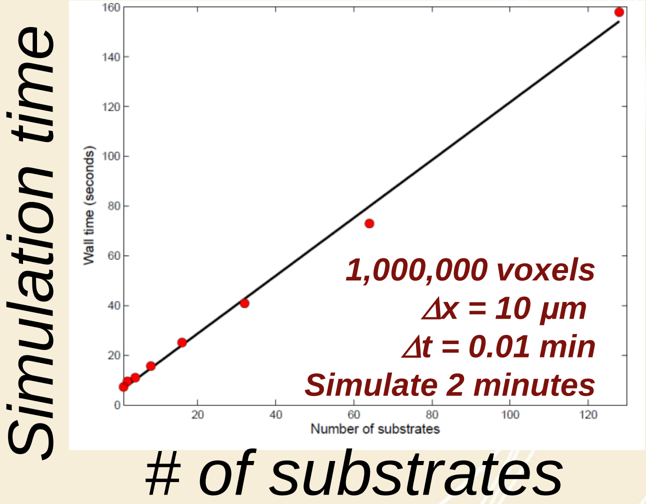
Method:

- 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





1,000,000 voxels
Δx = 10 µm
Δt = 0.01 min
Simulate 2 minutes

PhysiCell: A multicellular framework

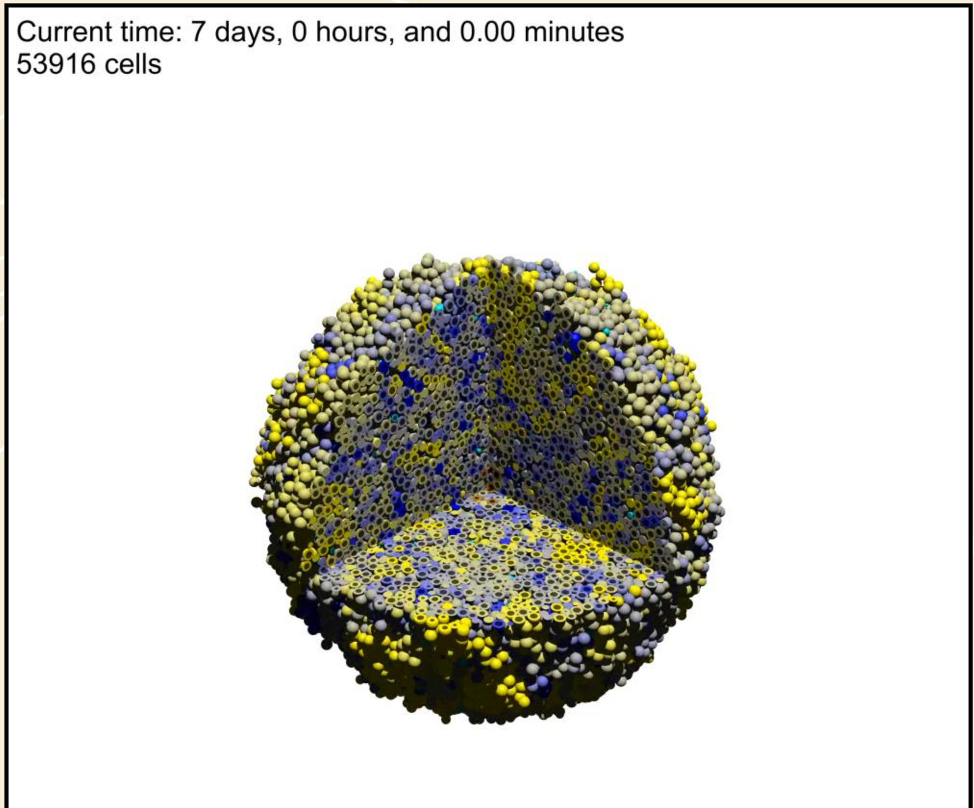
Design goal: Simulate 10^6 or more cells in 3D on desktops or single HPC nodes

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



Competition in a 3-D tumor

[View on YouTube (8K)]

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

DOI: 10.1371/journal.pcbi.1005991

2019 PLoS Computational Biology Research Prize *for* Public Impact

Try this model yourself! (2D)

https://nanohub.org/tools/pc4heterogen

Example:

tumor-immune interactions

Simple model of cancer immune response

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

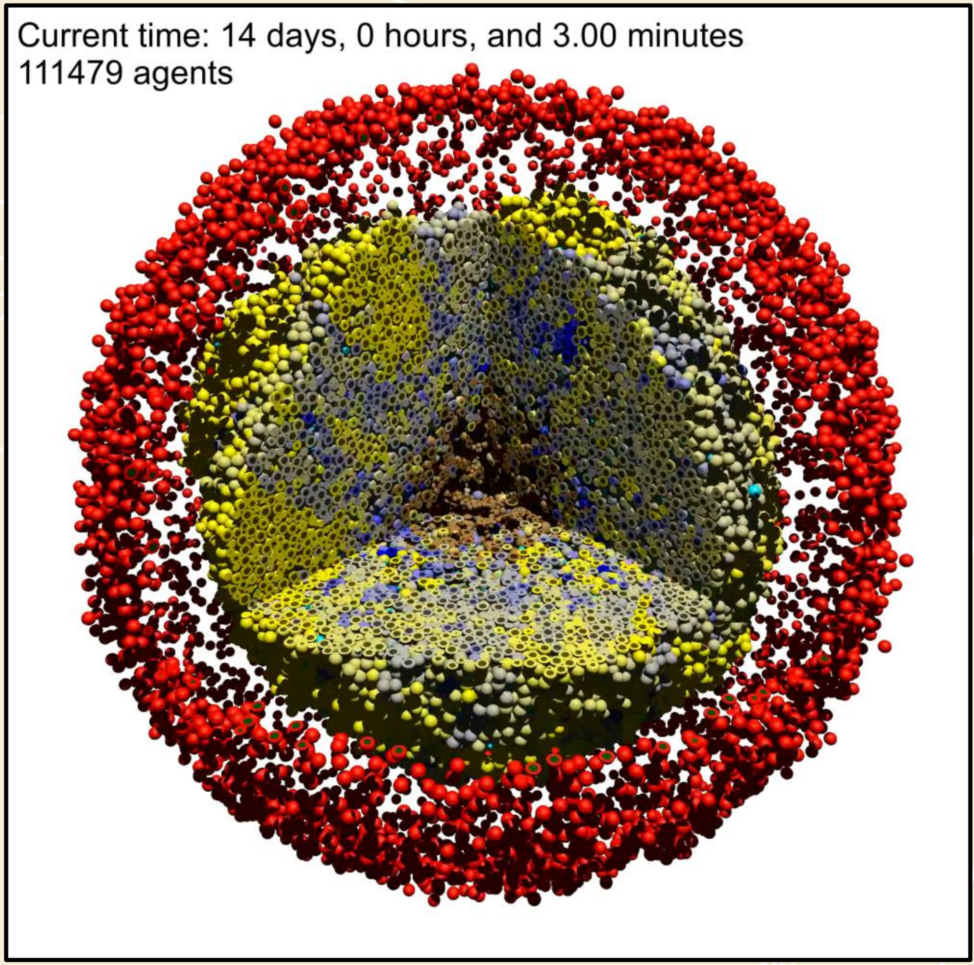
Immune cells (red):

- 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

Scaling up from demo to science …

- Early **insight**: immune cell homing is ***non-intuitive***
- Key immune cell parameters:
  - Random **motility** bias (biased random walk):
    - How much randomness to we allow in motility?
  - Immune cell **attachment rate**:
    - How quickly do immune cells form new adhesions, instead of wandering?
  - Immune cell **attachment lifetime**:
    - How long do immune cells try to kill before giving up?
- **Combinatorics**:
  - 3 parameters, 3 levels per parameter
  - $3^3 = 27$ simulations
- Simulations are **stochastic**! Need at least 10x replicates for each condition!
  - $3^3 \times 10 = 270$ simulations
  - 2 days per simulation → **1.5 years** of computing!!

We need high-throughput computing to do the science!

What we really need:

Extreme-Scale Model Exploration

EMEWS:

EXTREME-SCALE MODEL EXPLORATION WITH SWIFT

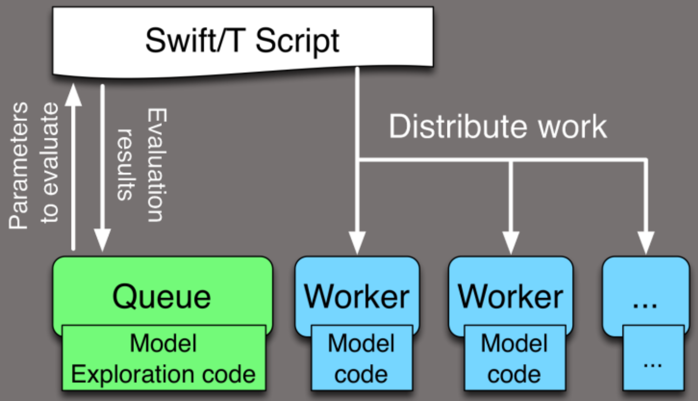
LIMITS OF CURRENT MODEL EXPLORATION APPROACHES

- For **full impact** and **confidence in results**, we need **robust characterization of model parameter spaces**.

- These full characterizations are difficult in practice:
  - Large parameter spaces require adaptive sampling
  - Constraints on which methods are feasible
  - Hand tuning may be “good enough”
  - *Ad hoc* approaches require "heroics" and are hard to generalize
  - Expertise mismatch:
    - Domain experts don't have the HPC expertise to scale
    - HPC experts don't have the scientific domain expertise
  - Large-scale investigation viewed as too "expensive"

- **Result:** Scientists avoid entire classes of "off limits" investigations

Extreme-scale Model Exploration with Swift



Swift/T Script

Parameters to evaluate

Evaluation results

Distribute work

Queue

Model Exploration code

Worker

Model code

Worker

Model code

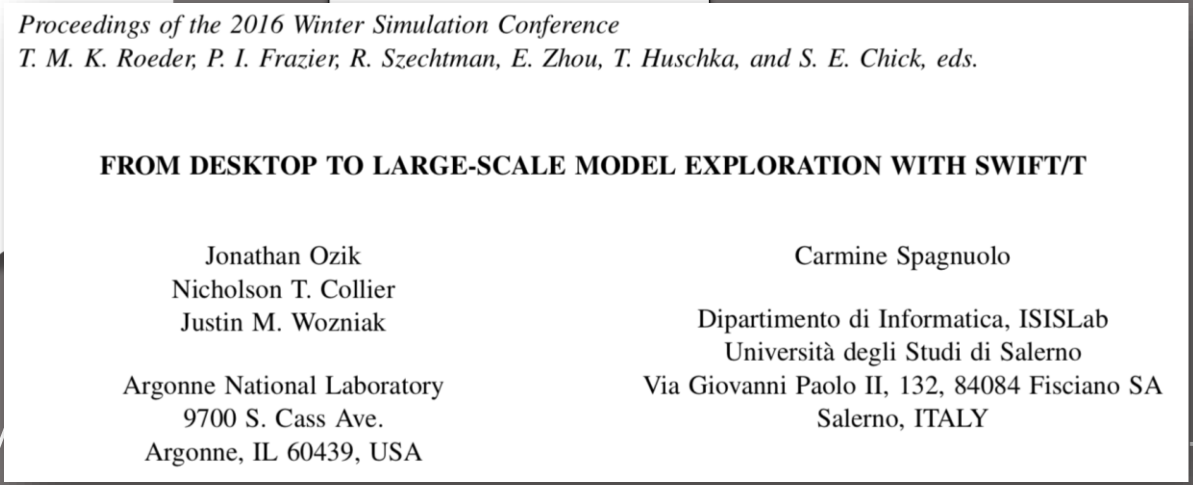
...

...

Multi-language ME and models: R, Python, Java, Julia, C++,…

http://emews.org

Extreme-scale Model Exploration with Swift



Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

FROM DESKTOP TO LARGE-SCALE MODEL EXPLORATION WITH SWIFT/T

Jonathan Ozik
Nicholson T. Collier
Justin M. Wozniak

Argonne National Laboratory
9700 S. Cass Ave.
Argonne, IL 60439, USA

Carmine Spagnuolo

Dipartimento di Informatica, ISISLab
Università degli Studi di Salerno
Via Giovanni Paolo II, 132, 84084 Fisciano SA
Salerno, ITALY

M

++,…

http://emews.org

Ozik et at al. 2016. "From Desktop to Large-Scale Model Exploration with Swift/T." In *Proc. Winter Simulation Conference*.
Available at: https://www.informs-sim.org/wsc16papers/019.pdf

Winner:
2018 R&D 100 Award
(SWIFT/T)

BENEFITS OF DIRECTLY INCORPORATING EXTERNAL CODE

- No need to port the exploration logic into the workflow language
  - **results:** Remove effort overhead, reduce "translation" errors

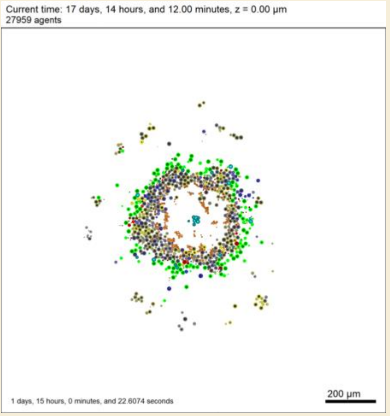
- With reduced "lock-in", easy access to the latest ML methods
  - **Python:** DEAP, scikit-learn, Keras
  - **R:** caret, randomForest, EasyABC, hetGP
  - **result:** easily compare utility and performance of new methods

- ME algorithms are only minimally aware of EMEWS context
  - **result:** can still use methods from non-massively parallel origins

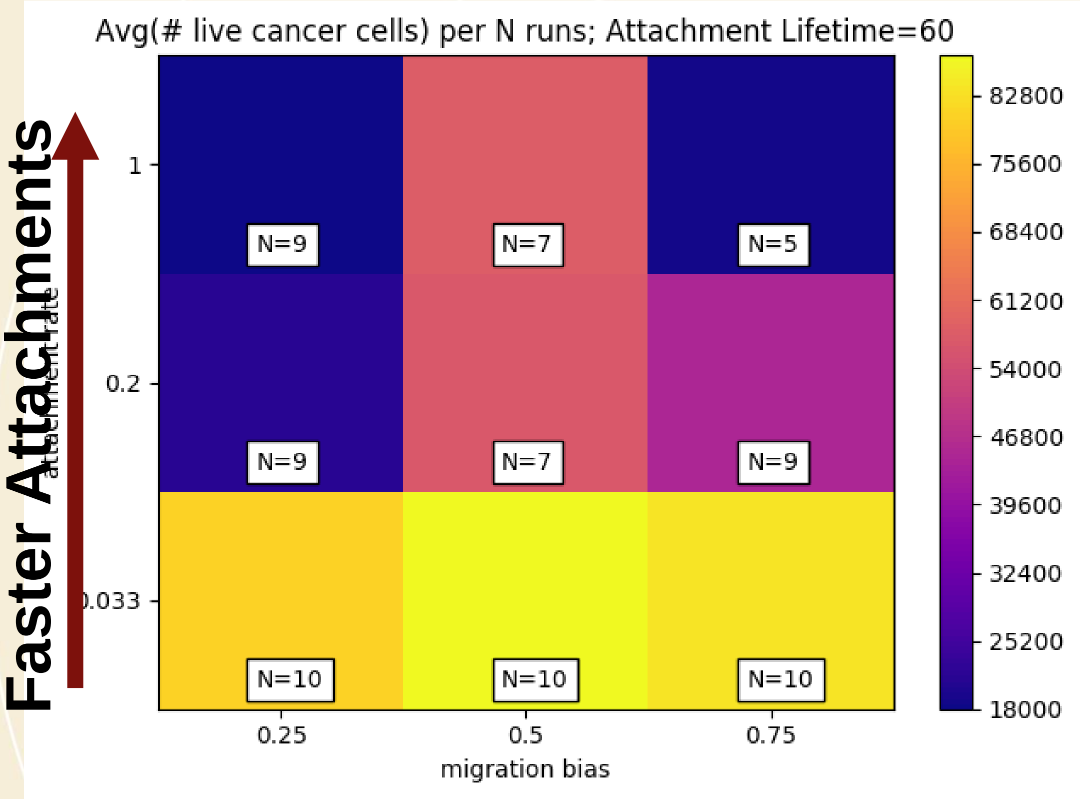
First results:

3D parameter survey

Example: 270 3D simulations in 1 weekend

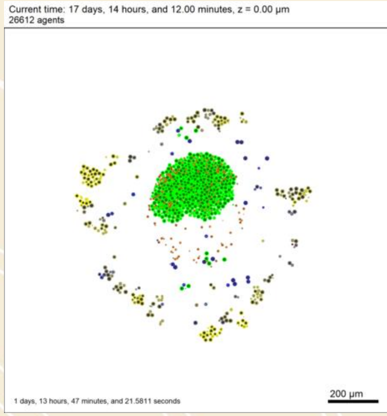

Current time: 17 days, 14 hours, and 12.00 minutes, z = 0.00 μm
27959 agents
1 days, 15 hours, 0 minutes, and 22.6074 seconds
200 μm

Blue = better

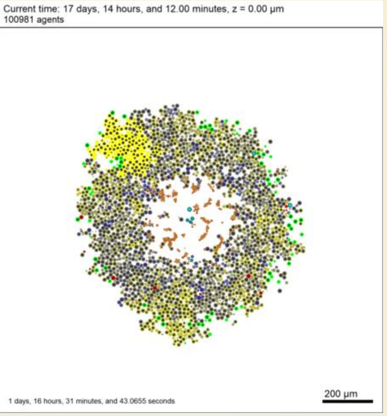


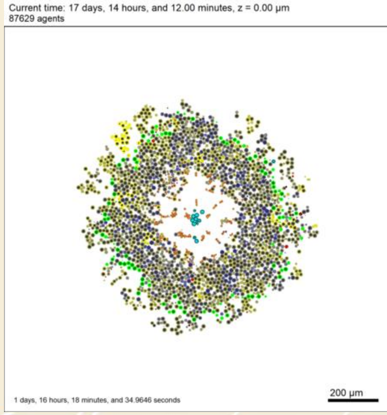
Faster Attachments

Less Random


Current time: 17 days, 14 hours, and 12.00 minutes, z = 0.00 μm
26612 agents
1 days, 13 hours, 47 minutes, and 21.5811 seconds
200 μm

Reference:
Ozik et al. (2018)


Current time: 17 days, 14 hours, and 12.00 minutes, z = 0.00 μm
100981 agents
1 days, 16 hours, 31 minutes, and 43.0655 seconds
200 μm


Current time: 17 days, 14 hours, and 12.00 minutes, z = 0.00 μm
87629 agents
1 days, 16 hours, 18 minutes, and 34.9646 seconds
200 μm

New results:

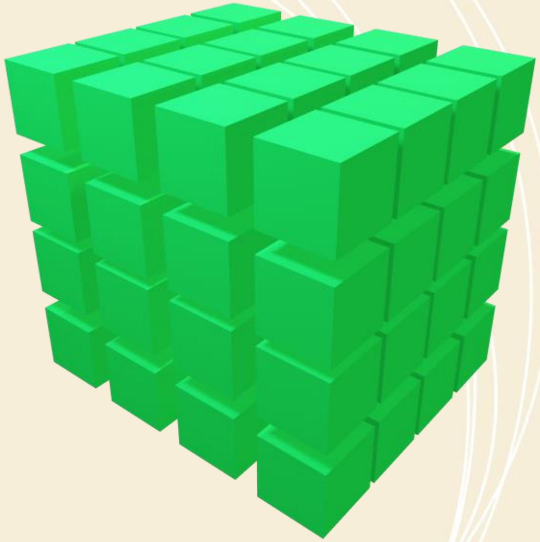
6-parameter design problem

Follow up study

- 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

$\Big\}$ **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.) …



Four scenarios to explore

- **Cancer control**

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

- **Cancer remission**

  2) Can we reduce cancer cells to 10% ($N_{\text{final}} \leq 0.1\ N_{\text{start}}$)

  3) Can we reduce cancer cells to 1% ($N_{\text{final}} \leq 0.01\ N_{\text{start}}$)

- **Treatment optimization:**

  4) Can we minimize N_{final}?

Approach:

Problem 4 is fairly traditional:

Use genetic algorithm (*)

Problems 1-3 are harder:

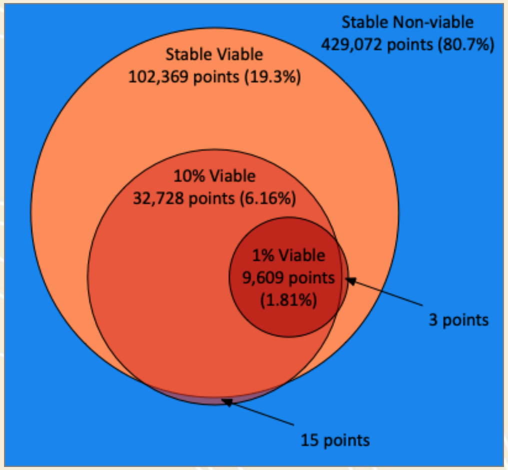
Can't densely sample 6-D design space! (Even on HTC!)

531,441 discrete points in design space

Use **active learning** to find the shape of the "valid design" region

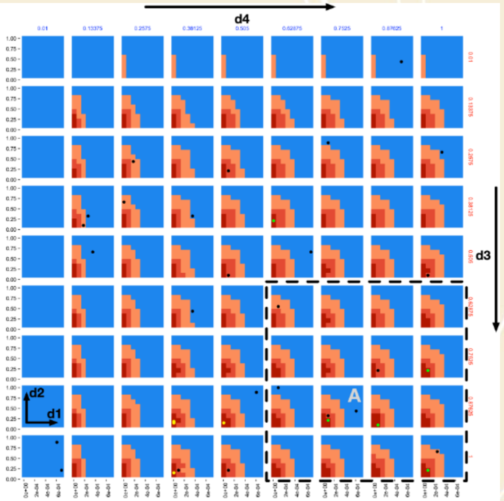
Results

- We explored **4 treatment scenarios:**
  - **Stable scenario** ($N_{final} \leq N_{start}$): 19.3%
  - **10% scenario** ($N_{final} \leq 0.1 \cdot N_{start}$): 6.2%
  - **1% scenario** ($N_{final} \leq 0.01 \cdot N_{start}$): 1.8%
  - **Optimal designs** (minimize N_{final}): $\subseteq$ 1% set



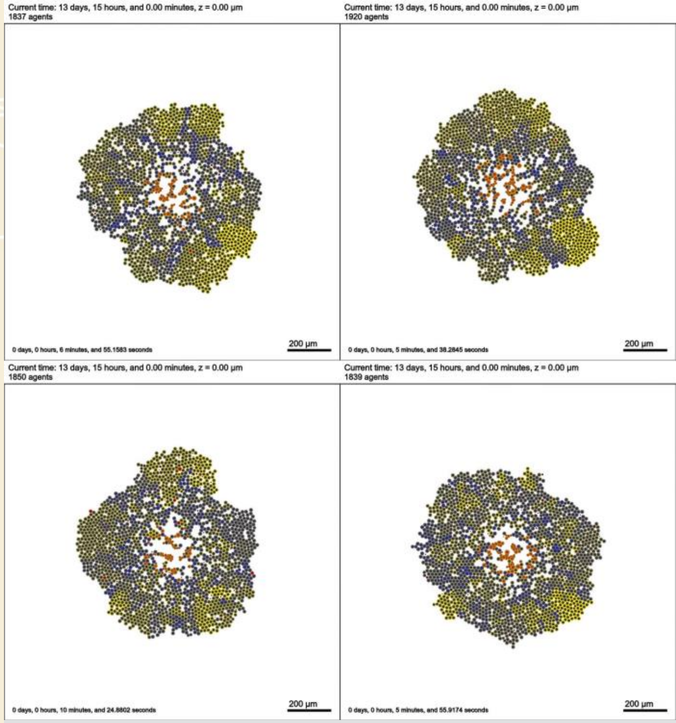
- HPC gives us the ***topology*** of the design space:
  - More **aggressive treatment goals** drastically **shrink the valid design space**
  - The optima aren't particularly robust:
    - Variability can easily push us outside the 1% case
    - Could strongly select for therapeutic resistance

Reference:
Ozik et al. (2019)



Results (continued)

- HPC + machine learning allows us to approach bigger classes of problems
  - ~ 48,000 core hours for each scenario
  - ~ 30,000 to 40,000 simulations per scenario
    - **Active learning:** Reduce from 10^7 to 10^4 simulations
  - ~ 250 (nonstop) days on high-end workstation
  - ~ 2 weeks (nonstop) on a smallish cluster
  - ~ 12 hours on a Cray



Sample designs from the study
[View on YouTube]

Try this model yourself! (2D)

nanohub.org/tools/pc4cancerimmune

Reference:
Ozik et al. (2019)

Benefits of active learning

- For each scenario (e.g., 10% scenario), we built a RF of binary DT classifiers:
  - **True**: points that meet the design goal (e.g., $\underline{N}_{\text{final}} \leq 0.1\ N_{\text{start}}$)
  - **False**: points that don't meet the design goal (e.g., $\underline{N}_{\text{final}} > 0.1\ N_{\text{start}}$)

- **Rank** the importance of **parameters** based on the **Gini** coefficients
  - Most important: apoptosis rate (relates to T-cell exhaustion)
  - Next most important: oncoprotein threshold (relates to immunogencity)

- Most optima were near the hypercube boundary
  - Barriers to therapy success are driven by biological and clinical constraints

- A basic model can **predict key drivers in treatment response** with **no molecular biology**

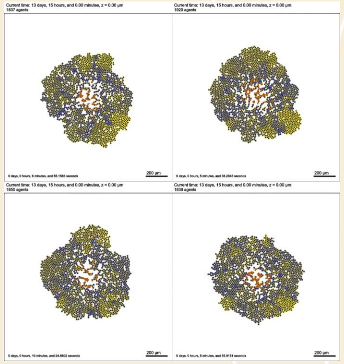
Reference:
Ozik et al. (2019)

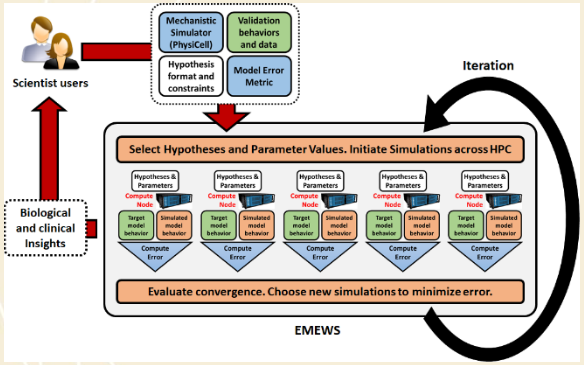
Future directions

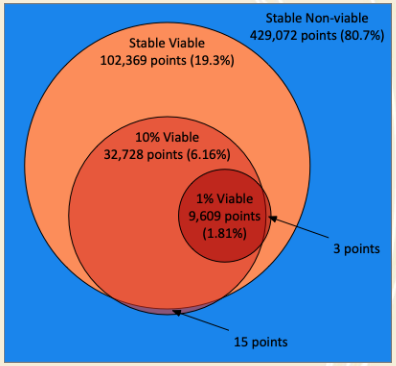
Multi-stage biosystems design

1. Work with domain experts in medicine, biology, chemistry, physics:
   - Expert observations and data drive model rules
   - **Choose design goals** (e.g., build a tissue, control cancer population)

2. Build and explore a multicellular simulation model
   - Run thousands (or millions!) of simulations on HPC
   - Find model rules that **achieve the design goal**




Scientist users
Mechanistic Simulator (PhysiCell)
Validation behaviors and data
Hypothesis format and constraints
Model Error Metric
Iteration
Select Hypotheses and Parameter Values. Initiate Simulations across HPC
Hypotheses & Parameters
Compute Node
Target model behavior
Simulated model behavior
Compute Error
Evaluate convergence. Choose new simulations to minimize error.
EMEWS
Biological and clinical Insights



3. Work with domain experts in synthetic biology, molecular engineering:
   - **Implement the cell programs**: growth factors, siRNA, synbio …

Improvements with next-gen computing

- If we can **improve performance**:
  - Faster diffusion solvers (e.g., via GPU computing)
  - Hybrid OpenMP-MPI for cell agents
  - AI accelerations

- And if we could **run on next-gen HPC** systems, we could:
  - Simulate **more immune cell types**
    - More sophisticated models of immune-immune interactions
    - More sophisticated models of immune differentiation
    - More sophisticated models of tumor-immune interactions
  - Add **molecular-scale effects** to each cell agent
    - Boolean networks or systems of ODEs for each cell
    - represent at SBML, parse, attach model instance to each cell
    - receptor and other signaling models
  - Run the high-parameter studies **in full 3D**

Moving towards the clinic: digital twins

Emerging joint initiative: DOE, NCI, Academia, National Labs

"Digital Twins" for predictive oncology

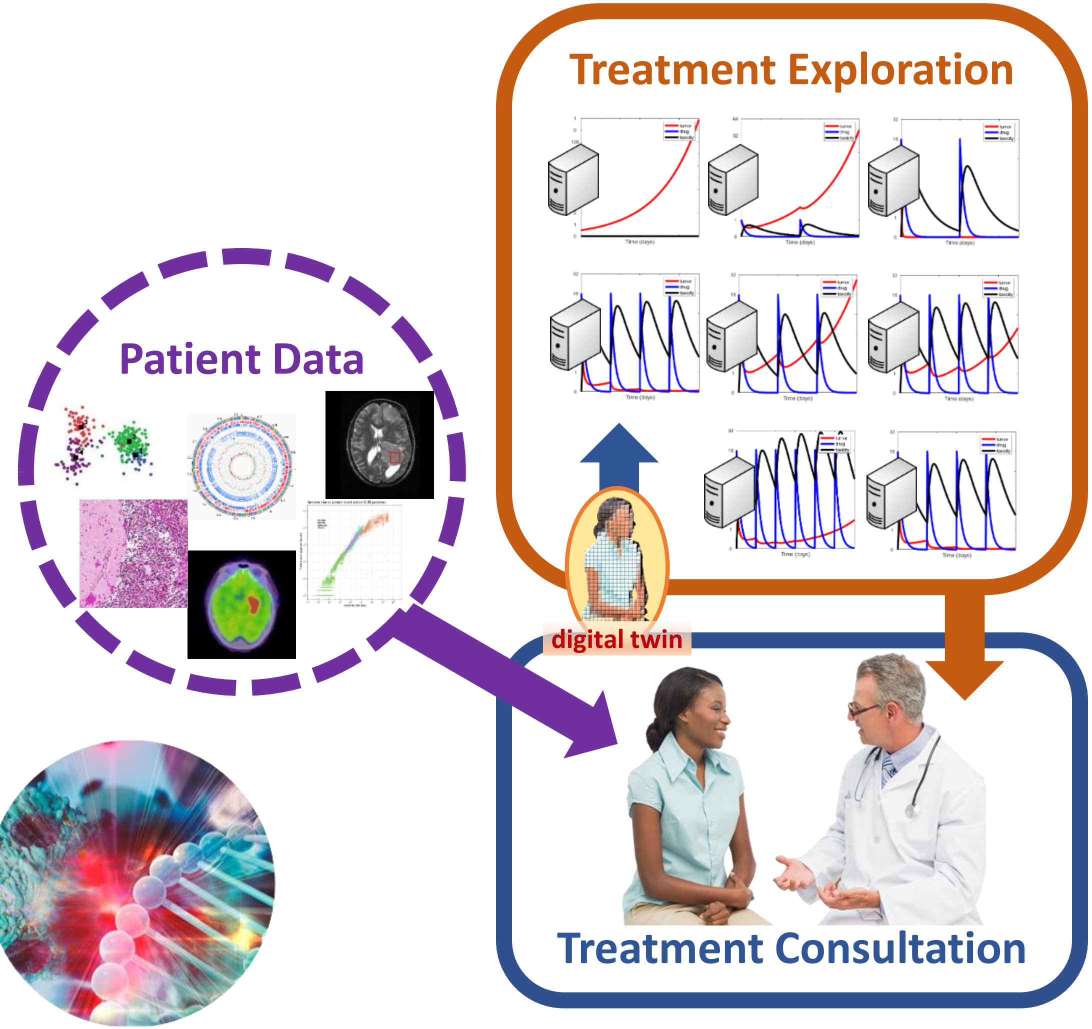
Team leads

Tina Hernandez-Boussard, Stanford
Paul Macklin, Indiana University
Tanveer Syeda-Mahmood, IBM Research
Ilya Shmulevich, Institute for Systems Biology

Jonathan Ozik, Argonne National Lab
Nicholson Collier, Argonne National Lab

Emily Greenspan, NCI
Carolyn Lauzon, DOE

High-throughput modeling with digital twins for the clinic



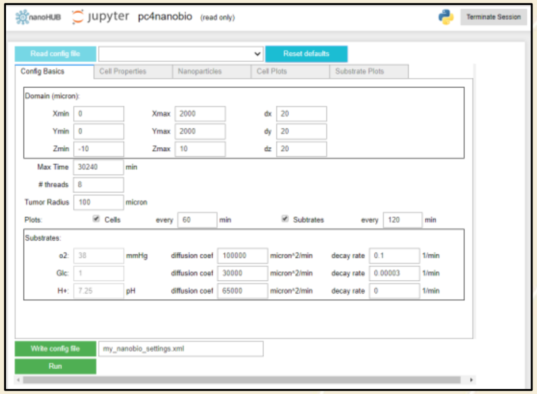
Digital Twin Concept

1. Patient and oncologist discuss **goals** and **preferences**
2. Clinicians build a "**digital twin**"
3. Clinicians use HPC to simulate **thousands of treatment options** on the virtual twin
4. Patient and clinician explore risks, benefits, side effects
5. They **choose a plan** and **monitor progress** against their digital twin

Opening up high-tech resources to the public

Jupyter-based GUIs

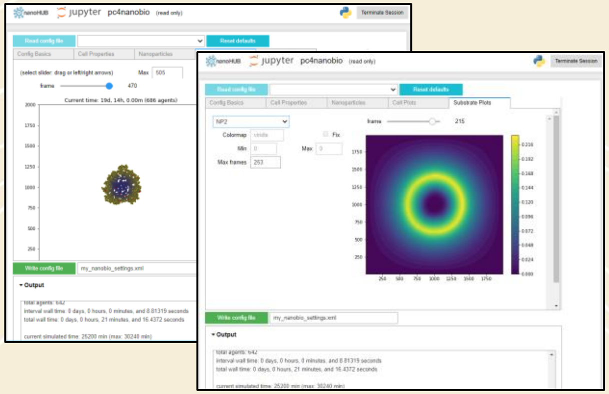
- **Goal:** make the simulator user-friendly, available without installing / compiling



Domain (micron):
Xmin 0 Xmax 2000 dx 20
Ymin 0 Ymax 2000 dy 20
Zmin -10 Zmax 10 dz 20
Max Time 30240 min
# threads 8
Tumor Radius 100 micron
Plots: ☑ Cells every 60 min ☑ Subtrates every 120 min
Substrates:
o2: 38 mmHg diffusion coef 100000 micron^2/min decay rate 0.1 1/min
Glc: 1 diffusion coef 30000 micron^2/min decay rate 0.00003 1/min
H+: 7.25 pH diffusion coef 65000 micron^2/min decay rate 0 1/min
my_nanobio_settings.xml

GUI: settings

Jupyter notebook



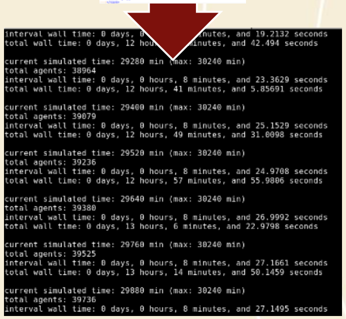
Max frames 253
my_nanobio_settings.xml

```
total agents: 642
interval wall time: 0 days, 0 hours, 0 minutes, and 8.81319 seconds
total wall time: 0 days, 0 hours, 21 minutes, and 16.4372 seconds

current simulated time: 25200 min (max: 30240 min)
```

GUI: output

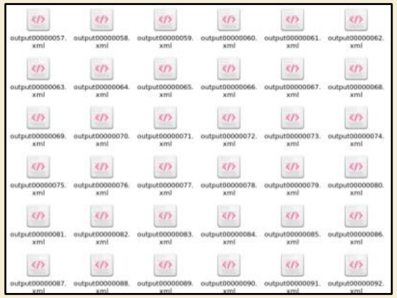
Jupyter notebook

XML config file



```
interval wall time: 0 days, 0                inutes, and 19.2132 seconds
total wall time: 0 days, 12 hou              minutes, and 42.494 seconds

current simulated time: 29280 min (max: 30240 min)
total agents: 38964
interval wall time: 0 days, 0 hours, 8 minutes, and 23.3629 seconds
total wall time: 0 days, 12 hours, 41 minutes, and 5.85691 seconds

current simulated time: 29400 min (max: 30240 min)
total agents: 39079
interval wall time: 0 days, 0 hours, 8 minutes, and 25.1529 seconds
total wall time: 0 days, 12 hours, 49 minutes, and 31.0098 seconds

current simulated time: 29520 min (max: 30240 min)
total agents: 39236
interval wall time: 0 days, 0 hours, 8 minutes, and 24.9708 seconds
total wall time: 0 days, 12 hours, 57 minutes, and 55.9806 seconds

current simulated time: 29640 min (max: 30240 min)
total agents: 39380
interval wall time: 0 days, 0 hours, 8 minutes, and 26.9992 seconds
total wall time: 0 days, 13 hours, 6 minutes, and 22.9798 seconds

current simulated time: 29760 min (max: 30240 min)
total agents: 39525
interval wall time: 0 days, 0 hours, 8 minutes, and 27.1661 seconds
total wall time: 0 days, 13 hours, 14 minutes, and 50.1459 seconds

current simulated time: 29880 min (max: 30240 min)
total agents: 39736
interval wall time: 0 days, 0 hours, 8 minutes, and 27.1495 seconds
```

PhysiCell simulation



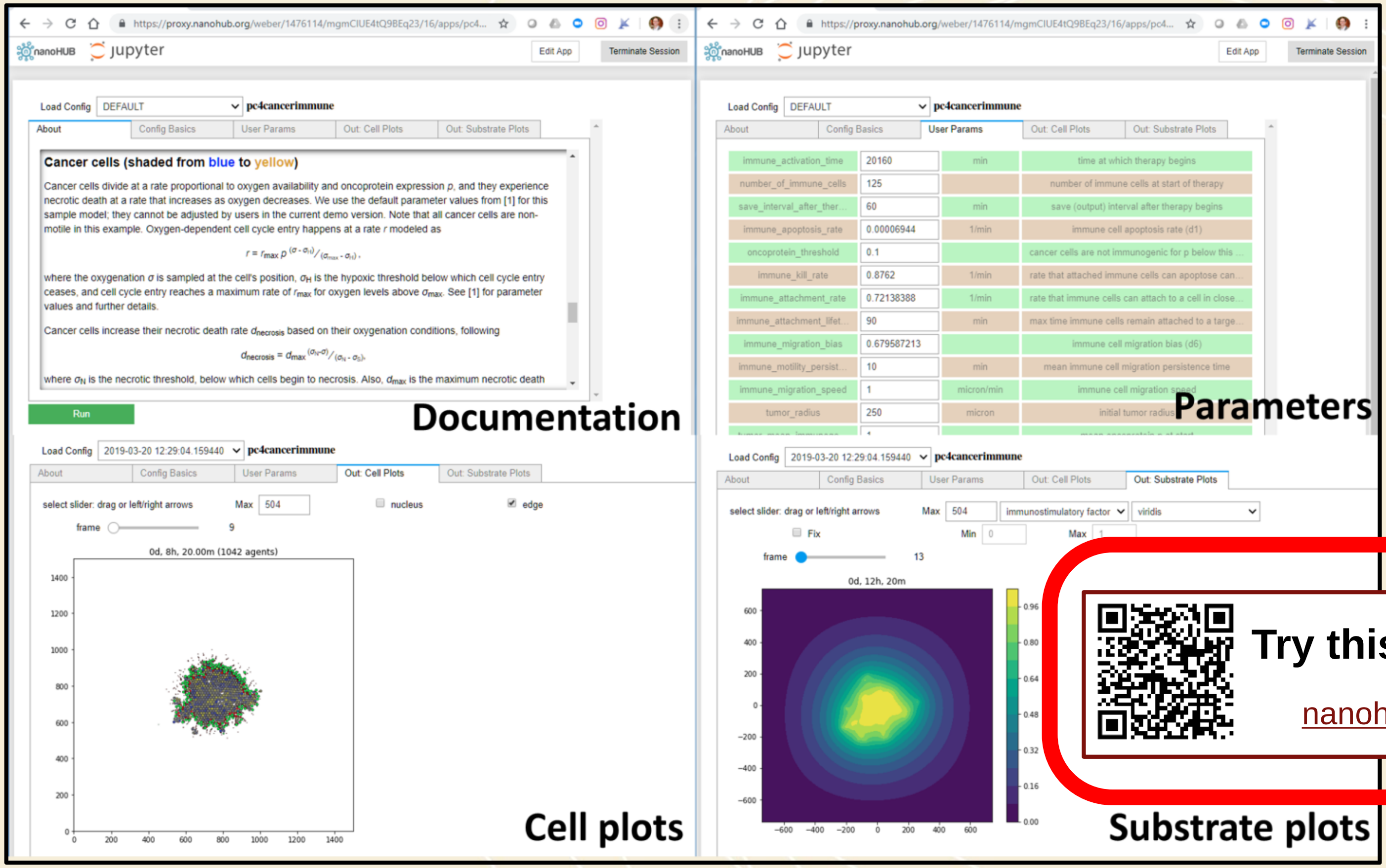
```
output00000057.xml  output00000058.xml  output00000059.xml  output00000060.xml  output00000061.xml  output00000062.xml
output00000063.xml  output00000064.xml  output00000065.xml  output00000066.xml  output00000067.xml  output00000068.xml
output00000069.xml  output00000070.xml  output00000071.xml  output00000072.xml  output00000073.xml  output00000074.xml
output00000075.xml  output00000076.xml  output00000077.xml  output00000078.xml  output00000079.xml  output00000080.xml
output00000081.xml  output00000082.xml  output00000083.xml  output00000084.xml  output00000085.xml  output00000086.xml
output00000087.xml  output00000088.xml  output00000089.xml  output00000090.xml  output00000091.xml  output00000092.xml
```

Simulation data

The Jupyter notebook and executable can be **cloud-hosted as an app**.
This allows **model sharing** without download, compiling, and other difficulties.

Use case: "Try this model yourself!"

- With xml2jupyter, we can automatically create a Jupyter-based GUI for any PhysiCell model, and host it on nanoHUB as an interactive model.
- The apps can easily be **included in talks, posters, and presentations**.



Load Config: DEFAULT — **pc4cancerimmune**

Cancer cells (shaded from blue to yellow)

Cancer cells divide at a rate proportional to oxygen availability and oncoprotein expression p, and they experience necrotic death at a rate that increases as oxygen decreases. We use the default parameter values from [1] for this sample model; they cannot be adjusted by users in the current demo version. Note that all cancer cells are non-motile in this example. Oxygen-dependent cell cycle entry happens at a rate r modeled as

$$r = r_{\max}\, p\, {}^{(\sigma - \sigma_H)}/_{(\sigma_{\max} - \sigma_H)},$$

where the oxygenation σ is sampled at the cell's position, σ_H is the hypoxic threshold below which cell cycle entry ceases, and cell cycle entry reaches a maximum rate of $r_{\max}$ for oxygen levels above $\sigma_{\max}$. See [1] for parameter values and further details.

Cancer cells increase their necrotic death rate $d_{necrosis}$ based on their oxygenation conditions, following

$$d_{necrosis} = d_{\max}\, {}^{(\sigma_N - \sigma)}/_{(\sigma_N - \sigma_S)},$$

where σ_N is the necrotic threshold, below which cells begin to necrosis. Also, $d_{\max}$ is the maximum necrotic death

Documentation

Load Config: DEFAULT — **pc4cancerimmune**

immune_activation_time	20160	min	time at which therapy begins
number_of_immune_cells	125		number of immune cells at start of therapy
save_interval_after_ther...	60	min	save (output) interval after therapy begins
immune_apoptosis_rate	0.00006944	1/min	immune cell apoptosis rate (d1)
oncoprotein_threshold	0.1		cancer cells are not immunogenic for p below this ...
immune_kill_rate	0.8762	1/min	rate that attached immune cells can apoptose can...
immune_attachment_rate	0.72138388	1/min	rate that immune cells can attach to a cell in close...
immune_attachment_lifet...	90	min	max time immune cells remain attached to a targe...
immune_migration_bias	0.679587213		immune cell migration bias (d6)
immune_motility_persist...	10	min	mean immune cell migration persistence time
immune_migration_speed	1	micron/min	immune cell migration speed
tumor_radius	250	micron	initial tumor radius

Parameters

Load Config: 2019-03-20 12:29:04.159440 — **pc4cancerimmune**

select slider: drag or left/right arrows — Max: 504 — □ nucleus — ☑ edge

frame: 9

Cell plots

Load Config: 2019-03-20 12:29:04.159440 — **pc4cancerimmune**

select slider: drag or left/right arrows — Max: 504 — immunostimulatory factor — viridis

□ Fix — Min: 0 — Max: 1

frame: 13

Substrate plots

Try this model yourself! (2D)

nanohub.org/tools/pc4cancerimmune

HPC-powered 3D simulations for the public

- Users run a cloud-hosted PhysiCell model with a friendliy GUI:

1. Set up a big 3D model via cloud interface
2. GUI initiates simulations on remote HPC.
3. Results delivered back to the GUI, just as before.

Untrained laypeople could run a sophisticated 3D simulation on the web, faster than trained scientist users can today.

The general public can try sophisticated models and HPC.

Some notes (and thank you!)

- **Collaborations** and **exchanges**
  - Seeking experimental collaborators for immunoctherapies
  - We're happy to help you adapt PhysiCell (+HPC, +nanoHUB) to projects
  - We write **letters of support** for **travel fellowships** to host IU visitors:
    - Learn PhysiCell, write models together, and share interactive models on nanoHUB.
    - So far: University of Sydney, Barcelona Supercomputing Center, EU-funded nanotherapy

- **(Possibly) hiring a postdoc**
  - I will likely have funding for a 3-year postdoc soon
  - Work on breast cancer metastasis
  - Seeking a math/computing savvy postdoc to push the work
  - Opportunities to refine PhysiCell and visualization

Some references

- **PhysiCell** Method paper -- *PLoS Computational Biology*
  - 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.
- **PhysiCell** + High-Throughput Computing (**HTC**) -- *BMC Bioinformatics*
  - 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** + **HPC** + **machine learning** -- *Molecular Systems Design and Engineering*
  - J. Ozik, N. Collier, R. Heiland, G. An, and P. Macklin. Learning-accelerated Discovery of Immune-Tumour Interactions. *Molec. Sys. Design Eng.*, 2019. **DOI**: 10.1039/c9me00036d
- **PhysiCell** + **Boolean** Networks -- *Bioinformatics*
  - 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.
- Automatically create **Jupyter-based GUIs** -- *Journal of Open Source Software*
  - R. Heiland, D. Mishler, T. Zhang, E. Bower, and P. Macklin. xml2jupyter: Mapping parameters between XML and Jupyter widgets. *J. Open Source Software*, 2019 (in review). **Preprint:** https://dx.doi.org/10.1101/601211