

A synthetic multicellular system to deliver a therapeutic "cargo"

1 Introduction

This app demonstrates how simple "rules" for chemical- and contact-based communication could be used for *active* delivery of therapeutic compounds, while overcoming traditional biotransport limits and using hypoxia to target delivery. It is based on a published PhysiCell example in Ghaffarizadeh et al.[2]. In this model, **cancer cells (green)** rapidly divide while consuming oxygen, which drives the emergence of hypoxic gradients. Greater availability of oxygen drives faster proliferation, while low oxygen can trigger necrotic death, creating a necrotic core (brown cells) in the center of the tumor. After initial growth, we "inject" a multicellular synthetic therapy.

In this (for now theoretical) synthetic therapy, **"worker cells" (red)** search for **"cargo cells" (blue)**, form focal adhesions, and haul their cargo towards hypoxic (oxygen-depleted) regions in the tissue. Once they reach a region of sufficiently low oxygen, they release their cargo and resume their initial search for more cargo. Released cargo cells secrete a chemotherapeutic compound that can kill nearby tumor cells. (Cancer cells are shaded according to their level of therapy-induced damage.)

This model and cloud-hosted demo are part of a course on computational multicellular systems biology created and taught by Dr. Paul Macklin in the Department of Intelligent Systems Engineering at Indiana University. It is also part of the education and outreach for the NSF-funded IU Engineered nanoBIO Node and the NCI-funded cancer systems biology grants. The models are built using PhysiCell: a C++ framework for multicellular systems biology.

2 Chemical components

All chemical signals move by chemical diffusion in the simulated environment, using BioFVM[1] to solve the reaction-diffusion equations. A key property is the *diffusion length scale*. Diffusion (with a diffusion coefficient D) helps spread a signal over large distances, while decay (and uptake) (with coefficient λ) eliminates the signal to slow its spread. These effects compete to determine the characteristic distance L that a chemical signal travels.

That length scale, L , is given by $L = \sqrt{D/\lambda}$.

Choosing D and λ can help to tune the chemical communication distance in these models.

2.1 Oxygen (σ)

This is released by the computational boundary (a simplified model of a far-field vasculature), and consumed by tumor cells. We use a far-field value of 38 mmHg (about 5%), which is a typical physioxenic condition. See [1] and [2] for more references. We use a length scale of 1000 μm , and cancer cell uptake gives a length scale of 100 μm in densely packed regions.

2.2 Chemoattractant (a):

This factor is released by undocked cargo cells to help guide chemotaxis by worker cells.

The chemoattractant is nondimensionalized so that $0 \leq a \leq 1$, and we initially assign a length scale of 100 μm .

2.3 Chemotherapeutic (c):

This factor is released by released cargo cells to kill cancer cells.

The therapeutic is nondimensionalized so that $0 \leq c \leq 1$, and we initially assign a length scale of 80 μm .

3 Cell agents

We use agent-based models to explore multicellular systems like this one. Each cell is modeled as a software agent with an independent state, and its own rules to change its behavior based on local environmental conditions and communication. We use PhysiCell as our modeling platform [1].

3.1 Key features in the agent-based models

Birth and death: Cells can proliferate (divide into two daughter cells of half size), with division rates regulated by cell rules. In our model, cells continuously work to grow towards a "target" mature volume.

Note that birth and death are stochastic events for each cell agent: if such a process occurs at rate r , then between now (t) and soon ($t + \Delta t$), the probability of the event occurring for that agent is $r\Delta t$.

Secretion and uptake: Cells can secrete chemical factors, or they can remove them (i.e., consume or uptake). This can contribute to gradients in chemical factors, and it's a key method of communication between cells. Moreover, cells can "sample" the chemical state of their surrounding environment. Secreting is sending a signal. Uptaking and sampling is receiving a signal.

Biased migration: Motile cells perform *biased migration*. After traveling for $T_{\text{persistence}}$ time (the mean persistence time), they choose a new migration direction $\mathbf{d}_{\text{migrate}}$. Based on the environmental conditions, the cell chooses a directional bias (intended direction) \mathbf{d}_{bias} and a random unit vector \mathbf{r} . The motile direction $\mathbf{d}_{\text{migrate}}$ is then determined by according to:

$$\mathbf{d}_{\text{migrate}} = b \mathbf{d}_{\text{bias}} + (1 - b) \mathbf{r}.$$

Here, b is a *bias* parameter (between 0 and 1) that determines how strongly biased migration is towards \mathbf{d}_{bias} . If $b=1$, then migration is deterministic towards \mathbf{d}_{bias} . If $b=0$, then migration is completely random (Brownian).

Once the migration direction is chosen, it is normalized, and multiplied by the cell's migration speed.

Mechanics: Cell agents can stick to one another within a prescribed interaction distance (some multiple of their radius), and they can exert a pushing force on neighbors. PhysiCell [1] uses potential functions to implement these simple mechanics. Notably, PhysiCell is an off-lattice model, meaning that cells can have variable sizes, and can move freely without grid artifacts.

They aren't required to move some prescribed number of spatial steps. They also can divide without checking for an open neighbor "site". Instead, they can divide and push their neighbors out of the way.

Focal adhesions: PhysiCell allows custom functions for any cell agents. We use this to model focal cell-cell adhesions as spring-like forces. If two cells i and j (with positions \mathbf{x}_i and \mathbf{x}_j) have a focal adhesion, then cell i is adhered to cell j with a spring-like force that we model as:

$$\mathbf{F}_{ij} = \varepsilon(\mathbf{x}_j - \mathbf{x}_i),$$

where ε is the elastic coefficient. Because PhysiCell has an inertialess formulation (see the method paper in [1]), this is added directly to the cell's velocity, and so ε has units min^{-1} . To model the finite deformability of cells and these elastic focal adhesions, we detach the cells if their distance ($\|\mathbf{x}_i - \mathbf{x}_j\|$) exceeds a set maximum distance r_{max} .

3.1.1 Worker cells (red)

Worker cells do not divide, but they can be assigned an apoptotic death rate to model normal "wear and tear." We model workers as non-adhesive except by spring-like focal adhesions; see focal adhesions. They are made more "repulsive" (by default, fivefold more repulsive) to allow them to more readily move through dense cellular regions. Worker cells consume oxygen, but at a reduced rate (by default, 10% of the rate of cancer cells).

Unattached worker cells use the following as their migration bias direction:

$$\mathbf{d}_{\text{bias}} = \nabla a,$$

where a is the chemoattractant released by unattached cargo cells. Note that ∇ is the gradient operator that points towards higher concentrations of a chemical factor.

Unattached worker cells test for near contact with unattached cargo cells. If they detect an unattached cargo cell with active receptor R ($R \approx 1$), they form a focal adhesion (see focal adhesions).

Attached worker cells change their migration bias direction to:

$$\mathbf{d}_{bias} = -\nabla\sigma,$$

where σ is the oxygen.

3.1.2 Cargo cells (blue)

Cargo cells do not divide, but they can be assigned an apoptotic death rate to model normal "wear and tear." We model cargo as non-adhesive except by spring-like focal adhesions; see focal adhesions. They are made more "repulsive" (by default, fivefold more repulsive) to allow them to more readily move through dense cellular regions. Cargo cells consume oxygen, but at a reduced rate (by default, 10% of the rate of cancer cells). Note that all cargo cells are non-motile.

Initially, all cargo cells express a receptor R with dimensionless value 1, which is used to control their chemical secretions. If $R = 1$, they secrete chemoattractant a to help guide worker cells. If $R = 0$ and they are unattached, then they secrete the therapeutic c .

Cargo cells permanently deactivate their receptor (set $R = 0$) once they are attached to a worker cell. They detach from worker cells if $\sigma < \sigma_{release}$.

3.1.3 Cancer cells (green)

Cancer cells divide at a rate proportional to oxygen availability, 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.

Beyond oxygen-driven birth and hypoxia-driven death, cancer cells have a basic damage-repair model. Exposure to the chemotherapeutic compound c causes damage (ω , which can be repaired. We simulate this as

$$\frac{d\omega}{dt} = r_{damage} c - r_{repair} \omega$$

where c is sampled at the cell's position. Cancer cells increase their apoptotic death rate by their therapeutic death rate $d_{therapy}$, which we model as

$$d_{therapy} = \frac{\omega}{\omega_{max}} d_{max},$$

where d_{max} is the maximum death rate that is reached when damage (ω) reaches ω_{max} .

In this model, we set ω_{max} by solving the damage model to find the steady state damage at a maximum value of c (which is 1):

$$\omega_{max} = r_{damage}/r_{repair}$$

3.2 Basic instructions

Modify the parameters in the "Config Basics" and "User Params" tabs and click the "Run" button.

To view simulation results, click the "Out: Plots" tab, and slide the bar to advance through simulation frames. Note that as the simulation runs, the "# frames" field will increase, so you can view more simulation frames.

Note that you can download full simulation data for further exploration in your tools of choice.

4 Additional information

To learn more about our work, please visit MathCancer.org.

Credits:

Scientific Staff: Randy Heiland, Research Associate, Intelligent Systems Engineering, Indiana University.

- Lead software architect:

Paul Macklin, Ph.D. , Associate Professor, Intelligent Systems Engineering, Indiana University.

- Scientific lead, developed PhysiCell simulation model:

The following undergraduate students have contributed to this project:

Jupyter notebook and UI development:

- Eric Bower (IU Intelligent Systems Engineering, B.S., 2018-present)
- Daniel Mishler (IU Intelligent Systems Engineering, B.S., 2018-present)
- Tyler Zhang (IU Intelligent Systems Engineering, B.S., 2018)

5 Sponsored by

- NSF EEC-1720625. Network for Computational Nanotechnology - Engineered nanoBIO Node
- Breast Cancer Research Foundation
- Jayne Koskinas Ted Giovanis Foundation for Health and Policy
- NIH U01CA232137

References

- [1] GHAFFARIZADEH, A., FRIEDMAN, S. H., AND MACKLIN, P. BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations. *Bioinformatics* 32, 8 (Apr. 2016), 1256–1258.
- [2] GHAFFARIZADEH, A., HEILAND, R., FRIEDMAN, S. H., MUMENTHALER, S. M., AND MACKLIN, P. PhysiCell: an open source physics-based cell simulator for 3-D multicellular systems. *PLoS Comput. Biol.* 14, 2 (2018), e1005991.