High-throughput cancer hypothesis testing with an integrated PhysiCell-EMEWS workflow

Jonathan Ozik
Argonne National Laboratory
Argonne, IL USA

Nicholson Collier
Argonne National Laboratory
Argonne, IL USA

Justin Wozniak
Argonne National Laboratory
Argonne, IL USA

Charles Macal
Argonne National Laboratory
Argonne, IL USA

Chase Cockrell
Dept. of Surgery
University of Chicago
Chicago, IL USA

Samuel H. Friedman
Opto-Knowledge Systems, Inc.
Torrance, CA USA

Ahmadreza Ghaffarizadeh
Lawrence J. Ellison Center for Transformative Medicin
University of Southern California
Los Angeles, CA USA

Randy Heiland
Intelligent Systems Engineering
Indiana University
Bloomington, IN USA

Gary An
Dept. of Surgery
University of Chicago
Chicago, IL USA
docgca@gmail.com

Paul Macklin
Intelligent Systems Engineering
Indiana University
Bloomington, IN USA
Paul.Macklin@MathCancer.org
http://MathCancer.org

***Abstract*—Cancer is a complex, multiscale dynamical system, with interactions between tumor cells and non-cancerous systems. Therapies act on this cancer-host system, sometimes with unexpected results. Systematic investigation of mechanistic models could help identify the factors driving a treatment's success or failure, but exploring mechanistic models over high-dimensional parameter spaces is computationally challenging. In this paper, we introduce a high throughput computing (HTC) framework that integrates a mechanistic 3-D multicellular simulator (PhysiCell) with an extreme-scale model exploration platform (EMEWS) to investigate high-dimensional parameter spaces. We show early results in adapting PhysiCell-EMEWS to 3-D cancer immunotherapy and show insights on therapeutic failure. We describe a PhysiCell-EMEWS workflow for high-throughput cancer hypothesis testing, where thou-sands of mechanistic simulations are compared against data-driven error metrics to perform hypothesis optimization. We close by discussing novel applications to synthetic multicellular systems for cancer therapy.**

Keywords—Agent-based model, PhysiCell, cancer, immunotherapy, high performance computing, EMEWS, high throughput computing, hypothesis testing

I. Introduction

Cancer is a complex, dynamical system operating on many spatial and temporal scales: processes include molecular interactions (e.g., gene expression and protein synthesis; nanoseconds to minutes), cell-scale processes (e.g., cycle progression and motility; minutes to hours), tissue-scale processes (e.g., tissue mechanics and biotransport; minutes to days), and organ and organism-scale processes (e.g., organ failure and clinical progression; weeks, months, and years). Cancer-host interactions dominate throughout these scales, including interactions between tumor cells and the vasculature (hypoxic tumor cells trigger growth of new blood vessels; new but dysfunctional blood vessels supply further growth substrates and can promote metastasis), between tumor cells and stromal cells (tumor cells can prompt tissue remodeling that facilitates tissue invasion), and between tumor cells and the immune system (immune cells can kill tumor cells, but tumor cells can co-opt inflammation to promote their survival). See the reviews in [2-7]. When designing and evaluating new cancer treatments, it is imperative to consider the impact on this complex multiscale cancer-host system.

Cancer-host interactions have been implicated in the poor (and sometimes surprising) clinical outcome of existing and new treatments. Chemotherapies fail when molecular-scale processes (e.g., DNA repair failures, mutations, or epigenetic alterations) cause resistant tumor clones to emerge (multicellular-scale birth-death processes) which can survive the treatment [7-12]. Anti-angiogenic therapies that target blood vessels were expected to be potent agents against cancer [13], but disrupting tissue perfusion inhibits drug delivery and increases hypoxia, which was subsequently shown to select for more aggressive tumor phenotypes, including alternative metabolism, chemoresistance, and increase tissue invasion [14-16]. On the other hand, medications originally developed for osteoporosis (bone loss) were found to reduce the incidence of bone metastases, through unclear mechanisms, but hypothesized to arise from tumor-osteoclast interactions [17-19]. Such examples underscore the need to evaluate and improve cancer treatments from a cancer-host systems perspective.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Recent successes of cancer immunotherapies—such as CAR (chimeric antigen receptor) T-cell treatments [20, 21]—have brought heightened attention to cancer immunology. In some patients, immune cell therapies have been impressively successful, while other patient populations have demonstrated disappointing outcomes; this variability of patient response arises in part from the poorly-understood, complex interactions between cancer and the immune system [22-27]. This suggests that better immune therapies could be designed through systematic investigations of tumor-immune interactions

II. KEY ELEMENTS FOR SYSTEMATIC AND MECHANISTIC INVESTIGATION OF CANCER IMMUNOTHERAPY

Mathematical and computational models have been developed to simulate cancer and its interactions with the immune system (e.g., [28-30]). Some have even been designed for single simulations of 10^9 cells or more [31-33], but to date they have lacked one or more of the critical elements to systematically investigate cancer-immune dynamics across high-dimensional parameter spaces (or hypothesis spaces) to identify the factors driving immunotherapy failure or success. These key elements include:

1. efficient 3-D simulation of diffusive biotransport of multiple (5 or more) growth substrates and signaling factors on mm^3-scale tissues, on a single compute node;
2. efficient simulation of 3-D multicellular systems (10^5 or more cells) that account for basic biomechanics, single-cell processes, cell-cell interactions, and flexible cell-scale hypotheses, on a single compute node;
3. a mechanistic model of an adaptive immune response to a 3-D heterogeneous tumor, on a single compute node;
4. efficient, high-throughput computing frameworks that can automate thousands of simulations through high-dimensional hypothesis spaces to efficiently investigate the model behavior by distributing them across HPC/HTC resources; and
5. clear metrics to quantitatively compare simulation behaviors, allowing the formulation of a hypothesis optimization problem.

In this paper, we present BioFVM (finite volume method for biological problems) and PhysiCell (physics-based multicellular simulator) to address (1)-(3), and we give an overview of how these codes were optimized to run on single compute nodes with OpenMP [1, 34]. We present a computational model of adaptive immune response to tumor cells and early simulation results from [1]. We detail the extreme-scale model exploration platform (EMEWS) framework for guiding agent-based model (ABM) exploration in high-dimensional parameter spaces [35], and we demonstrate early work to test PhysiCell in EMEWS. We outline the overall computational experimental workflow for systematic, high-throughput hypothesis testing and optimization, and we close with a discussion of ongoing and future work, with applications to developing synthetic multicellular cancer treatment systems. We note that the entire framework is available at http://PhysiCell.MathCancer.org (for PhysiCell) and http://www.mcs.anl.gov/~emews/tutorial/ (for EMEWS)

We thank the Breast Cancer Research Foundation, Jayne Koskinas Ted Giovanis Foundation for Health and Policy, National Cancer Institute, the Department of Energy, and National Science Foundation for generous support.

III. EFFICIENT 3-D MULTI-SUBSTRATE BIOTRANSPORT

In prior work [34], we developed BioFVM: an open source framework to simulate biological diffusion of multiple chemical substrates (a vector $\boldsymbol{\rho}$) in 3-D, governed by the vector of partial differential equations (PDEs)

$$\frac{\partial \boldsymbol{\rho}}{\partial t} = \boldsymbol{D}\nabla^2\boldsymbol{\rho} - \boldsymbol{\lambda}\boldsymbol{\rho} + \boldsymbol{S}(\boldsymbol{\rho}^* - \boldsymbol{\rho}) - \boldsymbol{U}\boldsymbol{\rho} + \sum_{\{\text{cells } i\}} \delta(\boldsymbol{x} - \boldsymbol{x}_i) W_i [\boldsymbol{S}_i(\boldsymbol{\rho}_i^* - \boldsymbol{\rho}) - \boldsymbol{U}_i\boldsymbol{\rho}].$$

Here, $\mathbf{D}$ is the vector of diffusion coefficients, $\boldsymbol{\lambda}$ gives the decay rates, $\mathbf{S}$ and $\mathbf{U}$ are vectors of bulk source and uptake rates, and for each cell i, $\mathbf{S}_i$ and $\mathbf{U}_i$ are its secretion and uptake rates, W_i is its volume, and $\mathbf{x}_i$ is its position. All vector-vector products (e.g., $\boldsymbol{\lambda}\boldsymbol{\rho}$) are component-wise, $\boldsymbol{\rho}^*$ denotes a vector of saturation densities (at which secretion or a source ceases), and δ is the Dirac delta function.

As detailed in [34], we solve this equation by a first-order operator splitting: we solve the bulk source and uptake equations first, followed by the cell-based sources and uptakes, followed by the diffusion-decay terms. We use first-order implicit time discretizations for numerically stable first-order accuracy. When solving the bulk source/decay term, we have an *independent* vector of linear ordinary differential equations (ODEs) in each computational voxel of the form:

$$\frac{\partial \boldsymbol{\rho}}{\partial t} = \boldsymbol{S}(\boldsymbol{\rho}^* - \boldsymbol{\rho}) - \boldsymbol{U}\boldsymbol{\rho}$$

Each of these sets of ODEs can be solved with the standard backwards Euler difference, giving a first-order accurate, stable solution. We trivially parallelize the solution by dividing the voxels across the processor cores with OpenMP: each thread works on a single voxel's set of ODEs. Moreover, we wrote the ODE solver to work vectorially, with a small set of BLAS (basic linear algebra subprograms) implemented to reduce memory allocation, copy, and deallocation operations. (We implemented specific BLASes as needed to keep the framework source small and minimize dependencies to facilitate cross-platform portability across Windows, Linux, OSX, and other operating systems.) We solved the cell-centered sources and sinks similarly, by dividing the solvers across the cells by OpenMP (one set of ODEs per cell); note that each cell will act on the substrates in the voxel containing the cell center, by the Dirac delta formulation.

We solve the diffusion-decay equation by the locally one-dimensional (LOD) method, which transforms a single 3-D PDE into a series of three 1-D PDEs (one PDE with respect to the *x* derivatives, one for the *y* derivatives, and one for the *z* derivatives) [36, 37]. In any *x*-, *y*-, or *z*-strip, using centered 2nd-order finite differences for the spatial derivative and backward 1st-order Euler differences yields a tridiagonal linear system for each substrate's PDE; because each PDE has the same form, we have a vector of tridiagonal linear systems. In [34], we solved this system with a vectorized Thomas algorithm [38]: an efficient O(*n*) direct linear solver for a single tridiagonal linear system, which we vectorized by performing all addition, multiplication, and division operations vectorially (with term-wise vector-vector multiplication and division). As a further optimization, we took advantage of that fact that $\mathbf{D}$ and $\boldsymbol{\lambda}$ are constant and noted that the forward sweep stage of the Thomas algorithm only depends upon $\mathbf{D}$, $\boldsymbol{\lambda}$, and the spatial mesh, but not on the prior or current solution. Thus, we could pre-compute and cache in memory the forward-sweep steps in the *x*-, *y*-, and *z*-directions to reduce the processing time. We tested on numerous computational problems, and found the overall method was first-order accurate and stable in time, and second-order accurate in space [34]. Moreover, we found that the computational speed scaled linearly in the number of PDEs solved, with a slope much less than one: Simulating 10 PDEs takes approximately 2.6 times more computational effort than a single PDE, whereas sequentially solving 10 PDEs requires approximately 10 times more effort than a single PDE. See further results in [34].

In testing, we have found that this system can simulate 5-10 diffusing substrates on 1 million computational voxels (sufficient to simulate 8 mm^3 at 20 μm resolution) on a quad-core desktop workstation with 2 GB of memory; the performance was faster on a single computenode with greater computational core counts. This CPU-based algorithm maximizes cross-platform compatibility, but we anticipate a GPU implementation would be at least an order of magnitude faster.

IV. Efficient 3-D Multicellular Simulations

In [1], we developed a 3-D agent-based model by extending BioFVM's *basic agents* (discrete cell-like agents with static positions, which could secrete and consume chemical substrates in the BioFVM environment) to create extensible software cell agents. Each cell has an independent, hierarchically-organized phenotype (the cell's behavioral state and parameters) [39, 40], user-settable function pointers to define hypotheses on the cell's phenotype, volume changes, cell cycling or death, mechanics, orientation, and motility, and user-customizable data. The cells' function pointers can be changed at any time in the simulation, allowing dynamical cell behavior and even switching between cell types. The overall program flow progresses as follows. In each time step:

1. Update the chemical diffusing fields by solving the PDEs above with BioFVM.
2. For each cell, update the phenotype by evaluating each cell's custom phenotype function. Also run the cells' cell cycle/death models, and volume update models. This step is parallelized across all the cells by OpenMP.
3. Serially process the cached lists of cells that must divide, and cells that must be removed (due to death). Separating this from step 2 preserved memory coherence.
4. For each cell, evaluate the mechanics and motility functions to calculate the cells' velocities. This step can be parallelized by OpenMP because the cell velocities are based upon relative positions.
5. For each cell, update the positions (using the second-order Adams-Bashforth discretiation) using the pre-computed velocities. This step is also parallelized by OpenMP.
6. Update time.

The cell velocity functions (adapted from [41]) requires computing *n-1* pairwise cell-cell mechanical interactions for all *n* cells, giving $O(n^2)$ computational performance—this would be prohibitive beyond 10^3 or 10^4 cells. However, biological cells have finite interaction distances, so we created an interaction testing data structure that placed each cell's memory address in a Cartesian mesh, and limited cell-cell mechanical interaction testing to the nearest interaction voxels. This reduced the computational effort to $O(n)$. PhysiCell uses separate time step sizes for biotransport (Δt ~ 0.01 min), cell mechanics (Δt ~ 0.1 min), and cell processes (Δt ~ 6 min) to take advantage of the multiple time scales. See [1] for further details.".

V. Mechanistic 3-D Simulation of Adaptive Immune Response to Heterogeneous Tumors

In [1], we developed an initial model of an adaptive immune response to a heterogeneous tumor. In the model each tumor cell agent was assigned a normally-distributed value of an idealized mutant oncoprotein *p*, which was modeled as increasing the cell's response to oxygen availability to enter the cell cycle. See the first Frame in Figure 1, where the cell's expression of *p* is shaded from blue (lowest, $p = 0$) to yellow (highest, $p = 2$), in a simulation of a ~5 mm^3 domain on a quad-core desktop workstation. By 14 days, the tumor has grown by an order of magnitude (from ~10^4 to 10^5 cells), there is clear selection for the cells with the most *p* (the tumor is visibly more yellow), oxygen transport limits have lead to the formation of a necrotic core (brown central region), and the initial spherical symmetry has been lost due to the formation of clonal foci (larger, more homogenous yellow regions).

At 14 days, we introduce 7500 immune cells (red) to model an adaptive tumor response. As a simplified model of MHC (a surface complex that presents a "signature" sampling of fragments of the cell's peptides, allowing immune cells to learn to recognize the body's own cells), we assumed cells with greater *p* expression were more immunogenic: more likely to present abnormal peptides on MHC and be recognized as targets for immune attack. All

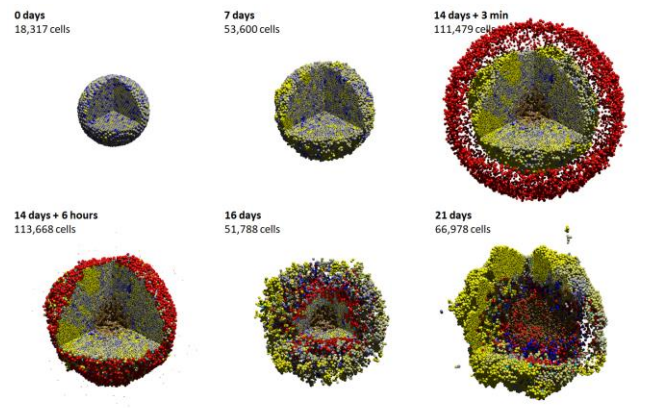

Fig. 1. 3D simulation of adaptive immune response to a heterogeneous tumor (cells ranging from blue (low proliferation and immunogenicity) to yellow (high proliferation and immunogenicity). A high-resolution animation can be viewed at https://www.youtube.com/watch?v=nJ2urSm4ilU. Adapted with permission from [1], which is currently under review at PLoS Computational Biology

tumor cells secreted an immunostimulatory factor that diffused through the domain (even *in situ* tumors are known to prompt immune cell homing [42]). Immune cells perform biased random migration (chemotaxis) along gradients of this factor, and test for collision with cells, and form tight adhesions with any cells that are found. See times 14 days + 3 minutes and 14 days + 6 hours in Figure 1. For any time interval $[t,t+\Delta t]$ while an immune cell i is attached to another cell j, the immune cell attempts to induce apoptosis (programmed cell death) with probability $r_i\, p_j\, \Delta t$, were r_i is the immune cell's killing rate for a normal immunogenicity $p = 1$, and p_j is the jth cell's oncoprotein expression; this models activation of a death receptor, such as FAS. For more background biology and references, see [1]. If an immune cell triggers apoptosis, it detaches and continues its search for new immunogenic targets. Otherwise, it remains attached, but with a similar stochastic process to regulate how long it remains attached.

By later simulation times (16 and 21 days in Figure 1),l we see that the immune cells continue migrating along the chemical gradient until reaching the center where the gradient is approximately flat. Due to the particular choice of motility parameters for the immune cells, they become temporarily trapped in the center, allowing tumor cells to evade therapy and re-establish the tumor. A high-resolution video of this simulation can be viewed at https://www.youtube.com/watch?v=nJ2urSm4ilU. In [1], we further discuss the biological lessons from this simulation: had the immune cell "homing" been weaker (i.e., more random, less biased along the chemical gradient), there would have been more mixing between the immune and tumor cells, leading to more cell-cell interactions, a greater probability of tumor cell killing, and a greater effective response. This simulation required approximately 2 days on a four-year-old desktop workstation, including time to save simulation outputs once every three simulated minutes. Simulations on a single compute node (with fewer I/O operations to save data, and more processor cores available) are significantly faster; simultaneously distributing thousands of such simulations across HPC/HTC resources is feasible

VI. Efficient Exploration of ABMs Across High-Dimensional Parameter Spaces

Our framework, Extreme-scale Model Exploration with Swift/T (EMEWS) [35] uses the general-purpose parallel scripting language Swift [43] to generate highly concurrent simulation workflows. These workflows enable the integration of external model exploration (ME) algorithms to coordinate the running and evaluation of large numbers of simulations. The general-purpose nature of the programming model allows the user to supplement the workflows with additional analysis and post-processing as well.

EMEWS enables the user to plug in both ME algorithms, such as IMIS, and scientific applications, such as CRSPIN. The ME algorithm can be expressed in Python, R, C, C++, Fortran, Julia, Tcl, or any language supported by Swift/T. The scientific application can be implemented as an external application called through the shell, in-memory libraries accessed directly by Swift (for faster invocation), or Python, R, Julia, and JVM language applications. Thus, researchers in various fields who may not be parallel programming experts can simply incorporate existing ME algorithms and run computational experiments on their existing scientific application without explicit parallel programming. A key feature of this approach is that neither the ME algorithm nor the scientific application is modified to fit the framework. This is implemented in a reusable way by connecting the parameter generating ME algorithm and output registration methods to interprocess communication mechanisms that allow these values to be exchanged with Swift/T. EMEWS currently provides this high-level queue-like interface with three implementations: EQ/Py, EQ/R and EQ/C (EMEWS Queues for Python, R, and C/C++).

We examined increasing the efficiency of parameter space search by utilizing Active Learning [44, 45] on an EMEWS implementation of an ABM of the innate immune response, the IIRABM. The IIRABM represents the host response to infection and generates system-level dynamics consistent with the clinical entity sepsis. We have used HPC-examination of the IIRABM to characterize global system properties that affect the ability to find treatments for sepsis (of which there are none) [46]; this work included an extensive parameter space characterization with respect to variables representing microbial virulence properties and host cardio-respiratory resilience. That parameter space characterization required running 70.4 million instances of the IIRABM (8800 parameter sets x 40 injury sizes x 100 stochastic replicates x 2 treatments), from which we identified boundaries for what considered clinically plausible parameter sets (n = 754) encompassing ~3 million instances.

We subsequently began to examined the potential increased efficiency of an Active Learning-based search using EMEWS. By the end of the initial training sequence the Active Learning algorithm only required sampling 20% of the total parameter space in order to achieve 99% accuracy in identifying clinically relevant parameter sets. As

we recognized that the increased efficiency was limited by the relatively coarse granularity of the original simulation data set, we are in the process of using synthetic data sets to further train the Active Learning algorithm. This experience, along with the structural similarity between the IIRABM and the Tumor-Immune cell ABM, suggests that implementation of the Tumor-Immune cell ABM on EMEWS and the use of Active Learning would greatly enhance the efficiency of parameter space characterization.

The structural similarity between the IIRABM and the Tumor-Immune cell ABM suggests that its implementation on EMEWS and the use of Active Learning would greatly enhance the efficiency of parameter space characterization.

A. *First application to PhysiCell*

Our first proof-of-concept PhysiCell workflow was a simple EMEWS-managed parameter sweep over the PhysiCell "Test Mechanics 1" example model, a convergence testing code which simulated mechanical relaxation of two initially overlapping cells; see the supplementary material in [1]. The workflow itself consists of five parts: The Test Mechanics 1 model compiled as a standalone application; a model input file containing individual parameter values, one per line; a bash script used to launch a model passing it the parameter value; the EMEWS Swift/T workflow script that performs the actual sweep; and lastly a bash script used to launch the workflow. The Test Mechanics 1 model was designed to test the convergence of the computational mechanics sub-model with respect to time; it takes a single parameter—the cell mechanics time step *dt*—as input, and it outputs the distance between the cells as several times throughout the relaxation process. The Swift/T workflow script reads this input file and launches a model run for each *dt* value, iterating over the lines of the input file. These model runs occur in parallel in dependence on the available resources. Each run occurs in its own working "instance" directory created by the workflow script, such that the model runs do not overwrite each other's output. The model itself is run as standalone application via a bash script that the EMEWS Swift/T script invokes, passing it a *dt* value. This bash script sets the instance directory as the model's working directory and then invokes the model passing the *dt* value to it. In this case, no modification of the model source code was required. (It would also have been possible to modify the model source code and compile it as a library, which would in turn be invoked using the Swift/T extension mechanism, making the model launch bash script redundant.) The last part of the workflow, the bash script used to launch the workflow script, performs some preliminary setup prior to calling the script, such as setting an experiment directory in which the instance directories mentioned above are created. It also provides defaults for running locally or on HPC/HTC resources which can be edited by the user.

VII. Formulation of the Hypothesis Testing Optimization Problem

Using the PhysiCell-EMEWS framework, we can perform high-throughput hypothesis testing: if the scientific end-users can supply:

1. A family of cell behavior hypotheses and constraints on their parameter values. For example:
   a. immune cells can exhibit any combination of random motility, chemotaxis towards tumor cells, or chemotaxis away from other immune cells
   b. attached immune cells can secrete immunoinhibitory or immunostimulatory factors
   c. tumor cells can secrete immunoinhibitory factors, but at a cost to cellular energy available for proliferation
   d. the microenvironment can have variable far-field oxygenation values.
2. A mechanistic computational model for simulating the cancer-host system under the hypotheses. For example:
   a. We implement the additional diffusion equations in BioFVM
   b. We implement the prior tumor cell immunogenicity model, and add a basic model of cell metabolism (e.g., as in [47]) with extra energy cost for secreting the immunoinhibitory factor
   c. We implement the prior immune cell adaptive response model but vary the cell motility according the specific hypotheses for migration bias along the various chemical gradients, the level of randomness, and we vary decrease the migration speed, adhesion rate, and cell killing rate under immunoinhibition.
3. A target or validation cell or system behaviors,

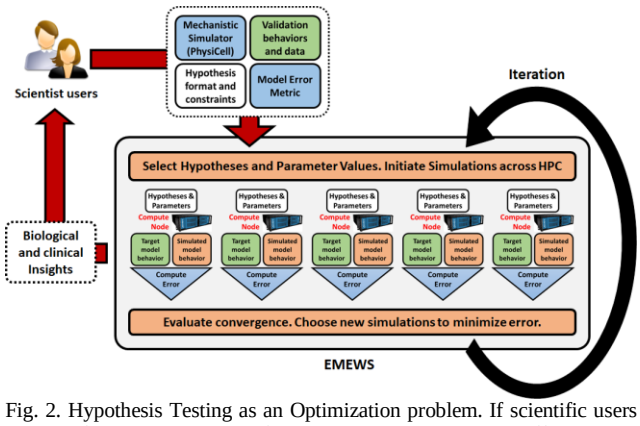

Fig. 2. Hypothesis Testing as an Optimization problem. If scientific users can (1) formulate a range of hypotheses, (2) supply an efficient 3-D mechanistic simulator (BioFVM+PhysiCell), (3) provide validation behaviors and/or data, and (4) supply an error metric, then the combined PhysiCell-EMEWS system can automatically explore the space of hypotheses, initiate simulations on HPC/HTC resources, collect data to evaluate the error metric, and then make further decisions on which hypotheses and parameter values to explore next. The framework iteratively sharpens hypotheses that bring new biological and clinical insights.

and/or validation data. For example:

   a. We seek hypotheses that result in emergence of immune-resistant tumors.

4. A model error metric to compare models and assess their match to target behavior. For example:
   a. For a set of hypotheses, we quantify the number of tumor cells after 48 hours of immune attack, the secretion level of the immunoinhibitory factor, and the mean immunogenicity (mutant oncoprotein)

Given these user inputs, PhysiCell-EMEWS can distribute simulations across the hypothesis space (each running independently on its own compute node, where they are optimized). For succinctness, we refer to a point in the hypothesis space as a single ***simulation ruleset***. Because these models are stochastic, EMEWS will initialize multiple simulations for each ruleset. EMEWS then collections the simulation outputs, evaluates the user-supplied metric against the target model behavior, and either reports the best hypothesis ruleset (if only one iteration is allowed), or repeats the process to refine the current best hypothesis ruleset (e.g., by a genetic algorithm). Each iteration is a **high-throughput hypothesis test**. And the overall iteration is **hypothesis optimization**. See Figure 2.

The output is a set of hypotheses *H* that lead to the desired cell behaviors. For example, in hypoxic conditions, we may see less selection for the immunoinhibitory secreting cells due to limited nutrients, unless the cells are under attack by many immune cells. This hypothesis could then be tested experimentally. If the hypothesis does not hold experimentally, we would refine the computational model (e.g., focusing more on hypoxic cell metabolic and motile adaptations.)

A. Early framework test

As an early test we created a fast 2-D tumor simulator that could simulate 48 hours of oxygen-limited tumor growth in 1-2 minutes. The framework integration proceeded as in Section VI.*A* above. To work through user-supplied elements:

1. Oxygenation conditions could vary from completely anoxic (0 mmHg) to typical values of well-oxygenated breast tissue (60 mmHg; see [34, 48]). The initial cell population could vary from 1 to 2000 cells.
2. PhysiCell was used to create small project that could read these two hypothesis parameters at the command line, initialize the simulation, and run to 48 hours without user input.
3. The target behavior was to maximize live cell fraction.
4. The model metric was the live cell fraction after 48 hours.

We ran a single iteration of the PhysiCell-EMEWS framework, with the following oxygenation values

0, 2.5, 5, 8, 10, 15, 38 or 60 mmHg

and the following initial cell counts:

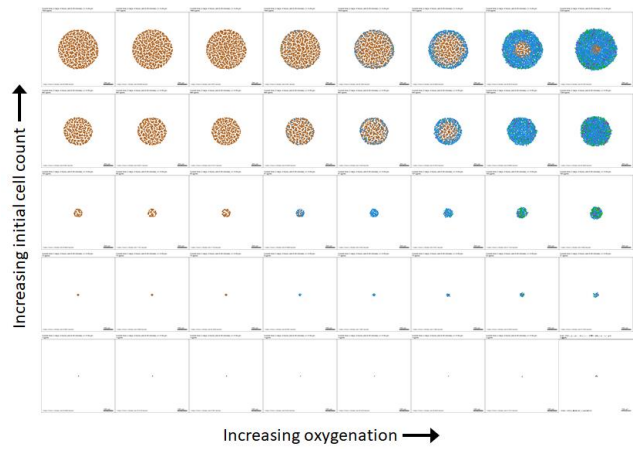

Fig. 3. Final simulation snapshots for the early framework test. Here necrotic cells (dead by oxygen starvation) are brown, non-cycling cells are blue, and cycling cells are green and magenta. Increasing the initial cell count increases the final cell count, but also decreases the final dead cell fraction (seen as the increasing prevalence of brown).

1, 10, 100, 1000, 2000

EMEWS saved the model outputs in separate directories, facilitating subsequent postprocessing analysis and visualization. We plot a 2-D array of the final simulation images in Figure 3 and the final live cell fractions in Figure 4. As expected, increasing the initial cell count always increases the final cell count (and overall tumor size) 48 hours later, but for any fixed oxygenation condition, this also leads to greater prevalence of necrosis, and a nonmonotonic effect on final live cell fraction (Figure 4).

In Figure 4, we plot the final live cell fraction as a function of the initial cell count, for each fixed oxygenation condition. For low oxygenation conditions (0, 2.5 mmHg), almost all cells are dead at 48 hours regardless of cell seeding choices. For intermediate oxygenation conditions (5 to 38 mmHg), the effect is nonmonotonic: for small initial cell populations (1 or 10 cells), stochastic apoptosis effects can sometimes leave a smaller final live fraction than a larger cell population; this highlights the importance of testing multiple simulation replicates for stochastic models. Past 100 initial cells, the stochastic effects are reduced, and increasing the initial cell count results in a lower final live fraction (due to oxygen depletion by the larger cell population and the emergence of a necrotic core). In particular, for these simulations increasing from 1000 to 2000 cells decreased the final live cell fraction. This behavior was not observed for high oxygenation (60 mmHg): no portions of the tumor ever drop below the necrotic threshold. Moreover, this simulated cell line has saturating proliferation above 38 mmHg pO_2 (tissue physioxia [48]), and so for sufficiently high initial oxygenation, the entire tumor stays about this threshold where there is no oxygen constraint to growth.

VIII. Discussion and Future Directions

We have demonstrated a realistic mechanistic tumor-immune interaction model (and more generally, a mechanis-

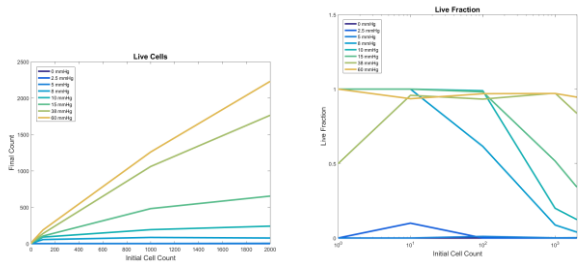

Fig. 4. Live tumor cell count (left) and live cell fraction (right) after 48 hours, as a function of oxygenation conditions (each curve is a different condition) and initial cell count (horizontal axis). For intermediate oxygenation conditions, increasing the initial cell count increases the final live cell count (left) but decreases the live cell fraction (right). Once oxygenation is high enough, any initial cell count yields nearly 100% live fraction at 48 hours.

tic agent-based multicellular systems biology modeling platform) that has an appropriate balance of flexibility, speed, and realism for efficient single simulations, that predict the emergent systems behaviors for a given set of cancer hypotheses. It is self-contained code (it can be distributed as a ZIP file), enabling very simple deployment in EMEWS.

We have shown a previously-developed extreme-scale model exploration and optimization platform, and demonstrated that it can compatibly deploy PhysiCell for high-throughput model exploration. We have outlined the overall platform to perform high-throughput hypothesis testing on the PhysiCell-EMEWS system, and we gave an early example on a simple (but spatially nontrivial) model system of hypoxic tumor growth.

In our next steps, we will further develop the mechanistic tumor-immune interaction model as described above, and import it into the EMEWS framework. We will explore the emergent tumor response to immune therapy under a variety of immune cell hypotheses and cancer phenotypes. Ultimately, we will generate hypotheses that elucidate the most and least ideal patient characteristics for immunotherapies.

We note that performance could be further improved. In particular, the diffusion solver (BioFVM) is well-suited to a GPU implementation by OpenCL, which could leverage GPU resources available on today's typical HPC/HTC compute nodes. Scientifically, complex molecular-scale systems biology is typically written as SBML (systems biology markup language) models [49]. To integrate these into high-throughput multiscale mechanistic hypothesis testing, we plan to implement an SBML model integrator (e.g., using the C-code generation functions of COPASI [50]).

Also, the generalized description of hypotheses is not yet fully mature. Standards have emerged to describe molecular-scale systems biology (generally systems of ODEs) as SBML [49], and more recently to express multicellular biology as MultiCellDS [39, 40], but cell-cell interaction rules will likely require a different description, such as by using elements of the cell behavior ontology (CBO) [51].

Cancer biology—particularly cancer-immune interactions—is driven by complex dynamical, multiscale systems that frequently yield surprising emergent behaviors that can impair treatment. High-throughput model investigation and hypothesis testing affords a new paradigm to attacking these complex problems, gaining new insights, and improving cancer treatment strategies.

We close by noting that this framework has applications beyond cancer. In general, testing multiscale hypotheses in high throughput is valuable in determining the rules underlying (often puzzling) experimental data, and even to evaluate the limitations of experiments themselves [52, 53]. Moreover, we envision that the PhysiCell-EMEWS framework could be used as a multicellular design tool: for any given multicellular design including single-cell and cell-cell interaction rules (which map onto hypotheses in this framework), PhysiCell-EMEWS can test the emergent multicellular behavior against the target behavior (the design goal), and iteratively tune the cell rules to achieve the design goal. In [1], we designed cell-cell interaction rules to create a multicellular cargo delivery system to (1) actively move cells to designated target areas, and then to (2) actively deliver a cell-secreted cancer therapeutic beyond regular drug transport limits to hypoxic cancer regions. In that work, we manually tuned the model rules to achieve this (as yet unoptimized) design objective, requiring days of people-hours to configure, code, test, visualize, and evaluate. Integrating such problems into a high-throughput design testing framework such as PhysiCell-EMEWS would be of clear benefit.

Acknowledgments

We thank the Breast Cancer Research Foundation, the Jayne Koskinas Ted Giovanis Foundation for Health and Policy, the National Cancer Institute (1R01CA180149), the National Institute of General Medical Science (1S10OD018495-01), the Department of Energy (National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH1123 and from Lawrence Livermore National Laboratory under Award #B616283), and the National Science Foundation (1720625) for generous support.

This manuscript was created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.

References

[1] A. Ghaffarizadeh, S. H. Friedman, S. M. Mumenthaler, and P. Macklin, "PhysiCell: an Open Source Physics-Based Cell Simulator for 3-D Multicellular Systems," *bioRxiv*, 2017.

[2] T. S. Deisboeck, Z. Wang, P. Macklin, and V. Cristini, "Multiscale cancer modeling," *Annu Rev Biomed Eng*, vol. 13, pp. 127-55, Aug 15 2011.

[3] J. S. Lowengrub, H. B. Frieboes, F. Jin, Y. L. Chuang, X. Li, P. Macklin, *et al.*, "Nonlinear modelling of cancer: bridging the gap between cells and tumours," *Nonlinearity*, vol. 23, pp. R1-R9, 2010.

[4] Y. Kam, K. A. Rejniak, and A. R. Anderson, "Cellular modeling of cancer invasion: integration of in silico and in vitro approaches," *J Cell Physiol*, vol. 227, pp. 431-8, Feb 2012.

[5] K. A. Rejniak and A. R. Anderson, "State of the art in computational modelling of cancer," *Math Med Biol*, vol. 29, pp. 1-2, Mar 2012.

[6] P. Macklin, H. B. Frieboes, J. L. Sparks, A. Ghaffarizadeh, S. H. Friedman, E. F. Juarez, *et al.*, "Progress Towards Computational 3-D Multicellular Systems Biology," *Adv Exp Med Biol*, vol. 936, pp. 225-246, 2016.

[7] P. Macklin, "Biological background," in *Multiscale Modeling of Cancer: An Integrated Experimental and Mathematical Modeling Approach*, V. Cristini and J. S. Lowengrub, Eds., ed Cambridge, UK: Cambridge University Press, 2010, pp. 8-23.

[8] G. Xiong, M. Feng, G. Yang, S. Zheng, X. Song, Z. Cao, *et al.*, "The underlying mechanisms of non-coding RNAs in the chemoresistance of pancreatic cancer," *Cancer Lett*, vol. 397, pp. 94-102, Jul 01 2017.

[9] K. M. Sakthivel and S. Hariharan, "Regulatory players of DNA damage repair mechanisms: Role in Cancer Chemoresistance," *Biomed Pharmacother*, vol. 93, pp. 1238-1245, Sep 2017.

[10] "Identifying mechanisms for cancer cell chemoresistance," *Biotechnol Bioeng*, vol. 114, p. 1914, Sep 2017.

[11] A. Martinez-Cardus, M. Vizoso, S. Moran, and J. L. Manzano, "Epigenetic mechanisms involved in melanoma pathogenesis and chemoresistance," *Ann Transl Med*, vol. 3, p. 209, Sep 2015.

[12] L. N. Abdullah and E. K. Chow, "Mechanisms of chemoresistance in cancer stem cells," *Clin Transl Med*, vol. 2, p. 3, Jan 17 2013.

[13] R. Kerbel and J. Folkman, "Clinical translation of angiogenesis inhibitors," *Nat Rev Cancer*, vol. 2, pp. 727-39, Oct 2002.

[14] H. L. Kindler, D. Niedzwiecki, D. Hollis, S. Sutherland, D. Schrag, H. Hurwitz, *et al.*, "Gemcitabine Plus Bevacizumab Compared With Gemcitabine Plus Placebo in Patients With Advanced Pancreatic Cancer: Phase III Trial of the Cancer and Leukemia Group B (CALGB 80303)," *Journal of Clinical Oncology*, vol. 28, pp. 3617-3622, 2010/08/01 2010.

[15] O. Keunen, M. Johansson, A. Oudin, M. Sanzey, S. A. Rahim, F. Fack, *et al.*, "Anti-VEGF treatment reduces blood supply and increases tumor cell invasion in glioblastoma," *Proc Natl Acad Sci U S A*, vol. 108, pp. 3749-54, Mar 01 2011.

[16] A. McIntyre and A. L. Harris, "Metabolic and hypoxic adaptation to anti-angiogenic therapy: a target for induced essentiality," *EMBO Mol Med*, vol. 7, pp. 368-79, Apr 2015.

[17] I. J. Diel, E. F. Solomayer, S. D. Costa, C. Gollan, R. Goerner, D. Wallwiener, *et al.*, "Reduction in new metastases in breast cancer with adjuvant clodronate treatment," *N Engl J Med*, vol. 339, pp. 357-63, Aug 06 1998.

[18] G. M. Oades, J. Coxon, and K. W. Colston, "The potential role of bisphosphonates in prostate cancer," *Prostate Cancer Prostatic Dis*, vol. 5, pp. 264-72, 2002.

[19] A. Mathew and A. Brufsky, "Decreased risk of breast cancer associated with oral bisphosphonate therapy," *Breast Cancer (Dove Med Press)*, vol. 4, pp. 75-81, May 23 2012.

[20] A. Holzinger, M. Barden, and H. Abken, "The growing world of CAR T cell trials: a systematic review," *Cancer Immunol Immunother*, vol. 65, pp. 1433-1450, Dec 2016.

[21] M. Haji-Fatahaliha, M. Hosseini, A. Akbarian, S. Sadreddini, F. Jadidi-Niaragh, and M. Yousefi, "CAR-modified T-cell therapy for cancer: an updated review," *Artif Cells Nanomed Biotechnol*, vol. 44, pp. 1339-49, Sep 2016.

[22] I. Mellman, G. Coukos, and G. Dranoff, "Cancer immunotherapy comes of age," *Nature*, vol. 480, pp. 480-9, Dec 21 2011.

[23] P. F. Robbins, R. A. Morgan, S. A. Feldman, J. C. Yang, R. M. Sherry, M. E. Dudley, *et al.*, "Tumor regression in patients with metastatic synovial cell sarcoma and melanoma using genetically engineered lymphocytes reactive with NY-ESO-1," *J Clin Oncol*, vol. 29, pp. 917-24, Mar 01 2011.

[24] Y. Li, F. Li, F. Jiang, X. Lv, R. Zhang, A. Lu, *et al.*, "A Mini-Review for Cancer Immunotherapy: Molecular Understanding of PD-1/PD-L1 Pathway & Translational Blockade of Immune Checkpoints," *Int J Mol Sci*, vol. 17, Jul 18 2016.

[25] D. M. Pardoll, "The blockade of immune checkpoints in cancer immunotherapy," *Nat Rev Cancer*, vol. 12, pp. 252-64, Mar 22 2012.

[26] W. H. Fridman, F. Pages, C. Sautes-Fridman, and J. Galon, "The immune contexture in human tumours: impact on clinical outcome," *Nat Rev Cancer*, vol. 12, pp. 298-306, Mar 15 2012.

[27] K. E. de Visser, A. Eichten, and L. M. Coussens, "Paradoxical roles of the immune system during cancer development," *Nat Rev Cancer*, vol. 6, pp. 24-37, Jan 2006.

[28] R. Eftimie, J. L. Bramson, and D. J. Earn, "Interactions between the immune system and cancer: a brief review of non-spatial mathematical models," *Bull Math Biol*, vol. 73, pp. 2-32, Jan 2011.

[29] J. A. Sherratt and M. A. Nowak, "Oncogenes, anti-oncogenes and the immune response to cancer: a mathematical model," *Proc Biol Sci*, vol. 248, pp. 261-71, Jun 22 1992.

[30] L. G. dePillis, A. Eladdadi, and A. E. Radunskaya, "Modeling cancer-immune responses to therapy," *J Pharmacokinet Pharmacodyn*, vol. 41, pp. 461-78, Oct 2014.

[31] S. Kang, S. Kahan, J. McDermott, N. Flann, and I. Shmulevich, "Biocellion: accelerating computer

simulation of multicellular biological system models," *Bioinformatics*, vol. 30, pp. 3101-8, Nov 01 2014.

[32] M. Cytowski and Z. Szymanska, "Large-Scale Parallel Simulations of 3D Cell Colony Dynamics," *Computing in Science & Engineering*, vol. 16, pp. 86-95, 2014.

[33] M. Cytowski and Z. Szymańska, "Enabling Large Scale Individual-Based Modelling through High Performance Computing," *ITM Web of Conferences*, vol. 5, p. 00014, 2015.

[34] A. Ghaffarizadeh, S. H. Friedman, and P. Macklin, "BioFVM: an efficient, parallelized diffusive transport solver for 3-D biological simulations," *Bioinformatics*, vol. 32, pp. 1256-8, Apr 15 2016.

[35] J. Ozik, N. T. Collier, J. M. Wozniak, and C. Spagnuolo, "From desktop to Large-Scale Model Exploration with Swift/T," in *2016 Winter Simulation Conference (WSC)*, 2016, pp. 206-220.

[36] G. I. Marchuk, "Splitting and alternating direction methods," in *Handbook of Numerical Analysis*. vol. 1, ed: Elsevier, 1990, pp. 197-462.

[37] N. N. Yanenko, "Simple Schemes in Fractional Steps for the Integration of Parabolic Equations," in *The Method of Fractional Steps: The Solution of Problems of Mathematical Physics in Several Variables*, M. Holt, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 1971, pp. 17-41.

[38] L. H. Thomas, "Elliptic problems in linear difference equations over a network," *Watson Sci. Comput. Lab. Rept., Columbia University, New York*, vol. 1, 1949.

[39] S. H. Friedman, A. R. A. Anderson, D. M. Bortz, A. G. Fletcher, H. B. Frieboes, A. Ghaffarizadeh, *et al.*, "MultiCellDS: a standard and a community for sharing multicellular data," *bioRxiv*, 2016.

[40] S. H. Friedman, A. R. A. Anderson, D. M. Bortz, A. G. Fletcher, H. B. Frieboes, A. Ghaffarizadeh, *et al.*, "MultiCellDS: a community-developed standard for curating microenvironment-dependent multicellular data," *bioRxiv*, 2016.

[41] P. Macklin, M. E. Edgerton, A. M. Thompson, and V. Cristini, "Patient-calibrated agent-based modelling of ductal carcinoma in situ (DCIS): from microscopic measurements to macroscopic predictions of clinical progression," *J Theor Biol*, vol. 301, pp. 122-40, May 21 2012.

[42] P. Macklin, S. Mumenthaler, and J. Lowengrub, "Modeling Multiscale Necrotic and Calcified Tissue Biomechanics in Cancer Patients: Application to Ductal Carcinoma In Situ (DCIS)," in *Multiscale Computer Modeling in Biomechanics and Biomedical Engineering*, A. Gefen, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 349-380.

[43] J. M. Wozniak, T. G. Armstrong, M. Wilde, D. S. Katz, E. Lusk, and I. T. Foster, "Swift/T: Large-Scale Application Composition via Distributed-Memory Dataflow Processing," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, pp. 95-102.

[44] B. Settles, "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, pp. 1-114, 2012/06/30 2012.

[45] M. Cevik, M. A. Ergun, N. K. Stout, A. Trentham-Dietz, M. Craven, and O. Alagoz, "Using Active Learning for Speeding up Calibration in Simulation Models," *Medical Decision Making*, vol. 36, pp. 581-593, 2016.

[46] C. Cockrell and G. An, "Sepsis reconsidered: Identifying novel metrics for behavioral landscape characterization with a high-performance computing implementation of an agent-based model," *J Theor Biol*, vol. 430, pp. 157-168, Oct 07 2017.

[47] R. A. Gatenby, K. Smallbone, P. K. Maini, F. Rose, J. Averill, R. B. Nagle, *et al.*, "Cellular adaptations to hypoxia and acidosis during somatic evolution of breast cancer," *Br J Cancer*, vol. 97, pp. 646-53, Sep 03 2007.

[48] S. R. McKeown, "Defining normoxia, physoxia and hypoxia in tumours-implications for treatment response," *Br J Radiol*, vol. 87, p. 20130676, Mar 2014.

[49] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, *et al.*, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, pp. 524-31, Mar 01 2003.

[50] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, *et al.*, "COPASI--a COmplex PAthway SImulator," *Bioinformatics*, vol. 22, pp. 3067-74, Dec 15 2006.

[51] J. P. Sluka, A. Shirinifard, M. Swat, A. Cosmanescu, R. W. Heiland, and J. A. Glazier, "The cell behavior ontology: describing the intrinsic biological behaviors of real and model cells seen as active agents," *Bioinformatics*, vol. 30, pp. 2367-74, Aug 15 2014.

[52] P. Macklin, "When Seeing Isn't Believing: How Math Can Guide Our Interpretation of Measurements and Experiments," *Cell Syst*, vol. 5, pp. 92-94, Aug 23 2017.

[53] S. H. Friedman and P. Macklin, "Computational investigation of biological and technical variability in high throughput phenotyping and cell line identification," *bioRxiv*, vol. 175703, 2017.