Multi-Agent Simulation of Self-Organizing Behaviors of Alveolar Cells In Vitro

Sean H. J. KIM, Shahab SHEIKH-BAHAEI, and C. Anthony HUNT

Abstract- **We present an agent-based model of cultured lung alveolar type II (AT II) cells and simulation results that provide early insight into generative principles underpinning alveolar morphogenesis in 3D cell cultures. We also describe a supporting framework and system-level agents that manage simulation experiments. The AT II cell-mimetic analogue comprises discrete components that represent cells and basic parts composing cell cultures. Cells are represented as quasi-autonomous agents that act based on decision logic and a set of operating principles articulated as axioms. By adhering strictly to those principles, cell agents self-organize and develop into multi-cell structures that resemble alveolar-like cysts in vitro. While it is premature to assign a specific biological mapping to the in silico operating principles, the findings enforce the idea that complex morphogenetic phenomena are a consequence of adherence to a small set of epigenetic principles. We expect more advanced adaptations will provide a rigorous platform to unravel the mechanistic bases of alveolar development and regeneration.**

***Index Terms*—Multi-agent, self-organization, autonomous systems design, agent-directed, simulation framework, behavior programming, systems biology**

1. Introduction

Alveolar morphogenesis is a fundamental feature of mammalian lung development and repair [1]. It comprises various processes by which terminal airway cells self-organize into primary functional lung units. An essential, yet poorly understood, aspect of alveolarization concerns how the various cell processes and actions give rise to stereotypical cyst formations. What are the basic principles of cell operation that govern alveolar growth? How do they come about? Can the morphogenic principles be translated into therapeutic strategies for lung injury or disease? Here we introduce an agent-based, discrete event simulation model and supporting methods to probe possible answers. Albeit simplistic, the model can reproduce basic developmental attributes and offers a logic-based, cell-level explanation of alveolar growth in vitro.

Pulmonary alveoli consist of type I and II cells [2]. Alveolar type I cells are flat and provide about 95% of alveolar surface for respiration. Alveolar type II (AT II) cells are approximately cuboidal and cover the remaining surface. While they provide little respiratory surface, AT II cells are essential for alveolar function, homeostasis, and regeneration [3], so they are one of the most studied cell types in lung and alveolar morphogenesis research.

Most ex vivo studies of alveolar development have relied on two-dimensional (2D) culture methods, even though they are incapable of emulating the physiological three-dimensional (3D) environment. There are no methods to isolate and grow whole alveoli ex vivo, however different 3D methods have been applied successfully to grow AT II cells into alveolar-like organoids [4][5]. In [5] the cells form alveolar-like cysts (ALCs) and secrete surfactant into the cyst lumen similar to the in vivo condition. ALCs grow exclusively by cell aggregation and subsequent hollowing (cell-cell separation). Apparently, ALC growth requires cellular mechanisms distinct from those engaged in cyst development of other epithelial cell types [6]. That finding suggests that their development might involve cell operations that are somewhat different from those identified previously [7]. A major goal of this study is to begin elucidating an operating theory about principles that drive AT II cyst growth ex vivo.

Object-oriented, discrete simulation modeling methods like cellular automata [8], cellular Potts models [9], and agent-based models (ABMs) [10] have gained popularity in studies that directly address cell-level mechanics of animal development. One example is a multi-scale ABM that simulates mesoendoderm migration of Xenopus explant ex vivo [11]. Another addresses mesenchymal cell behavior and molecular dynamics of in vitro precartilage cell condensation [12]. Bodenstein et al. [13] developed a multi-cell model to explain a critical event in early chick embryogenesis. Galle et al. [14] reported an ABM simulating epithelial cell colony growth, with an explicit representation of physicochemical determinants of cell survival and proliferation. Another study incorporated a mathematical model of growth factor diffusion and signaling into an established 2D epithelial cell ABM to examine proliferative behaviors under different conditions [15]. Notably, Grant et al. [7] developed a Madin-Darby canine kidney (MDCK) epithelial cell mimetic analogue, which provided useful insight into cell-level mechanisms that account for multiple phenotypes in four different culture conditions.

The object-oriented class of models and methods are what Fisher and Henzinger [16] have referred to as executable biology. They comprise discrete components and mechanisms (in the form of executable algorithms) that map logically to wet-lab counterparts. We refer to such a system as an analogue to help distinguish this class of models from traditional, inductive models. Analogues are executed and measured in the same way as their referent.

Manuscript received April 1, 2009. This work was supported in part by the CDH Research Foundation and a Graduate Fellowship to S. H. J. Kim from the International Foundation for Ethical Research. This paper is extended from "Agent-Based Modeling of Alveolar Type II Cyst Formation In Vitro" published at *Spring Simulation Multiconference 2009, Agent-Directed Simulation Symposium,* San Diego, CA, USA, March, 2009.

S. H. J. Kim (e-mail: seanhjk@berkeley.edu) and S. Sheikh-Bahaei (e-mail: shahabsb@berkeley.edu) are with the UCSF/UC Berkeley Joint Graduate Group in Bioengineering, University of California, Berkeley, CA 94720, USA.

C. A. Hunt is with Department of Bioengineering and Therapeutic Sciences, University of California, San Francisco, CA 94143, USA (e-mail: a.hunt@ucsf.edu).

Data accumulated during executions are compared against data taken from the referent. When an analogue fails validation, we revise it, validate it against its predecessor (cross-model validation) and then against referent attributes. When satisfaction is achieved, a case can be made for concretizable software to wet-lab mappings at both behavioral and mechanistic levels. The methods provide for establishment of plausible reductive hierarchies between lower level mechanisms and higher-level phenomena by growing useful, more detailed software analogues from a predecessor.

The analogues differ from conventional models of inductive type. An inductive model is usually built by first analyzing data, creating a mapping between the envisioned system structure and components of the data, and then representing the generation of those data components with mathematical equations. Inductive analytical models test hypotheses about data. The method relies heavily on the researcher's knowledge combined with induction. When successful, it creates models that can extrapolate beyond the original data, making the method ideal for prediction. The inductive and executable biology methods present different yet complementary approaches to exploring explanations of biological phenomena.

Here, we adapt, refine, and advance current ABM methods, drawing on those articulated by Grant et al. [7], to develop and begin validating analogues of alveolar-like cyst (ALC) growth in 3D cell cultures. The analogues are a multi-agent system composed of quasi-autonomous cell agents and passive objects that represent the extracellular cell culture composition within a software framework supported by other agents [17]. Internal cell state and neighboring object configuration determine simulated cell action. Simple decision logic and cell agent axioms define the preconditions and corresponding cell actions. Following a simple rule set, cell agents self-organize into stable cyst-like structures and mimic key growth characteristics of cultured AT II cells.

2. Biology

Pulmonary alveoli (~250 μm in diameter; human) have an elaborate anatomy consisting of an epithelium, basement membrane, and surrounding capillary network [2]. The alveolar epithelium encloses a hollow air space, called lumen. Three cell types compose the epithelium: type I and II pneumocytes, and macrophages. AT I cells provide thin, flat surface through which oxygen diffusion occurs. AT II cells have multiple functions essential to alveolar homeostasis. The alveolar basement membrane and associated extracellular matrix (ECM) surround and anchor the epithelium to interstitial connective tissue. Densely networked capillaries cover the alveoli to form a blood-air barrier, which prevents blood from entering alveoli while permitting oxygen flow.

AT II cells are considered defenders of the alveolus [3]. Their primary function is to produce and secrete pulmonary surfactant along the inner alveolar lining. Surfactant, composed of lipids and proteins, helps maintain proper surface tension and prevents alveolar collapse during ventilation. It also plays important roles in fluid balance and host defense.

3. Conceptual Model

We first catalogued basic AT II growth attributes in 3D matrix, and then narrowed the list into an initial set of target attributes for model development and validation. Subcellular molecular details were presumed to conflate into cell level events and processes. Doing so allowed us to define attributes that map directly to experimental in vitro observations and measurements. Next we conceptually discretized cultures so that only those components essential in achieving the targeted attributes were represented. Specifically, the in vitro cultures were represented as having five components: cells, clusters, ECM (matrix hereafter), ECM-free region (free or luminal space hereafter), and a space to contain them. Hereafter, we use small caps when describing model components and processes to distinguish them clearly from their in vitro counterparts. CELL represents an individual AT II cell. CLUSTER represents a coherent aggregate of cells. MATRIX represents either culture medium or ECM. FREE SPACE represents regions devoid of both ECM elements and cells. It also represents luminal material and the material in pockets enclosed by cells. The latter are called LUMINAL SPACE when distinction from FREE SPACE is useful. CULTURE represents cell culture as a whole containing the culture components described.

4. Agent-Based Model

We constructed AT II cell-mimetic analogues and supporting modules following ABM and discrete event simulation (DES) design principles. Fig. 1 depicts the system's architecture and component methods. Main system components are EXPERIMENT MANAGER, OBSERVER, CULTURE, CELL, CLUSTER, MATRIX and FREE SPACE. Additional components that extended functionality include CULTURE graphical user interface (GUI) and DIFFUSER.

4.1 Experimentation Agents

4.1.1. Experiment manager

EXPERIMENT MANAGER is the top-level agent. It manages experiment setup, execution, and data processing. From user specified options, EXPERIMENT MANAGER launches a simulation in one of three experimentation modes: default, visual, and batch. An experiment in default mode is simply a single execution. In visual mode, a CULTURE GUI is created and a console launched to allow visualizing and accessing a live simulation. Batch mode enables automatic construction and execution of multiple experiments, as well as processing and analysis of recorded measurements. EXPERIMENT MANAGER generates a set of parameter files from user-defined specifications which delimit the parameter space from which individual CULTURE

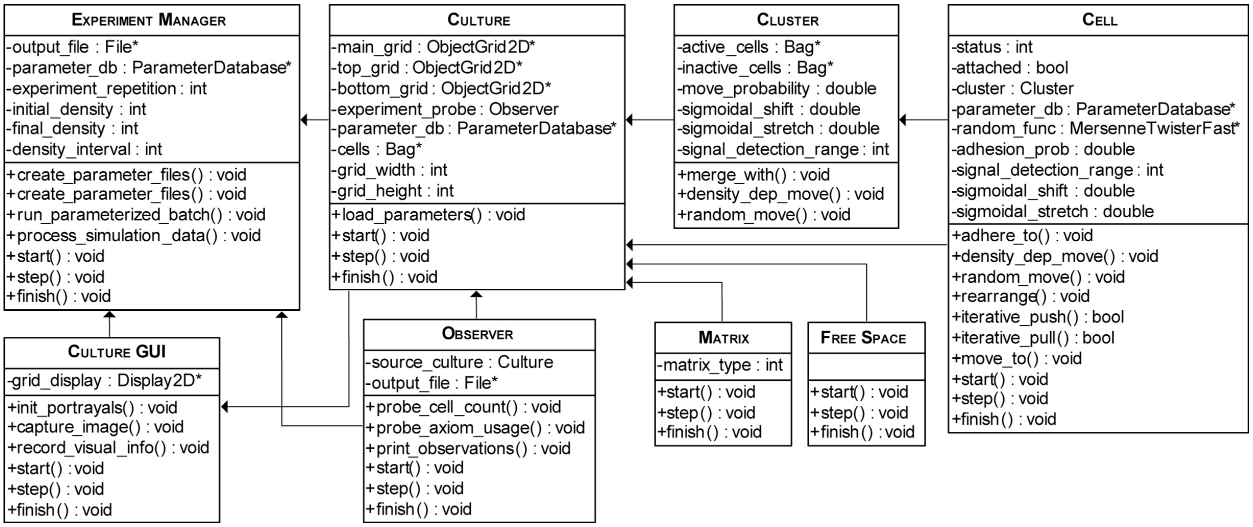

Fig. 1. Uniform Modeling Language diagram of the system's architecture. EXPERIMENT MANAGER is a system-level agent that prepares parameter files, manages experiment execution, and processes output data for analysis and summary. OBSERVER is an agent that tracks individual CELL activities and measures CULTURE attributes such as the number of CLUSTERS that form during simulation and their sizes. CULTURE is the simulation proper and represents an arbitrary section of an in vitro cell culture. It maintains a 2D hexagonal grid in which CELLS, MATRIX, and FREE SPACE objects are instantiated. Those three object types correspond to, respectively, cells, extracellular matrix, and the region (e.g., cyst lumen) devoid of cell and extracellular matrix. MATRIX and FREE SPACE are passive objects. CELLS are active agents with a set of axioms and decision logic that determine their action based on their internal state and neighboring object configurations. CULTURE GUI provides a graphical interface to visualize and probe CULTURE during simulation. The system was implemented using MASON which provides standard libraries, object classes and methods (*).

parameter values are generated. Once parameter files are generated, EXPERIMENT MANAGER automatically executes a batch of experiments, each corresponding to a different parameter file. It instantiates and communicates with OBSERVER to conduct and record measurements during simulation. User specifies the number of repetitions for each experiment. Repetitions are executed sequentially. After completion of all experiments, basic analytic operations are automatically applied to collect and summarize experimental data.

4.1.2. Observer

OBSERVER is primarily responsible for recording CULTURE measurements. OBSERVER creation is a system option selected only when detailed measurements are needed. The agent has access to CULTURE, its grid spaces, and individual CELLS. OBSERVER is stepped and its probe method called at the end of every CULTURE simulation cycle. The probe method scans the CULTURE internals and performs measurements. Recorded measurements include total CELL population, CELL activities in terms of axiom usage, and migration trajectories of individual CELLS. These measures are recorded as time series vectors; the data are written to summary files at simulation's end. Using CULTURE GUI functionalities, OBSERVER captures time-lapse CULTURE images and stores them in multiple formats for post-processing.

4.2 AT II Culture Components

The core AT II model comprises five component types: MATRIX, FREE SPACE, CELL, CLUSTER, and CULTURE. MATRIX and FREE SPACE are passive objects. CELLS are active agents with a set of rules that govern their actions. Cellular processes of interest, such as migration and adhesion, are represented as discrete events. CLUSTERS are simple representations of aggregated, adhered cells, and their actions are governed by their own logic. CULTURE is a simulation proper and represents the in vitro system as a whole. It maintains a 2D hexagonal grid (CULTURE space), which is populated by MATRIX, FREE SPACE, and CELLS. The grid, and hence the CULTURE, represents an observable 2D cross-section of an in vitro culture. CULTURE GUI and DIFFUSER components extend CULTURE functionalities.

4.2.1. Cell agent

CELLS mimic specified behaviors of alveolar type II cells in cultures. They are quasi-autonomous agents that follow their own agenda and schedule their own events. Their behaviors are governed by a decision logic (Algorithm 1) and a set of rules—axioms—that reflect observed in vitro behaviors. The term 'axiom' emphasizes that computer programs are mathematical, formal systems, and the initial mechanistic premises within simulations are analogous to axioms in formal systems. For this study, an axiom is an assumption about what conclusion can be drawn from what precondition for the purposes of further analysis or deduction. Standard logic methods can be applied to reformulate the axioms to enforce constraints or reduce redundancy when needed [18]. Within a simulation, every CELL carries out exactly one action during each simulation cycle. CELL action is dictated by internal CELL state and neighboring object composition. To achieve the initial set of target attributes, we defined what we judged to be a minimal set of actions: attach to an adjacent CELL, migrate, and rearrange within a CLUSTER. Every CELL maintains a state variable indicating whether it is a member of a CLUSTER.

Algorithm 1: CELL agent decision logic

```
IF this CELL is active on CULTURE grid THEN
   N = {neighboring objects}
   IF N has CELL object type THEN
      FOR each CELL neighbor C in N
         p = UniformDistribution(0,1)
         IF p < threshold THEN
            Attach to C
         END IF
      END FOR
   END IF
   IF not attached to any CELL neighbor THEN
      Migrate
   ELSE
      Rearrange
   END IF
END IF
```

When stepped, a CELL first determines its current neighboring object composition and arrangement. If it has no CELL neighbors, it classifies itself as single and migrates to a neighboring non-CELL location. A CELL can migrate randomly, CHEMOTACTICALLY, or along a CELL density gradient as specified in simulation control. Random CELL migration implements a simple, unbiased random walk. In CHEMOTACTIC mode, a CELL is capable of sensing CELL-produced ATTRACTANT concentration in its immediate local environment and moving towards the most concentrated region. The CELL density-based mechanism enables CELLS to determine local CELL density within a defined radius and actively move along the density gradient. The mechanism does not yet map to a specific, known biological mechanism: it is an abstract placeholder for whatever non-chemotactic mechanisms (e.g., ones based on ECM remodeling or long-range intercellular connection) enable AT II cells to sense other cells in their surroundings, obtain directional cues, and migrate based on that information.

Each migration mode is encoded as a separate object method. Every CELL maintains a state variable specifying its current migration mode. CELLS also have a parameter that can be used to introduce random movements while CELLS are in a directional migration mode. The parameter specifies the probability of electing to move randomly when the CELL is in directional migration mode.

CELL migration speed is specified parametrically: a parameter specifies the average CELL speed in grid units per simulation cycle. A parameter value = 1 results in an average speed of one grid unit per simulation cycle. Setting the parameter to zero abolishes migration. Non-integer parameter values (e.g., 0.5 or 2.8) require approximation. In the current implementation, non-integer speeds are resolved as follows. A CELL specified to migrate 0.5 units per cycle has a 50 percent chance of moving one unit at each cycle. Consequently, its instantaneous speed is either zero or one unit per cycle. However, on average, the CELL moves 0.5 units per cycle. The same method is used to accommodate non-integer CLUSTER migration speeds.

When one or more CELL neighbors are in contact, the decision-making CELL attaches to each neighbor with a parametrically controlled probability, *p*, to form a CLUSTER. We used a pseudo-random number generator (PNG) to draw *p* values from a uniform distribution, *U(0,1)*. We defined two probability thresholds for CELL-CELL attachment. One controlled the attachment probability of single, non-clustered CELLS; the other controlled attachment probability of clustered CELLS.

Following attachment, CELLS switch to clustered state. Clustered CELLS follow the axioms in Table 1 to rearrange their relative positions within the parent CLUSTER, a process that is essential for ALC formation. Axioms define

Table 1: CELL agent axioms for inter-CLUSTER rearrangement

Axiom	Precondition			Action
	CELL	MATRIX	FREE SPACE	
1a	≥ 1	0	0	Push out a CELL neighbor; move to the vacated location; leave behind a FREE SPACE object
1b	0	≥ 1	0	Do nothing
1c	0	0	≥ 1	Move to an adjacent FREE SPACE; leave behind a FREE SPACE object
2a	1 or 2*	≥ 1	0	Move to an adjacent MATRIX next to a CELL neighbor; leave behind a MATRIX object
2b	2**	≥ 1	0	Move to an adjacent MATRIX next to a CELL neighbor; pull the other CELL neighbor into the vacated location
2c	3	≥ 1	0	Do nothing
2d	4	≥ 1	0	Move to an adjacent MATRIX; leave behind a MATRIX object
2e	5	≥ 1	0	Move to an adjacent MATRIX; leave behind a FREE SPACE object
2f	≥ 1	0	≥ 1	Push out a CELL neighbor; move to the vacated location; leave behind a FREE SPACE object
2g	0	≥ 1	≥ 1	Do nothing
3a	≥ 1	≥ 1	1	Move to the adjacent MATRIX; leave behind a FREE SPACE object
3b	1†	≥ 1	≥ 1	Move to an adjacent FREE SPACE; pull the CELL neighbor into the vacated location
3c‡	≥ 1	≥ 1	≥ 1	Do nothing

*adjacent; **non-adjacent; †adjacent to a MATRIX neighbor; ‡for all other configurations of three object types preconditions and corresponding CELL actions. Preconditions are based on neighboring object types and their specific configurations. Rearrangement actions are based on experimental insights from the literature and expert opinion. For cases that lack relevant biological information, we rely on basic assumptions about epithelial cell behavior. One assumption is that cells act to establish and maintain a preferred local environment. Another is their mandate to achieve three surface types: apical (lumen), basal (matrix), and lateral (cell contact) [6].

4.2.2. Cluster agent

CLUSTER is an agent and a composite object containing member CELLS and LUMINAL SPACE objects. It represents coherent multi-CELL structures and simulates simple features of their collective behavior. A CLUSTER is created when two or more CELLS attach. Single CELLS that establish attachments to a CLUSTER's member CELLS are added to the CLUSTER. Individual CLUSTERS that are adjacent and detected by member CELLS, can combine to form a larger aggregate.

A CLUSTER schedules its own events, which run at the same frequency of CELL events. CLUSTER events are managed by the master event schedule; their execution ordering is pseudo-random and independent of CELL events. A CLUSTER deactivates and withdraws from simulation when its membership diminishes to one; the remaining CELL reverts to single CELL status.

Each CLUSTER uses an identical step function to determine its action. The step function is scheduled every simulation cycle. A CLUSTER can either migrate a certain distance or do nothing. In vitro, multi-cell aggregates tend to migrate as a coherent body. That collective movement is driven by inherent, coordinated, individual cell actions, but to simplify implementation we directed CLUSTER to handle collective migration. When stepped, a CLUSTER moves with some probability if its overall shape has been altered by the rearranging CELL actions. Migration speed and the probability of movement are specified parametrically. Similar to a CELL, a CLUSTER can adopt one of three migration modes: random, CHEMOTACTIC, and CELL density-based. For simplicity, it adopts the majority migration mode of its member CELLS. To further simplify design, CLUSTER movement maintains overall shape and relative positions of the composing CELLS. CLUSTER movement stops when the movement is blocked by non-member CELLS or other CLUSTERS. For object interaction, each CLUSTER is provided with standard class methods to query and access member CELLS and associated LUMINAL SPACE. Because CLUSTERS do not exist on a separate grid, they do not have or need their own positioning information.

4.2.3. Culture

A CULTURE maps to an arbitrary culture section within one well of a multi-well culture plate. It has a master event schedule, PNGs, and its own start and end methods that are called automatically at simulation's start and end.

The start method initializes CULTURE content according to specifications provided in a parameter file. The CULTURE uses a standard 2D hexagonal grid to provide the space in which its objects reside. The grid has toroidal topologies. Following grid initialization, CULTURE components—CELLS, MATRIX, and FREE SPACE—are placed on the grid and an initial scheduling of CELLS is created on the master event schedule. For simplicity, each grid position is occupied by one object. That condition can be easily changed when the need arises.

Simulation starts following completion of the start method call. As execution advances, the event schedule is stepped for a number of simulation cycles or until a stop signal is produced. Simulation time advances in discrete steps. Its course unit is simulation cycle, during which everything in the simulation has one opportunity to update. Ordering of events to occur within the same simulation cycle is pseudo-random. Having objects update pseudo-randomly simulates the parallel operation of cells in culture. It also makes execution outcomes nondeterministic while building in a degree of uncertainty, a fundamental characteristic of living systems.

At the end of simulation, the finish method is executed to close data files and clear the system. Visualization and user interaction are provided by a CULTURE GUI. It extends CULTURE with display and controller methods, which enable the user to start a simulation or pause and access live states of CULTURE grid content during simulation. CULTURE GUI also supports automatic recording of sequential images in multiple formats for post-simulation image processing.

4.2.4. Diffuser

A DIFFUSER is a CULTURE extension for simulating dispersion of extracellular substances. A DIFFUSER object is created only when CHEMOTACTIC migration mode is enabled. It contains a grid and a step function to compute diffusion. The same hexagonal 2D grid type is used and aligned with the CULTURE grid; however, the grid contains only numerical values. The CULTURE start function initializes the DIFFUSER with the specified initial ATTRACTANT levels. The DIFFUSER object is stepped and its diffusion algorithm is executed a parameter-specified number of times within each simulation cycle. The diffusion algorithm provides a simple discrete approximation of diffusion in a continuous space using parametrically defined diffusion and loss rates:

$$A_i(t+1) = (1-e)[A_i(t) + d(\tilde{A}_i(t) - A_i(t))] \tag{1}$$

where d and e are the diffusion and loss rates, t is the diffuser step counter, $A_i(t)$ is the ATTRACTANT level at grid position i, and $\tilde{A}_i(t)$ is the average level across grid position i and its six neighboring locations. Maximum rates = 1. ATTRACTANT levels are capped at a maximum listed in CULTURE specifications.

Table 2: Key model parameters and respective values

Parameter	Value(s)
Initial CELL population	100~6,000
Maximum CELL push/pull iteration	5
CELL speed in grid units	1
Single CELL attachment probability	0.2
Clustered CELL attachment probability	0.01
Local CELL density radius	5
Diffusion step multiples*	25
Diffusion rate	0.4
Evaporation rate	0.05
Maximum SOLUTE level	50,000
SOLUTE production per simulation cycle	3,000~8,000

* Diffusion algorithm iterations per simulation cycle

5. EXPERIMENTS AND RESULTS

We conducted exploratory simulation experiments to study ALC growth at varying CELL densities. We varied the initial CELL population from 100 to 6,000 in increments of 100. CULTURE grid height and width were set to 100 units each. All three CELL migration modes were used at each CELL density. We executed 100 Monte Carlo (MC) simulations per CELL density and migration mode. Each MC run lasted 100 simulation cycles corresponding to ~147 hours in vitro. Model parameters used are provided in Table 2. To simulate in vitro embedded culture, CULTURE grid was initialized and filled with MATRIX objects. Next, CELLS were instantiated and distributed randomly on the grid, replacing existing MATRIX objects as needed. We used PNG seeds, set to system time, to randomize simulation events. CELLS and CLUSTERS were selected at random (one at a time without replacement) and carried through their actions each simulation cycle.

Fig. 2 summarizes simulation outcomes in comparison with referent measures. CELLS self-organized into multi-CELL structures that resemble ALCs (Fig. 2A, B) by adhering strictly to their axiomatic principles of operation. CELL migration and aggregation enabled formation of multiple CLUSTERS. CLUSTERS used CELL rearrangement to develop into LUMINAL SPACE-enclosing CYSTS. LUMINAL SPACES were devoid of CELLS and MATRIX, and were enclosed by a continuous CELL monolayer. Generally, ALCS maintained convex polygonal shapes, which map to a roundish shape in continuous 2D cross-sections.

ALC sizes increased monotonically with initial CELL densities (Fig. 2C). In vitro, overall changes represent more than a twofold increase in ALC diameter; changes in silico varied depending on CELL migration mode. At the lowest CELL density corresponding to 1 x 10^4 cells/cm^2, all three CELL migration modes produced small ALCS, mostly in the 20-to-40 µm diameter range, which was comparable to in vitro measurements (1 grid unit = 8.5 µm in vitro). CELLS migrating randomly failed to mimic the relatively steep rise in ALC diameter following an increase in the initial cell density. In contrast with random migration, CHEMOTAXIS and CELL density-based migration produced ALCS that were similar in size to their in vitro counterparts. Directionally migrating CELLS tended to aggregate and form multi-CELL structures earlier, compared to randomly migrating CELLS. The difference was most evident in sparsely populated CULTURES. Both directional migration mechanisms gave rise to convergent CELL movement that closely mimicked in vitro aggregation patterns (not shown). Of the two, the CELL density-based migration led to simulation outcomes closest to the in vitro measures.

Fig. 3 shows clustered CELL axiom usage in CELL density-based migration mode. Essentially same use patterns were observed in random and CHEMOTACTIC modes. As shown, Axioms 2a and 3c were used most frequently, followed by Axioms 2b and 2c. CELLS executing Axiom 2a, 2b, and 2c had only CELL and MATRIX neighbors; they

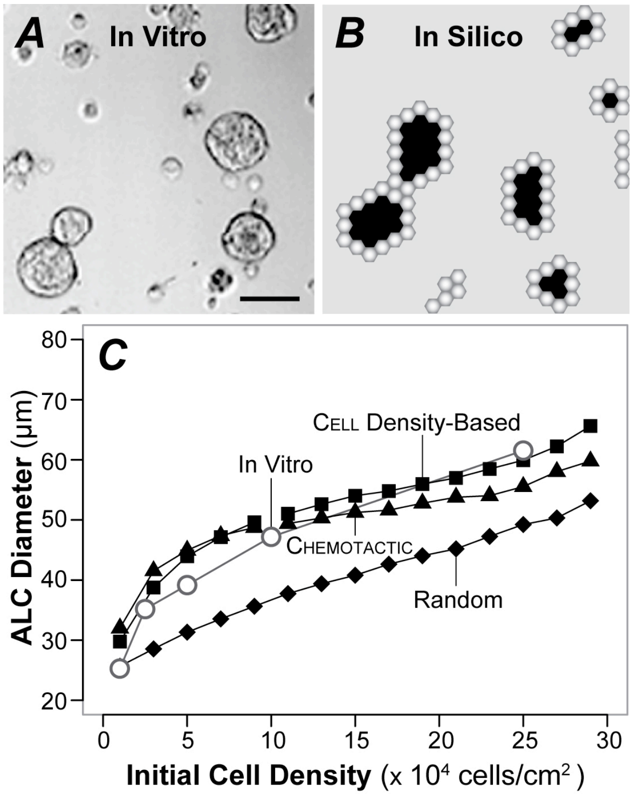

Fig. 2. Alveolar-like cyst (ALC) growth in simulated and in vitro cell cultures. (A) Phase-contrast image after 4 days in 2% Matrigel [5]; courtesy of W. Yu and K. Mostov. Cells were plated initially at 10 x 10^4 cells/cm^2. ALCs had roundish shapes without obvious depressions or dimples. Bar, ~ 50 µm. (B) CULTURE images after 100 simulation cycles; the initial CELL population was set to 2,000 CELLS. Like in vitro, the ALCS maintained convexity and had no dimples. Gray and black spaces represent MATRIX and FREE (or LUMINAL) SPACE, respectively. (C) Both in vitro and in silico, ALC diameter increased monotonically with the initial cell density. Circles: in vitro measurements (mean). We tested three different CELL migration mechanisms: random migration (diamonds), CHEMOTAXIS (triangles), and CELL density-based migration (squares). In the first, CELLS migrated randomly without directional bias or persistence. In the second, CELLS migrated along a diffusive CHEMOATTRACTANT gradient. In the third, CELLS migrated along a local CELL density gradient. In all three scenarios, CELLS developed normal ALCS. CELLS migrating along a CELL density gradient achieved outcomes most similar to the in vitro measures. Mean outcome of 100 Monte Carlo runs each lasting 100 simulation cycles.

were members of developing clusters that did not yet have a luminal space. Cells in maturing ALCs executed Axiom 3c, so not surprisingly that axiom was executed most frequently as the simulation progressed.

Usage patterns changed dynamically over time reflective of the changes in a cell's extracellular composition and arrangement. Although relative axiom use patterns were qualitatively similar for all initial cell densities, the specific details were both simulation cycle and initial cell density dependent. Cells in sparsely populated cultures exhibited more frequent and extended use of Axioms 2b and 2c, whereas in densely populated cultures, use frequencies of Axioms 2e, 2f, 3a, and 3b increased several fold.

The remaining axioms exhibited infrequent use, but that did not mean that they were unimportant. In fact, several infrequently used axioms were critical to the formation of morphologically normal ALCs. Blocking execution of each of these axioms disrupted normal ALC growth (not shown). For example, blockage of Axiom 1c or 2f led to frequent appearance of isolated clumps of cells within ALC lumens. When executions of Axioms 1a and 2e were blocked, cell aggregates failed to develop a luminal space and no ALCs formed. Without Axiom 3a, the cells formed ALCs having irregular, nonconvex shapes.

6. Discussion

Epithelial cyst formation has been studied extensively using various cell lines under 3D culture conditions [6]. When cultured in 3D, those cell types develop spherical, lumen-enclosing cysts similar to those observed in AT II cultures. Unlike AT II cultures, proliferation and apoptosis (cell death) play dominant roles in epithelial cyst development. Interestingly, MDCK cells can develop cysts by mechanisms similar to those of ALC growth under certain conditions, indicating that one of multiple mechanisms is in use depending on circumstances. Comparison of cell decision logic between AT II and MDCK models [7] reflects the observed differences in cyst growth mechanisms. Notwithstanding those differences, morphological similarities suggest that common principles of cell operation may underpin epithelial cyst development. For example, both models suggest a common theme whereby every cell strives to maximize the number of adhesive contacts along its lateral surface. Cells arranged in a cyst structure satisfy such mandates. We expect successful validation of

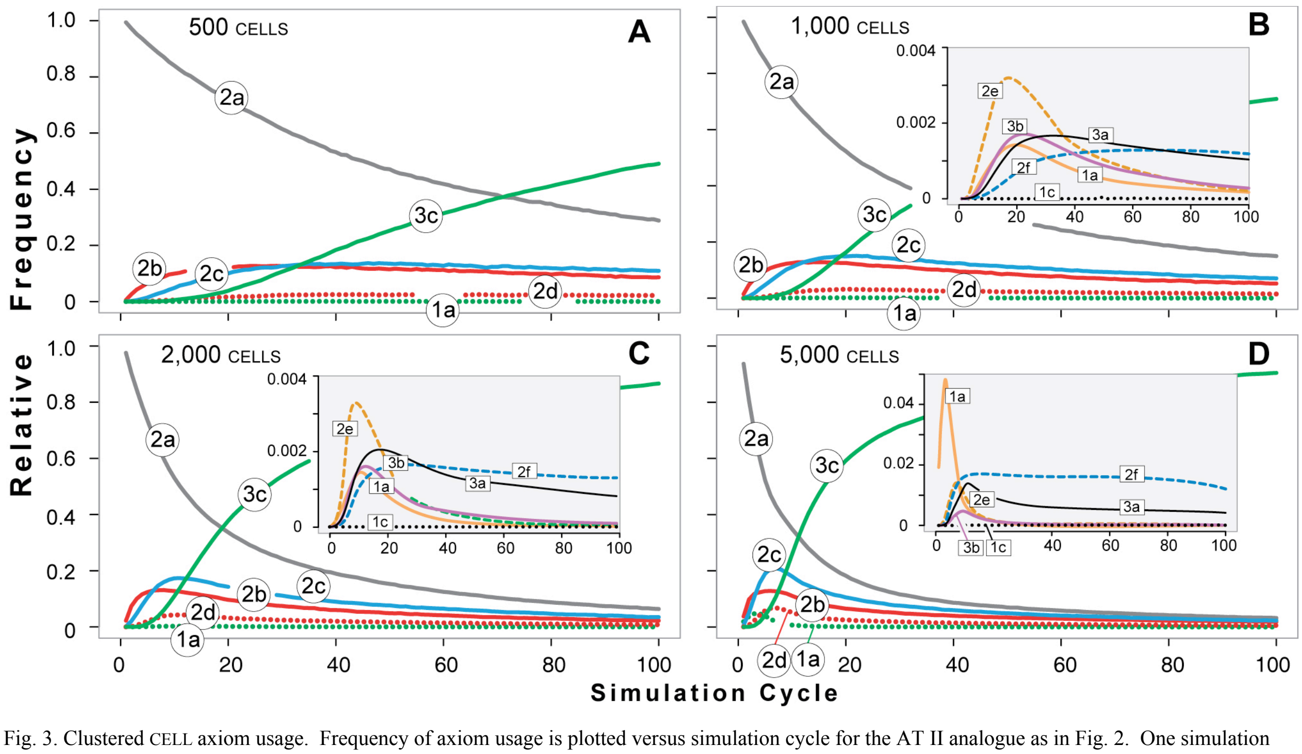

Fig. 3. Clustered cell axiom usage. Frequency of axiom usage is plotted versus simulation cycle for the AT II analogue as in Fig. 2. One simulation cycle maps to ~ 1.47 hours in vitro. Relative axiom use depended on initial cell densities, which are listed in each panel. The variance in use frequency across simulation cycles for the less frequently used axioms was large. In the inserts, trend lines were used to make patterns more evident. Early in simulations, Axioms 2a, 2b, and 2c were used most frequently as cells rearranged themselves and condensed into packed clusters. Axioms 1a and 2e were executed most often early in ALC development to provide for luminal space creation. As simulations progressed and ALCs matured, Axiom 3c (do nothing) was executed more frequently: stable structures were forming and for most cells, no further rearrangement was needed. (A–B) At low-to-moderate cell densities, Axioms 2b and 2c also applied often when cell clusters were unable to grow further and develop into ALCs. The remaining axioms showed only limited usage (insets), yet they were essential in achieving targeted attributes. For example, Axiom 1c was essential in enabling cells trapped within the luminal space to merge with its parent cluster. (C–D) In densely populated cultures, usage of Axioms 2e, 2f, 3a, and 3b increased several fold (insets), especially early in simulation. ALCs developed and matured sooner in dense cultures as indicated by the earlier increases in Axiom 3c usage.

extensions of 2D behaviors into three dimensions justifies the hypothesis of an analogous cell behavior in a 3D culture environment.

Cell migration is the principal process by which AT II cells develop multi-cell clusters, and that is also the case for simulations. However, it is largely unknown what migration mechanism(s) are involved in vitro and how changes in migration patterns affect ALC development. To explore those issues we challenged our model using different cell migration methods and compared outcomes. First was random CELL motility implemented as a simple, random walk. That represents the simplest mechanism; it required no assumptions about molecular or physical cues that might guide AT II cell migration. Our simulation results suggest that random motility alone is not an effective means for cell aggregation.

In addition, we explored two directional migration modes, one driven by CELL density gradients and the other by CHEMOTAXIS, and found that both are adequate drivers of CLUSTER formation and ALC growth. The results show that of the two, CHEMOTAXIS is somewhat less effective particularly in densely populated CULTURES, in which CELLS lost directional persistence due to rapid fluctuations in local ATTRACTANT concentrations. In contrast, the CELL density-based mechanism enabled CELLS to maintain persistent directionality in densely populated conditions. The observed differences could be attributed to the analogue's chosen spatial discretization and implementation details, or unknown artifacts. In vitro and in vivo chemotaxis also could involve local gradients of multiple chemotactic agents, which additionally might modulate cell processes like cellular adhesion. In addition, the current model precludes simultaneous use of multiple migration mechanisms, yet it is possible that AT II cells might employ multiple mechanisms and switch between those to direct and optimize their movement. Experimental findings of related cell types like MCF10A indicate that they can adopt counter modes of migration, random versus directionally persistent, which are regulated by a mechanism dependent on Rac1 protein activity [19]. Such issues can be probed further in a future study as more wet-lab information becomes available.

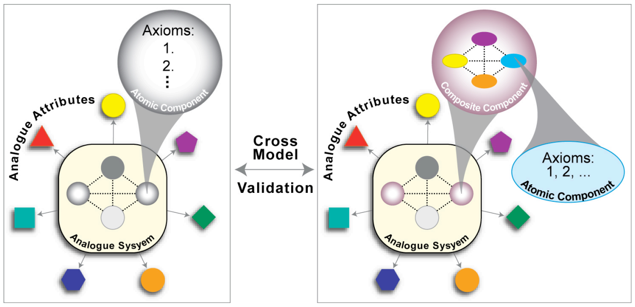

Fig. 4. Model refinement and cross-model validation. CELLS are the main actors of the current AT II analogue. They are atomic components whose actions are governed by axioms. CELL actions and interactions cause phenotypic attributes. Because CELLS and other components are discrete objects, they can be replaced easily with new or revised objects. Replacements can be composite objects composed of other objects. Details at different scales can be incorporated into the analogue in like manner. When so doing, one must provide appropriate similarity metrics and demonstrate cross-model validation (original versus revised) to insure that the change does not alter analogue phenotype measurably.

When and how does an AT cell choose to switch from one activity to another? Why does it choose one action rather than another? Are several action options always available to each cell? The class of models presented herein provides a platform to hypothesize, challenge, and refine plausible answers. The causal chain of events responsible for most simulation events can be explored in detail, and assessments can be made as to whether critical events are biotic (supportable by in vitro evidence) or not.

Thus far, we have focused on basic AT II cell attributes in one specific culture type. When the conditions change, AT II cells develop morphologies and attributes that are different from those observed in the 3D cell culture. Inclusion of those attributes automatically invalidates the current analogue. An advantage of the approach used is the ease with which additional attributes and details can be added during the iterative model refinement protocol described below. Having achieved the initial attributes, we can proceed to elaborate the model to include actual subcellular details that will map to cell locomotion pathways and intercellular signaling networks. From an engineering perspective, doing so is a relatively easy task. It can be achieved simply by swapping a current component (e.g., atomic CELL agent) for a more detailed composite agent (Fig. 4). The replacement could also occur at the intracomponent level, for example by replacing CELL axioms with more detailed logic based on interacting components. A challenging task will be to insure cross-model validation between different analogue variants.

For further development, we suggest following the iterative refinement protocol used successfully herein and in previous studies [20-22]. The protocol supports adhering to the guideline of parsimony that is important when building a complex model. The protocol steps are:

1) Start with a small but diverse set of referent attributes, static and dynamic. They comprise the initial targeted attribute list.

2) Posit coarse-grained discretized mechanisms, requiring as few components as is reasonable that may generate analogous phenomena. Make components that map to living counterparts quasi-autonomous.

3) Instantiate analogue components and mechanisms in software capable of conducting automated experiments.

4) Conduct experiments that measure a variety of phenomena generated during execution. So doing establishes the degree of in silico–in vitro phenotype overlap, and lack thereof.

5) Achieve a degree of validation by satisfying a prespecified level of similarity between in silico and targeted in vitro attributes.

6) Add one or more new attributes (measurable phenomena) to the targeted list until the analogue in step 5 is falsified. Added attributes need to be at a similar level to and close to those already present so that it seems feasible

to achieve the expanded target attribute with as little re-engineering of components as possible. Once the analogue in step 5 is falsified, return to step 2.

The protocol articulates a process for developing, testing, and refining mechanistic explanations or hypotheses about biological phenomena. It is straightforward and so can be used for evolving any mechanistically focused, agent-based, biomimetic analogue of the executable biology type [16].

Finally, the nature and organization of software components within the AT II analogue framework were designed to facilitate iterative refinement. As the process continues, following each round of validation, more of what we know or think we know becomes instantiated in the analogue. After many such rounds, the analogue will begin transforming into an executable knowledge embodiment. To achieve that vision, it is essential that biomimetic components function (quasi-) autonomously, all or part of the time. That is why CELLS and CLUSTERS are agents. Everything that a CELL needs to function (in a specified software environment) is contained within its code. Absent that property, the mappings from software mechanisms to AT II in vitro mechanisms are not concretizable.

7. Conclusions

The AT II analogue establishes a foundation for future studies aimed at articulating the generative principles of lung alveolar development. Guided by additional data, the model can easily accommodate new or more detailed system characteristics. We anticipate that in silico experimentation using this class of agent-based model will provide a fruitful new strategy to dissect the operational basis of morphogenesis.

8. Implementation Tools

The model framework was implemented using MASON v11. MASON [23] is a discrete event, multi-agent simulation library coded in Java. We used R 2.5.1 [24] for data analysis and generation of summary figures. Batch simulation experiments were performed on a small-scale Beowulf cluster system consisting of one master node and seven client nodes, each equipped with a single Intel Pentium 4 3.0-GHz CPU and a 2-GB SDRAM memory. For model development, testing, analysis, and simulation image processing, we used personal computers. Computer codes and project files are available from the authors upon request.

Acknowledgement

We would like to thank Wei Yu and Keith Mostov for generously providing us with in vitro data, and members of the UCSF BioSystems and Mostov groups for helpful suggestions and discussions. We especially thank Mark Grant and Sunwoo Park for their contributions in designing and implementing the model and the simulation framework.

References

[1] Cardoso, W.V. 2001. “Molecular Regulation of Lung Development.” Annu Rev Physiol 63: 471-494.

[2] Crystal, R.G.; J.B. West; P.J. Barnes; W.R. Weibel. 1997. The Lung: Scientific Foundations. Lippincott Williams & Wilkins, New York, USA.

[3] Fehrenbach, H. 2001. “Alveolar Epithelial Type II Cell: Defender of the Alveolus Revisited.” Respir Res 2: 33-46.

[4] Shami, S.G.; J.D. Aghajanian; R.L. Sanders. 1984. “Organotypic Culture of Fetal Lung Type II Alveolar Epithelial Cells: Applications to Pulmonary Toxicology.” Environ Health Perspect 56: 87-94.

[5] Yu, W.; X. Fang; A. Ewald; K. Wong; C.A. Hunt; Z. Werb; M.A. Matthay; K. Mostov. 2007. “Formation of Cysts by Alveolar Type II Cells in Three-dimensional Culture Reveals a Novel Mechanism for Epithelial Morphogenesis.” Mol Biol Cell 18(5): 1693-1700.

[6] O’Brien, L.E.; M.M. Zegers; K.E. Mostov. 2002. “Building Epithelial Architecture: Insights from Three-Dimensional Culture Models.” Nat Rev Mol Cell Biol 3: 531-537.

[7] Grant, M.R.; K.E. Mostov; T.D. Tlsty; C.A. Hunt. 2006. “Simulating Properties of In Vitro Epithelial Cell Morphogenesis.” PLoS Comput Biol 2(10): e129.

[8] Deutsch, A.; S. Dormann. 2005. Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis. Birkhäuser, Boston, USA.

[9] Merksa, R.M.; J.A. Glazier. 2005. “A Cell-Centered Approach to Developmental Biology.” Physica A 352(1): 113-130.

[10] Thorne, B.C.; A.M. Bailey; S.M. Peirce. 2007. “Combining Experiments with Multi-Cell Agent-Based Modeling to Study Biological Tissue Patterning.” Brief Bioinform 8(4): 245-257.

[11] Robertson, S.H.; C.K. Smith; A.L. Langhans; S.E. McLinden; M.A. Oberhardt, et al. 2007. “Multiscale Computational Analysis of Xenopus laevis Morphogenesis Reveals Key Insights of Systems-Level Behavior.” BMC Syst Biol 1: 46.

[12] Christley, S.; M.S. Alber; S.A. Newman. 2007. “Patterns of Mesenchymal Condensation in a Multiscale, Discrete Stochastic Model.” PLoS Comput Biol 3(4): e76.

[13] Bodenstein, L.; C.D. Stern. 2005. “Formation of the Chick Primitive Streak as Studied in Computer Simulations.” J Theor Biol 233(2): 253-269.

[14] Galle, J.; M. Hoffmann; G. Aust. 2009. “From Single Cells to Tissue Architecture-A Bottom-up Approach to Modelling the Spatio-temporal Organisation of Complex Multi-Cellular Systems.” J Math Biol 58: 261-283.

[15] Walker, D.; S. Wood; J. Southgate; M. Holcombe; R. Smallwood. 2006. “An Integrated Agent-Mathematical Model of the Effect of Intercellular Signalling via the Epidermal Growth Factor Receptor on Cell Proliferation.” J Theor Biol 242: 774-789.

[16] Fisher, J.; T.A. Henzinger. 2007. “Executable Cell Biology.” Nat Biotechnol 25: 1239-1249.

[17] Kim, S.H.J.; W. Yu; K. Mostov; M.A. Matthay; C.A. Hunt. 2009. “A Computational Approach to Understand In Vitro Alveolar Morphogenesis.” PLoS ONE 4(3): e4819.

[18] Hogger, C.J. 1990. Essentials of Logic Programming. Oxford University Press, New York, USA.

[19] Pankov, R.; Y. Endo; S. Even-Ram; M. Araki; K. Clark; E. Cukierman; K. Matsumoto; K.M. Yamada. 2005. “A Rac Switch Regulates Random versus Directionally Persistent Cell Migration.” Cell Biol 170: 793-802.

[20] Tang, J.; K.F. Ley; C.A. Hunt. 2007. “Dynamics of In Silico Leukocyte Rolling, Activation, and Adhesion.” BMC Syst Biol 1: 14.

[21] Engelberg, J.A.; G.E. Ropella; C.A. Hunt. 2008. “Essential Operating Principles for Tumor Spheroid Growth.” BMC Syst Biol 2: 110.

[22] Lam, T.N.; C.A. Hunt. 2009. “Discovering Plausible Mechanistic Details of Hepatic Drug Interactions.” Drug Metab Dispos 37: 237-246.

[23] http://cs.gmu.edu/~eclab/projects/mason/.

[24] http://www.r-project.org/.

Sean H. J. Kim received B.A. degree in Computer Science and Mathematics in 2001 from University of California, Berkeley, USA. He is currently a doctoral student in the UCSF/UC Berkeley Joint Graduate Group in Bioengineering at University of California, San Francisco and Berkeley. His research interests include multi-agent, discrete event-oriented modeling and simulation, systems biology, theoretical formalisms for biology, and computational methods in drug development. His technical knowledge draws from previous software engineering experience at Cadence Design Systems and Avanti Corporation in the electronic design automation industry.

Shahab Sheikh-Bahaei received B.S. degree in Electrical Engineering in Dec. 1999 from Isfahan University of Technology, Isfahan, Iran; and M.S. degree in Control Engineering in Dec. 2003 from University of New Mexico, Albuquerque, New Mexico. He is currently a doctoral candidate in the UCSF/UC-Berkeley Joint Graduate Group in Bioengineering. His areas of interest include modeling and simulation, multi-agent systems, self-organizing systems, complex systems modeling and artificial intelligence.

C. Anthony Hunt received B.S. degrees in Chemistry and Applied Biology from the Georgia Institute of Technology and the Ph.D. degree from the University of Florida. His research transitioned from wet-lab to computational in 1998. He directs the BioSystems Group. The group's research focuses on developing and using advanced modeling and simulation methods to help unravel the complexities of biological systems used in biomedical and pharmaceutical research. Dr. Hunt is a member of several scientific and engineering societies, including ACM, IEEE Engineering in Medicine and Biology Society, Society for Computer Simulation International, and Biomedical Engineering Society. He is a member and fellow of the American Association for the Advancement of Science and the American Association of Pharmaceutical Scientists.