Building multi-scale models with PhysiBoSS, an agent-based modeling tool

Marco Ruscone[1,2,3,✉], **Andrea Checcoli**[4,5], **Randy Heiland**[6], **Emmanuel Barillot**[1,2,3], **Paul Macklin**[6], **Laurence Calzone**[1,2,3], and **Vincent Noël**[1,2,3,✉]

[1] Institut Curie, Université PSL, F-75005, Paris, France
[2] INSERM, U900, F-75005, Paris, France
[3] Mines ParisTech, Université PSL, F-75005, Paris, France
[4] Centre de Recherche des Cordeliers, Equipe labellisé par la Ligue contre le cancer, Université de Paris Cité, Sorbonne Université, Inserm U1138, Institut Universitaire de France, Paris, France
[5] Metabolomics and Cell Biology Platforms, Institut Gustave Roussy, Villejuif, France
[6] Department of Intelligent Systems Engineering, Indiana University, Bloomington, IN, USA

Multi-scale models provide a unique tool to study complex processes considering events that occur at different scales across space and time. In the context of biological systems, such models can represent mechanisms which happen both at the intracellular level, and at the extracellular scale, where cells communicate and coordinate with other cells. They aim at understanding the impact of deregulations that are observed in complex diseases, describing the interplay between a pathological tissue and the immune system, and suggest strategies to revert the diseased phenotypes. The construction of these multi-scale models remains a very complex task. A large part of the complexity is due to the difficulty to program the model in languages such as C++ or Python, which may require expert knowledge. Simplifying this process through the use of structured description formalisms coupled with graphical interfaces is a crucial step in making these models more accessible to the broader scientific community. This tutorial introduces four examples of multi-scale models, which rely on the multi-scale framework PhysiBoSS, an add-on of PhysiCell that include intracellular description as continuous time Boolean models to the agent-based approach. The tutorial demonstrates how to easily construct such models, primarily relying on XML model definition files generated by PhysiCell Studio, a Graphical User Interface designed for PhysiCell.

Multi-scale modeling | Agent-based modeling | Boolean modeling

Correspondence: ***vincent.noel@curie.fr, marco.ruscone@curie.fr***

Introduction

Multi-scale modeling is an important tool in understanding complex biological systems, as it allows for the study of events occurring at various scales, both spatial and temporal. Such models are particularly useful in investigating the interplay between intracellular level mechanisms, and intercellular interactions where cells communicate and coordinate. This is especially pertinent in the context of cancer, where multi-scale models can be useful when studying the cross-talks between the microenvironment components, offering insights into the mechanisms of disease progression and potential therapeutic strategies. In this context, hybrid agent-based models result in a broader representation of biological systems, as they blend discrete agent-based techniques with continuous mathematical models. This approach allows for a detailed depiction of individual cell behaviors while simultaneously capturing the broader, continuous dynamics of the biological environment. Such models are instrumental in accurately simulating the intricate interactions within cancerous tissues, shedding light on the complex interplay of cellular and molecular factors. However, developing these models can be challenging, often requiring proficiency in programming languages like C++ or Python, which might not be accessible to all researchers. An important advancement in this field is the introduction of PhysiBoSS(1, 2). This add-on to PhysiCell(3) enhances the modeling process by integrating intracellular descriptions into the agent-based approach. PhysiBoSS utilizes MaBoSS(4, 5), a tool that models signaling pathways as Boolean networks, thus simplifying the description of intracellular models. In addition to this, PhysiCell-Studio(6), a graphical interface compatible with both PhysiCell and PhysiBoSS, further streamlines model development, catering to users with varying programming expertise. Despite these advancements, building a model can still result complex for non-computational researchers that approach the softwares for the first time. However, we have streamlined the process significantly, making it more accessible and user-friendly. The models presented in this paper showcase a range of complexities and features, each highlighting a different aspect of multi-scale modeling challenges and their solutions. In the format of a detailed tutorial, this paper guides readers through the practical implementation of these models, demonstrating their utility in cancer research.

Methods

Agent-based modelling with PhysiCell. Agent-based modeling is a computational approach that aims at using autonomous, interacting agents to study the behaviors of a system. The agent represents a single individual with a set of behaviors that can react to other agents or to the surrounding environment. Agent-based models allow for the study of the emergence of complex population events from a simple set of agent's behaviors. In the field of medical science, an agent can represent a cell that can interact with other cells or its microenvironment. With this approach it is possible to represent different biological scenarios, study collective cellular behaviors and testing hypotheses in-silico. At present, many agent-based frameworks are available, with different characteristics to better answer different modeling needs (7). In this context, the C++ software PhysiCell provides a center-based approach, simulating mechanical and phenotypical cell dynamics, as well as the diffusion of substrates to represent cellular respiration, paracrine communication and more. PhysiCell enables the customization of the simulations through a general configuration XML file. Moreover, it proposes a dictionary of signals and behaviors as a straightforward method to help the user to map the stimuli perceived by an agent and its reaction (8).

Logical modelling with MaBoSS. In the study of biological systems, logical modeling provides an efficient way to study and represent complex behavioral patterns. This method involves representing biological entities, such as genes, proteins, or full pathways, as nodes within a network. Using a Boolean approach, each node can take two values, 0 for absent or inactive and 1 for present or active, and the update of these variables is monitored by logical rules that link all the inputs of a node with the logical connectors OR, AND and NOT. The type of model can be used to explore patients' responses by simulating various initial conditions, and account for mutations observed in patients by forcing the values of nodes in the model. MaBoSS is a C++ software for simulating Boolean models using continuous time Markov processes (4, 5). It uses an asynchronous update scheme, which allows the description of heterogeneous responses. By associating transition rates to each node, for both activation and inactivation, it generates continuous trajectories with a notion of physical time. MaBoSS uses two files for describing the model : the BND file with contains the information about the boolean network, and the CFG file which contains the simulation settings.

PhysiBoSS framework. PhysiBoSS is a addon of PhysiCell that integrates a MaBoSS engine inside each agent. This new approach allows to characterize the cell with a specific Boolean model that will represent the intracellular signaling dynamics. The Boolean network can be the same for all the cells or different per cell types. At each simulation step the agent cell can collect different stimuli that will modify some specific nodes of the network (input nodes). Next the MaBoSS engine will compute the model trajectory that can cause the switch of the so-called phenotypic nodes (or output nodes). Those nodes can then trigger some specific cell actions (motility, secretion, uptake, death and so on). PhysiBoSS uses as input the same configuration files of PhysiCell, and the .bnd and .cfg MaBoSS input files.

Mapping agent-based to intracellular models. PhysiCell provides a dictionary of signals and a dictionary for behaviors, aimed at giving better accessibility to all the signals perceived by each agent-cell and all possible behaviour that an agent-cell can express. PhysiBoSS uses this data structures to simplify the connection between PhysiCell and MaBoSS, giving access to the PhysiCell/MaBoSS mapping through the configuration file and so, drastically diminishing the amount of C++ code necessary to develop a model. Mapping can be of two types : input mapping, which links a PhysiCell signal to a MaBoSS (input) node by using activation thresholds, or output mapping, which links a MaBoSS (output) node to a PhysiCell behavior by using values representing the Boolean state. Implementation details about the mapping are available in the supplementary, section S1.1.

Time synchronisation. The intracellular model is updated periodically, according to the value of `intracellular_dt`. The `scaling` parameter is also available to match the time scale of the intracellular model to the time scale of the agent-based model. Finally, in order to account for biological phenomenons such as cellular desynchronisation, an option is available for stochastic update time. More information about the implementation of time is PhysiBoSS is available in the supplementary, section S1.2 and S1.3.

Results

PhysiBoSS allows the simulation of models which combine intracellular molecular description (with MaBoSS) and physical intercellular communication (with PhysiCell). With this approach, it is possible to study the impact of events that occur inside the cell at the level of the population, and the effect a treatment may have considering physical features. This paper aims at demonstrating the construction of such multi-scale models to answer biological questions.

We present four examples of multi-scale models: (1) a modified version of a previously published model of cell fate decision processes in response to death receptor engagement and the effect of a TNF-treatment on these decisions, (2) a model of cell cycle allowing to investigate the effect of perturbations in signalling, (3) a simple model of immune cell differentiation. For all these models, step-by-step procedures are provided as supplementary material to build these models and use them as templates for any other project. We also included in the supplementary materials an improved version of a model of cell invasion already published (9).

Cell fate model upon TNF treatment. Upon cell death receptor engagement, different phenotypes can be triggered depending on the status of some cell components. Programmed cell death, through necroptosis or apoptosis, or survival through the NF-κB pathway can be activated. An existing Boolean model describing the effect of death receptors (10) was integrated into PhysiBoSS (1) to study the effect of a TNF treatment on a population of interacting cells by varying the type of treatments (continuous vs. pulsating) and the composition of the population (to explore the efficacy of the treatment of an heterogeneous population). The model presented here is a version of the initially published one modified to fit the evolution of the tool.

Analysis of the intracellular model. The intracellular model considers two receptors, Fas and TNF, and studies the conditions that lead to either survival (*Survival* or programmed cell death (*Non_apoptotic_Cell_Death* or *NonACD* and *Apoptosis*) (see Supplementary Material, section S9). With MaBoSS(5), the proportion of the three cell fates can be quantified and differences appear with varying initial conditions or types of treatments: upon continuous TNF receptor activation, most of the cells (95%) will trigger apoptosis, while a small population of cells will activate either necrosis (3%) or NF-κB-driven survival (2%); when cells are treated in a pulsating manner (every 40 hours for 20 hours), the simulation of a population of individual non-interacting cells shows very little difference, even though, in contrast with the continuous treatment, at time 100, all cells have undergone apoptosis (Figure S11).

This model can also reproduce the effect of gene mutations on the cell fate distribution. For example, the double mutant *IKK++/cFLIP++* shows a shift of phenotypes following TNF treatment to only obtain resistant cells, with NF-κB fully active.

Integration of the Boolean model in PhysiBoSS. When integrating a Boolean model into PhysiBoSS, there are several aspects to consider: (1) the time synchronization of the two models, (2) the connection between the Boolean intracellular model and the agent-based model.

The synchronization of the two time scales is a tricky task as intracellular and extracellular events may not have the same scales. The two parameters controlling timing are: `scaling` and `intracellular_dt`. Since the standard PhysiCell simulation time unit is in minutes, while the cell fate model's unit is in hours, the `scaling` parameter needs to be set to 60, thus converting the MaBoSS model unit to minutes. The second parameter specifies how often the simulation should be updated. In this specific case, the asymptotic behavior of the system is considered, which is reached after 24h, setting the parameter `intracellular_dt` to 1440 min (24h = 1440min). In order to avoid having all our cells responding in synchrony to the TNF treatment, we set the value of `time_stochasticity`, a parameter responsible for producing slightly different periodic updates, to 0.5.

The next step consists in the mapping of the two models, described by three rules. The first mapping rule is an input rule, which describes the effect of the TNF in the vicinity of the cell in the activation of the TNF input node of the intracellular model. The following mapping rules are output rules, connecting the intracellular phenotypes to behaviors of the PhysiCell simulation. In the cell fate model, there are three outputs, two of which corresponding to the two death phenotypes. The first one will link the *Apoptosis* node to the `Apoptosis` behavior, which in PhysiCell is controlled by a fixed activation rate. In order to be uniquely controlled by the *Apoptosis* node, we set this rate to 0 when the node is inactive, and to a very high value (1e+6) when the node is active (thus ensuring the apoptosis is activated at the next update). The second rule is similar for the necrosis node (*NonACD*) which is linked to the activation rate controlling the PhysiCell's `Necrosis` behavior. Finally, the last phenotype, *Survival* is left without any mapping, as it represent the complement of the two death phenotypes, do it can be described as a resistant phenotype to the TNF treatment. Variations on this cell fate model exists where the NFkB pathway is linked to a autocrine secretion of TNF, which could create a feedback loop in our model(11). A brief description on how to create this behaviour is described in the supplementary, section S3.8.

To simulate the TNF treatment in time, a function was added, controlled by user parameters. Note that PhysiPKPD (12), a future add-on of PhysiCell, will facilitate the simulation of any type of treatments. For the prolonged TNF treatment, the parameter `treatment_duration` was set to 11520 minutes (8 days), more than our simulation max time. In Figure 1A, it can be observed that, while most of the population is killed either by apoptosis or necrosis at day 2, a resistant population emerges and leads to a large proliferating population at day 8.

To reproduce the effect of a pulsatile treatment, the parameters `treatment_duration` and `treatment_period` were modified to simulate a treatment of 2000 minutes happening every 3440 minutes. In Figure 1B, the size of the population of tumor cells decreases after each treatment. The purpose of such treatments could be preventing the formation of a resistant population, as well as reducing the toxicity of the TNF treatment.

Finally, to explore more functionalities of PhysiBoSS, we also produced a version of the model accounting for the observed necrotic core of the tumor due to the lack of oxygen (see supplementary materials, section S3.6) , and describing the impact of *IKK++*,*cFLIP++* mutation on the outcome of the treatment. (supplementary materials, section S3.7).

By reproducing this model from the original PhysiBoSS into PhysiBoSS 2.2, we showed how the usability of this framework was improved to facilitate its use, allowing a much wider user base to easily build complex models.

Boolean cell cycle model. The cell cycle is a complex system, controlled by cyclins and cyclins dependant kinases (CDKs) which are acting as checkpoints to validate that the necessary steps were performed and that the cycle can progress. The loss of control in the proliferation is one of the hallmarks of cancer, that may be due to some alterations in the signaling pathways that lead to the transcription of cell cycle genes. PhysiCell however represents this cycle as a very simple process, where each phase as a fixed duration, and no signalling in involved to pertub it. With this example, we wanted to integrate with PhysiBoSS a more realistic cell cycle model, and show how we can reproduce the effect of known mutations. To this end, we used a published Boolean model of cell cycle from Sizek et al.(13) as a intracellular model, and link it to the transitions between the different phases to control the progression of the PhysiCell cell cycle.

Analysis of the model. In their work, Sizek and collegues built a Boolean model that reproduces the cell cycle progression, including apoptosis and growth signals. The model is composed by 87 nodes and captures PI3K/AKT1 activity dur-

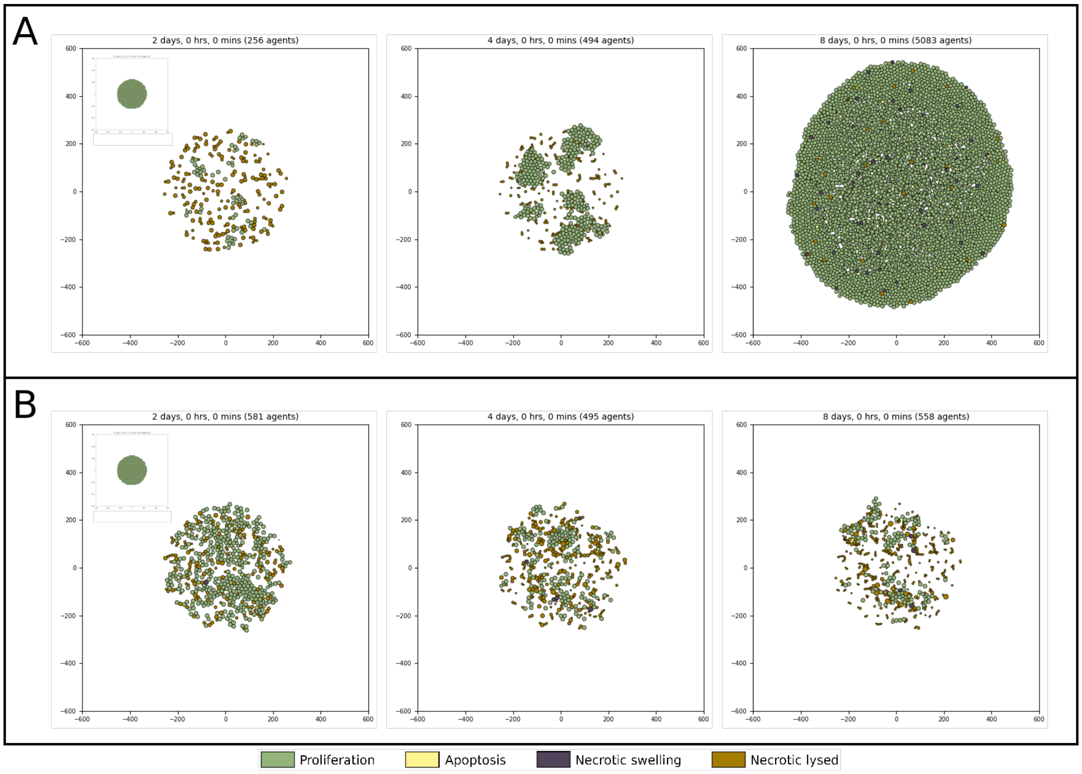

Fig. 1. Simulation of the cell fate model upon TNF treatment, at t=2,4,8 days. A) MaBoSS simulation of a prolonged and continuous TNF treatment. B) MaBoSS simulation of pulses of TNF treatment.

ing cell cycle and its role in the deregulation of PLK1 and FOXO3. The perturbations can lead to different cell fates such as G2 arrest and mitotic catastrophe characterized by a sustained activity of Cyclin B and Casp2 activation, respectively. In order to integrate the Boolean network into PhysiBoSS, it was necessary to perform a complete analysis of the model to decide which nodes would be responsible of the switch between cell cycle phases. The analysis was performed using MaBoSS (5) and included in a Jupyter Notebook where we simulated the wild type model with different initial conditions, and mutants (see suplementary materials, Cell_cycle_boolean_analysis.pdf).

The model analysis shows an interplay between components of the cell cycle and the apoptotic pathway, highlighting the role of Casp3, a read-out of cell death, which spontaneously and gradually gets activated after several cycles. The model is able to reproduce the sequential activation of the cyclins: Cyclin E, Cyclin A and Cyclin B, and their oscillation, until Casp3 gets fully activated. However, we observed that this sequence is not always preserved and leads to imcomplete cell cycle, such as Cyclin E and Cyclin A activation not follows by a Cyclin B activation.

The initial model reported published mutations and reproduced their phenotypes, which were then confirmed with the MaBoSS simulations (see notebook). Among these mutations, we focused on the role of PLK1, FOXO3, p110 and PI3K. Loss function of PLK1 (*PLK1* node is set to 0) leads to an overactivation of Cyclin B, indicating that the cells may be stuck in G2 phase, with no observed apoptosis. A knockout of FOXO3 (*Foxo3* node is set to 0) leads to cytokynesis. In this condition some cells are unable to separate the cytoplasm and to complete division and start apoptosis, while others complete the cell cycle but the next cycles are slowed down. Finally the double knock-in mutation of PI3K and p110 (PI3K and p110 nodes are set to 1) show an increase in the activity of AKT leading to an increase in the proliferation rate.

Among the in-built cell cycle models proposed by PhysiCell, we selected one of the simplest, the Flow Cytometry model, composed by 3 phases and 3 rates. In this model, a cell starts at the default phase "G0G1" and enter the cell cycle with a rate r01 to reach the "S" phase. From "S" phase, it moves to "G2M" phase with a rate r12. Finally, the cell divides and returns to "G0G1" phase at a rate r20.

With PhysiBoSS, it is possible to associate the transition rates of a cell cycle phase, to the state of a node of the Boolean model. To facilitate this pairing, we included in the Sizek model 3 nodes that matches the 3 transition of the Flow Cytometry model: *G0G1_entry*, *S_entry* and *G2M_entry*. The state of these nodes is determined by the activity of one or more Cyclins: CyclinD1 and CyclinA control the *G0G1_entry*, CyclinA and CyclinE control the *S_entry* and finally CyclinB controls the *G2M_entry*. The introduction of these three read out nodes does not affect the behavior of the network, but provide a single Boolean node for each transition between the three phases.

Integration of the Boolean model in PhysiBoSS. To include the Sizek model into PhysiBoSS, we again mainly have to describe two aspects of their connection : the time synchro-

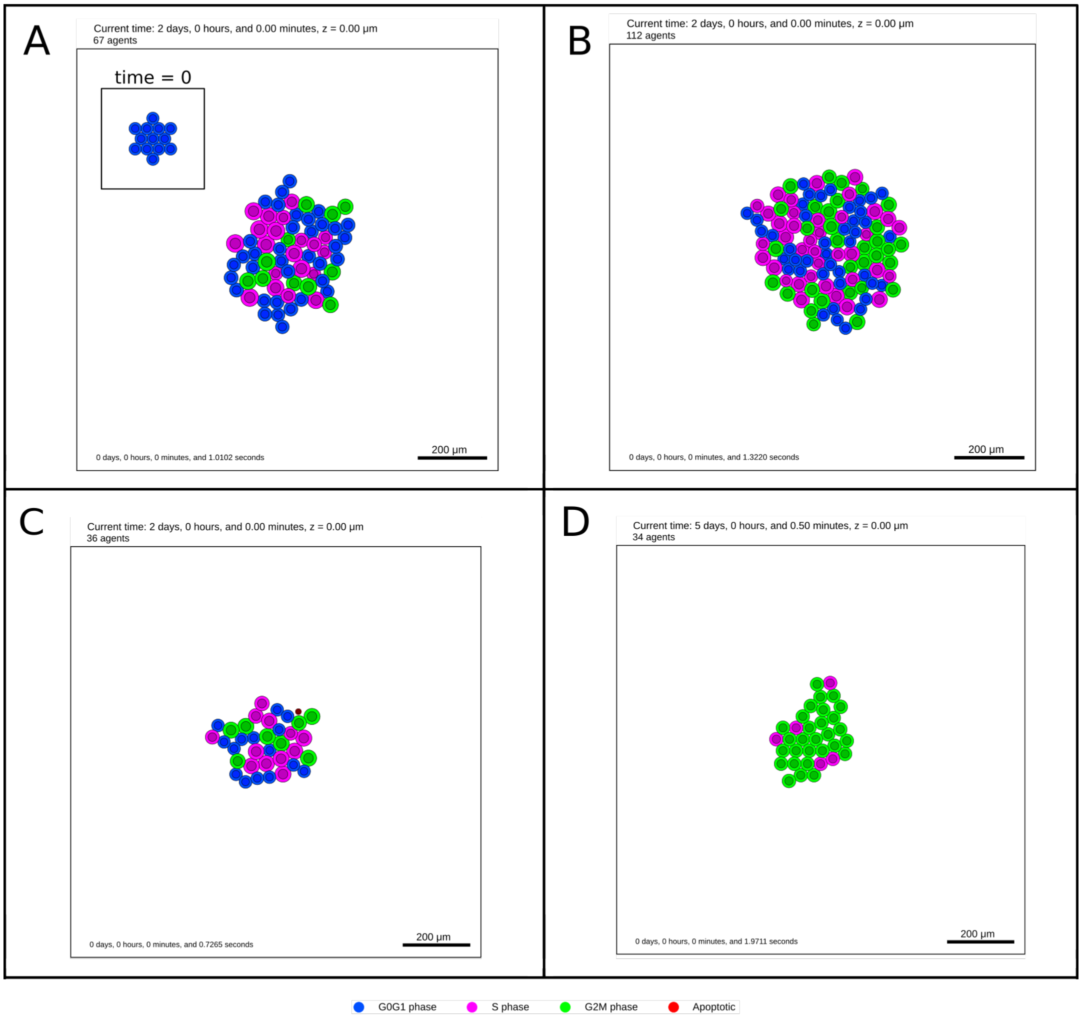

Fig. 2. Simulation of the Sizek cell cycle model. A) Wild type simulation at both time 0 and after 48 simulated hours. B) Double knock-in of PI3K and p110 accelerate the cell cycle frequency, rising the number of cell division, with 112 cells after 48 hours vs. 67 cells in Wild Type condition. C) FoxO3 knock-out simulation slows down the cell cycle, diminishing the number of cell divisions, with 36 cells after 48 simulated hours. D) Plk1 knock-out simulation causes the majority of cells to be stuck in G2/M phase. All the simulations were executed with a value of *scaling* of 37.5 and *intracellular_dt* of 2.5. Figure adapted from PhysiCell Studio.

nisation, and the mapping. To synchronize the time between the two models, we started by considering a cell cycle duration of 24h. Since a full cell cycle in MaBoSS is achieved in 24 units of time, we proceeded to set the scaling value to 60, similarly to what was done with the previous TNF model. However, this choice did not result in a 24h cell cycle, but to a longer one of 39h, due to the some incomplete cycles in the Sizek model. In order to fix this, we calculated a correction for the scaling factor, setting it to 40 and reproducing the expected cellular behaviors (see supplementary materials, section S4.2). The time interval is set to a small value (`intracellular_dt` = 1 min) as, contrary to the previous model, here it is important in this model to capture transient effects. The model does not take into account environmental conditions, making irrelevant the mapping of input nodes. However, it is possible to specify in the intracellular configuration the initial state of the inputs of the model, such as the node *Trail* (death signal) or *GF* (growth factor). We proceeded to connect the previously defined phenotype nodes to the corresponding behaviors, associated to the controls of the cell cycle transition rates. The *S_entry* node is connected to the behavior `Cycle entry`, *G2M_entry* to `exit from cycle phase 1`, *G0G1_entry* to `exit from cycle phase 2`. Finally the node *Casp3* is connected to the behavior `apoptosis` which concretely modifies the rate of activation of the apoptotic death model. The basal value of all the rates is set to 0. When one of the nodes regulating the phenotype is activated, the transition rate is fixed to a very high value (1e+6) in order to immediately trigger the phase switch or the apoptotic death. When the node is inhibited, it restores the basal value of the transition rate.

The PhysiBoSS simulations assumed an initial population of 13 cells (agents) growing to 67 in 48 hours (Figure2A). The phases follow a proper order in individual cells, but not all cells are in the same phase of the cycle as expected in a desynchronized population of cells.

We further tested the impact of single and double mutations at the population level, by selecting the appropriate node to mutate and assigning it a value of 0 (knock-out) or 1 (knock-in). *Plk1* knock-out (*Plk1* node fixed to 0), as expected from the MaBoSS analysis, causes the majority of the cells to get stucked in G2/M phase, in a cell cycle arrest (Figure 2D). *FoxO3* knock-out (*Foxo3* node fixed to 0), allows the cells to go through one cell cycle before either dying or slowing down the proliferation. The cells are not arrested in a specific phase of the cycle, but they keep proliferating at a very low rate (Figure 2C). Finally, the double mutant *PI3K* and *p110* overexpressed (*PI3K* and *p110* nodes fixed to 1) results in a high increase of the proliferation rate, bringing the final number of cells after 48 hours from 13 to about 110 (Figure 2B). In conclusion, the multiscale model of a detailed molecular description of the cell cycle reproduces the complexity of the cell cycle at the single and multicellular level, allowing not only to modify the duration of the cell cycle, but to perform mutations and explore multiple initial conditions (corresponding different extracellular contexts).

Immune cell differentiation. The examples previously presented assumed that all cells were of the same type. With PhysiBoSS, users can also consider interactions among several cell types but even model cell differentiation, where a cell of a specific type can transition into a different, user-defined cell type. To show how a multicellular model can be built, we considered a simple model of immune cell differentiation that encompasses six different cell types and relies on two different Boolean models.

Analysis of the intracellular models. The Boolean model for cell differentiation is adapted from a previously published model of Corral-Jara and colleagues (15), which describes the processes of T cell differentiation. The model is based on experiments performed on naive CD4+ T cells (referred to as T0), which, upon the effect of external stimuli, can differentiate into either a Type 1 helper cell (Th1), a T helper 17 cell (Th17), or a regulatory T cell (Treg). In contrast with the models we presented above, Corral-Jara's model has been designed to be a multi-level model. Some nodes reflect this feature: while on simulation tools like GINsim, multi-level activity can be simulated on a single node, in MaBoSS this is not possible and a specific node is created to allow different activity values and their different downstream signaling (e.g: *MHCII_b1*,*MHCII_b2*). We also use a simple phenomenological model for dendritic cells, describing in a few nodes their behavior. The model (Figure S24) encompass a total of 5 nodes, of which 4 inputs (Maturation, Contact, CCR7 and CCR7_h) and 1 phenotype nodes (Migration). A more complex model can later replace this simple model.

In this model, under the chemoattractant effect of the CC motif chemokine ligand 21 (CCL21), a cytokine constitutively expressed in secondary lymphoid organs (such as lymph nodes), a population of mature dendritic cells (mDCs) is attracted towards the draining lymph node. The CC chemokine receptor type 7 (CCR7), whose expression is increased on mDCs surface (represented by the node *CCD7*), allow them to migrate to the lymph nodes by binding to CCL21 (16, 17) and activating the node *Migration*. Once in the lymph node, the node *CCR7_h* representing a very high CCL21 concentration is activated which deactivate the node *Migration*.

mDCs also express a set of ligands that are capable of triggering the differentiation of the T0 cell population into 3 different subsets of CD4+ T cells, once in the lymph nodes. Among these ligands, we can cite Interleukin-12 (IL-12), Interleukin-1β (IL-1β), and other cytokines as Interleukin-6 (IL-6) or Transforming growth factor beta (TGF-β). We choose not to include all those nodes in our mDC model, and just represent them by a single node *Contact*. Within the Corral-Jara's MaBoSS model, these nodes are already present as input nodes. The model also includes 3 master transcription factors considered as markers of differentiated T cells: RORgt (Th17), FOXP3 (Treg) and Tbet (Th1). Based on these nodes, we built three phenotype nodes (*Th1*, *Treg*, *Th17*), used later as output nodes, to better represent the different cell types. The relative logical equations have been constructed to avoid overlap between phenotypes, so that each one is mutually exclusive. A larger analysis of the Corral-Jara's model is included in a Jupyter notebook in supplementary materials, including a mutant analysis to search for perturbations modifying the balance of cell differentiation.

Integration of the Boolean models in PhysiBoSS. In this model, we need many cells types : naive T cells (T0), dendritic cells, type 1 helper cells (Th1), T helper 17 cells, regulatory T cells and finally lymphoid endothelial cells. For the integration of the two Boolean models presented above into PhysiBoSS, two intracellular models are created for the naive TCells and the dendritic cells (see supplementary materials, Figure S25) and S26). As for the TNF example, the asymptotic behaviors of both the naive T cell and dendritic cell are considered. Based on the MaBoSS simulations, the two parameters, `scaling` and `intracellular_dt` are set to 1 and 6, respectively.

For simplicity, we create one single endothelial cell secreting CCL21, located in an area representing the lymph node. We also create a population of T0 cells in the same area, as well as a distant population of dendritic cells (Figure 3A). The initial state of the PhysiBoSS simulation assumes that the dendritic cells are mature, a condition in which they are expressing CCR7, a receptor that drives the migration of mature DCs (mDCs) towards secondary lymphoid structures (i.e., the lymph nodes). We then create one input mapping in the dendritic cells, linking the substrate `CCL21` to the node *CCR7*. Upon activation of the *CCR7* node within the DC network, mDCs move towards the source of CCL21, following its gradient combined with a random walk. The percentage of random walk in the movement can be changed, from a full deterministic movement (Figure 3C) to a completely stochastic movement. This can affect the result of the simulation, leading to mDCs needing more time to reach the lymph node (as shown in Figure 3D), and could be linked to the quantity of CCR7 expressed by the dendritic cells.

Once in the lymph node, we described the DC as going back to a random walk. To achieve this, we create a second input

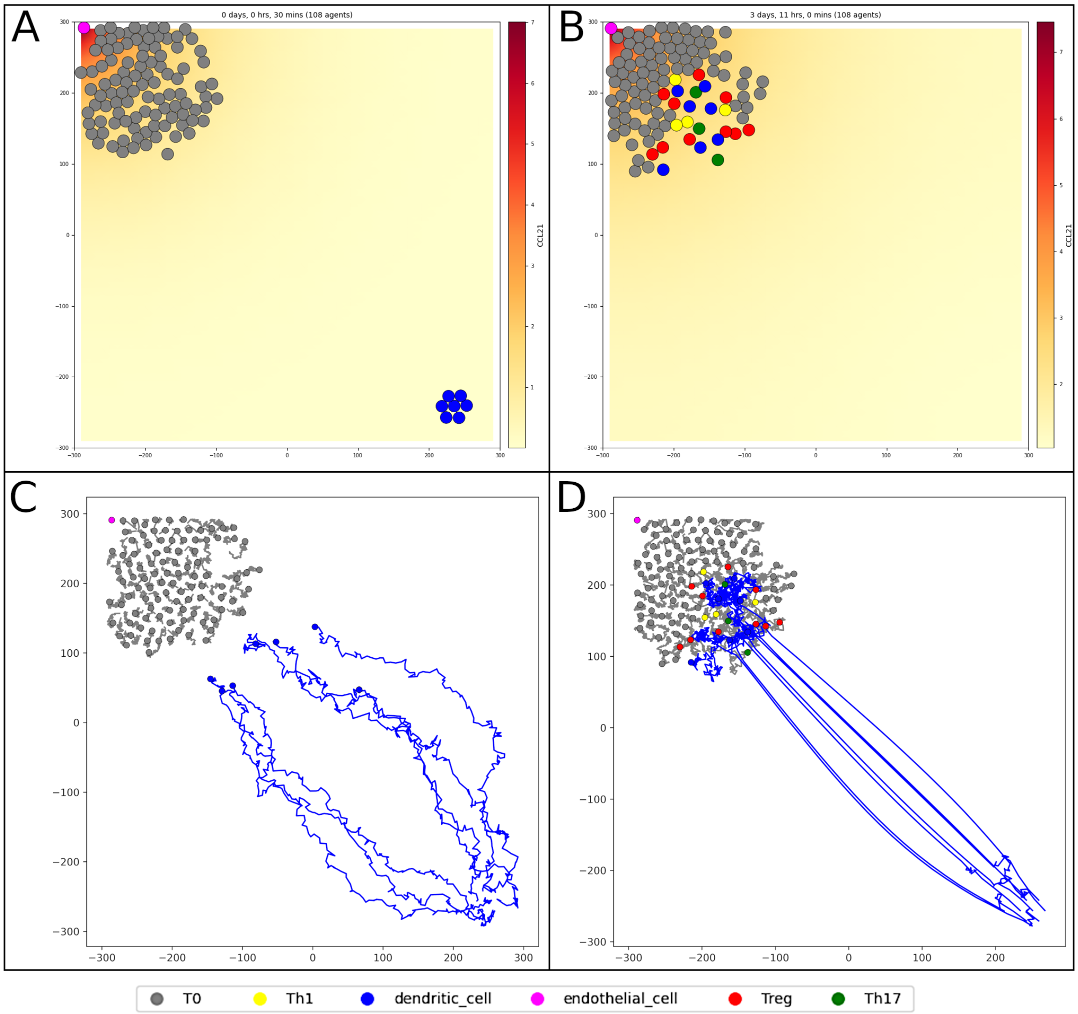

Fig. 3. Simulation of the T-cell differentiation model in 2 and 3 dimensions. A) Initial population of T-cell (gray), with an endothelial cell (pink) secreting CC21. A population of dendritic cells (blue) is attracted towards the source of CCL21. B) Upon contact, the dendritic cells trigger the receptors of the naive T-cell, which start the differentiation process according to the outputs of the intracellular model, into Treg (red), Th1 (yellow) and Th17 (green). C) and D) 2D simulation of the physical trajectories of the dendritic cells over time for different value of migration bias, respectively 0.2 and 1, plotted using PhysiCOOL(14).

mapping rule, which activates the *CCR7_h* node only when a high concentration of CCL21 is detected. This input inhibit the *Migration* node, deactivating the chemotactic movement.

When mDCs and T0 cells are in contact, the differentiation process of naive T cells is triggered. The mDCs secrete major cytokines that are essential to mediate first the contact between DCs and T0 cells (set consists in IL-12, IL-1β, IL-6, TGF-β and IL-23), and then the cascades leading to the three subsets of differentiated T cells, Th1, Th17 and Treg. For the sake of simplicity, instead of allowing each agent representing a dendritic cell to release cytokines, we encoded such interactions by activating the input corresponding to the cytokines within the T0 model. To do this, we created many input mappings which connect the contact of dendritic cell with a T0 to the activation of the input nodes within the latter, representing the cytokines that should be released by the dendritic cells. In addition to the above-mentionned list of cytokines, input nodes triggered upon contact between dendritic cells and T0 inlcude also: *MHCII_b1*, *MHCII_b2*, *CD80*, *CD4* and *PIP2*. The activation of these nodes will trigger the differentiation into Th1, Th17 or Treg. To achieve this, we added three output mapping rules, linking the transformation into these cell types to the phenotype nodes *Th1*, *Th17*, *Treg* presented precedently. None of these new cell types has an intracellular model, so upon differentiation, it loses all the T0 properties. This choice was made to allow the implementation of specific behaviors for the different T-cell types in future version of the PhysiBoSS model. The model has been tested for single and combination of double mutants, to search for possible targets that can influence the probability of differentiation for the three cell types (See supplementary materials, section S5.4). Although the intracellular model proved to be robust, some combinations of mutants, can cause the probability of Th1 and Th17 differentiation to vary by about 10%. On a small cell population, the effect of these mutations does not have a big impact, but it can if a higher initial population number is considered for T0.

A. Update of the tumor invasion model. Advanced features and new functionalities can be hard-coded in a model in the appropriate sections. The model of cancer cell invasion presented in (9) is a perfect example of extended functionality through the manipulation of the PhysiCell code. By default, the substrates created by PhysiCell have no physical interactions with the agents. In the model proposed in our previous publication, a function to simulate repulsion and adhesion with a substrate (extracellular matrix), was implemented directly in the code. Using the new functionalities provided by this new PhysiBoSS version, we adapted and simplified the tumor invasion model. Due to its complexity, we refer to the supplementary materials (section S6) for the implementation and analysis of the model.

Discussion

In this paper, we presented new functionalities of PhysiBoSS, which are drastically simplifying the process of creating models. We shows that, using the new mapping system, we can now easily connect the agent-based model to Boolean intracellular models. While previous version of PhysiBoSS required knowledge in C++ programming to allow the creation of model, with this new version the user can completely rely on PhysiCell Studio, the graphical interface of PhysiCell, to build a model from existing templates. For some specific functionalities which still requiring to write code, such as the treatment mechanism, new addons of PhysiCell are being developed to simplify their accessibility.

While simple, we believe that the three models presented here cover enough functionalities to give a good overview in PhysiBoSS, and provide good examples to start from. We are providing in the supplementary materials a step by step guide for installing PhysiBoSS and PhysiCell-Studio, and for building these models in an effort to allow newcomers to follow the process of creating them. The example on cancer invasion in supplementary shows a better real world example, and its comparison with the original models shows the simplicity and power of the mapping system.

Still, building these models remains a very difficult endeavor, in very big part because of the many parameters required for tuning their behavior. In this tutorial, we didn't want to put too much emphasis on this, but it is a real challenge. New methods are needed in this field, and we believe the use of surrogate models (18, 19) will prove itself fundamental. Despite these challenges, we hope that the improvements in PhysiBoSS presented here will allow a larger community of users to enter the multi-scale modeling world.

Funding

This work was funded by the PerMedCoE project which is part of the European Union's Horizon 2020 research and innovation program under the grant agreement n951773. This work also received funding from the Inserm amorçage project.

Acknowledgments

We thank John Metzcar for testing PhysiBoSS implementation and Furkan Kurtoglu and Arnau Montagud for the fruitful discussion about PhysiBoSS' mapping.

Bibliography

1. Gaelle Letort, Arnau Montagud, Gautier Stoll, Randy Heiland, Emmanuel Barillot, Paul Macklin, Andrei Zinovyev, and Laurence Calzone. PhysiBoSS: a multi-scale agent-based modelling framework integrating physical dimension and cell signalling. *Bioinformatics*, 35 (7):1188–1196, April 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty766.
2. Miguel Ponce-de Leon, Arnau Montagud, Vincent Noël, Gerard Pradas, Annika Meert, Emmanuel Barillot, Laurence Calzone, and Alfonso Valencia. PhysiBoSS 2.0: a sustainable integration of stochastic Boolean and agent-based modelling frameworks, April 2022. Pages: 2022.01.06.468363 Section: New Results.
3. Ahmadreza Ghaffarizadeh, Randy Heiland, Samuel H. Friedman, Shannon M. Mumenthaler, and Paul Macklin. PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. *PLOS Computational Biology*, 14(2):e1005991, February 2018. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005991. Publisher: Public Library of Science.
4. Gautier Stoll, Eric Viara, Emmanuel Barillot, and Laurence Calzone. Continuous time boolean modeling for biological signaling: application of Gillespie algorithm. *BMC Systems Biology*, 6(1):116, August 2012. ISSN 1752-0509. doi: 10.1186/1752-0509-6-116.
5. Gautier Stoll, Barthélémy Caron, Eric Viara, Aurélien Dugourd, Andrei Zinovyev, Aurélien Naldi, Guido Kroemer, Emmanuel Barillot, and Laurence Calzone. MaBoSS 2.0: an environment for stochastic Boolean modeling. *Bioinformatics*, 33(14):2226–2228, July 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx123.
6. Randy Heiland, Daniel R. Bergman, Blair Lyons, Julie Cass, Heber L. Rocha, Marco Ruscone, Vincent Noël, and Paul Macklin. Physicell studio: a graphical tool to make agent-based modeling more accessible. *bioRxiv*, 2023. doi: 10.1101/2023.10.24.563727.
7. John Metzcar, Yafei Wang, Randy Heiland, and Paul Macklin. A review of cell-based computational modeling in cancer biology. *JCO clinical cancer informatics*, 2:1–13, 2019.
8. Jeanette AI Johnson, Genevieve L Stein-O'Brien, Max Booth, Randy Heiland, Furkan Kurtoglu, Daniel Bergman, Elmar Bucher, Atul Deshpande, Andre Forjaz, Michael Getz, et al. Digitize your biology! modeling multicellular systems through interpretable cell behavior. *bioRxiv*, pages 2023–09, 2023.
9. Marco Ruscone, Arnau Montagud, Philippe Chavrier, Olivier Destaing, Isabelle Bonnet, Andrei Zinovyev, Emmanuel Barillot, Vincent Noël, and Laurence Calzone. Multiscale model of the different modes of cancer cell invasion. *Bioinformatics*, 39(6):btad374, 2023.
10. Laurence Calzone, Laurent Tournier, Simon Fourquet, Denis Thieffry, Boris Zhivotovsky, Emmanuel Barillot, and Andrei Zinovyev. Mathematical Modelling of Cell-Fate Decision in Response to Death Receptor Engagement. *PLOS Computational Biology*, 6(3):e1000702, March 2010. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000702. Publisher: Public Library of Science.
11. Gautier Stoll, Aurélien Naldi, Vincent Noël, Eric Viara, Emmanuel Barillot, Guido Kroemer, Denis Thieffry, and Laurence Calzone. Upmaboss: A novel framework for dynamic cell population modeling. *Frontiers in Molecular Biosciences*, 9, 2022. ISSN 2296-889X. doi: 10.3389/fmolb.2022.800152.
12. Daniel Bergman, Lauren Marazzi, Mukti Chowkwale, Supriya Bidanta, Tarunendu Mapder, Jialun Li, et al. Physipkpd: A pharmacokinetics and pharmacodynamics module for physicell. *Gigabyte*, 2022, 2022.
13. Herbert Sizek, Andrew Hamel, Dávid Deritei, Sarah Campbell, and Erzsébet Ravasz Regan. Boolean model of growth signaling, cell cycle and apoptosis predicts the molecular mechanism of aberrant cell cycle progression driven by hyperactive PI3K. *PLOS Computational Biology*, 15(3):e1006402, March 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006402. Publisher: Public Library of Science.
14. Inês G Gonçalves, David A Hormuth II, Sandhya Prabhakaran, Caleb M Phillips, and José Manuel García-Aznar. Physicool: A generalized framework for model calibration and optimization of modeling projects. *Gigabyte*, 2023:1–11, 2023.
15. Karla Fabiola Corral-Jara, Camille Chauvin, Wassim Abou-Jaoudé, Maximilien Grandclaudon, Aurélien Naldi, Vassili Soumelis, and Denis Thieffry. Interplay between SMAD2 and STAT5A is a critical determinant of IL-17A/IL-17F differential expression. *Molecular Biomedicine*, 2(1):9, April 2021. ISSN 2662-8651. doi: 10.1186/s43556-021-00034-3.
16. Alfonso Martín-Fontecha, Silvia Sebastiani, Uta E. Höpken, Mariagrazia Uguccioni, Martin Lipp, Antonio Lanzavecchia, and Sallusto Federica. Regulation of dendritic cell migration to the draining lymph node: impact on t lymphocyte traffic and priming. *Journal of Experimental Medicine*, pages 615–621, Aug 2003.
17. Michele Weber, Robert Hauschild, Jan Schwarz, Christine Moussion, Ingrid de Vries, Daniel F. Legler, Sanjiv A. Luther, Tobias Bollenbach, and Michael Sixt. Interstitial dendritic cell guidance by haptotactic chemokine gradients. *Science*, 339(6117):328–332, 2013. doi: 10.1126/science.1228456.
18. Richard J Preen, Larry Bull, and Andrew Adamatzky. Towards an evolvable cancer treatment simulator. *Biosystems*, 182:1–7, 2019.
19. Heber L. Rocha, João Vitor de O. Silva, Renato S. Silva, Ernesto A.B.F. Lima, and Regina C. Almeida. Bayesian inference using gaussian process surrogates in cancer modeling. *Computer Methods in Applied Mechanics and Engineering*, 399:115412, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.115412.