NEXUS: A MULTI-SCALE SIMULATOR FOR BIOLOGICAL CONTROL AND CAUSAL DISCOVERY

Amogh Singh[*]
University of Cambridge

Patrick Schwab
GSK.ai

Kim Branson
GSK.ai

Pietro Liò
University of Cambridge

Arash Mehrjou[†]
GSK.ai
Max Planck Institute for Intelligent Systems

ABSTRACT

Biological systems exhibit complex multi-scale dynamics spanning subcellular gene regulatory processes to tissue-level spatial organization. However, causal analysis and control of biological systems face fundamental challenges due to the predominantly static nature of experimental measurements and the difficulty of collecting large-scale interventional datasets. We present NEXUS, a hierarchical biological simulator that addresses these limitations by generating realistic, multi-modal biological data across cellular and subcellular scales. NEXUS integrates gene regulatory network dynamics for RNA and protein simulation with spatial models of cell movement, chemical diffusion, and reactions. Built on the JAX framework, NEXUS leverages automatic vectorization and just-in-time compilation to achieve significant speedups over existing simulators, particularly for large cell populations. The simulator supports extensive customization, backpropagation for parameter learning, and generation of interventional data, enabling large-scale dataset creation for causal discovery and analysis. Critically, NEXUS provides a testbed for advancing control strategies in biological systems, particularly hierarchical optimal control where interventions are applied simultaneously at multiple levels to steer the system towards desired states. We demonstrate NEXUS's accuracy through validation against established simulators and real-world biological data, showing its potential to advance both causal reasoning and control strategies in biological systems.

1 INTRODUCTION

Causal analysis of biological systems plays a crucial role in understanding disease mechanisms and developing targeted therapeutic interventions. By identifying causal relationships between biological entities, researchers can design precise interventions such as gene editing (Mehrjou et al., 2021), protein inhibition, and activation strategies (Nada et al., 2024). However, performing reliable causal inference on biological data faces significant methodological and practical challenges (Chevalley et al., 2023; 2024).

Biological systems are inherently dynamic, with processes unfolding across multiple temporal and spatial scales (Resat et al., 2011). Yet experimental measurements typically capture only static snapshots of these complex systems, providing limited insight into temporal dynamics and causal mechanisms (Somer et al., 2026). Furthermore, causal analysis methods often require large-scale datasets with diverse interventional conditions, which are difficult and expensive to obtain through experimental means alone (Luna et al., 2021; Chevalley et al., 2025b). The scarcity of interventional data, combined with the static nature of most biological measurements, constrains our ability to learn causal models and validate intervention strategies (Yuan & Shou, 2022; Chevalley et al., 2025a).

[*] Correspondence: as3600@cam.ac.uk

[†] Correspondence: arash@distantvantagepoint.com. Senior and supervising author.

To address these limitations, we introduce NEXUS, a hierarchical biological simulator designed to generate realistic, multi-modal biological data across cellular and subcellular scales. NEXUS is highly configurable, computationally efficient, and capable of producing interventional datasets that are essential for causal discovery and analysis. The simulator is built on dynamic biological models and supports automatic differentiation, enabling backpropagation for learning parameters from existing experimental data.

NEXUS operates at two complementary levels of biological organization. At the subcellular level, the simulator models gene regulatory networks to compute RNA and protein concentrations, capturing the intricate dynamics of transcriptional regulation and protein synthesis. At the cellular level, NEXUS simulates tissue-scale phenomena including cell movement, chemical diffusion, and biochemical reactions. These two levels can be integrated such that each cell's internal molecular dynamics are governed by the subcellular simulator while spatial interactions and tissue-level processes are handled by the cellular simulator.

The capabilities of NEXUS enable several important applications. Researchers can explore the effects of different parameter configurations, study system responses to various interventions, and perform gradient-based optimization to learn model parameters from experimental observations. By generating large-scale synthetic datasets that capture realistic biological dynamics, NEXUS facilitates the development and validation of causal discovery algorithms, ultimately advancing our understanding of biological systems and improving intervention design.

Beyond causal discovery, NEXUS provides a critical testbed for advancing control strategies in biological systems. The hierarchical architecture of NEXUS supports a research direction we refer to as *controllable biology*, in which interventions are applied simultaneously at multiple scales to steer biological systems towards desired states. The theory of multiscale hierarchical control has been extensively developed across diverse domains, providing a principled foundation for coordinating interventions across interacting levels of complex systems (Clarke et al., 2025; Dehghani, 2018; Mehrjou). Within this framework, control strategies can target both subcellular processes such as gene expression modulation and cellular-level processes such as chemical signaling, allowing for coordinated multi-scale interventions. The simulator's support for automatic differentiation and gradient-based optimization makes it particularly well-suited for learning optimal control policies that account for the complex interactions between different biological scales. This capability opens new avenues for designing therapeutic interventions that leverage the hierarchical nature of biological systems to achieve more effective and precise control outcomes.

2 RELATED WORKS

Existing simulators focus largely on either subcellular or cellular-level dynamics. At the subcellular level, SERGIO (Dibaeinia & Sinha, 2020) models gene concentrations from production, decay, and regulatory influences in a GRN, with stochastic RNA dynamics and optional noise for instrument-specific outputs. scMultiSim (Li et al., 2025) extends this with RNA velocities, chromatin assembly, and a kinetic model (cell- and gene-identity vectors) plus technical noise for single-cell sequencing. At the cellular and tissue level, lattice-based tools include Biocellion (Kang et al., 2014), which uses a grid and user-defined agents that can create or delete entities, and scMultiSim's cellular automata for spatial cell arrangements. Lattice-free simulators include CX3D (Zubler & Douglas, 2009), which uses triangulated sub-fields for diffusion and reactions and force-based cell movement for neuronal development, and PhysiCell (Ghaffarizadeh et al., 2018), which couples PDEs for diffusion and reactions with agent-based cell cycling, death, volume regulation, and motility, and models state transitions as a Markov chain.

These tools capture biological processes at individual scales but offer limited multi-scale integration and causal analysis support, and few support backpropagation or interventional data generation. NEXUS addresses these gaps with a unified framework that integrates subcellular and cellular simulation and supports automatic differentiation for parameter learning and intervention analysis, serving as both a data generator and a testbed for causal discovery and control. A detailed capability comparison is given in Appendix A.5 (Table 2).

3 Methodology

NEXUS is designed as a hierarchical simulator capable of modeling biological processes across multiple levels of granularity. The architecture consists of two integrated systems operating at different scales. At the cellular level, NEXUS simulates cell movement, chemical diffusion, and reactions within tissue environments. At the subcellular level, the simulator models RNA dynamics through gene regulatory networks and computes protein concentrations resulting from RNA transcription and translation.

This section describes the mathematical foundations and implementation details of both simulator components. The Results section presents validation experiments and performance comparisons demonstrating NEXUS's accuracy and computational efficiency.

3.1 Subcellular Simulator

The subcellular component of NEXUS, referred to as NEXUS-GRN, simulates RNA and protein concentrations across multiple cells using gene regulatory networks to model transcriptional regulation. The development of this component incorporates concepts from SERGIO (Dibaeinia & Sinha, 2020) for RNA concentration simulation and Dynprot (Kuchta et al., 2018) for protein concentration computation. RNA concentrations are determined by the concentrations of regulatory genes and the RNA's own decay rate, while protein concentrations are derived from the corresponding RNA concentrations and protein-specific half-lives.

NEXUS-GRN extends the capabilities of SERGIO and Dynprot by combining their functionalities and providing extensive customization options. These include configurable Hill coefficients, variable decay rates for individual genes, and optional Wiener noise injection to model stochastic biological processes.

3.1.1 RNA concentration

Following Dibaeinia et al. (Dibaeinia & Sinha, 2020), RNA concentration is computed as a function of the concentrations of its regulators and its own decay. Master regulators are produced at a fixed basal rate independent of other species. The simulator initializes all RNAs at their steady state concentrations, which are computed in advance to reduce runtime. The steady state expressions are as

$$E[x_i] = \begin{cases} \frac{b_i}{\lambda_i} & \text{if gene } i \text{ is a Master Regulator} \\ \frac{\sum_j p_{ij}(E[x_j])}{\lambda_i} & \text{if gene } i \text{ is not a Master Regulator.} \end{cases} \tag{1}$$

The equation for production rates of each RNA, calculated as a function of its regulators and its own basal rate, is given as

$$P_i = \sum_j p_{ij} + b_i \tag{2}$$

$$p_{ij} = \begin{cases} K_{ij} \frac{x_j^{n_{ij}}}{h_{ij}^{n_{ij}} + x_j^{n_{ij}}} & \text{if regulator } j \text{ is activator} \\ K_{ij} \left(1 - \frac{x_j^{n_{ij}}}{h_{ij}^{n_{ij}} + x_j^{n_{ij}}}\right) & \text{if regulator } j \text{ is repressor.} \end{cases}$$

Given the production rate P_i, the concentration $(x_i)_t$ of gene i at time t evolves in discrete time according to

$$\begin{aligned} (x_i)_{t+1} = (x_i)_t &+ (P_i(t) - \lambda_i x_i(t))\Delta t \\ &+ q_i \left(\sqrt{P_i(t)}\Delta W_\alpha + \sqrt{\lambda_i x_i(t)}\Delta W_\beta \right), \end{aligned} \tag{3}$$

where ΔW_α and ΔW_β are independent Wiener processes with $\Delta W = \sqrt{\Delta t} \cdot N(0, 1)$. The remaining parameters are b_i (basal production rate), K_{ij} (regulatory influence of gene j on gene i), n_{ij} (Hill coefficient), h_{ij} (half response), and λ_i (decay rate).

3.1.2 Protein Concentration

Similar to the computation of the RNA concentration, the protein concentration also starts from a steady state. The protein concentration depends only on the transcribing RNA. The equations relating to the steady state computation and concentration update are given as

$$E[Pr_i(t)] = \frac{k_{\text{trans},i} \cdot E[G_i(t)]}{k_{d,i}} \tag{4}$$

$$\frac{d[Pr_i(t)]}{dt} = k_{\text{trans},i}[G_i(t)] - k_{d,i}[Pr_i(t)] \tag{5}$$

where $Pr_i(t)$ is the concentration of protein i at time t, $k_{\text{trans},i}$ is the translation rate, $k_{d,i}$ is the degradation rate, and $G_i(t)$ is the mRNA concentration.

3.2 Cellular Simulator

The cellular component of Nexus simulates tissue-level dynamics involving multiple cells within a spatial environment. This simulator supports both lattice-based and lattice-free cell modeling approaches and captures cell movement, cell cycle progression, chemical production and decay through reactions, and chemical transport via diffusion. The subcellular and cellular simulators can be integrated such that each cell's internal RNA and protein concentrations are governed by the subcellular simulator while spatial interactions and tissue-level processes are handled by the cellular simulator.

The simulator is capable of simulating different types of cells, where each type of cell is defined by a set of parameters. During initialization, the cells are populated according to the configured proportions and then simulated using the parameters defined for that cell type.

3.2.1 Cell Movement

Each cell is capable of movement along the three axes and is controlled by cell-cell attraction and repulsion forces, a residual drift force and a random induced force. The cell-cell forces are attractive in most of the cases and the repulsive force is only used to prevent or resolve overlaps between the cells. A residual drift force is computed where the cell inherits a fraction of the force it was experiencing in the previous time-step, and a random force can also be applied on the cells. The strengths of the three forces can be modified to fine-tune and test different configurations. The equations relating to the computation of inter-cellular forces is given in A.2

3.2.2 Cell Cycle

Cells cycle through interphase, mitosis, programmed cell death (apoptosis), and sudden cell death (necrosis). Cycle length is drawn from a normal distribution centered on a configured value, and each cell is assigned interphase and mitosis durations that determine its state during simulation. At mitosis completion, a cell splits into two equal-mass daughter cells, both reset to interphase; daughter cell cycle parameters are drawn from a normal distribution centered on the parent's values. Cells may undergo apoptosis or necrosis at configurable random probabilities at each stage. Necrosis sets mass, volume, and radius to zero (sudden shrinking), while apoptosis shrinks cells gradually with a configurable decomposition rate. Dead cells are excluded from inter-cellular force computation.

3.2.3 Diffusion

The diffusion of chemicals across the tissue is simulated by splitting the tissue into *fields* and the chemical movement and reactions are simulated at field-level. Diffusion is performed between a field and its neighbours which is currently setup as a fixed gird, dividing the available space into multiple fields. Each step of the simulator performs a computation of the chemical flux across the neighbouring fields and is used for moving the chemicals across fields. The diffusion process is configured to preserve the total mass in the system by operating at the mass-level to compute the concentrations. The simulator also includes an additional check used to prevent reduction of masses to negative values which adjusts the flux in case a post-diffusion negative-mass situation is detected.

The change in concentration through diffusion is computed as

$$\begin{aligned} J_{i,t} &= \sum_{j \in \text{neighbors}} -P_{ij}[C_{j,t} - C_{i,t}] \\ C_{i,t+\Delta t} &= C_{i,t} - \nabla \cdot J_{i,t} \Delta t \end{aligned}$$

where $J_{i \to j}$ is the flux from field i to field j, with $J_{i \to j,t} = -J_{j \to i,t}$ and $P_{i,j} = P_{j,i}$. The intermediate steps leading to these results are shown in A.3.

NEXUS also supports diffusion of chemicals between the cells and the fields. Currently, the proteins from the cells can diffuse into the fields and vice-versa and is calculated using the same method as described above. Each cell is associated to the field it is closest at each timestep to compute the flux.

3.2.4 REACTIONS

The simulator is currently capable of performing zero, first and second order reactions of the configured chemicals. The reactions take place at a field-level where the chemicals inside a field are used in performing a reaction. NEXUS can be configured to perform reactions with a probability defined in the equation below. The order of execution of the reactions is randomly generated at each step in order to reduce irregular outputs due to a fixed order of execution.

$$\text{Prob}(\text{reaction}_i) = \frac{K_{m,i}}{\sum_j K_{m,j}} \left[1 - e^{-\Delta t \sum_j K_{m,j}}\right]$$

Here the reaction order $m \in \{0, 1, 2\}$ is the number of concentration factors in the rate law (zero for constant rate, one for unimolecular, two for bimolecular). The term $K_{m,i}$ is the rate constant for reaction i at order m, the sum $\sum_j K_{m,j}$ runs over all reactions of order m in the field, and Δt is the time step. The factor in brackets is the probability that any reaction of order m occurs in the interval Δt; the fraction selects reaction i among those.

The simulator generates a reaction matrix for each defined reaction using the reaction order, reactants and products. For computational efficiency, each chemical's concentration is updated independently assuming it is the only chemical in a reaction of the given order. The concentration updates for reaction orders 0, 1, and 2 are:

$$\begin{aligned} \lambda &= k_0 \Delta t \\ \gamma &= \text{Sample}(\text{Poisson}(\lambda)) \\ [A]^{t+\Delta t} &= [A]^t + \gamma \\ \ln[A]^{t+\Delta t} &= \ln[A]^t + \gamma \\ \frac{1}{[A]^{t+\Delta t}} &= \frac{1}{[A]^t} - \gamma \end{aligned}$$

where the first relation defines λ for zero-order updates (second relation), and γ is sampled from the Poisson distribution based on the reaction rate constant k and time step Δt for first-order (third relation) and second-order (fourth relation) reactions. The signs are adjusted for reactants.

3.3 CELL-FIELD DIFFUSION

The cell-field diffusion is the link between the two levels of hierarchy in the simulator, allowing for diffusion of chemicals between cell and fields and between fields. The tissue is split into fields in the simulation and each cell is assigned a field based on their proximity to the field using a clustering approach. The flux between each cell and its associated field is used to compute the flux and update the concentration of the cell and the cumulative flux of the field updates the concentration of the

Simulator	9 Cells	900 Cells	2700 Cells
SERGIO	1.1251 ± 0.0091	112.0576 ± 0.5646	332.7912 ± 3.3610
Nexus-GRN	5.0755 ± 0.1804	5.0956 ± 0.0902	6.3008 ± 0.2558

Table 1: Run time comparison of SERGIO and Nexus-GRN averaged over 5 runs

field. For flux calculation and concentration updating, a similar process as shown in 3.2.3 is used, with the field's flux being calculated as

$$J_i^f = \sum_{j \in \text{contained cells}} -J_{i,j}$$

where $J_{i,j}$ is the flux between the cell j and the field i it is a part of.

4 Results

4.1 GRN Simulator

To validate the accuracy of Nexus-GRN, we compare the computed steady state concentrations with results from SERGIO (Dibaeinia & Sinha, 2020) using identical input configurations. SERGIO expects two input files: one specifying the master regulators and their basal rates, and one specifying the remaining genes, their regulators, and transcription rates. We use the SERGIO input files `Regs_cID_4.txt` (master regulators) and `Interaction_cID_4.txt` (remaining genes). Agreement between GRNSim and SERGIO is assessed using the NumPy routine `allclose()`, which tests element-wise equality within given relative and absolute tolerances; a formal definition is given in Appendix A.4. Both simulators are run on the same inputs for 9 and 2700 cells, and the outputs match when `allclose()` is called with relative tolerance 10^{-6} and absolute tolerance 10^{-32}.

Nexus demonstrates significant computational advantages over SERGIO when simulating RNA concentrations, particularly for large cell populations. Table 1 presents runtime comparisons between SERGIO and Nexus-GRN across different cell population sizes.

While Nexus exhibits higher overhead for small cell populations due to vectorization, just-in-time compilation, and initialization costs, it achieves substantial speedups for large-scale simulations. These performance gains result from JAX's automatic vectorization and just-in-time compilation capabilities, which become increasingly beneficial as the number of cells grows.

For verifying the protein simulator, we use real-world data from Beyer et al. (Beyer et al., 2004) containing the mRNA and protein concentrations, protein transcriptional rates, and the half-lives of approximately 15 proteins from yeast. The gene interaction network is taken from the Saccharomyces Genome Database (SGD) (Cherry et al., 1997; Engel et al., 2025). The resultant network has 1913 nodes, each representing a pair of RNA-protein, with 56283 edges representing the regulator-target links. As we do not have data about the decay rates, the regulatory contributions of genes (K_{ij}), protein decay rates and other variables, we run an initial experiment where we initialize 100 cells and set these values randomly for each cell. To identify the cell with the closest output to the true values, we compute the MSE loss of the predicted values of each cell with the data collected by Beyer et al.(Beyer et al., 2004). The concentrations are normalized independently before being used in the comparisons. In the best performing cell, the MSE loss is 1.8499368 on the steady state concentration of the RNA and 0.28618234 on the protein subset with known half lives.

To improve the fit of the output values, A few parameters for the Nexus-GRN simulator is modified to be closer to the real world values. Specifically, the decay rates for the genes are set to a uniform distribution with a mean of $5.6e-4$ and a standard deviation of $1e-4$ to be closer to the real-world decay rate of yeast BioNumbers Database. We evaluate different ranges for other parameters and find that sampling basal rates within $[1, 10]$ and sampling regulatory influence of genes, K_i, within $[-2, 2]$ gives the lowest loss on the RNA. We identify the cell that has the lowest MSE loss over the proteins whose half lives are known and compute the MSE losses. The loss over RNA is 1.8746623,

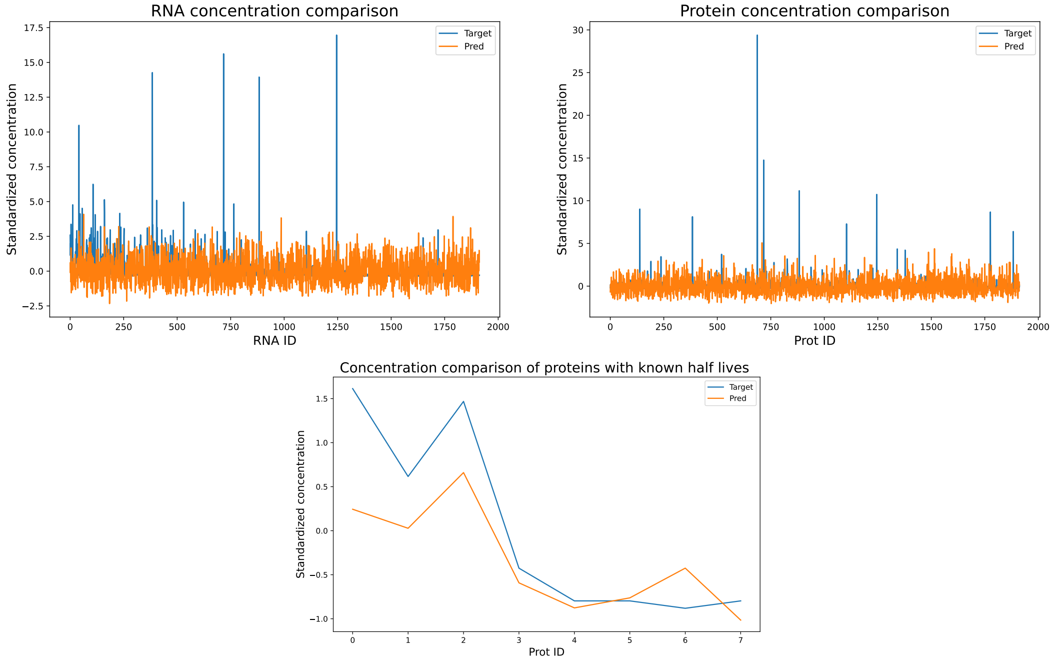


Figure 1: Visualization of chemical concentration during cell-field diffusion with random initialization of protein concentration

over protein is 2.025427 and across the proteins with known half lives is 0.14554834. The plots are shown in Figure 1.

There is a slight increase in the RNA loss in the second cell as we pick the cell with the lowest MSE loss over the protein subset. In both the cases, there is an elevated MSE loss as many simulation parameters are unknown and are hard to estimate despite knowing the possible range of parameters. We notice a lower MSE loss on the proteins with known half-lives as it did not require an estimation of the parameters. Despite these limitations, NEXUS demonstrates its ability to generate simulations that closely approximate real-world biological data.

4.2 SPATIAL SIMULATOR

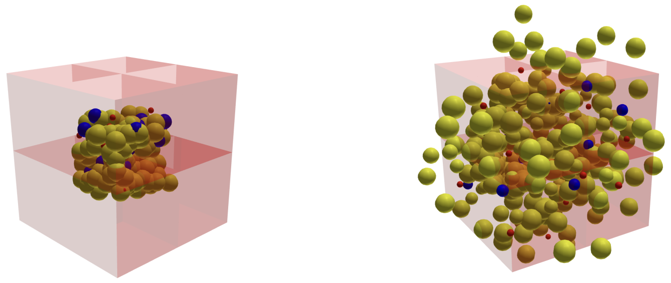
The validation of the spatial simulator output is performed by inspecting the positions of the cell and studying the behaviour of the cells as a result of the forces and cell cycling. By selecting the appropriate parameters for attraction and repulsion forces, the simulator can reproduce multiple cell arrangements ranging from loose to tightly packed configurations, as shown in Figure 2.

In the visualization, the live cells are shown in yellow, the cells undergoing programmed cell death are shown in blue and the cell that have undergone necrosis are shown in red. The radius of the sphere is the simulated radius of the cell and changes size as the cell cycles through different states. The red segments represent the fields arranged in a grid that compute chemical concentration as a result of diffusion between adjacent fields and chemical reactions inside the field.

Running the simulation with cell death and cell splitting disabled, the arrangement reaches a stable configuration with minimal movement of the cells. The equilibrium state demonstrates the convergence of the cell positions to a layout that is stable when being acted upon the inter-cellular forces and can be useful to study the effect of cell splitting or cell death on cell arrangements.

4.3 CELL-FIELD DIFFUSION

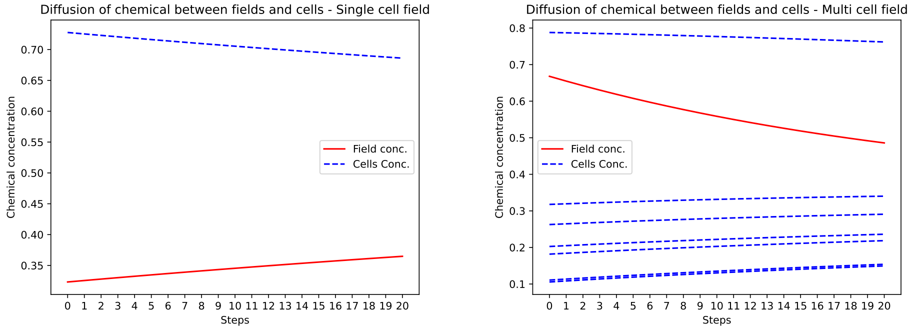
One of the core objectives of NEXUS is to perform a multi-scale simulation that integrates across different levels. The cell-field diffusion is one of the mechanisms that demonstrates this integration,



(a) Strong inter-cellular attraction

(b) Weak inter-cellular attraction

Figure 2: Visualization of simulator outputs with different magnitude of inter-cellular forces



(a) Single cell field

(b) Multi cell field

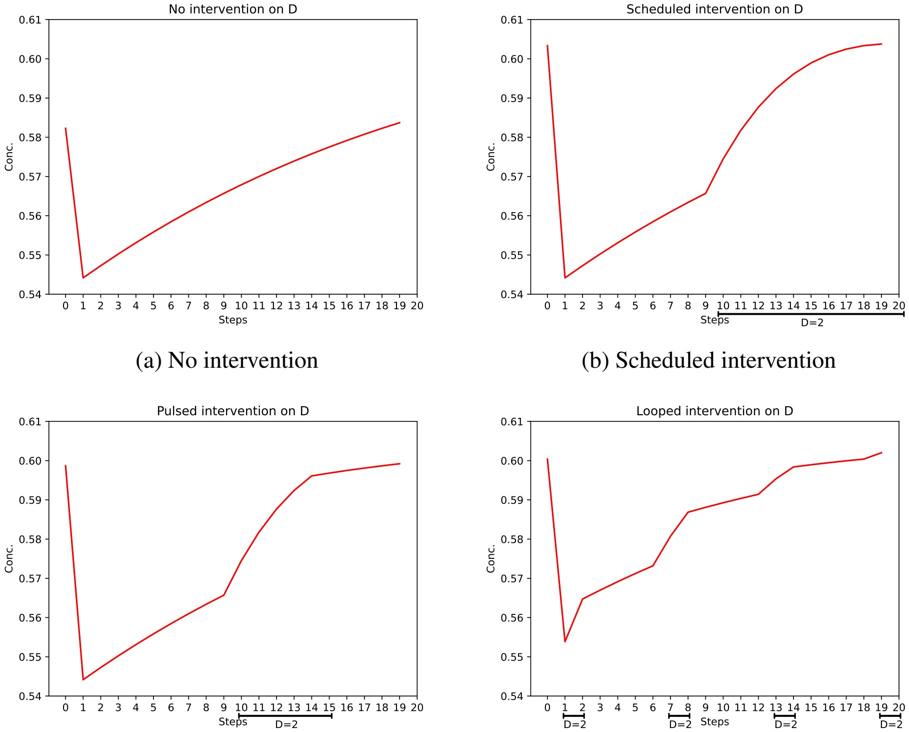
Figure 3: Visualization of chemical concentration during cell-field diffusion with random initialization of protein concentration

allowing for movement of chemicals, specifically proteins in case of cell-field diffusion, across the two levels. To better study the working of this behaviour, cell movement is disabled to prevent a cell from being assigned to different fields during the simulation and inter-field diffusion and reactions are disabled to prevent modification of the chemical concentrations. The concentration of a single chemical in the field and its associated cells across the simulation run are plotted with the protein concentrations initialized to random values for better visualization. Figure 3a shows the concentration of a field with a single cell inside it while Figure 3b shows the concentration of a field and the multiple cells within. In Figure 3a, the concentration of the chemical is lower in the field than in the cell and therefore we observe a gradual reduction of the concentration in the cell as it is moving to the field.

These demonstrate the link between the cellular and sub-cellular models, where chemicals can diffuse between the cells and the field. This allows chemicals to diffuse out of a cell and into a field from where it can diffuse into another cell. Similarly, the chemical can diffuse between fields and into a cell in a different field and is not limited to moving within the same field.

4.4 Intervention experiments

NEXUS supports interventions on model parameters, enabling systematic study of how such changes affect system outputs. Crucially, these are *mechanism-changing* interventions; rather than merely setting a variable to a fixed value (as in the do-calculus), they modify the dynamical mechanism



(a) No intervention

(b) Scheduled intervention

(c) Pulsed intervention

(d) Looped intervention

Figure 4: Interventions supported by NEXUS

that governs the evolution of the system. This distinction is central in real-world applications. For instance, in biology and pharmacology, an intervention such as a drug does not simply fix a state variable but alters the underlying dynamics (e.g., reaction rates or transport coefficients). NEXUS implements several temporal intervention profiles. The *scheduled* profile applies a parameter change at a specified time-step, *pulse* holds the change between given start and end time-steps, and *loop* applies the change repeatedly with configurable start time, period, and inter-loop gap. Figure 4 illustrates how interventions on the diffusion constant D alter the trajectory of a chemical's concentration.

5 DISCUSSION

This work introduces NEXUS, a hierarchical biological simulator using a dynamical systems approach to generate realistic data for causal reasoning. The subcellular simulator supports multiple cell types with distinct parameters for movement and cell cycling. Validation shows RNA concentrations match SERGIO (Dibaeinia & Sinha, 2020) outputs, while protein concentrations align with real-world yeast data (Beyer et al., 2004). NEXUS simulates cell arrangement, movement, and cycling, enabling data generation for causal discovery algorithm evaluation and targeted intervention studies.

NEXUS leverages auto-vectorization and just-in-time compilation for performance, enabling the subcellular simulator to scale better than SERGIO for large cell populations and handle large-scale simulations efficiently. Future work will integrate tissue-level chemical dynamics with subcellular processes, improve the intervention API for runtime parameter changes, enable parameter learning from experimental data via backpropagation to generate targeted perturbation datasets, and expand the available ODE solvers beyond the current implementation.

REFERENCES

Andreas Beyer, Jens Hollunder, Heinz-Peter Nasheuer, and Thomas Wilhelm. Post-transcriptional Expression Regulation in the Yeast Saccharomyces cerevisiae on a Genomic Scale *. *Molecular & Cellular Proteomics*, 3(11):1083–1092, November 2004. ISSN 1535-9476, 1535-9484. doi: 10.1074/mcp.M400099-MCP200. URL `https://www.mcponline.org/article/S1535-9476(20)33117-0/abstract`. Publisher: Elsevier.

BioNumbers Database. Median mRNA decay constant - Budding yeast Saccharomyces ce - BNID 105510 — bionumbers.hms.harvard.edu. URL `https://bionumbers.hms.harvard.edu/bionumber.aspx?id=105510&ver=5&trm={S}accharomyces+cerevisiae+decay&org=`.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/jax-ml/jax`.

Chris Bradley, Andy Bowery, Randall Britten, Vincent Budelmann, Oscar Camara, Richard Christie, Andrew Cookson, Alejandro F. Frangi, Thiranja Babarenda Gamage, Thomas Heidlauf, Sebastian Krittian, David Ladd, Caton Little, Kumar Mithraratne, Martyn Nash, David Nickerson, Poul Nielsen, Øyvind Nordbø, Stig Omholt, Ali Pashaei, David Paterson, Vijayaraghavan Rajagopal, Adam Reeve, Oliver Röhrle, Soroush Safaei, Rafael Sebastián, Martin Steghöfer, Tim Wu, Ting Yu, Heye Zhang, and Peter Hunter. OpenCMISS: A multi-physics & multiscale computational infrastructure for the VPH/Physiome project. *Progress in Biophysics and Molecular Biology*, 107(1):32–47, October 2011. ISSN 0079-6107. doi: 10.1016/j.pbiomolbio.2011.06.015. URL `https://www.sciencedirect.com/science/article/pii/S0079610711000629`.

Lukas Breitwieser, Ahmad Hesam, Jean de Montigny, Vasileios Vavourakis, Alexandros Iosif, Jack Jennings, Marcus Kaiser, Marco Manca, Alberto Di Meglio, Zaid Al-Ars, Fons Rademakers, Onur Mutlu, and Roman Bauer. BioDynaMo: a modular platform for high-performance agent-based simulation. *Bioinformatics*, 38(2):453–460, January 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab649. URL `https://doi.org/10.1093/bioinformatics/btab649`.

J. Michael Cherry, Catherine Ball, Shuai Weng, Gail Juvik, Rita Schmidt, Caroline Adler, Barbara Dunn, Selina Dwight, Linda Riles, Robert K. Mortimer, and David Botstein. Genetic and physical maps of Saccharomyces cerevisiae. *Nature*, 387(6632 Suppl):67–73, May 1997. ISSN 0028-0836. URL `https://pmc.ncbi.nlm.nih.gov/articles/PMC3057085/`.

Mathieu Chevalley, Jacob Sackett-Sanders, Yusuf Roohani, Pascal Notin, Artemy Bakulin, Dariusz Brzezinski, Kaiwen Deng, Yuanfang Guan, Justin Hong, Michael Ibrahim, et al. The causalbench challenge: A machine learning contest for gene network inference from single-cell perturbation data. *arXiv preprint arXiv:2308.15395*, 2023.

Mathieu Chevalley, Patrick Schwab, and Arash Mehrjou. Deriving causal order from single-variable interventions: Guarantees & algorithm. *arXiv preprint arXiv:2405.18314*, 2024.

Mathieu Chevalley, Arash Mehrjou, and Patrick Schwab. Theoretical guarantees for causal discovery on large random graphs. *arXiv preprint arXiv:2511.02536*, 2025a.

Mathieu Chevalley, Yusuf H Roohani, Arash Mehrjou, Jure Leskovec, and Patrick Schwab. A large-scale benchmark for network inference from single-cell perturbation data. *Communications Biology*, 8(1):412, 2025b.

James Clarke, Jake McGrath, Colin Johnson, and José Alvarado. Control across scales: signals, information, and adaptive biological mechanical function. *arXiv preprint arXiv:2509.03418*, 2025.

Fergus R. Cooper, Ruth E. Baker, Miguel O. Bernabeu, Rafel Bordas, Louise Bowler, Alfonso Bueno-Orovio, Helen M. Byrne, Valentina Carapella, Louie Cardone-Noott, Jonathan Cooper, Sara Dutta, Benjamin D. Evans, Alexander G. Fletcher, James A. Grogan, Wenxian Guo, Daniel G. Harvey, Maurice Hendrix, David Kay, Jochen Kursawe, Philip K. Maini, Beth McMillan, Gary R. Mirams, James M. Osborne, Pras Pathmanathan, Joe M. Pitt-Francis, Martin Robinson, Blanca Rodriguez, Raymond J. Spiteri, and David J. Gavaghan. Chaste: Cancer, Heart and Soft Tissue Environment. *Journal of Open Source Software*, 5(47):1848, March 2020. ISSN 2475-9066. doi: 10.21105/joss.01848. URL https://joss.theoj.org/papers/10.21105/joss.01848.

Nima Dehghani. Theoretical principles of multiscale spatiotemporal control of neuronal networks: a complex systems perspective. *Frontiers in Computational Neuroscience*, 12:81, 2018.

Payam Dibaeinia and Saurabh Sinha. SERGIO: A Single-Cell Expression Simulator Guided by Gene Regulatory Networks. *Cell Systems*, 11(3):252–271.e11, September 2020. ISSN 2405-4712. doi: 10.1016/j.cels.2020.08.003. URL https://www.sciencedirect.com/science/article/pii/S2405471220302878.

Stacia R. Engel, Suzi Aleksander, Robert S. Nash, Edith D. Wong, Shuai Weng, Stuart R. Miyasato, Gavin Sherlock, and J. Michael Cherry. Saccharomyces Genome Database: advances in genome annotation, expanded biochemical pathways, and other key enhancements. *Genetics*, 229(3): iyae185, March 2025. ISSN 1943-2631. doi: 10.1093/genetics/iyae185.

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. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005991.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Seunghwa Kang, Simon Kahan, Jason McDermott, Nicholas Flann, and Ilya Shmulevich. Biocellion: accelerating computer simulation of multicellular biological system models. *Bioinformatics*, 30(21):3101–3108, November 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu498. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC4609016/.

Krzysztof Kuchta, Joanna Towpik, Anna Biernacka, Jan Kutner, Andrzej Kudlicki, Krzysztof Ginalski, and Maga Rowicka. Predicting proteome dynamics using gene expression data. *Scientific Reports*, 8(1):13866, September 2018. ISSN 2045-2322. doi: 10.1038/s41598-018-31752-4. URL https://www.nature.com/articles/s41598-018-31752-4. Publisher: Nature Publishing Group.

Hechen Li, Ziqi Zhang, Michael Squires, Xi Chen, and Xiuwei Zhang. scMultiSim: simulation of single-cell multi-omics and spatial data guided by gene regulatory networks and cell–cell interactions. *Nature Methods*, 22(5):982–993, May 2025. ISSN 1548-7105. doi: 10.1038/s41592-025-02651-0. URL https://www.nature.com/articles/s41592-025-02651-0.

Augustin Luna, Metin Can Siper, Anil Korkut, Funda Durupinar, Ugur Dogrusoz, Joseph E Aslan, Chris Sander, Emek Demir, and Ozgun Babur. Analyzing causal relationships in proteomic profiles using causalpath. *STAR protocols*, 2(4), 2021.

Arash Mehrjou. Attractome: From theory to generative models for cancer dynamics and control. In *ICLR 2026 Workshop on Machine Learning for Genomics Explorations*.

Arash Mehrjou, Ashkan Soleymani, Andrew Jesson, Pascal Notin, Yarin Gal, Stefan Bauer, and Patrick Schwab. Genedisco: A benchmark for experimental design in drug discovery. 2021.

Hossam Nada, Yongseok Choi, Sungdo Kim, Kwon Su Jeong, Nicholas A Meanwell, and Kyeong Lee. New insights into protein–protein interaction modulators in drug discovery and therapeutic advance. *Signal transduction and targeted therapy*, 9(1):341, 2024.

Haluk Resat, Michelle N Costa, and Harish Shankaran. Spatial aspects in biological system simulations. In *Methods in enzymology*, volume 487, pp. 485–511. Elsevier, 2011.

Jonathan Somer, Shie Mannor, and Uri Alon. Temporal tissue dynamics from a spatial snapshot. *Nature*, pp. 1–10, 2026.

Jörn Starruß, Walter de Back, Lutz Brusch, and Andreas Deutsch. Morpheus: a user-friendly modeling environment for multiscale and multicellular systems biology. *Bioinformatics*, 30 (9):1331–1332, May 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt772. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3998129/.

Maciej H. Swat, Gilberto L. Thomas, Julio M. Belmonte, Abbas Shirinifard, Dimitrij Hmeljak, and James A. Glazier. Multi-Scale Modeling of Tissues Using CompuCell3D. *Methods in cell biology*, 110:325–366, 2012. ISSN 0091-679X. doi: 10.1016/B978-0-12-388403-9.00013-8. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3612985/.

Alex Eric Yuan and Wenying Shou. Data-driven causal analysis of observational biological time series. *Elife*, 11:e72518, 2022.

Frederic Zubler and Rodney Douglas. A framework for modeling the growth and development of neurons and networks. *Frontiers in Computational Neuroscience*, 3, November 2009. ISSN 1662-5188. doi: 10.3389/neuro.10.025.2009. URL https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/neuro.10.025.2009/full.

A Appendix

A.1 Implementation Details

NEXUS is implemented in Python using the JAX framework Bradbury et al. (2018), which provides automatic vectorization, just-in-time compilation, and automatic differentiation capabilities. These features enable both computational acceleration and gradient-based parameter learning. Configuration management is handled through Hydra, facilitating flexible simulator configuration and composition of multiple configuration files. Simulation outputs are stored using TileDB, a database system optimized for multi-dimensional arrays that supports chunking, compression, indexing, and parallel read-write operations.

The stored simulation data can be analyzed by accessing the TileDB files directly or visualized through an interactive React-based web server that uses Deck.gl to visualize the spatial locations of the cell and their movement during the simulation, the concentrations of the fields and the sub-cellular RNA and protein concentrations. The visualization interface enables users to explore data for selected cells or chemical species across different time points and spatial locations.

A.2 Calculation of inter-cellular forces

The inter-cellular attraction and repulsion forces are computed as

$$\bar{f}^a_{i,j} = \text{pos}_i - \text{pos}_j, \quad \bar{f}^r_{i,j} = -\bar{f}^a_{i,j}$$

$$f^r_{i,j} = \frac{2.0 \cdot [\text{Pos}(i) - \text{Pos}(j)]}{\Delta} \cdot [M(i) + M(j)]$$

$$f^a_{i,j} = \frac{M(i) \cdot M(j)}{\|\text{Pos}(i) - \text{Pos}(j)\|^2}$$

where $\bar{f}^a_{i,j}$ and $\bar{f}^r_{i,j}$ are attraction and repulsion force vectors with magnitudes $f^a_{i,j}$ and $f^r_{i,j}$, $M(i)$ and $\text{Pos}(i)$ are the mass and position of cell i.

A.3 Computation of diffusion

The concentration and flux are computed as:

$$C_1 = M_1/V_1, \quad C_2 = M_2/V_2$$
$$J = -D\nabla C \approx \frac{|C_2 - C_1|}{\|\mathrm{pos}_2 - \mathrm{pos}_1\|}$$

where D is the diffusion constant. The mass update preserves total mass:

$$\Delta M = J_{i\to j} A\Delta t = -D\nabla C \cdot A\Delta t$$
$$M^i_{t+\Delta t} = M^i_t - \Delta M, \quad M^j_{t+\Delta t} = M^j_t + \Delta M$$
$$M^j_t + M^i_t = M^j_{t+\Delta t} + M^i_{t+\Delta t} = M^j_0 + M^i_0.$$

The discrete diffusion update is

$$J_{i,t} = \sum_{j \in \text{neighbors}} -P_{ij}[C_{j,t} - C_{i,t}]$$
$$\Delta C_{i,t} = -\nabla \cdot J_{i,t}\Delta t$$
$$C_{i,t+\Delta t} = C_{i,t} + \Delta C_{i,t},$$

with $J_{i\to j,t} = -J_{j\to i,t}$ and $P_{i,j} = P_{j,i}$.

A.4 Working of $allclose()$

The $allclose()$ function in Numpy Harris et al. (2020) checks if two arrays are equal element-wise within a specified absolute and relative tolerance. For two arrays to be equal given an absolute tolerance, $atol$, and a relative tolerance, $rtol$, they must satisfy the condition in Equation 6.

$$|a - b| \le atol + rtol * |b| \tag{6}$$

A.5 Comparison of Simulators

This section provides a detailed comparison of Nexus with existing biological simulators across key capabilities. Table 2 summarizes the comparison; we discuss each simulator category and their relationship to Nexus below.

Subcellular simulators. SERGIO (Dibaeinia & Sinha, 2020) focuses exclusively on gene regulatory network dynamics at the subcellular level, modeling RNA concentrations through stochastic differential equations guided by user-provided regulatory networks. While SERGIO is interpretable, works out-of-the-box, and can incorporate biological or technical noise to match experimental instrument outputs, it operates solely at the subcellular scale. SERGIO lacks multi-scale integration with tissue-level processes, does not model protein dynamics, provides no support for causal discovery workflows, lacks backpropagation capabilities for parameter learning, and does not scale efficiently to large cell populations. Unlike Nexus, SERGIO cannot integrate subcellular gene expression with spatial cell arrangements or chemical diffusion processes.

Multi-scale cellular simulators. Several simulators address cellular and tissue-level dynamics but lack detailed subcellular modeling. PhysiCell (Ghaffarizadeh et al., 2018) combines agent-based cell modeling with PDE-based diffusion and reactions, supporting multi-scale phenomena through cell cycling, apoptosis, necrosis, volume regulation, and motility. PhysiCell tracks detailed cellular properties including fluid volumes, solid volumes, and nuclear-cytoplasmic partitioning, with cell state transitions modeled as Markov chains. However, PhysiCell lacks subcellular gene regulatory network modeling, causal discovery tools, and backpropagation support, limiting its utility for causal analysis and parameter learning from experimental data. CompuCell3D (Swat et al., 2012) uses the Glazier-Graner model for multi-scale tissue simulation and offers extensibility through its plugin architecture, but similarly lacks subcellular detail and causal analysis capabilities. OpenCMISS (Bradley et al., 2011) employs PDE-based approaches for multi-scale modeling of physiological systems but does not integrate subcellular processes or support causal discovery workflows. While these tools excel at tissue-level dynamics, they cannot model the gene regulatory networks and protein dynamics that NEXUS captures at the subcellular level.

Single-scale cellular simulators. Several tools focus on cellular dynamics without comprehensive multi-scale integration. Biocellion (Kang et al., 2014) provides scalable agent-based simulation with parallel computing support, enabling efficient simulation of large cell populations through high-performance computing frameworks. However, Biocellion operates primarily at the cellular level without subcellular gene regulatory network detail. BioDynaMo (Breitwieser et al., 2022) offers agent-based modeling with some multi-scale capabilities (indicated by • in the table), yet lacks causal discovery and backpropagation support. Chaste (Cooper et al., 2020), Cortex3D (Zubler & Douglas, 2009), and Morpheus (Starruß et al., 2014) are ODE-based simulators that handle cellular-level dynamics; Chaste and Morpheus show partial multi-scale support, but none integrate subcellular gene regulatory networks or provide causal discovery tools. These simulators are well-suited for specific cellular-level questions but cannot address the hierarchical biological processes that NEXUS models across subcellular, cellular, and tissue scales.

NEXUS's unique capabilities. In contrast to the simulators above, NEXUS uniquely combines subcellular gene regulatory network modeling with cellular and tissue-level dynamics in a unified hierarchical framework. Unlike SERGIO, NEXUS integrates protein dynamics and spatial processes, enabling simulation of how gene expression influences cell behavior and tissue organization. Unlike PhysiCell and CompuCell3D, NEXUS includes detailed subcellular gene expression modeling, allowing researchers to study how regulatory networks drive cellular phenotypes. Critically, NEXUS is the only simulator in this comparison that supports causal discovery workflows (indicated by • in the table), enabling systematic generation of interventional datasets and analysis of causal relationships. NEXUS's backpropagation support enables parameter learning from experimental data, while its deep learning integration facilitates seamless connection with modern ML pipelines. NEXUS also supports modeling of feedback loops between scales, is highly extensible through its modular architecture, and scales efficiently to large cell populations through JAX-based auto-vectorization and just-in-time compilation. These capabilities position NEXUS as a comprehensive tool for both realistic data generation and causal analysis in biological systems, addressing gaps that existing simulators cannot fill.

Simulator	**Simulator Type**	Multi Scale	Causal Discovery	Modeling Loops	Backprop support	Interpretable	Works out-of-the-box	DL Integration	Easy to extend	Scalable
SERGIO	SDE-based GRN simulator	✗	✗	✗	✗	✓	✓	✗	✗	✗
BioDynaMo	Agent-based	●	✗	✗	✗	✓	✗	✗	✗	✓
PhysiCell	Agent + PDE	✓	✗	✗	✗	✓	✗	✗	✗	✓
CompuCell3D	Glazier-Graner model	✓	✗	✗	✗	✓	✗	✗	✓	✗
Chaste	ODE-based	●	✗	✗	✗	✓	✓	✗	✗	✓
Cortex3D	ODE-based	✗	✗	✗	✗	✓	✓	✗	✗	✓
Morpheus	ODE-based	●	✗	✗	✗	✓	✓	✗	✗	✗
Biocellion	Agent-based	●	✗	✗	✗	✓	✗	✗	✗	✓
OpenCMISS	PDE-based	✓	✗	✗	✗	✓	✓	✗	✗	✓
NEXUS	Multi-scale ODE-based sim	✓	●	✓	✓	✓	✓	●	✓	✓

Table 2: Comparison of the capabilities of different simulators