PLOS Computational Biology

SOFTWARE

AgentBasedModeling.jl: A tool for stochastic simulation of structured population dynamics

Paul Piho, Philipp Thomas*

Department of Mathematics, Imperial College London, London, United Kingdom

* p.thomas@imperial.ac.uk

Abstract

Agent-based modeling is a powerful approach for understanding cellular systems. Yet, many existing frameworks treat cell-level and population behaviors separately, overlooking their interplay. We present AgentBasedModeling.jl, a Julia package for simulating stochastic, continuous-time agent-based models that integrate intracellular processes with population dynamics. In our stochastic framework, agents evolve according to general continuous-time jump-diffusion dynamics and interact via continuous-rate jump processes. It supports flexible specification of the underlying measure-valued process of intracellular reaction networks and cell growth, capturing events such as cell division, death, intercellular communication, and environmental interactions. We demonstrate the use of the package by validating it on models of stochastic gene expression in growing cells and providing new insights into cell-cell communication and stochastic phage infection. Our tool provides a flexible and efficient platform for exploring how single-cell stochasticity drives emergent population-level behaviors in structured biological systems.

Citation: Piho P, Thomas P (2026) AgentBasedModeling.jl: A tool for stochastic simulation of structured population dynamics. PLoS Comput Biol 22(7): e1014366. https://doi.org/10.1371/journal.pcbi.1014366

Editor: Qing Nie, University of California Irvine, UNITED STATES OF AMERICA

Received: September 15, 2025

Accepted: May 27, 2026

Published: July 29, 2026

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pcbi.1014366

1. Introduction

Interacting autonomous agents, such as living cells, can exhibit complex dynamics requiring extensive computer simulation. The diversity and emergent phenomena observed in cell populations arise from the interactions among agents with heterogeneous characteristics, known as population structure. Such structures present states that manifest in the composition of cells, including DNA, proteins, metabolites, and the overall cell physiology, which encompasses cell age, size, and growth rate, all of which are responsive to external signals and subject to internal variations. Agent-based approaches in cellular systems range from gene expression [1–4] to cell populations, multi-cellular organisms [5], cancer [6–9] and immune cell interactions, and extend to complex systems such as individual-based models of epidemics [10–13].

Data availability statement: AgentBasedModeling.jl is a Julia package and available with the source code and usage examples at https://github.com/pihop/AgentBasedModeling.jl.

Funding: UKRI supported this work through a Future Leaders Fellowship (MR/T018429/1 to PT). The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Simulation methods for cellular systems are gaining momentum as more data combine with computing power, for example, to enable simulation-based inferences [14–17]. While many tools exist for constructing and simulating agent-based models, such as *Agents.jl* [18], MESA [19], Repast Simphony [20] and NetLogo [21], these tools are often geared towards spatial structure. Their limitation lies in their focus on incremental, time-stepped progression or probabilistically distributed event times, which require cellular states not to change between events, as assumed in Gillespie's algorithm [22]. Living cells rarely meet such conditions. When cell states are dynamic, these methods do not sample the underlying stochastic process exactly and are thus not universally valid for cellular population dynamics.

Existing approaches overcome these limitations by discretizing state evolution into a series of steps or pseudo-compartments [23–26]. These approaches introduce additional network complexity and approximation errors that can be difficult to control in practice. In the context of structured cell population dynamics that describe agents with different ages, life cycles, genetic differences, or biochemical makeups [27], exact approaches that employ continuous-rate formulations of agent-based systems remain limited [28,29]. Several established cell-based agent-based frameworks, like BSim [30], PhysiCell [31], and Chaste [32], support hybrid models for simulating cells with internal state dynamics governed by ordinary differential equations (ODEs), embedded within spatial environments that include physical constraints and diffusible signals. These tools are particularly suited to modeling bacterial populations, multicellular development, and tumors.

Jump-diffusion processes [33] provide a generic continuous-rate framework for modeling stochastic cellular dynamics implemented in standard packages such as *JumpProcesses.jl* [34] or simulators of individual cells like *SGNS2* [28] and *PyEcoLib* [29]. These tools have not yet been applied to structured cell populations, which have traditionally been described by deterministic partial differential equations [35] or measure-valued stochastic processes [36,37]. In the agent-based world, such modeling considerations have led to the development of tailored frameworks, such as multi-scale models for cell populations [7,30,38], where the intracellular dynamics of cells evolve continuously, specified by ordinary differential equations. However, general and extensible tools for structured populations described by measure-valued stochastic processes, which couple stochastic state dynamics of interacting agents in continuous time, are still limited.

Here, we introduce *AgentBasedModeling.jl*, which enables easy specification and simulation of stochastic agent-based cell models where internal agent dynamics are modeled as jump-diffusion processes and influence population-level interactions (illustrated in Fig 1). We implement an exact simulation algorithm, allowing the rates of interactions between agents to depend on the continuously evolving internal states of the agents. Our tool integrates easily with the existing mathematical modeling tools of *ModelingToolkit.jl* [39] for differential equations and *Catalyst.jl* [40] for stochastic reaction networks, thus providing a flexible and scalable modeling framework for structured cell populations in *Julia*.

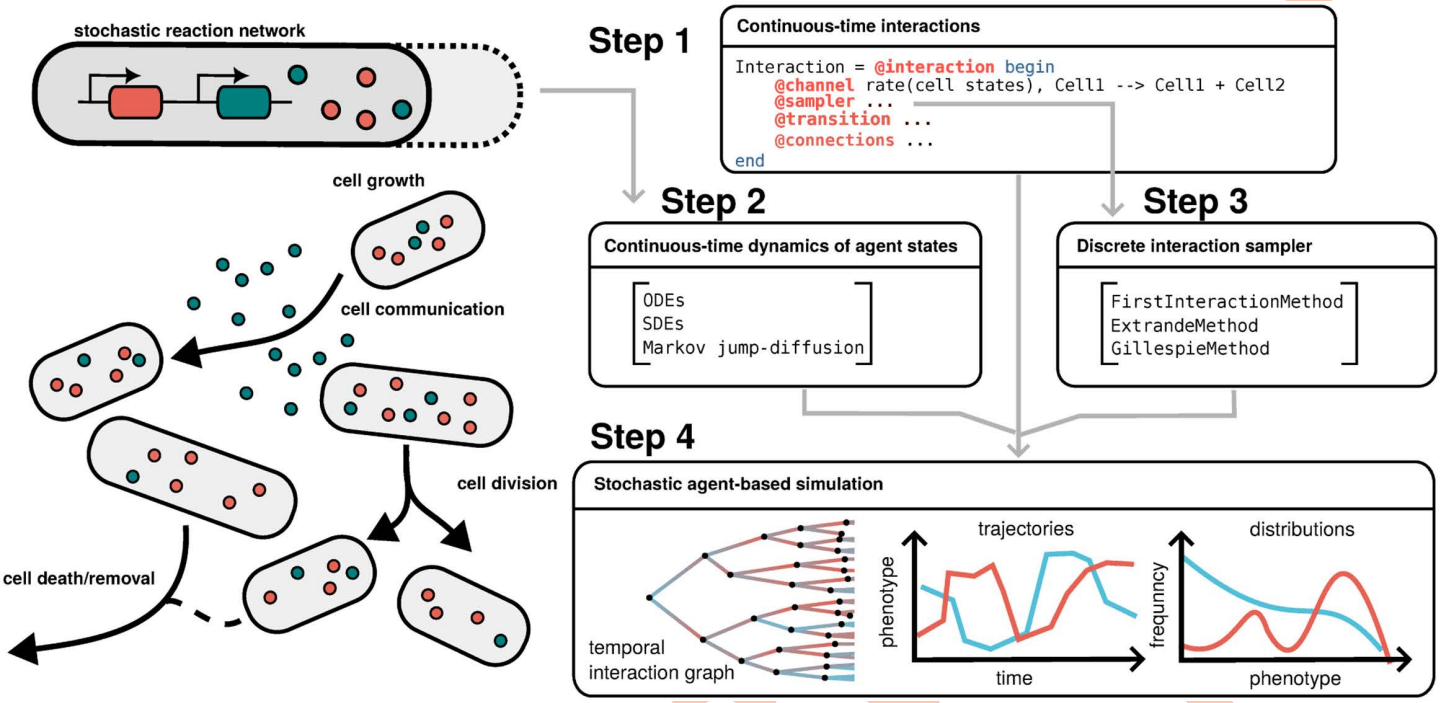

Fig 1. Overview of the AgentBasedModeling.jl workflow. Agent-based models of cells interacting via cell-cell communication, cell division or death. Cell states may arise from the coupling of intracellular reaction networks, cell growth and other modeled factors (left). The package combines cell interactions in continuous time (STEP 1) with exact stochastic sampling of the temporal interactions cells' state dynamics (STEP 2 and 3) and outputs temporal graphs, trajectories or distributions (STEP 4).

https://doi.org/10.1371/journal.pcbi.1014366.g001

2. Design and implementation

2.1. A framework for agent-based modeling of structured cell populations

We consider interactions between agents to depend on the agent type and their internal states that evolve continuously over time. With the i^{th} agent we associate an agent type S_j, where $j \in 1, ..., M$, and a state vector $\mathbf{x}_i(t)$, which is a realization of a Markov jump-diffusion process:

$$\mathrm{d}\mathbf{x}_i(t) = \mathbf{f}_j(\mathbf{x}_i(t), t)\mathrm{d}t + \sigma_j(\mathbf{x}_i(t), t)\mathrm{d}\mathbf{W}_i(t) + \sum_r \nu_{rj}\mathrm{d}Y_{ri}(t), \tag{1}$$

where $\mathbf{f}_j$ is a deterministic drift vector corresponding to ordinary differential equation (ODE); $\sigma_j(\mathbf{x}_i(t))\mathrm{d}\mathbf{W}_i(t)$ is state-dependent Gaussian white noise corresponding to the stochastic differential equation part (SDE); and $\mathrm{d}Y_{ri}$ are Poisson process jumps with path-dependent intensity $\int_0^t \mathrm{d}s w_{rj}(\mathbf{x}_i(s))$ and increments ν_{rj} that define the jump process of agent type S_j.

We define an interaction channel c that takes k_c input agents of types $S_{n_1^c}, ..., S_{n_{k_c}^c}$ with internal states $\mathbf{x}_1^-, ..., \mathbf{x}_{k_c}^-$, which we denote as $S_{n_1^c}[\mathbf{x}_1^-], ..., S_{n_{k_c}^c}[\mathbf{x}_{k_c}^-]$, and creates l_c output agents of types $S_{m_1^c}, ..., S_{m_{l_c}^c}$ with states $\mathbf{x}_1^+, ...\mathbf{x}_{l_c}^+$:

$$\begin{aligned} &S_{n_1^c}[\mathbf{x}_1^-] + ... + S_{n_{k_c}^c}[\mathbf{x}_{k_c}^-] \xrightarrow{r_c\left(\mathbf{x}_1^-,...,\mathbf{x}_{k_c}^-,t\right)} S_{m_1^c}[\mathbf{x}_1^+] + ... + S_{m_{l_c}^c}[\mathbf{x}_{l_c}^+] \\ &\mathbf{x}_1^+, ..., \mathbf{x}_{l_c}^+ \sim B_c(\bullet|\mathbf{x}_1^-, ..., \mathbf{x}_{k_c}^-), \end{aligned} \tag{2}$$

where $n_1^c, ..., n_{k_c}^c, m_1^c, ..., m_{l_c}^c \in \{1, ..., M\}$ are the indices of agent types involved in the interaction. The rate function r_c of the rule is computed using the internal states of the input agents $\mathbf{x}_1^-, ..., \mathbf{x}_{k_c}^-$. The states of the output agents $\mathbf{x}_1^+, ...\mathbf{x}_{l_c}^+$ are initialized probabilistically according to the transition kernel B_c that defines the conditional probability of creating output agent states $\mathbf{x}_1^+, ...\mathbf{x}_{l_c}^+$ given the input agent states $\mathbf{x}_1^-, ...\mathbf{x}_{k_c}^-$.

In simple terms, an interaction channel represents a reaction or rule, analogous to the definition of reactions in biochemical kinetics. However, here the agents' internal states modulate both the interaction rate function and the transition kernel. Thus, mapping a channel to a set of reactions amounts to enumerating all possible agent states, potentially yielding an infinite number of reactions. This map is effectively expressed through the interaction channel analogous to rule-based frameworks [23,41,42]. For our purpose, we consider cells as agents whose populations comprise different cell types. For example, we can describe asymmetric stem cell division of a stem cell S_1 differentiating towards S_2, via the rule $S_1[\mathbf{x}_1^-] \xrightarrow{r_c(\mathbf{x}_1^-)} S_1[\mathbf{x}_1^+] + S_2[\mathbf{x}_2^+]$, which could depend on the cell cycle position $\mathbf{x}_1^+$ of the mother cell. Similarly, we could consider cell-cell interaction via rules such as $S_1[\mathbf{x}_1^-] + S_2[\mathbf{x}_2^-] \xrightarrow{r_c(\mathbf{x}_1^-, \mathbf{x}_2^-)} S_1[\mathbf{x}_1^+] + S_2[\mathbf{x}_2^+]$, where $\mathbf{x}_{1,2}^-$ encode ligand-receptor interactions and $(\mathbf{x}_1^+, \mathbf{x}_2^+) \sim B_c(\cdot|\mathbf{x}_1^-, \mathbf{x}_2^-)$ encodes the resulting transcriptional changes.

2.2. *AgentBasedModeling.jl* facilitates model specification and integrates easily with deterministic and stochastic modeling tools

AgentBasedModeling.jl package allows for convenient model specification and simulation of such models. The first step (Fig 1 Step 1) in the specification concern the definition of interaction channels (Eq 2). These are implemented with the `@interaction` environment with the lines of code following the specification:

$$\texttt{@channel}\ r_c(\mathbf{x}_1^-, ..., \mathbf{x}_{k_c}^-, t),\ S_{n_1^c} + ... + S_{n_{k_c}^c} \to S_{m_1^c} + ... + S_{m_{l_c}^c}$$
$$\texttt{@connection}\ (\mathbf{x}_1^- \Rightarrow \mathbf{x}_1(t), ..., \mathbf{x}_{k_c}^- \Rightarrow \mathbf{x}_{k_c}(t))$$
$$\texttt{@transition}\ (\mathbf{x}_1(t) \Rightarrow \mathbf{x}_1^+, ..., \mathbf{x}_{l_c}(t) \Rightarrow \mathbf{x}_{l_c}^+)$$

where $\mathbf{x}_1^+, ..., \mathbf{x}_{l_c}^+$ are sampled from the transition kernel $B_c(\bullet|\mathbf{x}_1^-, ..., \mathbf{x}_{k_c}^-)$. The `@channel` line defines the rate function r_c of the interaction and the interaction stoichiometry while the `@transition` line defines the initializations of the output agents given the transition kernel B. The `@connections` line is used to indicate which traits of which agents correspond to the symbols used in the expressions. The tuple $(\mathbf{x}_1^- \Rightarrow \mathbf{x}_1(t))$ denotes that the value of vector $\mathbf{x}_1^-$ corresponds to the state $\mathbf{x}_1(t)$ of the first input agent at time t. These rules are matched to the individual instances of agents in the simulation.

The second step (Fig 1, Step 2) in the specification is to define the internal state dynamics for all agent types (Eq 1). This is done as an ODE, SDE, jump process, or a combination of them using *ModelingToolkit.jl* [39] and *Catalyst.jl* [40], leveraging the existing *Julia* programming language ecosystem.

2.3. *AgentBasedModeling.jl* enables exact simulation of stochastic agent-based models

AgentBasedModeling.jl provides a stochastic simulation algorithm to exactly simulate any agent-based model. The outline of the algorithm is given in Algorithm 1 and is based on the first reaction method for simulating Markov jump processes [43]. To sample the next interaction time in a simulation time-interval $[t, t + \Delta t]$, the algorithm computes the trajectories of agent states in that interval. This is dependent on the model given for the agent state dynamics and can, for example, involve solving a system of ODEs, SDEs, or sampling a trajectory of a jump-diffusion process for each agent in the population. We then construct a set of possible interaction instances between agents for each interaction channel. For each channel, the simulation algorithm samples and executes the instance with the fastest interaction time.

Algorithm 1. Stochastic simulator for the population models.

Require: Simulation time-span $[T_0, T]$, lookup horizon Δt.
Let $t \leftarrow T_0$ and $t_w \leftarrow T_0 + \Delta t$ and consider the time window $[t, t_w]$.
Let $\mathcal{S}$ be the initial population of agents at time t defined by pairs of agent types and states at time t.
For all agents in $\mathcal{S}$ simulate the state trajectories in the window $[t, t_w]$.
Let C be a set of interaction channels. For each $c \in C$ with input agent types $S_{n_1^c}, \cdots, S_{n_{k_c}^c}$ construct the set A_c of all without replacement combinations of agents in the population $\mathcal{S}$ that match the input types.
while $t \leq T$ **do**
For all channels c let $f_c(a, s)$ be the interaction rate for a combination of agents a at time $s \in [t, t_w]$ and let $\bar{f}_c(a)$ denote the upper bound of an interaction rate for all $s \in [t, t_w]$. Utilize one the following algorithms: the first reaction method where for each $a \in A_c$ use the thinning algorithm [44] with rate $f_c(a, s)$ and bound $\bar{f}_c(a)$ to sample next interaction times $t_a \in [t, t_w]$ and choose the time t_c and input agents a corresponding to $\text{argmin}_a t_a$; or apply the Extrande algorithm [45] with rate bound $\sum_r \bar{f}_c(a)$ to sample an interaction time t_c and the input agents a.
Find the channel c with the least next interaction time $\hat{t} = \min_c t_c$ and the corresponding combination of input agents a_c.
if $\hat{t} \leq t_w$ **then**
Given the set of input agents $a_c \subset \mathcal{S}$ with states $\mathbf{x}_{n_1^c}(\hat{t}), \ldots, \mathbf{x}_{n_{k_c}^c}(\hat{t})$ at time $\hat{t}$ and output agent types $S_{m_1^c}, \ldots, S_{m_{l_c}^c}$, create output agents with the given types and states $\mathbf{x}^+_{m_1^c}, \ldots, \mathbf{x}^+_{m_{l_c}^c}$ at time $\hat{t}$ sampled from the kernel $B_c(\bullet | \mathbf{x}_{n_1^c}(\hat{t}), \ldots, \mathbf{x}_{n_{k_c}^c}(\hat{t}))$. Remove input agents a_c from $\mathcal{S}$.
Simulate the state trajectories of the output agents in the time interval $[\hat{t}, t_w]$, add the output agents to $\mathcal{S}$, and let $t \leftarrow \hat{t}$.
else
Let $t \leftarrow t_w$, $t_w \leftarrow t + \Delta t$ and simulate the state trajectories of all agents in $\mathcal{S}$ in time window $[t, t_w]$.
end if
end while

The package offers three algorithms for sampling the next interaction instance of a given channel, and different algorithms can be combined within a simulation model for each channel (Fig 1 Step 3). Simulation algorithms are specified within the `@interaction` keyword using the `@sampler` keyword. The first method `FirstInteractionMethod()` samples interaction times for each instance of an interaction using the thinning algorithm [44]. The second algorithm uses the Extrande method [45] to sample the next interaction time for each interaction channel and can be used by specifying `ExtrandeMethod(...)`. An alternative but potentially unchecked approximation to these aforementioned methods is the `GillespieMethod()`.

The algorithms `FirstInteractionMethod(bfn, L)` and `ExtrandeMethod(bfn, L)` require specifying a symbolic function `bfn` returning a constant upper bound in the time interval [*t*, *t*+*L*], where *L* is a user-specified lookup time horizon. For convenience, the bound can be computed automatically using the optional `boundtype` keyword, e.g., `ExtrandeMethod(L; boundtype=unknown)`, which can assume `:increasing`, `:decreasing` or `:unknown` values. In these cases, the interaction rate function is bounded by assuming monotonically increasing or decreasing functions of time, i.e., the bound in a time interval [*t*, *t*+*L*] is the interaction rate at time *t*+*L* or *t*, respectively. When the keyword argument is `:unknown`, which is the default setting, the algorithm computes the upper bound of the interaction rate by evaluating the rate at each timestep in the interval [*t*, *t*+*L*] produced by simulating the agent state dynamics and taking the maximum of the computed rates on the fly.

Although our simulation methods are routed in Markovian dynamics, state-dependent interaction rates generate effective time-dependent event rates, leading to non-exponential waiting-time distributions even when the full process remains Markovian in an appropriately extended state space. In *AgentBasedModeling.jl*, such dynamics can be represented by augmenting agents with internal state variables that modulate event propensities. This provides a natural framework for

modelling history-dependent cellular events, such as size-dependent cell division or infection-age-dependent lysis events (see Section 3).

2.4. *AgentBasedModeling.jl* provides tailored simulation outputs

Agents and their states can be tracked at individual and population levels (Fig 1, Step 4). We distinguish (i) snapshot and (ii) trajectory simulation, (iii) interaction snapshots, and (iv) temporal interaction networks, which we will describe below. Biological data examples of snapshot measurements include cell size, growth, or gene expression obtained from smFISH, flow cytometry, single-cell RNA sequencing, or multi-omics technologies. Trajectories present common measurements in population dynamics such as the microbiome, cancer growth, infections, or drug responses. Interaction snapshots represent agent states sampled at interaction events, such as cell birth, division, or cell-cell interactions. Temporal interaction networks are the most complex output, enabling the tracking of cellular agents and states, such as those observed in time-lapse microscopy or lineage tracing [46]. The handling of these outputs in the package is described below.

A simulation is processed by specifying a model, a set of parameters, and an initial condition. The simplest output is a snapshot of the state of the population at the simulation endpoint, which we denote as a **snapshot simulation**:

```
result = simulate(model, init, params)
result.final
```

which requires specification of a `model`, initial conditions `init` and parameters `params`. Furthermore, the package can provide **trajectory simulation** representing a series of population snapshots at equidistant time points. These can represent counts of agents and types or snapshots across the agents' states. For example, for an agent A with state s this can be obtained through using

```
params = SimulationParameters(...,Δ;t,...;  snapshot = [PopulationSnapshot(A), StateSnapshot(A,s),...])
```

where `PopulationSnapshot(A)` counts agents of type A and `StateSnapshot(A, s)` provides a snapshot of the agent state every timestep Δt.

Other outputs concern **interaction snapshots** that track states of the input and output agents of specified interactions when events occur. These can be accessed by selecting the states in the `@interaction` environment using `@saveinstate` and `@saveoutstate` Keywords:

$$\texttt{@saveinstate}\ (S_{n_1}, x_1)\quad (S_{n_2}, x_2)\quad ...(S_{n_k}, x_k)$$
$$\texttt{@saveoutstate}\ (S_{m_1}, x_1)\quad (S_{m_2}, x_2)\quad ...(S_{m_l}, x_l)$$

Finally, the most complex output is a **temporal interaction graph** that represents a function-valued directed graph whose edges represent trajectories of agent states and vertices of sampled interactions between these agents:

```
vertices = result.interactions
edges = result.agents.
```

As the tool is implemented as a package for the *Julia* language, we can utilize existing statistical and visualization packages for analyses. For example, visualisation of such a graph can be conveniently implemented using *MetaGraphsNext.jl* from the *Graphs.jl* ecosystem.

3. Results

We demonstrate the use and functionality of *AgentBasedModeling.jl* with three examples of varying complexity. We first provide a hands-on, step-by-step example of gene expression model with a simple interaction structure and complex state models, coupling intracellular reaction networks with cell size, growth and cell division. We then investigate how cell-to-cell communication with feedback can lead to population-level synchronous state switching. Finally, we investigate the dynamics of infection-lysis by bacteriophages as a stochastic extinction event.

3.1. Stochastic gene expression is coupled to growth and division

We consider a stochastic agent-based model of gene expression coupled to cell growth (Fig 2a) using a combination of *ModelingToolkit.jl*, *Catalyst.jl* and *AgentBasedModeling.jl* (S1 Code). Such models explain concentration homeostasis [47–49] and account for extrinsic noise in cells [17,50,51]. We first consider the intracellular dynamics of cells whose state is given by cell size s, added size Δ, and protein count p that are specified using the `@abm_variables` keyword. Cell size s increases deterministically with exponential growth rate α and can be modeled by an ODE. Proteins p are expressed in geometrically distributed bursts m modeled by a Markov jump process. The `@reaction_network` keyword provided by *Catalyst.jl* combines these dynamics, which are then passed to `AgentDynamics` structure from *AgentBasedModeling.jl* as shown in Step 3.1.

Step 1: Stochastic gene expression with cell division and growth

```
using Catalyst, Distributions, AgentBasedModeling
@variables t
@abm_variables Δ(t) s(t) p(t)
@parameters α kprod b
D=Differential(t)
@register_symbolic Distributions.Geometric(a)
m=rand(Distributions.Geometric(1/(1+b*s)))
CellDynamics=@reaction_network begin
  @species p(t)
  @variables Δ(t) s(t)
  @equations begin
    D(s) $α*s
    D(Δ) $α*s
  end
  kprod, 0 -->$m*p
end
Cell=AgentDynamics(CellDynamics,())
```

Cells divide via the adder mechanism, where a stochastic size increment is added between divisions, as observed in many cell types. In Step 3.1, we implement the interaction channel via the `@interaction` keyword as follows. First, we assign variables Δ, s, p to the agent states via the `@connections` keyword. The interaction channel can be written as

$$\text{Cell}[\Delta, s, p] \xrightarrow{\gamma_{\text{division}}(s,\Delta,\alpha)} \text{Cell}[0, s_1^+, p_1^+] + \text{Cell}[0, s_2^+, p_2^+], \tag{3}$$

$$s_1^+ \sim \beta s, \quad s_2^+ \sim (1-\beta)s, \quad p_1^+ \sim \text{Binomial}(\beta, p), \quad p_2^+ \sim p - p_1^+. \tag{4}$$

with a size-dependent division rate

$$\gamma_{\text{division}}(s, \Delta, \alpha) = \underbrace{\alpha s}_{\text{growth}} \underbrace{\gamma_{\text{size}}(\Delta)}_{\text{cell size control}}, \tag{5}$$

where $\gamma_{\text{size}}(\Delta)$ denotes the cell size control (see S1 Code) and $\beta \sim \text{Beta}()$ denotes variability in the placement of the division site [52]. The keyword `@channel` specifies the channel stoichiometry and the division rate. During cell division, cells are split, and proteins are partitioned equally between the two daughter cells. The transition kernel $B(s_1, p_1, s_2, p_2|s, p)$

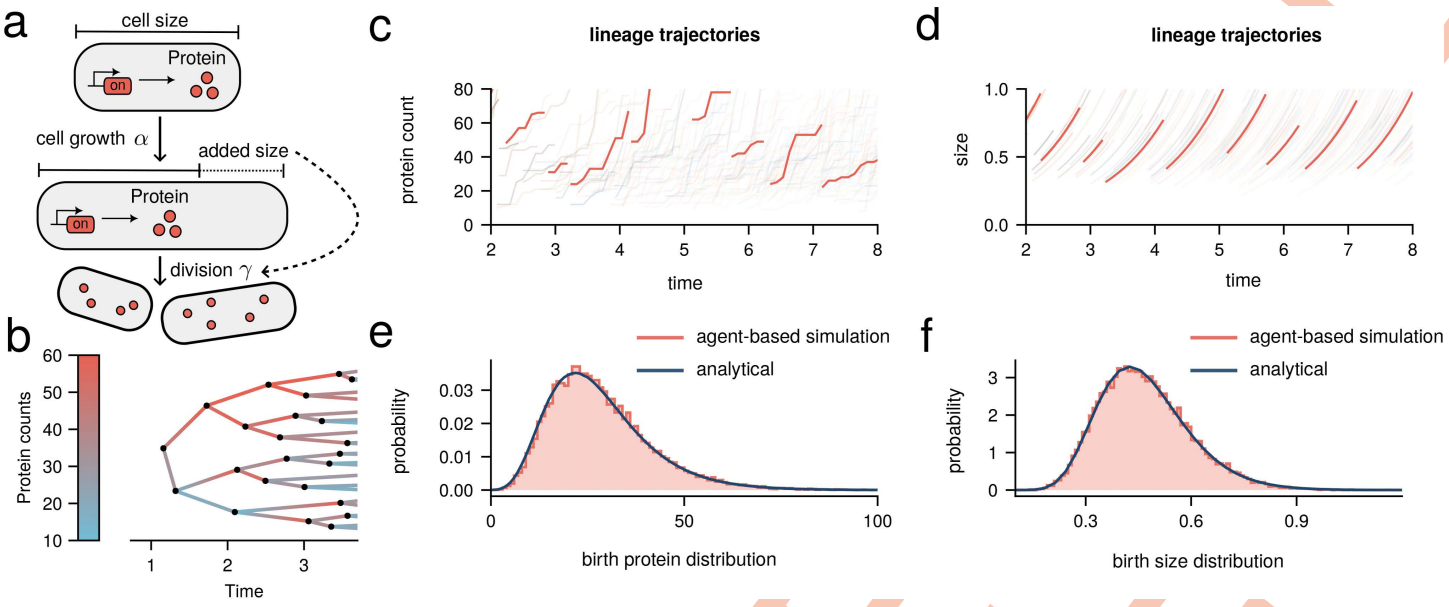

Fig 2. Stochastic gene expression is coupled to cell size in growing and dividing cells. (a) Cartoon of the model involving bursty protein expression in growing and dividing cells with growth rate α and division rate γ. Molecules are stochastically partitioned at cell division. **(b)** Lineage tree of growing cell population with protein levels superimposed. **(c-d)** Simulation of protein content and cell size for a lineage tree (shaded). A representative single cell tracked over several divisions is highlighted (red). **(e-f)** Simulated birth protein and size distributions at cell birth across lineage trees (red) are compared with the analytical computations (black).

https://doi.org/10.1371/journal.pcbi.1014366.g002

is sampled from beta and binomial distributions within the `@transition` keyword. It defines the probability of the two daughter cells inheriting sizes s_1, s_2 and protein counts p_1 and p_2 respectively, given the mother cell divides with size s and protein count p. We sample this interaction channel with the Extrande method specified via the `@sampler` keyword. For convenience, the optional keyword `boundtype=:increasing` computes the bound automatically, assuming the division rate and cell size are monotonically increasing with time. The keywords `@saveinstate` and `@saveoutstate` record the cell states of cell size s and protein levels p before and after division for analysis.

Step 2: Stochastic gene expression with cell division and growth

```
@abm_variables C(t)
@parameters Cs CΔ Cp β B L

division=@interaction begin
  @channel γdivision($Cs, $CΔ, $α), $C -->2*$C
  @sampler ExtrandeMethod($L; boundtype=:increasing)
  @connections (($CΔ =>$Δ, $Cs =>$s, $Cp =>$p),)
  @variable $β=rand($Beta(100))
  @variable $B=rand($Binomial($Cp, $β))
  @transition (
    ($Δ => 0.0, $s => $β*$Cs, $p => $B),
    ($Δ => 0.0, $s => (1-$β)* $Cs, $p => $Cp - $B))
  @saveinstate ($C, $s) ($C, $p)
  @saveoutstate ($C, $s) ($C, $p)
end
```

Finally, in Step 3, the interactions and state dynamics are composed into a simulation model with the `AgentsModel` function. The function `simulate()` performs the simulation with parameters passed by `SimulationParameters()`.

Step 3: Stochastic gene expression with cell division and growth

```
model=AgentsModel([division,], Dict(C =>Cell,))
initial_population = [C => (p =>25.0, s =>0.40, Δ =>0.0),]
timespan = (0.0, 20.0) # Simulation timespan.
Δt=1.0 # Simulator timestep.
params=SimulationParameters(
  [α =>1.0, kprod =>10.0, b =>6.0, L =>0.01], timespan, Δt)
res=simulate(model, init, params);
```

The stochastic simulation performed by *AgentBasedModeling.jl* results in a lineage tree of cells (Fig 2b). Cell state trajectories display growing cell size and protein levels interrupted by stochastic cell division events (Fig 2c-d). We validated our implementation by computing protein distribution and size at cell birth analytically [47,52], which are in excellent agreement with our exact simulation algorithm (Fig 2e-f).

3.2. Cell-cell communication enables collective switching

Cell-cell communication regulates interactions between cells from microbial communities to complex multicellular organisms. Bacteria release small, diffusible molecules that enable density-dependent regulatory mechanisms, known as quorum sensing. The process coordinates collective behaviors such as biofilm formation and virulence [53]. Similarly, in multicellular organisms, differentiation is orchestrated by signaling pathways that guide cells toward specific lineages. In disease, dysregulated communication allows tumor cells to hijack normal differentiation cues and reprogram their environment [54,55]. While deterministic models are traditionally used to model these phenotypes [56], these systems lend themselves naturally to agent-based modeling [57,58].

We here explore a general model of communication that includes stochastic effects of cell-cell coupling present in small and finite populations (Fig 3a). The model involves cellular agents that expand exponentially in cell size s with rate α and produce inducer molecules x whose levels feed back on their production via a Hill-type production rate $f(x,s)$:

$$\varnothing \xrightarrow{f(x,s)} x, \quad \frac{\partial s}{\partial t} = \alpha s. \tag{6}$$

The autoinducer is synthesized within cells and then secreted into the extracellular environment, and extracellular molecules y can then be sensed and taken up by surrounding cells

$$\mathrm{Env}[y] + \mathrm{Cell}[\Delta, s, x] \underset{}{\overset{\beta}{\rightleftharpoons}} \mathrm{Env}[y+1] + \mathrm{Cell}[\Delta, s, x-1]. \tag{7}$$

We further assume that cells grow in microchambers that support a defined population size through dilution at cell division, keeping the population size constant. Once a cell divides, a random cell from the surrounding population is removed to make space for the new daughter cell:

$$\mathrm{Cell}[\Delta_1, s_1, x_1] + \mathrm{Cell}[\cdot] \xrightarrow{\gamma_{\mathrm{division}}(s_1, \Delta_1, \alpha_1)} \mathrm{Cell}[0, s_1^+, x_1^+] + \mathrm{Cell}[0, s_2^+, x_2^+], \tag{8}$$

$$s_1^+ \sim \beta s, \quad s_2^+ \sim (1-\beta)s, \quad x_1^+ \sim \mathrm{Binomial}(\beta, x), \quad x_2^+ \sim x - x_1^+. \tag{9}$$

Cell division follows the stochastic adder rule and dilutes molecules via binomial partitioning, as before.

We simulated time courses following cells over many cell divisions in the turbidostat (S2 Code). Tracing a cell that was alive at the end of the simulation backward in time, we observed that the environment followed intracellular levels but

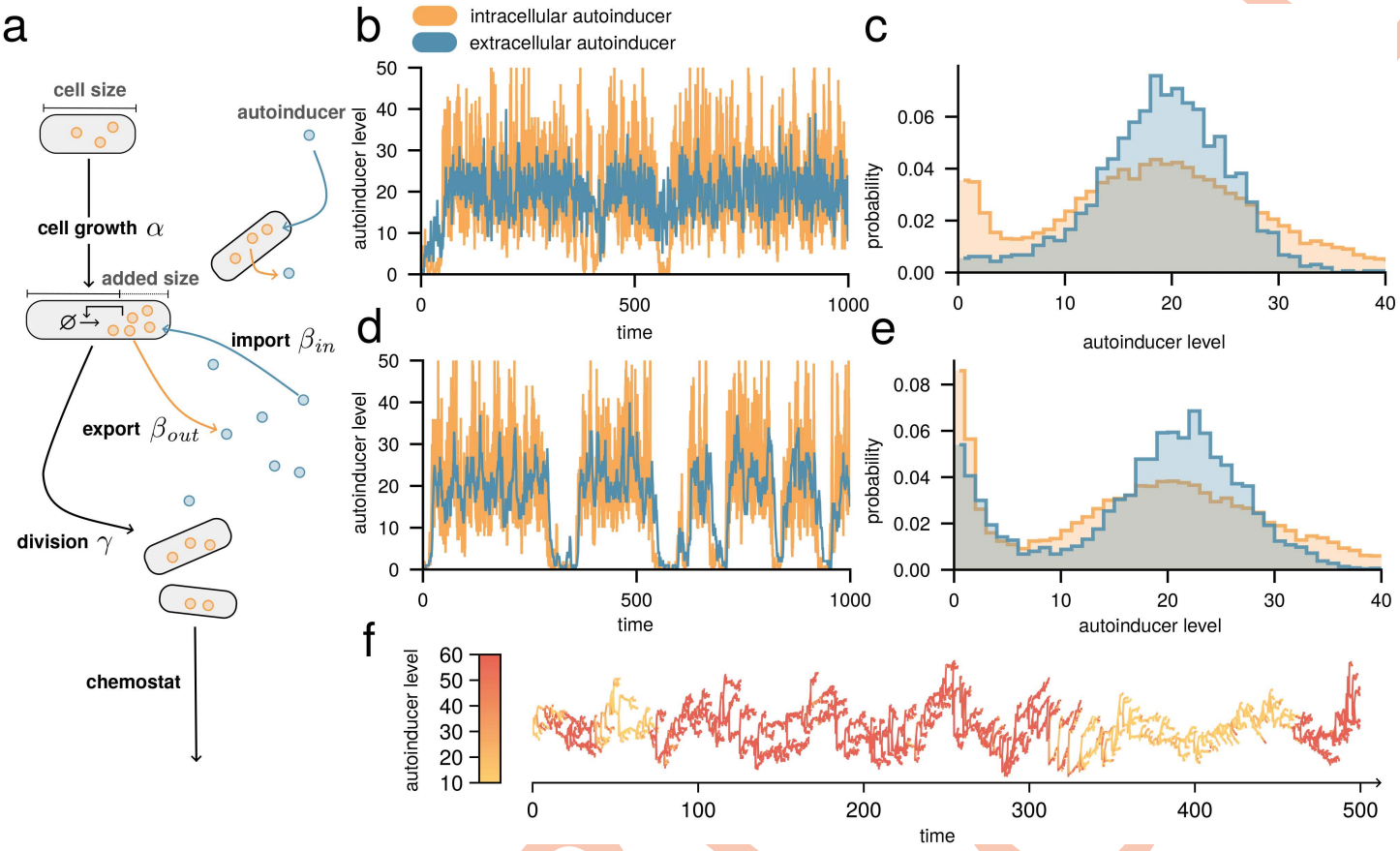

Fig 3. Cell-cell signaling enables collective switching in an agent-based cell model with diffusible autoinducer. (a) Illustration of the model involving import and export of an autoinducer molecule produced in cells growing with rate α and dividing with rate γ. We assume competition via a turbidostat, maintaining a constant cell number. **(b)** Trajectories of extracellular and intracellular levels of a random cell traced backwards in time (population of 20 cells). **(c)** Distribution of extracellular signaling molecule is unimodal while intracellular autoinducer shows bimodality. **(d)** Trajectories of extracellular (orange) and intracellular levels (blue) show significant temporal correlations at low population size (5 cells). **(e)** The corresponding distributions of extra- and intracellular autoinducer are bimodal. **(f)** Lineage of a cell population undergoing cell division and dilution in a turbidostat. Superimposed autoinducer level shows synchronised population-level transitions between active and repressed cell states.

https://doi.org/10.1371/journal.pcbi.1014366.g003

was generally less variable because it integrates signals from cells (Fig 3b). The distributions of the extracellular autoinducer were unimodal, while the intracellular levels showed bimodal distributions (Fig 3c). This suggests that cells coexist in the inactive and induced states and that the environment follows the mean across cells. Surprisingly, we observed a significant increase in temporal correlations between the extra- and intracellular levels when we decreased the population size in our simulations to what would be appropriate for a mother machine (Fig 3d). Moreover, extracellular levels were bimodal similar to intracellular levels (Fig 3e). The simulations suggested that cells collectively occupy inactive or induced states that reflect low and high environmental inducer states (Fig 3f).

3.3. Lineage fate after bacteriophage infection is determined by stochastic phage burst kinetics

Bacteriophage λ is a paradigm for studying cellular decision-making, both at the level of individual cellular pathways and at the level of population dynamics [59]. While these pathways have been described in the manner of Gillespie using

detailed stochastic reaction network models [59–62], their population dynamics has been explored using deterministic ODE models [63–66].

We here extend a recently experimentally validated model of infection-lysis dynamics during the lytic cycle of bacteriophage λ [64], incorporating multi-stage infection dynamics and Erlang-distributed lysis timing. Our stochastic model consists of discrete cell agents (Cell) that take up nutrients to grow and divide, and interact with phages dispersed in the environment (Env). The model explicitly accounts for stochasticity in intracellular phage replication, cell growth and division rates. The environment contains nutrients M and extracellular phages P and a cell's state consists of the age τ, cell size s, intracellular nutrients N, and intracellular phages Q.

Cells and the environment interact through interaction channels that model phage infection, cell lysis, cell division, and nutrient uptake from the environment. Cells become infected with phages (possibly multiple times)

$$\mathrm{Env}[M, P] + \mathrm{Cell}[\tau, s, N, Q] \xrightarrow{\epsilon} \mathrm{Env}[M, P-1] + \mathrm{Cell}[\tau^+, s, N, Q+1], \tag{10}$$

where the age τ is reset upon the first infection $\tau^+ \sim \tau$ if $Q>0$ and zero otherwise. Once a cell is infected, it lyses with an age- and infection-dependent rate

$$\mathrm{Env}[M, P] + \mathrm{Cell}[\tau, s, N, Q] \xrightarrow{\delta(\tau,Q)} \mathrm{Env}[M, P+Q], \tag{11}$$

and releases a burst of phages into the environment. Lysis occurs after a random time following a Gamma distribution φ with a fixed mean and coefficient of variation measured experimentally [67,68]. This non-Markovian infection dynamics can be implemented through an infection-age dependent lysis rate. The corresponding rate function is

$$\delta(\tau, Q) = \begin{cases} \varphi(\tau) / \int_\tau^\infty \mathrm{d}s\varphi(s) & Q > 0, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

models non-exponential infection times. These could be obtained from multi-stage infection dynamics with Erlang-distributed lysis times [64–66,69], which is a special case of the Gamma distribution. Meanwhile, uninfected cells will be able to divide

$$\mathrm{Cell}[\tau, s, N, Q=0] \xrightarrow{\gamma(\tau,s,\alpha)} \mathrm{Cell}[0, s_1^+, N_1^+, 0] + \mathrm{Cell}[0, s_2^+, N_2^+, 0], \tag{13}$$

$$s_1^+ \sim \beta s, \quad s_2^+ \sim (1-\beta)s \quad N_1^+ \sim \mathrm{Binomial}(\beta, N), \quad N_2^+ \sim N - N_1^+, \tag{14}$$

and partition intracellular nutrients between the daughter cells, similar to the gene expression model in Section 3.1. Uptake is modeled by importing nutrients from the environment into the cell:

$$\mathrm{Env}[M, P] + \mathrm{Cell}[\tau, s, N, Q] \xrightarrow{\beta} \mathrm{Env}[M-1, P] + \mathrm{Cell}[\tau, s, N+1, Q], \tag{15}$$

Intracellular nutrients are then consumed through metabolic processes fueling growth, but can also be directed towards replicating phages in infected cells:

$$Q + N \xrightarrow{k/s} 2Q, \quad N \xrightarrow{\alpha(N)} \varnothing, \quad \frac{\partial s}{\partial t} = \alpha(N)s, \tag{16}$$

where $\alpha(N)$ is the cellular growth rate following Monod's equation.

We simulated the model starting from a single cell and a fixed number of phages in the environment (S4 Code). The cell population initially expanded, peaked, and then rapidly collapsed (Fig 4b), a pattern also observed in deterministic models [64]. Surprisingly, in our stochastic model, both the peak size and timing were highly variable (Fig 4c). Peak size was inversely related to large bursts of phages released into the environment by cell lysis events (Fig 4d). We found that the distribution of extinction times was bimodal, displaying subpopulations that experienced early and late collapse (Fig 4e). Early collapse correlated with phage bursts from a few infected cells (blue lines, Fig 4d). We therefore analyzed the burst distributions from individual lysis events in these populations that we tracked using the keyword `@saveinstate` (S3 Code). Burst distributions were much wider for the early-collapsing lineages, exhibiting an order-of-magnitude increase in the distribution mode. Such burst stochasticity is not accounted for in deterministic models but has been quantified experimentally [70]. Recent advances in single-cell microfluidics [71,72] provide a fine-grained picture of phage-replication dynamics, and the modeling framework presented here could facilitate integrating these new single-cell insights into agent-based models of population dynamics.

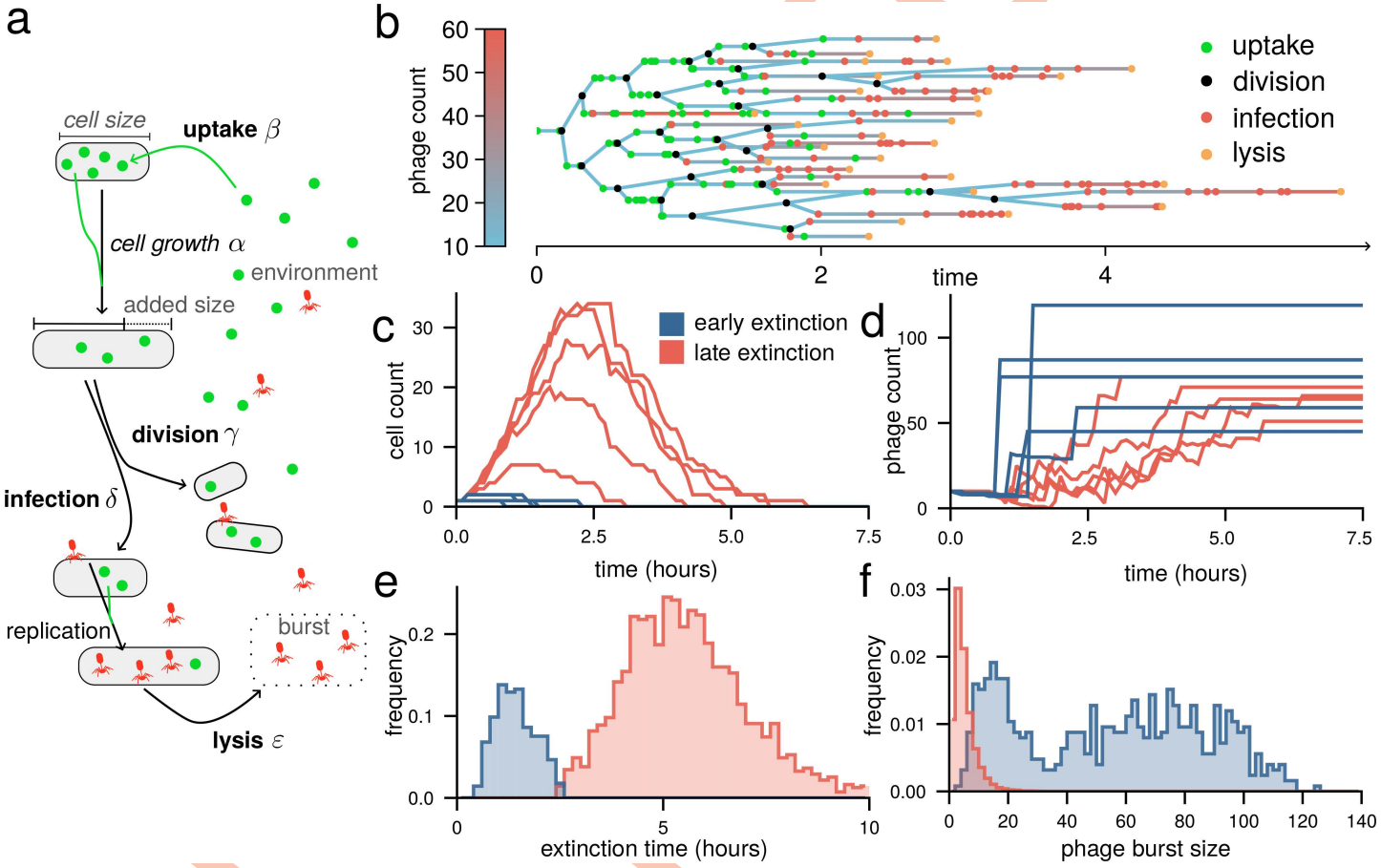

Fig 4. Stochastic lysis kinetics induces bimodal extinction time distributions in an agent-based phage infection model. **(a)** Model illustration of cellular agents that grow and divide after taking up nutrients from the environment. Phages infect cells and stochastically replicate inside cells in a nutrient-dependent manner, and lyse cells, releasing bursts of phages. **(b)** Lineage following cell division, nutrient uptake, infection and lysis across a micro-colony of cells. Line color represents intracellular phage levels. **(c)** Population trajectory following infection of a single cell. **(d)** Burst of phages following infection is stochastic. **(e)** Extinction times show bimodal distribution with populations divided in early and late extinction. **(f)** Phage burst distributions of early and late extinct populations.

https://doi.org/10.1371/journal.pcbi.1014366.g004

3.4. Discussion of findings and extensions

Our first example — modeling stochastic gene expression in growing and dividing cells — demonstrates how gene expression can be coupled to cell growth and division processes. Cells in this model grow exponentially and divide according to an "adder" cell size control mechanism, while proteins are synthesized in bursty, size-dependent reactions. We validated the exact simulations against analytical predictions for protein distributions at birth, finding direct agreement with our implementation. While we recovered known single-cell patterns, such as size-protein distributions, our model could easily be extended to include metabolism or gene regulation. Our simulation preserves trajectories, tracking individual cells over time, enabling other lineage-tree analyses similar to those in live-cell microscopy or lineage-tracing experiments.

An alternative to the model with added size-dependent division rate, is a multi-stage process that also accounts for the same adder behavior, and these models have been used to study the effect of gene replication on mRNA and protein expression [48,73–75]. We coupled a simple model of bursty gene expression to a multi-stage model of division and replication using *AgentBasedModeling.jl* (S4 Code and Section A in S1 Appendix). The multi-stage model reproduces the same cell size distributions, but it alters the gene expression distributions (Fig A in S1 Appendix). This is because the model lacks stochastic concentration homeostasis, which also prevents it from admitting simple analytical solutions [47,48].

The second case study — cell communication — illustrates how the framework can model intercellular signaling common to a broad range of microbial and mammalian systems. Cells exchange molecules with the environment, creating a shared pool that cells respond to in a stochastic, density-dependent manner via a positive feedback mechanism. Our simulation results reveal that feedback and stochasticity can give rise to complex, emergent behavior such as the coexistence of induced and repressed states at the single-cell and population level. Notably, at low population sizes, we observed that the temporal correlation between intra- and extracellular levels increased when cells switched synchronously between global on- and off-states. These transitions are driven by intrinsic noise, which is typically ignored in mean-field models.

Our third application – bacteriophage λ infection and lysis – further illustrates how stochastic intracellular processes can determine the population fate. As cells grow and divide, phages replicate stochastically within the host. Upon reaching a critical state, infected cells lyse, releasing a burst of phages that can infect other cells. We explicitly modeled lysis timing and burst variability, which are supported by experimental measurements but typically neglected or incorporated using the linear chain trick into deterministic models [64–66,69]. Simulations revealed that the timing and magnitude of early lysis bursts determine the population trajectory. We observed a bimodal distribution of extinction times: some populations collapsed early after large initial bursts, while others grew for longer before eventually collapsing. This heterogeneity is not captured by mean-field models, which predict extinction only over infinite-time horizons as they average over burst timings and lysis events. By contrast, our agent-based framework can trace stochastic events and cell histories to identify rare, high-impact lysis events that determine population fate.

We investigated whether bimodal extinction distributions are unique to phage infections or a general feature of stochastic infection models. We developed an agent-based SIR model with infection-age-dependent rates and population heterogeneity [76], including factors like incubation periods, age structure, and vaccination status (Section B in S1 Appendix, S5 Code, Fig Ba in S1 Appendix), which have previously been studied using partial differential equations [35]. Extinction occurred either early without outbreak or after the first or second wave due to low infection numbers, leading to multimodal extinction time distributions similar to what we observed in the phage model (Fig Bb in S1 Appendix). While multimodality persisted when ignoring population structure using Gillespie simulations [77], the age-structured model predicted a lower outbreak probability and delays in extinction times of outbreaks consistent with incubation effects (Fig Bc in S1 Appendix). These findings highlight the interplay of stochasticity and population structure underpinning infection dynamics.

Together, these case studies highlight the versatility of *AgentBasedModeling.jl*. Across very different biological scenarios—intracellular gene regulation, intercellular signaling, and host-pathogen interactions—our framework provides a unified formalism of structured population dynamics through evolving internal states and stochastic interactions.

4. Availability and future directions

Cellular systems can be understood as heterogeneous, interacting cellular agents whose behaviors are governed by both internal biological network dynamics and external environmental cues. Capturing such complexity requires frameworks that integrate fine-grained mechanistic intracellular processes with stochastic interactions across a cell population. Here, we introduced *AgentBasedModeling.jl*, a Julia-based platform that enables efficient, flexible simulation of structured populations governed by stochastic, continuous-time agent-based models. The Julia package is freely available with open source code and usage examples at https://github.com/pihop/AgentBasedModeling.jl.

A key strength of our computational framework is its ability to represent cells as agents whose internal dynamics evolve according to jump-diffusions and interact through continuous-rate processes that respond to cell states. This enables biologically realistic modeling of systems where internal fluctuations—such as noise in gene expression, metabolism and growth, or stochastic infection and cell death directly influence population-level behavior. By combining state-of-the-art modeling tools (*ModelingToolkit.jl* and *Catalyst.jl*) with a novel event-driven simulation algorithm, *AgentBasedModeling.jl* presents a powerful and extensible tool for computational biology.

Compared to existing simulation tools often tailored to spatial models [18–20], *AgentBasedModeling.jl* offers several advantages. First, it allows continuous-time simulation with exact handling of interactions with non-exponential waiting times via thinning-based samplers. Second, it supports hybrid dynamics, including ODEs, SDEs, and discrete jumps, enabling realistic mechanistic models of intracellular behavior. Third, our tool extends Markov jump processes of single agents, such as provided by *PyEcoLib* [29] or *JumpProcesses.jl*, to populations of interacting agents. Finally, the interface is composable and integrates with the broader Julia ecosystem, supporting parameter inference [78], visualization [79], and deployment in reproducible pipelines.

We anticipate our framework to advance the state-of-the-art of computational cell models. The ability to simulate and track individual cells as agents makes it suitable for simulation-based inference [15,16], especially in settings where lineage trees, time-lapse imaging, or single-cell omics data are available. The modular design allows rapid model construction and exploration of alternative hypotheses, which is valuable for both theoretical investigations and synthetic biology applications. Our tool could further facilitate the integration of stochastic population models with existing agent-based simulation environments. Many current agent-based frameworks rely on fixed discrete update times, whereas biologically relevant events such as intracellular reactions, growth, division, death, and cell-cell interactions occur concurrently in continuous time. This mismatch can distort event timing and ordering across many interacting cells, with consequences for emergent behavior even with advanced updating schedules. Continuous-rate, event-driven formulations such as those implemented in *AgentBasedModeling.jl* offer a natural route to overcoming these limitations and to coupling mechanistic intracellular models with larger-scale stochastic population simulations. Future extensions could include explicit spatial structure, history-dependent epigenetic rules, and adaptive agent behavior, allowing more faithful models of development, immunity, and ecological systems.

AgentBasedModeling.jl fills a critical gap in computational biology by enabling exact agent-based simulation and easy model specification of complex, stochastic intracellular dynamics that drive emergent population behavior.

Supporting information

S1 Appendix. Supporting information describing stochastic gene expression model coupled to a multi-stage division and replication (Section A) and epidemic model of infection age with heterogeneous mixing (Section B).
(PDF)

S1 Code. Pluto notebook: Stochastic gene expression in growing and dividing cells.
(JL)

S2 Code. Pluto notebook: Cell-cell communication.
(JL)

S3 Code. Pluto notebook: Bacteriophage infection.
(JL)

S4 Code. Pluto notebook: Stochastic gene expression model coupled to a multi-stage division and replication.
(JL)

S5 Code. Pluto notebook: Susceptible-infected-recovered epidemic model.
(JL)

Acknowledgments

The authors thank Torkel Lohman, Johannes Pausch and Francesco Puccioni for useful discussions.

Author contributions

Conceptualization: Paul Piho, Philipp Thomas.

Formal analysis: Paul Piho, Philipp Thomas.

Funding acquisition: Philipp Thomas.

Methodology: Paul Piho, Philipp Thomas.

Software: Paul Piho.

Validation: Paul Piho, Philipp Thomas.

Visualization: Paul Piho.

Writing – original draft: Paul Piho, Philipp Thomas.

Writing – review & editing: Paul Piho, Philipp Thomas.

References

1. Thomas P. Making sense of snapshot data: ergodic principle for clonal cell populations. J R Soc Interface. 2017;14(136):20170467. https://doi.org/10.1098/rsif.2017.0467 PMID: 29187636
2. García MR, Vázquez JA, Teixeira IG, Alonso AA. Stochastic individual-based modeling of bacterial growth and division using flow cytometry. Front Microbiol. 2018;8:2626. https://doi.org/10.3389/fmicb.2017.02626 PMID: 29354110
3. Ruess J, Ballif G, Aditya C. Stochastic chemical kinetics of cell fate decision systems: from single cells to populations and back. J Chem Phys. 2023;159(18):184103. https://doi.org/10.1063/5.0160529 PMID: 37937934
4. Piho P, Thomas P. Feedback between stochastic gene networks and population dynamics enables cellular decision-making. Sci Adv. 2024;10(21):eadl4895. https://doi.org/10.1126/sciadv.adl4895 PMID: 38787956
5. Pleyer J, Fleck C. Agent-based models in cellular systems. Front Phys. 2023;10. https://doi.org/10.3389/fphy.2022.968409
6. An G, Fitzpatrick BG, Christley S, Federico P, Kanarek A, Neilan RM, et al. Optimization and control of agent-based models in biology: a perspective. Bull Math Biol. 2017;79:63–87. https://doi.org/10.1007/s11538-016-0225-6
7. Cooper FR, Baker RE, Bernabeu MO, Bordas R, Bowler L, Bueno-Orovio A, et al. Chaste: cancer, heart and soft tissue environment. J Open Source Softw. 2020;5(47):1848. https://doi.org/10.21105/joss.01848 PMID: 37192932
8. Tripathi S, Chakraborty P, Levine H, Jolly MK. A mechanism for epithelial-mesenchymal heterogeneity in a population of cancer cells. PLoS Comput Biol. 2020;16(2):e1007619. https://doi.org/10.1371/journal.pcbi.1007619 PMID: 32040502
9. Puccioni F, Pausch J, Piho P, Thomas P. Survival resonances during fractional killing of cell populations. Phys Rev Lett. 2024;133(19):198401. https://doi.org/10.1103/PhysRevLett.133.198401 PMID: 39576926

10. Hoertel N, Blachier M, Blanco C, Olfson M, Massetti M, Rico MS, et al. A stochastic agent-based model of the SARS-CoV-2 epidemic in France. Nat Med. 2020;26(9):1417–21. https://doi.org/10.1038/s41591-020-1001-6 PMID: 32665655
11. Di Domenico L, Pullano G, Sabbatini CE, Boëlle P-Y, Colizza V. Impact of lockdown on COVID-19 epidemic in Île-de-France and possible exit strategies. BMC Med. 2020;18(1):240. https://doi.org/10.1186/s12916-020-01698-4 PMID: 32727547
12. Kerr CC, Stuart RM, Mistry D, Abeysuriya RG, Rosenfeld K, Hart GR, et al. Covasim: an agent-based model of COVID-19 dynamics and interventions. PLoS Comput Biol. 2021;17(7):e1009149. https://doi.org/10.1371/journal.pcbi.1009149 PMID: 34310589
13. Hinch R, Probert WJM, Nurtay A, Kendall M, Wymant C, Hall M, et al. OpenABM-Covid19-An agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. PLoS Comput Biol. 2021;17(7):e1009146. https://doi.org/10.1371/journal.pcbi.1009146 PMID: 34252083
14. Tankhilevich E, Ish-Horowicz J, Hameed T, Roesch E, Kleijn I, Stumpf MP, et al. GpABC: A Julia Package for Approximate Bayesian Computation with Gaussian Process Emulation. 2019. https://doi.org/10.1101/769299
15. Cranmer K, Brehmer J, Louppe G. The frontier of simulation-based inference. Proc Natl Acad Sci U S A. 2020;117(48):30055–62. https://doi.org/10.1073/pnas.1912789117 PMID: 32471948
16. Jørgensen ACS, Ghosh A, Sturrock M, Shahrezaei V. Efficient Bayesian inference for stochastic agent-based models. PLoS Comput Biol. 2022;18(10):e1009508. https://doi.org/10.1371/journal.pcbi.1009508 PMID: 36197919
17. Tang W, Jørgensen ACS, Marguerat S, Thomas P, Shahrezaei V. Modelling capture efficiency of single-cell RNA-sequencing data improves inference of transcriptome-wide burst kinetics. Bioinformatics. 2023;39(7):btad395. https://doi.org/10.1093/bioinformatics/btad395 PMID: 37354494
18. Datseris G, Vahdati AR, DuBois TC. Agents.Jl: A Performant and Feature-Full Agent-Based Modeling Software of Minimal Code Complexity. Simulation. 2022;1019–31. https://doi.org/10.1177/00375497211068820
19. Kazil J, Masad D, Crooks A. Utilizing Python for agent-based modeling: the mesa framework. Soc Cult Behav Model. 2020. p. 308–17. https://doi.org/10.1007/978-3-030-61255-9_30
20. North MJ, Collier NT, Ozik J, Tatara ER, Macal CM, Bragen M, et al. Complex adaptive systems modeling with Repast Simphony. Complex Adapt Syst Model. 2013;1:3. https://doi.org/10.1186/2194-3206-1-3
21. Wilensky U. NetLogo. 1999.
22. Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J Comput Phys. 1976;22(4):403–34. https://doi.org/10.1016/0021-9991(76)90041-3
23. Maus C, Rybacki S, Uhrmacher AM. Rule-based multi-level modeling of cell biological systems. BMC Syst Biol. 2011;5:166. https://doi.org/10.1186/1752-0509-5-166 PMID: 22005019
24. Yates CA, Ford MJ, Mort RL. A Multi-stage Representation of Cell Proliferation as a Markov Process. Bull Math Biol. 2017;79(12):2905–28. https://doi.org/10.1007/s11538-017-0356-4 PMID: 29030804
25. Helms T, Warnke T, Uhrmacher AM. Multi-level modeling and simulation of cellular systems: an introduction to ML-rules. Methods Mol Biol. 2019;1945:141–60. https://doi.org/10.1007/978-1-4939-9102-0_6 PMID: 30945245
26. Hurtado PJ, Kirosingh AS. Generalizations of the "Linear Chain Trick": incorporating more flexible dwell time distributions into mean field ODE models. J Math Biol. 2019;79(5):1831–83. https://doi.org/10.1007/s00285-019-01412-w PMID: 31410551
27. Cushing JM. An introduction to structured population dynamics: Outgrowth of a series of lectures given at a conference held at North Carolina University, Raleigh, during June of 1997. 1998. https://doi.org/10.1137/1.9781611970005
28. Lloyd-Price J, Gupta A, Ribeiro AS. SGNS2: a compartmentalized stochastic chemical kinetics simulator for dynamic cell populations. Bioinformatics. 2012;28(22):3004–5. https://doi.org/10.1093/bioinformatics/bts556 PMID: 23014631
29. Nieto C, Blanco SC, Vargas-García C, Singh A, Juan Manuel P. PyEcoLib: a python library for simulating stochastic cell size dynamics. Phys Biol. 2023;20(4). https://doi.org/10.1088/1478-3975/acd897 PMID: 37224818
30. Matyjaszkiewicz A, Fiore G, Annunziata F, Grierson CS, Savery NJ, Marucci L, et al. BSim 2.0: an advanced agent-based cell simulator. ACS Synth Biol. 2017;6(10):1969–72. https://doi.org/10.1021/acssynbio.7b00121 PMID: 28585809
31. Ghaffarizadeh A, Heiland R, Friedman SH, Mumenthaler SM, Macklin P. PhysiCell: An open source physics-based cell simulator for 3-D multicellular systems. PLoS Comput Biol. 2018;14(2):e1005991. https://doi.org/10.1371/journal.pcbi.1005991 PMID: 29474446
32. Cooper FR, Baker RE, Bernabeu MO, Bordas R, Bowler L, Bueno-Orovio A, et al. Chaste: cancer, heart and soft tissue environment. J Open Source Softw. 2020;5(47):1848. https://doi.org/10.21105/joss.01848 PMID: 37192932
33. Merton RC. Option pricing when underlying stock returns are discontinuous. J Financ Econ. 1976;3(1–2):125–44. https://doi.org/10.1016/0304-405x(76)90022-2
34. Zagatti GA, Isaacson SA, Rackauckas C, Ilin V, Ng S, Bressan S. Extending JumpProcesses.Jl for Fast Point Process Simulation with Time-Varying Intensities. Proceedings of the JuliaCon Conference. 2024;6:133. https://doi.org/10.21105/jcon.00133
35. Inaba H. Age-Structured Population Dynamics in Demography and Epidemiology. 2017. https://doi.org/10.1007/978-981-10-0188-8
36. Donnelly P, Kurtz TG. Particle representations for measure-valued population models. Ann Probab. 1999;27(1). https://doi.org/10.1214/aop/1022677258

37. Bansaye V, Méléard S. Stochastic Models for Structured Populations. 2015. https://doi.org/10.1007/978-3-319-21711-6
38. Dang Y, Grundel DAJ, Youk H. Cellular dialogues: cell-cell communication through diffusible molecules yields dynamic spatial patterns. Cell Syst. 2020;10(1):82–98.e7. https://doi.org/10.1016/j.cels.2019.12.001 PMID: 31954659
39. Ma Y, Gowda S, Anantharaman R, Laughman C, Shah V, Rackauckas C. ModelingToolkit: A Composable Graph Transformation System For Equation-Based Modeling. arXiv preprint arXiv:2103.05244. 2022. https://doi.org/10.48550/arXiv.2103.05244
40. Loman TE, Ma Y, Ilin V, Gowda S, Korsbo N, Yewale N, et al. Catalyst: fast and flexible modeling of reaction networks. PLoS Comput Biol. 2023;19(10):e1011530. https://doi.org/10.1371/journal.pcbi.1011530 PMID: 37851697
41. Blinov ML, Faeder JR, Goldstein B, Hlavacek WS. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. Bioinformatics. 2004;20(17):3289–91. https://doi.org/10.1093/bioinformatics/bth378 PMID: 15217809
42. Danos V, Feret J, Fontana W, Krivine J. Scalable Simulation of Cellular Signaling Networks. Programm Lang Syst. 2007;4807:139–57. https://doi.org/10.1007/978-3-540-76637-7_10
43. Gillespie DT. Exact stochastic simulation of coupled chemical reactions. J Phys Chem. 1977;81(25):2340–61. https://doi.org/10.1021/j100540a008
44. Lewis PAW, Shedler GS. Simulation of nonhomogeneous poisson processes by thinning. Naval Res Logist. 1979;26(3):403–13. https://doi.org/10.1002/nav.3800260304
45. Voliotis M, Thomas P, Grima R, Bowsher CG. Stochastic simulation of biomolecular networks in dynamic environments. PLoS Comput Biol. 2016;12(6):e1004923. https://doi.org/10.1371/journal.pcbi.1004923 PMID: 27248512
46. Hicks DG, Speed TP, Yassin M, Russell SM. Maps of variability in cell lineage trees. PLoS Comput Biol. 2019;15(2):e1006745. https://doi.org/10.1371/journal.pcbi.1006745 PMID: 30753182
47. Thomas P, Shahrezaei V. Coordination of gene expression noise with cell size: analytical results for agent-based models of growing cell populations. J R Soc Interface. 2021;18(178):20210274. https://doi.org/10.1098/rsif.2021.0274 PMID: 34034535
48. Jia C, Grima R. Coupling gene expression dynamics to cell size dynamics and cell cycle events: Exact and approximate solutions of the extended telegraph model. iScience. 2022;26(1):105746. https://doi.org/10.1016/j.isci.2022.105746 PMID: 36619980
49. Golding I, Amir A. Colloquium: Gene expression in growing cells: a biophysical primer. Rev Mod Phys. 2024;96(4). https://doi.org/10.1103/revmodphys.96.041001
50. Lunz D, Batt G, Ruess J, Bonnans JF. Beyond the chemical master equation: Stochastic chemical kinetics coupled with auxiliary processes. PLoS Comput Biol. 2021;17(7):e1009214. https://doi.org/10.1371/journal.pcbi.1009214 PMID: 34319979
51. Volteras D, Shahrezaei V, Thomas P. Global transcription regulation revealed from dynamical correlations in time-resolved single-cell RNA sequencing. Cell Syst. 2024;15(8):694–708.e12. https://doi.org/10.1016/j.cels.2024.07.002 PMID: 39121860
52. Thomas P. Analysis of cell size homeostasis at the single-cell and population level. Front Phys. 2018;6. https://doi.org/10.3389/fphy.2018.00064
53. Striednig B, Hilbi H. Bacterial quorum sensing and phenotypic heterogeneity: how the collective shapes the individual. Trends Microbiol. 2022;30(4):379–89. https://doi.org/10.1016/j.tim.2021.09.001 PMID: 34598862
54. Chen X, Chen L, Kürten CHL, Jabbari F, Vujanovic L, Ding Y, et al. An individualized causal framework for learning intercellular communication networks that define microenvironments of individual tumors. PLoS Comput Biol. 2022;18(12):e1010761. https://doi.org/10.1371/journal.pcbi.1010761 PMID: 36548438
55. Feng J, Goedegebuure SP, Zeng A, Bi Y, Wang T, Payne P, et al. sc2MeNetDrug: A computational tool to uncover inter-cell signaling targets and identify relevant drugs based on single cell RNA-seq data. PLoS Comput Biol. 2024;20(1):e1011785. https://doi.org/10.1371/journal.pcbi.1011785 PMID: 38181047
56. Ostovar G, Boedicker JQ. Phenotypic memory in quorum sensing. PLoS Comput Biol. 2024;20(7):e1011696. https://doi.org/10.1371/journal.pcbi.1011696 PMID: 38976753
57. Shu C-C, Chatterjee A, Dunny G, Hu W-S, Ramkrishna D. Bistability versus bimodal distributions in gene regulatory processes from population balance. PLoS Comput Biol. 2011;7(8):e1002140. https://doi.org/10.1371/journal.pcbi.1002140 PMID: 21901083
58. Weber M, Buceta J. Dynamics of the quorum sensing switch: stochastic and non-stationary effects. BMC Syst Biol. 2013;7:6. https://doi.org/10.1186/1752-0509-7-6 PMID: 23324134
59. Golding I. Single-cell studies of phage λ: hidden treasures under Occam's Rug. Annu Rev Virol. 2016;3(1):453–72. https://doi.org/10.1146/annurev-virology-110615-042127 PMID: 27482899
60. Arkin A, Ross J, McAdams HH. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. Genetics. 1998;149(4):1633–48. https://doi.org/10.1093/genetics/149.4.1633 PMID: 9691025
61. Robb ML, Shahrezaei V. Stochastic cellular fate decision making by multiple infecting lambda phage. PLoS One. 2014;9(8):e103636. https://doi.org/10.1371/journal.pone.0103636 PMID: 25105971
62. Joh RI, Weitz JS. To lyse or not to lyse: transient-mediated stochastic fate determination in cells infected by bacteriophages. PLoS Comput Biol. 2011;7(3):e1002006. https://doi.org/10.1371/journal.pcbi.1002006 PMID: 21423715
63. Weitz JS. Quantitative viral ecology: dynamics of viruses and their microbial hosts. 2016. https://doi.org/10.23943/princeton/9780691161549.001.0001

64. Geng Y, Nguyen TVP, Homaee E, Golding I. Using bacterial population dynamics to count phages and their lysogens. Nat Commun. 2024;15(1):7814. https://doi.org/10.1038/s41467-024-51913-6 PMID: 39242585

65. Hvid U, Mitarai N. Competitive advantages of T-even phage lysis inhibition in response to secondary infection. PLoS Comput Biol. 2024;20(7):e1012242. https://doi.org/10.1371/journal.pcbi.1012242 PMID: 38976747

66. Marchi J, Minh CNN, Debarbieux L, Weitz JS. Multi-strain phage induced clearance of bacterial infections. PLoS Comput Biol. 2025;21(2):e1012793. https://doi.org/10.1371/journal.pcbi.1012793 PMID: 39903766

67. Wang I-N. Lysis timing and bacteriophage fitness. Genetics. 2006;172(1):17–26. https://doi.org/10.1534/genetics.105.045922 PMID: 16219778

68. Singh A, Dennehy JJ. Stochastic holin expression can account for lysis time variation in the bacteriophage λ. J R Soc Interface. 2014;11(95):20140140. https://doi.org/10.1098/rsif.2014.0140 PMID: 24718449

69. Mitarai N, Brown S, Sneppen K. Population dynamics of phage and bacteria in spatially structured habitats using phage λ and Escherichia coli. J Bacteriol. 2016;198(12):1783–93. https://doi.org/10.1128/JB.00965-15 PMID: 27068593

70. Delbruck M. The burst size distribution in the growth of bacterial viruses (bacteriophages). J Bacteriol. 1945;50:131–5. https://doi.org/10.1128/JB.50.2.131-135.1945 PMID: 20989330

71. Attrill EL, Claydon R, Łapińska U, Recker M, Meaden S, Brown AT, et al. Individual bacteria in structured environments rely on phenotypic resistance to phage. PLoS Biol. 2021;19(10):e3001406. https://doi.org/10.1371/journal.pbio.3001406 PMID: 34637438

72. Wedd C, Yunusov T, Smith A, Li R, Hardo G, Hunter M, et al. Single-cell imaging of the lytic phage life cycle in bacteria. Biorxiv. 2024. https://doi.org/10.1101/2024.04.11.588870

73. Peterson JR, Cole JA, Fei J, Ha T, Luthey-Schulten ZA. Effects of DNA replication on mRNA noise. Proc Natl Acad Sci U S A. 2015;112(52):15886–91. https://doi.org/10.1073/pnas.1516246112 PMID: 26669443

74. Nieto C, Arias-Castro J, Sánchez C, Vargas-García C, Pedraza JM. Unification of cell division control strategies through continuous rate models. Phys Rev E. 2020;101(2):022401. https://doi.org/10.1103/PhysRevE.101.022401

75. Jia C, Grima R. Frequency domain analysis of fluctuations of mrna and protein copy numbers within a cell lineage: theory and experimental validation. Phys Rev X. 2021;11(2). https://doi.org/10.1103/physrevx.11.021032

76. Brauer F. Age of Infection Epidemic Models. Mathematical and Statistical Modeling for Emerging and Re-emerging Infectious Diseases. 2016. pp. 207–20. https://doi.org/10.1007/978-3-319-40413-4_13

77. Parsons TL, Bolker BM, Dushoff J, Earn DJD. The probability of epidemic burnout in the stochastic SIR model with vital dynamics. Proc Natl Acad Sci U S A. 2024;121(5):e2313708120. https://doi.org/10.1073/pnas.2313708120 PMID: 38277438

78. Tankhilevich E, Ish-Horowicz J, Hameed T, Roesch E, Kleijn I, Stumpf MPH, et al. GpABC: a Julia package for approximate Bayesian computation with Gaussian process emulation. Bioinformatics. 2020;36(10):3286–7. https://doi.org/10.1093/bioinformatics/btaa078 PMID: 32022854

79. Danisch S, Krumbiegel J. Makie.jl: Flexible high-performance data visualization for Julia. J Open Source Softw. 2021;6(65):3349. https://doi.org/10.21105/joss.03349