A modular framework for multiscale, multicellular, spatiotemporal modeling of acute primary viral infection and immune response in epithelial tissues and its application to drug therapy timing and effectiveness

A multiscale model of viral infection in epithelial tissues

T.J. Sego[1,2*], Josua O. Aponte-Serrano[1,2*], Juliano Ferrari Gianlupi[1,2], Samuel R. Heaps[1], Kira Breithaupt[1,3], Lutz Brusch[4], Jessica Crawshaw[5], James M. Osborne[5], Ellen M. Quardokus[1], Richard K. Plemper[6], James A. Glazier[1,2]

[1] Department of Intelligent Systems Engineering, Indiana University, Bloomington, IN, USA
[2] Biocomplexity Institute, Indiana University, Bloomington, IN, USA
[3] Cognitive Science Program, Indiana University, Bloomington, IN, USA
[4] Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany
[5] School of Mathematics and Statistics, University of Melbourne, Melbourne, 3010, Australia
[6] Institute for Biomedical Sciences, Georgia State University, Atlanta, GA, USA
[*] These Authors contributed equally to this study

Abstract

Simulations of tissue-specific effects of primary acute viral infections like COVID-19 are essential for understanding disease outcomes and optimizing therapies. Such simulations need to support continuous updating in response to rapid advances in understanding of infection mechanisms, and parallel development of components by multiple groups. We present an open-source platform for multiscale spatiotemporal simulation of an epithelial tissue, viral infection, cellular immune response and tissue damage, specifically designed to be modular and extensible to support continuous updating and parallel development. The base simulation of a simplified patch of epithelial tissue and immune response exhibits distinct patterns of infection dynamics from widespread infection, to recurrence, to clearance. Slower viral internalization and faster immune-cell recruitment slow infection and promote containment. Because antiviral drugs can have side effects and show reduced clinical effectiveness when given later during

infection, we studied the effects on progression of treatment potency and time-of-first treatment after infection. In simulations, even a low potency therapy with a drug which reduces the replication rate of viral RNA greatly decreases the total tissue damage and virus burden when given near the beginning of infection. Many combinations of dosage and treatment time lead to stochastic outcomes, with some simulation replicas showing clearance or control (treatment success), while others show rapid infection of all epithelial cells (treatment failure). Thus, while a high potency therapy usually is less effective when given later, treatments at late times are occasionally effective. We illustrate how to extend the platform to model specific virus types (*e.g.*, hepatitis C) and add additional cellular mechanisms (tissue recovery and variable cell susceptibility to infection), using our software modules and publicly-available software repository.

Author summary

This study presents an open-source, extensible, multiscale platform for simulating viral immune interactions in epithelial tissues, which enables the rapid development and deployment of sophisticated models of viruses, infection mechanisms and tissue types. The model is used to investigate how potential treatments influence disease progression. Simulation results suggest that drugs which interfere with virus replication (e.g., remdesivir) yield substantially better infection outcomes when administered prophylactically even at very low doses than when used at high doses as treatment for an infection that has already begun.

Introduction

The current global pandemic of COVID-19, caused by the novel coronavirus Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2), has motivated the study of beta coronavirus diseases at multiple spatial and temporal computational modeling scales [1]. The time course, severity of symptoms and complications from SARS-CoV-2 infection are highly variable from patient to patient [2]. Mathematical modeling methods integrate the available host- and pathogen-level data on disease dynamics that are required to understand the complex biology of infection and immune response to optimize therapeutic interventions [3–5]. Mathematical models and computer simulations built on spatial and ODE frameworks have been extensively used to study in-host progression of viral infection [6], with a recent acceleration in the development of spatial COVID-19 viral infection models in response to the global pandemic [7,8].

Building multiscale models of acute primary viral infection requires integration of submodels of multiple biological components across scales (*e.g.*, viral replication and internalization, immune system responses). Non-spatial, coupled ordinary differential equation (*ODE*) models can represent many aspects of pathogen-host interaction. In the context of viral infection dynamics, specialized ODE models can describe both the entire virus-host response at the tissue and organ levels and different stages of the viral replication cycle within cells, such as binding and internalization [9,10], viral genome replication and translation [11,12], assembly, packaging and release [13,14]. By fitting ODE models to clinical or experimental data, researchers have been able to estimate important parameters, such as the turnover rate of target cells, average lifetimes of viral

particles and infected cells and the rate of production of new viral particles by infected cells [15]. Other ODE models include pharmacokinetic models of drug availability [16].

The simplest non-spatial models assume that the distribution of the modelled quantities (*e.g.*, cells, viruses, chemical species) are uniformly distributed in space and time [17]. This assumption might not be realistic in solid tissues, where viruses and host immune cells are not usually distributed homogeneously and infection propagates locally [15] or in situations where transport limits the dynamics (*e.g.* the migration of antigen presenting cells to lymph nodes, the transmission of virus between organs or the microdosimetry of a drug therapy). By averaging over spatio-temporal and individual cell variations, non-spatial models may not accurately reflect the effects of tissue heterogeneity and resulting viral infection dynamics [18]. Compartmental ODE models, like physiologically based pharmacokinetic models (PBPK) models or multi-compartment tissue infection models, maintain some of the simplicity of single-compartment ODE models, while recognizing the critical role transport can play in viral infection, immune response and treatment [19].

However, the spread of, and response to, some viruses is highly spatially localized, both *in vitro* and *in vivo* [20,21]. For example, COVID-19 often begins with infection localized to the nose and throat and then spreads to the lungs [22], with the specific location, size and distribution of lesions affecting clinical outcomes [23]. Spatial models have been increasingly used to address such issues [24], including partial differential equations [25,26] fluid-dynamic models [27] and agent-based models (ABM) [28]. ABMs represent host cells as spatially located, individual agents, and propagation of the infection emerges from individual interactions between agents. ABMs are also well suited

for extending existing models by modular integration of biological subcomponents, and their model parameters should be validated by experiment and studied through sensitivity analysis [17]. ABMs have been developed to account for infection dynamics in different biological compartments (such as the lung and lymph nodes [29,30]) and to model disease progression of HIV [15,25,31–33] and dissemination of influenza virus to the lower respiratory tract [18,34].

Spatial models often predict significantly different viral and immune dynamics, parameter estimates and therapy efficacies from their non-spatial counterparts. Stochastic effects arising from spatial conditions, such as local availability of target and immune cells, greatly influence establishment of infection and lead to different infection outcomes [15]. Non-spatial models generally produce viral load titers higher than spatial models, and the peaks of infection happen significantly earlier [17,18]. Homogenous recruitment of immune cells in spatial models matches ODE models when the number of infected is large, but not at the beginning of the infection when the number of infected cells is small [18]. These differences can lead to inaccurate estimates of important parameters such as viral infectivity [35], viral diffusion [17] and the basic reproductive ratio [35]. The basic reproductive ratio is clinically needed to determine therapeutic effectiveness [35]. Microdosimetry is another area where spatial modeling is important, since spatial variation in bioavailability can lead to low concentrations in some regions of an infected tissue, which can promote the evolution of resistant viral strains [15].

In this paper, we consider *primary* infection, that is, infection by a virus which the immune system has not previously encountered, so that there is no initial adaptive immune response. We focus on *acute* cases (cases with relatively rapid onset and short

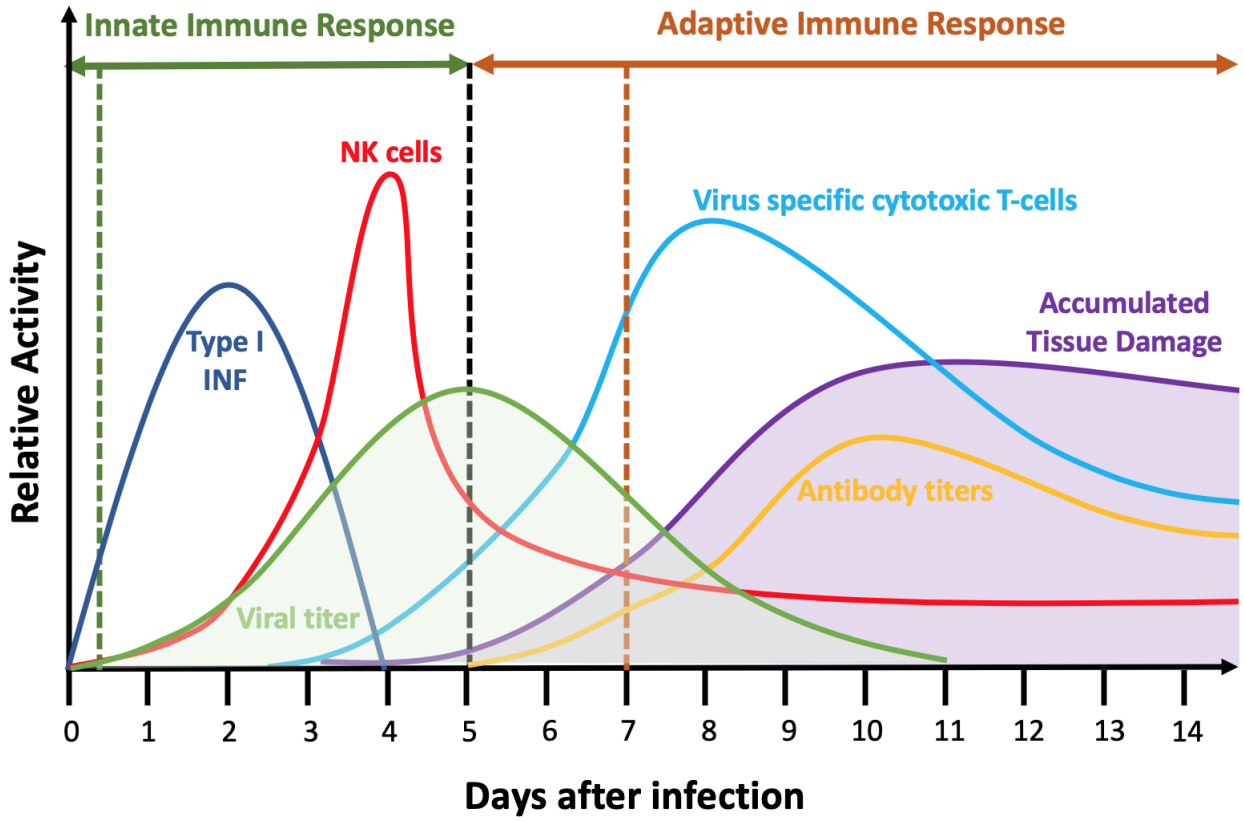
duration), in which a properly functioning immune system eventually eliminates the virus completely (*clearance*). Here we review relevant components of the immune system. Some of these components are included in this work and some are not modeled. However, by constructing a modular, extensible modeling framework and computational implementation, we enable the modeling of all these components.

While the details of infection vary by virus and patient [36], infection generally begins when a virus breaches the barrier of one or more tissues causing a limited number of target cells to be exposed and then internalize the virus (Fig 1). The virus begins to replicate within the initially infected target cells, but cells do not release any newly synthesized virus for a period of hours to days (the *eclipse* or *lag* phase of infection). Within hours, infected cells release proinflammatory cytokines and complement proteins as warning signals to neighboring cells [37,38]. Some of these cytokines, like Type 1 interferons, can induce autocrine and paracrine anti-viral responses (*e.g.*, inhibiting viral replication, viral entry or inducing cell death) [39]. Cytokines recruit circulating immune cells from the blood to the infected tissue and attract immune cells within the tissue by chemotaxis [40–42]. The early innate immune response activates a number of cell types including dendritic cells, macrophages, neutrophils, mast cells, basophils, eosinophils, leukocytes, and natural killer (*NK*) cells [43]. Many of these immune cells themselves release both pro- and anti-inflammatory signals. Immune signals also recruit circulating neutrophils in the blood and, later, activate cytotoxic innate immune cells like NK cells within the tissue, which kill infected cells through release of diffusible factors and contact-mediated interactions, respectively.

The temporal dynamics of the concentration of extracellular virus varies greatly among virus families, tissues and host species [44]. However, for many viruses, including influenza and coronaviruses, once infected cells begin to release virus, the amount of extracellular virus increases exponentially over a period of a few days, reaching a peak during an early phase of infection [45]. As the viral load increases, immune signaling increases rapidly (this increase is associated with the onset of fever and other symptoms) recruiting more circulating cells of the innate immune system to the infection site [46].

Immune signals from infected cells and innate immune cells help trigger the adaptive immune response. Macrophages and dendritic cells that have engulfed and degraded viral pathogens or fragments of dead infected cells (*i.e.*, phagocytosis) migrate over a period of days to nearby lymph nodes and serve as viral antigen presenting cells (*APC*s) to naive T-cells. Antigen presentation induces naive T-cell proliferation and differentiation into pathogen-specific memory and effector T-cells [47]. Effector T-cells migrate to the site of infection and induce apoptosis of infected cells by antigen recognition and contact killing. Both infected and uninfected cells in contact with dying cells can die through bystander-effect mechanisms. In acute infections, adaptive immune response leads to pathogen neutralization and clearance [48]. Viral loads usually decrease rapidly as adaptive immune cells like CD8+ T-cells enter the tissue and eliminate infected cells. Cells also begin to send out anti-inflammatory signals, shutting down the immune response as viral clearance proceeds. Antigen presentation within the immune system also induces activation of naive B-cell lymphocytes into antibody-producing memory B-cells and plasma cells, which leads to the production of antibodies. The adaptive immune response remembers its exposure to previous pathogens and

166 provides the body with pathogen-specific effector cells and antibodies which neutralize
167 and clear them, providing long term immunity [49]. Tissue damage results from virus and
168 cytokine-induced cell death (which is first noticeable after 2 or 3 days) and from killing of
169 infected and uninfected cells by immune cells, which increases steadily until the end of
170 viral clearance (7-10 days). Tissue recovery and healing start around the time of viral
171 clearance and may last for several weeks.
172



173

174 **Fig 1. Schematic of the innate and adaptive immune response during primary acute viral infection.**
175 Exposure to the virus occurs at time 0 and extracellular viral load begins to rise (shaded green curve). Initial innate
176 immune responses include phagocytosis of virus by neutrophils and macrophages, Type I interferon-induced antiviral
177 resistance (*IFN*) (dark blue) and killing of infected cells by Natural Killer (NK) cells and other cell types (red). The black
178 vertical dashed line denotes the transition between innate and adaptive immune responses. The adaptive immune
179 response is triggered both by cytokine signaling to the lymph nodes and the migration of antigen-presenting cells from

the tissue to the lymph nodes (not shown). In the later phases of infection, innate immune responses continue, but additional adaptive immune components come into play, including virus-specific cytotoxic T-cells (light blue) kill infected cells directly and also kill nearby cells through a variety of mechanisms. The orange vertical dashed line denotes the onset of the humoral adaptive immune response. B-cells produce virus-specific antibodies (orange line) which bind and inactivate virus directly and also allow its clearance and clearance of infected cells by other cell types. Tissue damage (shaded purple curve) accumulates due to cell death from direct responses to virus and from immune-cell killing by contact-mediated, diffusible factor-mediated and bystander-mediated mechanisms and eventually dissipates as cells proliferate to repair the damage (Adapted from [50,51]). The specific time course of all components varies among viruses, host tissues and host species, but the general sequence of events and immune response components are generally preserved.

In this paper we develop a framework for the multiscale multicellular spatiotemporal simulation of the complex processes of infection and immune response in a small patch of epithelial tissue. The model provides a representation of the complex biology that reproduces key observed emergent behaviors of infection dynamics. We create representations of the main types of components and biological mechanisms associated with acute, primary viral infection and immune response, with a special emphasis on modularity of mathematical forms and computational implementation to support the development of more detailed models in future work (*e.g.*, the creation of additional cell types, signals and detailed cell responses of various aspects of the immune response). We illustrate such an activity of development by integrating a detailed viral replication model for hepatitis C virus as an extension to the framework (see *Integration of an explicit RNA synthesis model allows the spatiotemporal modelling of hepatitis C virus infection*).

Our base model consists of three interconnected components (Fig 2A): an epithelium component, an extracellular environment component and a lymph node

component. The model represents the epithelium as a compact monolayer of initially identical immobile (which is appropriate for an epithelium) epithelial cells that it classifies by their current state of viral infection (*i.e.*, uninfected, infected, virus releasing, dead, Fig 2C). These epithelial cells are initially identical in their number of viral receptors (though we show how to include heterogeneity in *Heterogeneous susceptibility inhibits spread of infection*). The cells internalize extracellular virus, modulate their number of surface receptors, replicate virus, release virus and die in response to virus production, and release an extracellular cytokine signal when infected. Our model omits interferon-induced antiviral resistance, which can be implemented as a model extension using mechanisms demonstrated in this work (*e.g.*, secretion of cytokine, modulation of internalization parameters, see *Developing a Model Extension in CompuCell3D*).

The model represents the extracellular environment as a space above the epithelium which provides the space in which immune cells are recruited and move, and into which cells release viruses and chemicals. We include a single type of immune cell that exhibits many key immune-cell behaviors associated with macrophage, neutrophil, NK cell and T-cell roles, including activation, chemotaxis, relaying and amplification of cytokine signals, contact killing, secretion of diffusible killing factors, and bystander killing, to represent the main types of immune-cell mediated cell killing, rather than any particular immune cell phenotype. We omit macrophages, neutrophils and their phagocytosis and signaling, which can be represented using simple extensions of the immune-cell type. We do not model contact interactions between immune cells (*e.g.*, by T-cells and APCs). We also do not model tissue recovery, though we demonstrate examples of adding tissue recovery models in *Model extensions*.

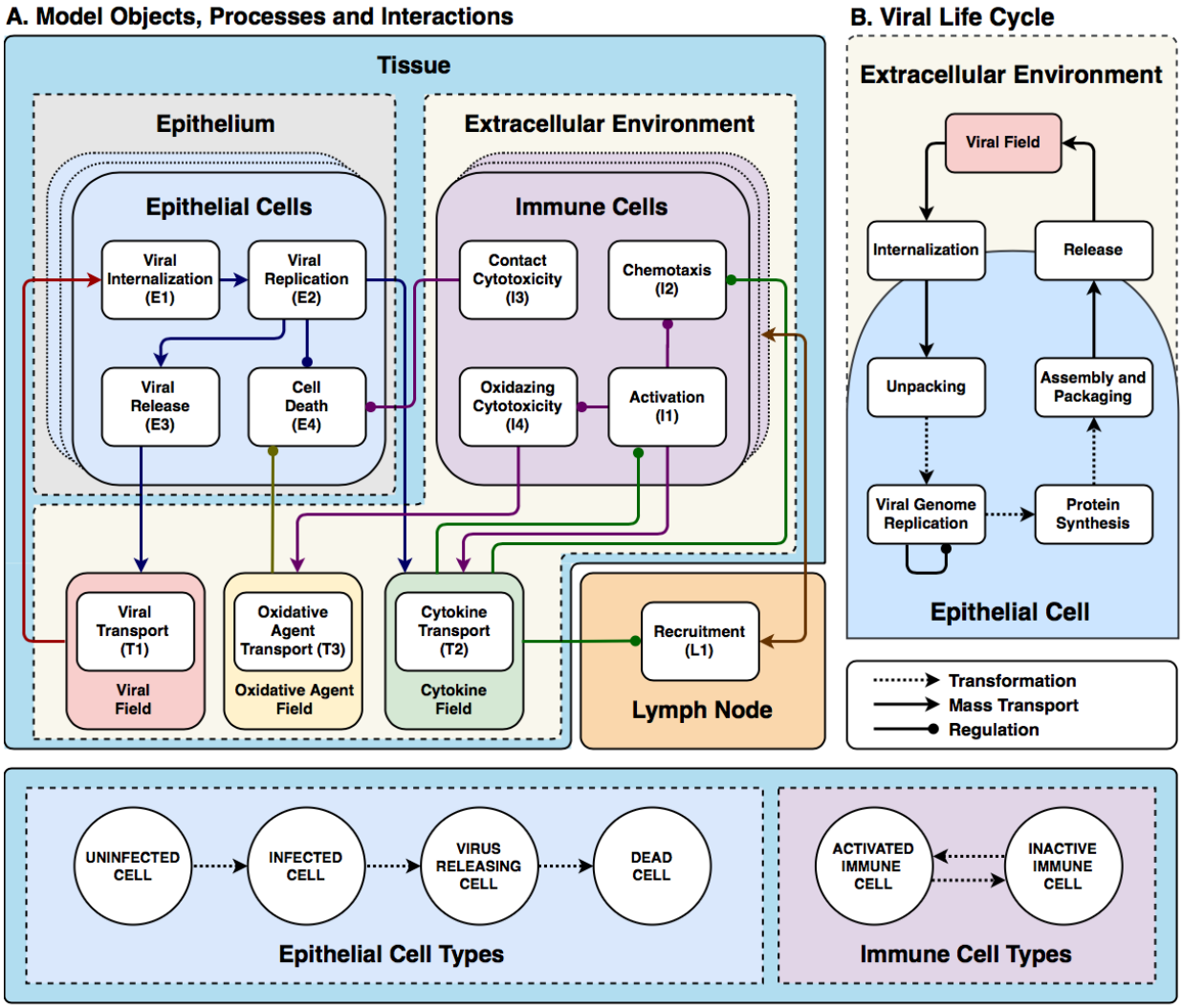


Fig 2. Full model schematic.

(A) Model objects, processes and interactions: Conceptual model of an epithelial tissue and lymph node. Schematic representation of the model objects, processes and interactions. Epithelial and immune cells refer to the two main classes of cells. Interactions occur within an extracellular environment, and a compartmental model of a lymph node controls immune-cell recruitment to the tissue. Together the epithelial-cell, extracellular-environment and immune-cell components represent the epithelial tissue. Each model object has associated processes that dictate its states and behaviors. Epithelial-cell processes include viral internalization (**E1**), viral replication (**E2**), viral release (**E3**) and cell death (**E4**). Immune cell processes include activation (**I1**), chemotaxis (**I2**), contact cytotoxicity (**I3**) and oxidative cytotoxicity (**I4**). *Activated* immune cells participate in oxidative cytotoxicity (**I4**) and secrete oxidative agents into the

oxidizing-agent field (**T3**). Activated cells become inactive after 1 hour. The virus field (**T1**), cytokine field (**T2**) and oxidizing-agent field (**T3**) describe spatially-varying densities of extracellular components. Field processes describe diffusive transport and clearance of material in the extracellular environment and activated transport to the lymph nodes. The lymph node is a single-compartment model whose pro- or anti-inflammatory state specifies the recruitment or removal rate (**L1**) of immune cells in the epithelial tissue. The transport of cytokines to the lymph node promotes its proinflammatory state. (B) Viral Life Cycle: Interactions in the viral internalisation, replication and release models. Schematic representation of inputs, outputs and interactions between stages of the viral replication model. Extracellular viral particles (represented as continuous fields) are internalized by the viral internalization model and initiate the viral replication model. The main stages of the viral replication model are: unpacking, viral genome replication, protein synthesis and viral assembly and packaging. The output of the viral replication model passes to the viral release model, which transfers newly assembled viral particles from the cells into the extracellular environment. (C) Cell types and transitions. Epithelial cells are of type *uninfected* if they have not yet internalized any virus (**E1**). They are of type *infected* if they have internalized virus, but are not releasing virus into the virus field (viral release **E3** is inactive). They are of type *virus releasing* if they are releasing virus into the extracellular virus field (*i.e.*, viral release **E3** is activated). Immune cells are initially *inactive* and do not participate in the oxidative cytotoxicity (**I4**) or chemotax towards higher concentrations of the cytokine field (**I2**). They become *activated* when they experience activation (**I1**). In all panels, dashed arrows with barbed heads represent transformations, solid arrows with barbed heads represent transport and solid arrows with lollipop heads represent regulation.

We simulate extracellular-virus particle density as a continuum field and particle transport and clearance as continuous diffusion and decay. We approximate the discrete process of a cell's internalization of a virus particle by a stochastic virus internalization event (**E1**) determined by time elapsed, the local concentration of the virus field, and the number of available cell-surface receptors on the cell. We simplify the complexity of viral replication into four steps: unpacking, viral genome replication, protein synthesis and packaging/assembly (**E2**, Fig 2B) [7,52–54]. The subcellular kinetics of viral replication determine the rate of release of new virus particles into the extracellular environment (**E3**). To represent the combined effect of the many types of virus-induced cell death, each

infected epithelial cell has a probability of dying that depends on the number of assembled viral particles inside the cell per unit time (**E4**).

We simplify the complex biochemistry of the many molecular signals active in epithelial tissues, which include chemokines, interferons and viral fragments, into a single generic extracellular cytokine field that acts both as a tissue-local and systemic pro-inflammatory signal. Infected epithelial cells and immune cells both secrete cytokine (**T2**). The cytokine field produces local immune effects such as activation (**I1**) and chemotaxis (**I2**) of immune cells. Activated immune cells can revert back to inactive immune cells when the cytokine signal ceases. The cytokine field also recruits immune cells to the tissue through long-distance signaling via the lymphatic system (**L1**). We model recruitment of immune cells to the simulation domain using an ordinary differential equation for the *immune signal* (***S***), which represents the balance between pro- and anti-inflammatory signaling and the delay due to antigen-presenting cell transport from the tissue through the lymphatic system to the lymph node and due to the time required for T-cell amplification. In the absence of infection, the lymph node maintains a small resident immune cell population in the tissue. Immune cells can cause epithelial cell death (**E4**) by three mechanisms: 1) contact cytotoxicity; 2) the bystander effect; and 3) through the release of an oxidative agent. Immune cells kill infected epithelial cells by contact cytotoxicity, in which case neighboring uninfected, infected and virus-releasing epithelial cells can also die through a bystander effect (**I3**). In regions of the tissue with high cytokine levels, immune cells secrete a diffusive oxidative agent to the environment (**T3**) that kills uninfected, infected and virus-releasing epithelial cells (**I4**).

Results

We begin by presenting our base multicellular model of viral infection in an epithelial tissue, along with a simulation for a baseline set of parameters and basic analyses. We then explore the simulation's dependence on critical parameters that are important to the dynamics of acute primary viral infection in airway epithelial cells. All simulations and spatial, population and system-level metrics presented in this section follow the specifications in *Simulation Specifications*. We performed simulations using the open-source modeling environment CompuCell3D [55]. *Downloading and running the simulation* provides instructions on how to run these simulations.

We initialize the simulations with periodic boundary conditions parallel to the plane of the sheet and Neumann boundary conditions perpendicular to the plane of the sheet. Initially, the extracellular environment does not contain any extracellular virus, cytokines, oxidative agents or immune cells. We introduce infection by creating a single infected epithelial cell at the center of the epithelial sheet, comparably to (but less than) initial conditions of similar works that model discrete cellular distributions [18,31] . To illustrate the full range of dynamics of viral infection in the presence of an immune response, we established a baseline set of parameters (Table 1) for which the immune response is strong enough to slow the spread of the infection, but insufficient to prevent widespread infection and death of all epithelial cells (Fig 3). While we have adjusted the parameters for the viral replication model to agree with reported time scales for SARS-CoV-2 replication *in vitro* [56], and we have selected parameter values in physiologically reasonable ranges, we have not attempted to match other model parameters to a specific tissue, virus or host species. Furthermore, this baseline parameter set is not unique with

respect to its purpose, in that many sets of parameters can generate appreciable but insufficient inhibition of spread of infection (see *Figs S17-S22*). Rather, as is shown in subsequent sections, this parameter set presents emergent dynamics of a theoretical virus and host immune response near, but not in, a regime of successful prevention of widespread infection, which is critical to showing the effects of underlying mechanisms on emergent dynamics and resulting outcomes.

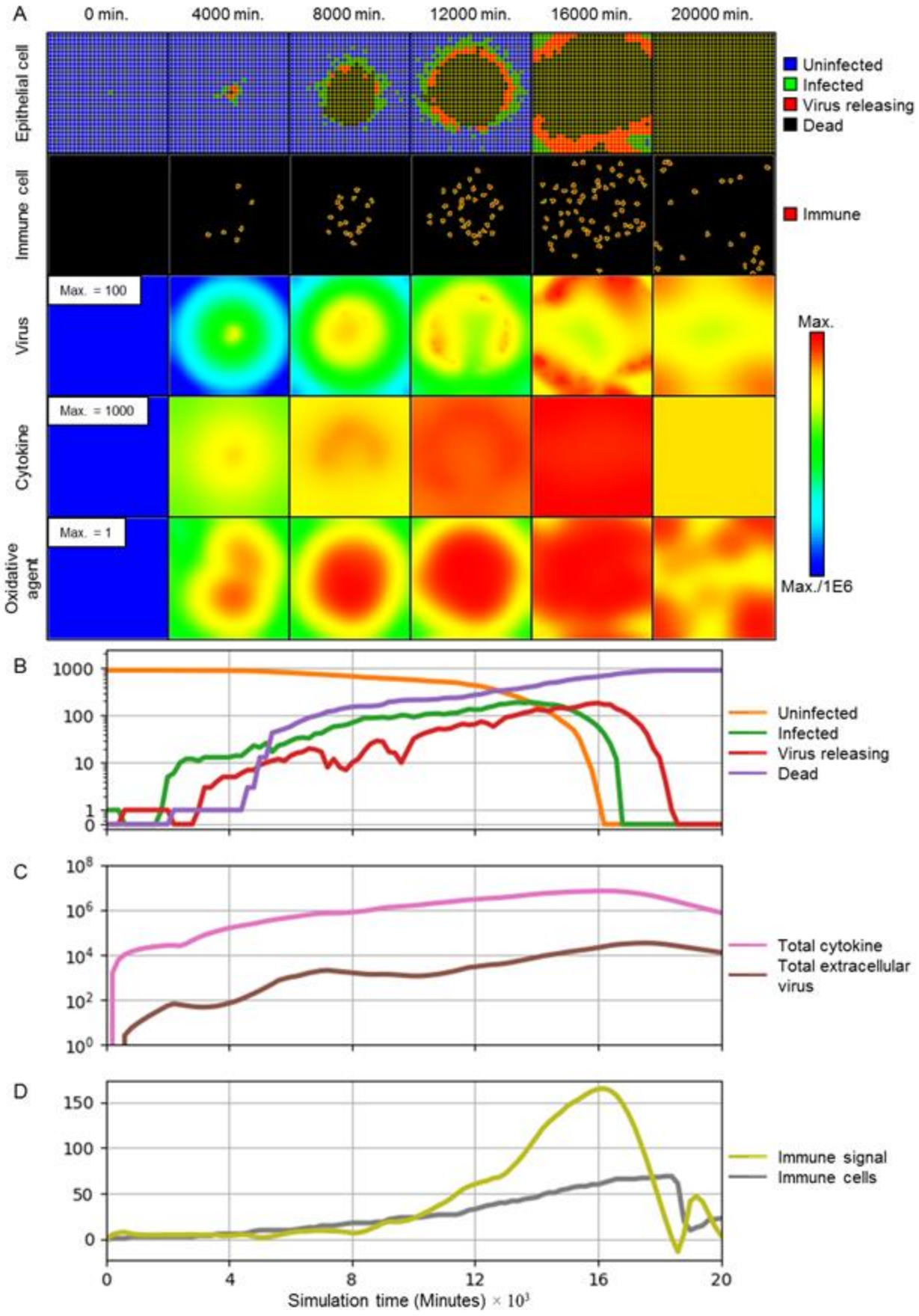


Fig 3. Simulation of the progression of infection in a patch of epithelial tissue of size 360 μm x 360 μm starting from a single infected cell for a representative simulation using the baseline parameters given in Table 1. (A) Snapshots of spatial configuration *vs* time, showing progression of a simulated infection. Columns, left to right: 0 minutes (time of initial infection), 4000 minutes (67 hours, 2 ¾ days) after infection, 8000 minutes (133 hours, 5 ½ days), 12000 minutes (200 hours, 8 ⅓ days), 16000 minutes (267 hours, 11 days), and 20000 (333 hours, 14 days) minutes. First row: epithelial-cell layer composed of uninfected (blue), infected (green), virus-releasing (red) and dead epithelial cells (black). Second row: position of immune cells in the extracellular environment layer. Third row: concentration of extracellular virus field. Fourth row: concentration of extracellular cytokine field. Fifth row: concentration of extracellular oxidative agent field. Fields are shaded on a logarithmic scale: red corresponds to the chosen maximum value specified in the first panel and blue corresponds to six orders of magnitude lower than the maximum value; colors saturate for values outside this range. (B-D) Simulation time series. (B) Number of uninfected (orange), infected (green), virus-releasing (red) and dead (purple) epithelial cells *vs* time on a logarithmic scale (0 values are overlaid at a non-logarithmic tick for clarity). (C) Total extracellular cytokine (magenta) and total extracellular virus (brown) *vs* time on a logarithmic scale. (D) Value of the immune recruitment signal S (yellow) and number of immune cells (grey) *vs* time on a linear scale. Simulations use periodic boundary conditions in the plane of the epithelial sheet, and Neumann conditions [57] normal to the epithelial sheet.

The infected cell immediately starts releasing cytokines into the extracellular environment. After an incubation period (~150 minutes, 2 ½ hours), the first infected epithelial cell (green) transitions from infected to virus releasing (red), and starts releasing viruses into the extracellular environment. Initial release of extracellular virus causes additional epithelial cells to become infected. Release of cytokines leads to delayed addition of immune cells to the simulation domain (Fig 3D). By 4000 minutes (67 hours, 2 ¾ days), the number of infected cells increases 10-fold and the epithelial cells start undergoing virally-induced death as the infection spreads radially outward from the initial site. The increase in the number of infected cells and the local cytokine concentration is accompanied by a similar increase in the local immune cell population. By 8000 minutes

(133 hours, 5 ½ days), the number of dead epithelial cells around the initial infection site increases sharply. Following this phase of rapid cell death, the number of infected, virus-releasing and dead epithelial cells continues to increase exponentially but at a slower rate. This transition results in the formation of an annular region of infected cells spreading radially outwards from the initial infection site (Fig 3A), analogous to the Fisher equation for deterministic front propagation [58]. Total extracellular virus and cytokine continue to increase exponentially. The increase in cytokine results in continued recruitment of additional immune cells. By 16000 minutes (267 hours, 11 days), the number of uninfected epithelial cells reaches zero and the number of infected and virus-releasing cells peaks. Despite the declining number of infected and virus-releasing epithelial cells, the delayed immune response continues to add immune cells to the tissue. After about 16000 minutes (267 hours, 11 days), the extracellular virus and the amount of cytokine decrease exponentially as the remaining virus-releasing epithelial cells die. By 20000 minutes (333 hours, 14 days), all epithelial cells die and many immune cells leave the tissue.

Classification of spatiotemporal infection dynamics

The rate at which infection propagates and the strength (speed and amplitude) of the immune response depend on multiple model parameters. Interplay between these rates leads to a wide spectrum of qualitatively-distinct spatiotemporal dynamics. The virus-receptor binding affinity k_{on} and the immune response delay coefficient β_{delay} are critical parameters affecting the rate of infection of epithelial cells and the strength of the

immune response, respectively. Larger k_{on} corresponds to a higher rate of infection propagation (increasing k_{on} increases the rate of internalization of extracellular viral particles into epithelial cells, see Equation (3) in *Models and methods*). Larger β_{delay} corresponds to weaker immune response (decreasing β_{delay} increases the strength of immune-cell recruitment, see Equations (12)-(14) in *Models and methods*).

Varying these two parameters around the baseline simulation values yields six patterns of spatiotemporal infection dynamics, ranging from unopposed infection to clearance (Fig 4). We defined these classes based on the transient dynamics and the final state of the simulation at 20000 minutes (333 hours, 14 days). We terminated the simulations at 20000 minutes (333 hours, 14 days) because we assume that, in real tissues, additional adaptive immune responses at this time generally lead to rapid viral clearance. As a result, any epithelial cells uninfected at the end of the simulation would likely remain uninfected if we included these additional immune mechanisms. We define the six patterns (*classes*) of infection dynamics as follows:

No immune response: a limiting case (corresponding to *in vitro* and organoid culture experiments on viral infection, which lack immune cells) that serves as a reference simulation showing the spread of unopposed infection. When the cellular immune response is absent, an infection front travels across the epithelium until all epithelial cells have died due to intracellular virus (Figs 4A and S1).

Widespread infection: when the immune response is weak (large β_{delay}) or the rate of infection propagation is large (large k_{on}), the immune cells kill enough infected epithelial cells to reduce both the concentration of extracellular virus and

the propagation of the infection front. However, at the end of the simulation no uninfected epithelial cells survive (Figs 3 and 4B).

Slowed infection: for moderate immune response (moderate β_{delay}) and a moderate rate of infection propagation (moderate k_{on}), both uninfected and infected epithelial cells and some extracellular virus remain at the end of the simulation (Fig 4C). These cases are functionally distinct from widespread infection, since even a single remaining uninfected epithelial cell could initiate tissue regeneration. In most cases of slowed infection, the numbers of infected cells and the extracellular virus continue to increase. A special case of slowed infection occurs when oscillations in the amount of virus (Fig S2A).

Containment: for strong immune response (small β_{delay}) and low to moderate rate of infection propagation (moderate k_{on}), a few infected and virus-releasing cells are present for most of the simulation. However, the immune cells eventually kill all infected and virus-releasing epithelial cells. At the end of the simulation, no infected or virus releasing cells remain, while uninfected cells survive and some extracellular virus remains in the extracellular environment (Fig 4D).

Recurrence: for strong immune response (small β_{delay}) and a fast infection propagation (large k_{on}), relatively few epithelial cells become infected early in the simulation. All infected and virus-releasing epithelial cells die. However, the remaining extracellular virus infects additional epithelial cells later on, restarting the spread of infection (Fig 4E).

Clearance: for strong immune response (small β_{delay}) and a slow infection propagation (small k_{on}), the number of infected and virus-releasing epithelial cells

417 goes to zero without recurrence and the extracellular virus drops below a threshold
418 (of 1/900 per cell in our analysis), rendering the frequency of recurrence negligible
419 (Fig 4F). A special case of clearance (**Failure to infect**) occurs when the initially
420 infected epithelial cells fail to infect any other epithelial cells (Fig S2B).
421
422 To quantitatively characterize simulation results, we measured the number of
423 uninfected, infected, virus-releasing and dead epithelial cells, the total number of immune
424 cells, the number of activated immune cells, the total amount of extracellular virus
425 (integral over the virus field), the total diffusive cytokine (integral over cytokine field), the
426 maximum total extracellular virus (over all simulation time) and the maximum total
427 diffusive cytokine (over all simulation time). Fig 4 shows these quantitative metrics,
428 together with a series of spatial configurations for all model components, corresponding
429 to each pattern of infection dynamics.
430

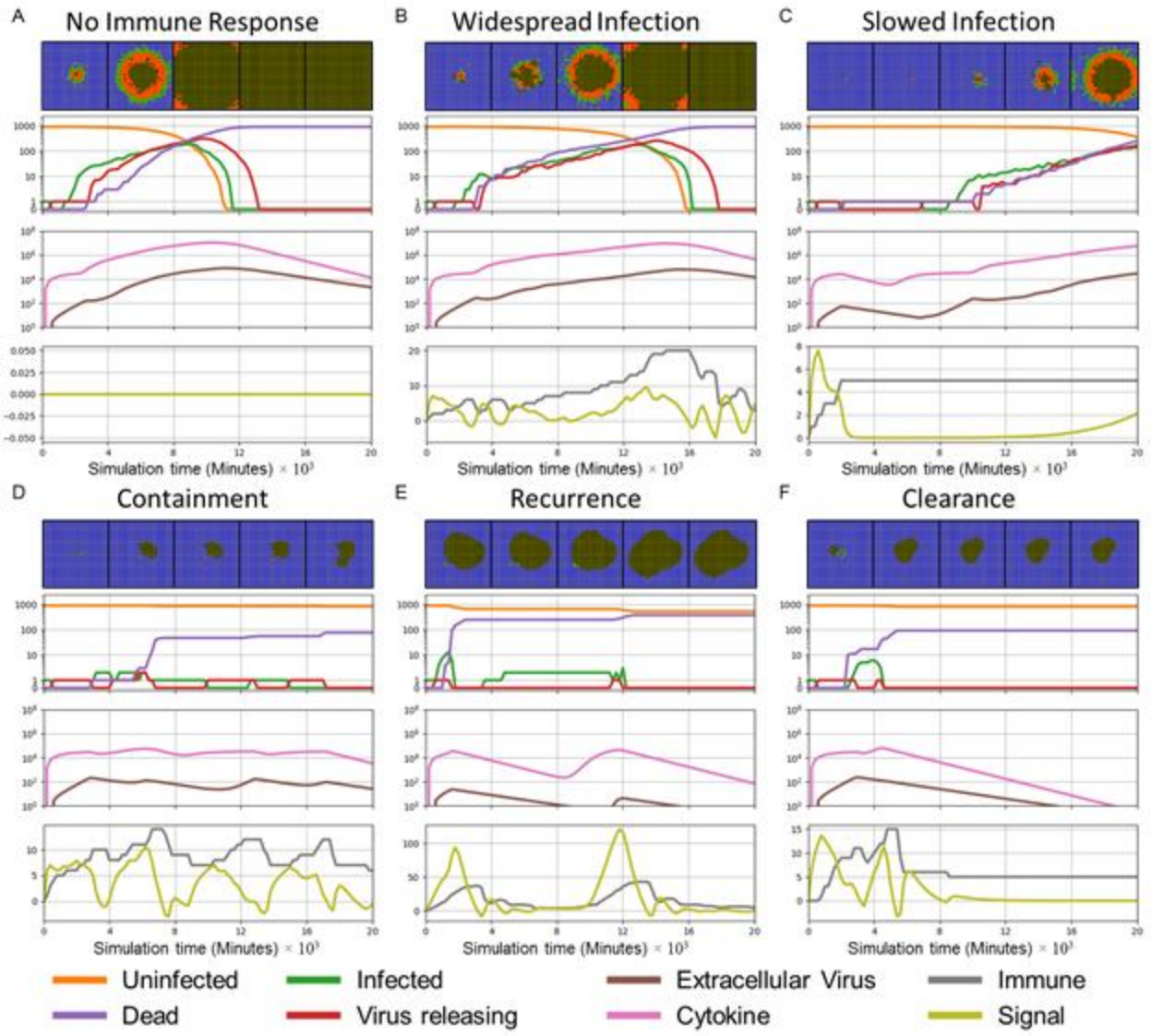


Fig 4. Patterns (classes) of spatiotemporal infection dynamics.

First row: snapshots of spatial configurations of the epithelial cells. Color coded: uninfected (blue), infected (green), virus releasing (red), dead (black). TImes from left to right 4000 minutes (67 hours, 2 ¾ days), 8000 minutes (133 hours, 5 ½ days), 12000 minutes (200 hours, 8 ⅓ days), 16000 (267 hours, 11 days) and 20000 minutes (333 hours, 14 days). The right border of each snapshot aligns with the corresponding time in the time series. Second row: number of uninfected (orange), infected (green), virus-releasing (red) and dead (purple) epithelial cells *vs* time on a logarithmic scale (with 0 included for clarity). Third row: total extracellular cytokine (magenta) and total extracellular virus (brown) *vs* time on a logarithmic scale. Fourth row: value of the immune recruitment signal S (yellow) and number of immune cells (grey) *vs* time on a linear scale. (A) No immune response: infection propagates unopposed until all epithelial cells have died from intracellular virus. (B) Widespread infection: weak immune response slows propagation of the infection,

but no uninfected cells survive at the end of the simulation. (C) Slowed infection: uninfected and infected epithelial cells coexist at the end of the simulation. (D) Containment of infection: no infected or virus-releasing epithelial cells remain, uninfected cells survive and virus remains in the extracellular environment at the end of the simulation. (E) Recurrence: the number of infected and virus releasing epithelial cells goes to zero, but persistent extracellular virus infects new epithelial cells later on. (F) Clearance: the number of infected and virus-releasing epithelial cells goes to zero and the level of extracellular virus is negligible at the end of the simulation. The model in (A) omits the immune response (components **L1**, **I1-I4**). All parameter values are as in Table 1 and Fig 3 except for k_{on} and β_{delay} (Table S1).

In the absence of an immune response, spread of the infection follows the diffusion of extracellular virus, resulting in concentric rings of different types of epithelial cells (see Fig 4A top row for epithelial cell types, Fig S1B shows the extracellular virus field). The viral propagation front, where uninfected epithelial cells transition to infected epithelial cells, moves radially outwards from the initial site of infection. Due to the stochastic internalization events, the front's outer contour is diffuse, with scattered infected epithelial cells ahead of the front, followed by a dense rim of infected and virus-releasing epithelial cells and a core of dead epithelial cells at the center. Similar waves, with a slower speed, are seen in the other cases where infection has occurred (Fig 4B and C top row).

Stronger immune response and lower rates of virus internalization promote containment of infection

To explore the effects of the rate of virus internalization and the strength of the immune response, we performed a multidimensional parameter sweep of the virus-receptor association affinity k_{on} and immune response delay coefficient β_{delay}. Variations in virus receptor association affinity represent factors that affect the binding affinity of cellular viral receptors (*e.g.*, ACE2 and TMPRSS-2 in the case of SARS-CoV-2) with a

467 virus (*e.g.*, mutations in viral coat protein or drugs to block viral entry). Variations in
468 immune response delay coefficient represent factors that affect the strength of the
469 systemic immune response (*e.g.*, anti-inflammatory corticosteroids, IL-7 treatment or age,
470 since older individuals tend to have slower adaptive immune responses)[36].

471 We ran ten simulation replicas for each parameter set, increasing and decreasing
472 the baseline parameter values 10-fold and 100-fold (Figs 5-7). For each simulation
473 replica, we examined the number of uninfected epithelial cells (Fig 5), the number of
474 infected epithelial cells (Fig 6), the total extracellular virus (Fig 7), the number of dead
475 epithelial cells (Fig S3), the number of virus-releasing cells (Fig S4) and the number of
476 immune cells (Fig S5). We identified regions of the parameter space where all ten
477 simulation replicas resulted in either containment/clearance (green-shaded subplots) or
478 widespread infection (orange-shaded subplots). In the intermediate regions (unshaded
479 subplots) different replicas for the same set of parameters exhibited other (and sometimes
480 multiple) classes of dynamics.

481

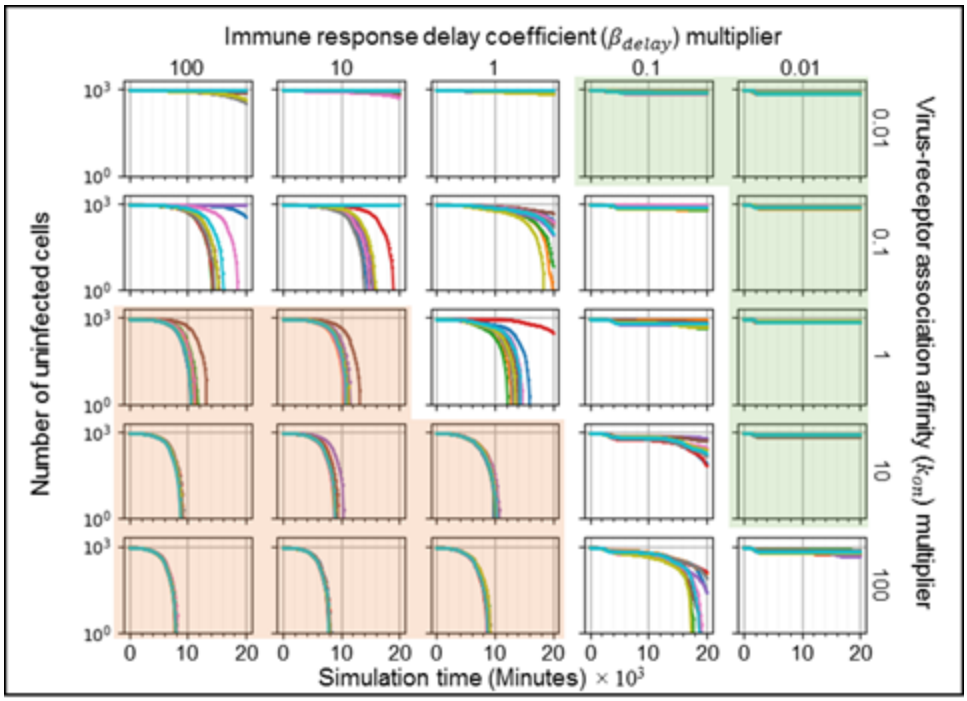


Fig 5. Sensitivity analysis of the number of uninfected epithelial cells vs time for variations in virus-receptor association affinity k_{on} and immune response delay coefficient β_{delay}, showing regions with distinct infection dynamics.

Logarithmic pairwise parameter sweep of the virus-receptor association affinity k_{on} and the immune response delay β_{delay} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1). The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes. See Fig 4 for the definitions of the classes of infection dynamics.

For large k_{on} and large β_{delay} (Figs 5-7, orange-shaded regions), all simulation replicas result in widespread infection and variability between simulation replicas is small. In this region, the initial release of virus into the extracellular environment results in a rapid increase in the number of infected and virus releasing epithelial cells early during the simulation, between 0 and 2000 minutes (0 to 33 hours, 0 to 1 ½ days) (Figs 6 and S4). Between 5000 and 20000 minutes (83 to 333 hours, 3 ½ to 14 days), the number of

uninfected epithelial cells rapidly decays to zero. As in all simulation replicas with a large β_{delay}, the immune recruitment signal (Fig S6) is less responsive to the cytokine signal produced by infected and virus-releasing epithelial cells and no significant recruitment of immune cells occurs throughout the simulations (Fig S5). The number of virus-releasing epithelial cells peaks around 7500 minutes (125 hours, 5 days), the level of extracellular virus peaks around 9000 minutes (150 hours, 6 ¼ days) and the number of dead epithelial cells peaks around 10000 minutes (167 hours, 7 days, Fig S3). With no remaining uninfected epithelial cells to infect, all remaining infected epithelial cells become virus-releasing epithelial cells, which then die, causing the total amount of extracellular virus to decrease.

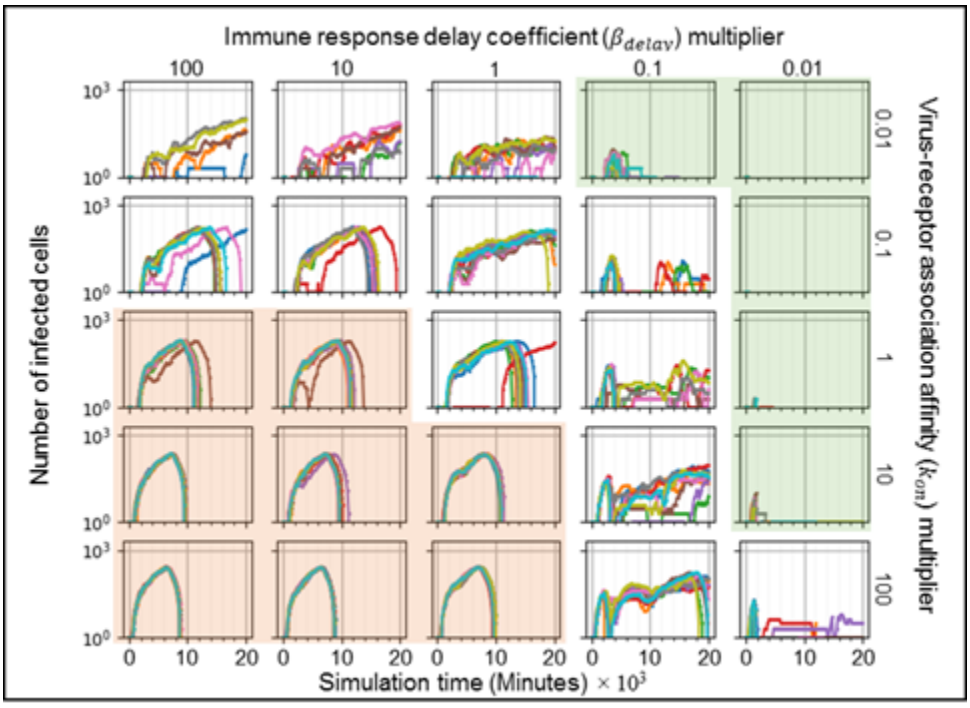


Fig 6. Sensitivity analysis of the number of infected epithelial cells vs time for variations in virus-receptor association affinity k_{on} and immune response delay coefficient β_{delay}, showing regions with distinct infection dynamics.

Same parameter sweep as Fig 5. The number of infected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes. See Fig 4 for the definitions of the classes of infection dynamics.

For small k_{on} and small β_{delay} (Figs 5-7, green-shaded subplots), all simulation replicas result in either clearance or containment and variability between simulation replicas is small. Initial release of virus to the extracellular environment results in a small change in the number of uninfected and infected epithelial cells. As in all simulation replicas with a small β_{delay}, the immune recruitment signal (Fig S6) is very sensitive to the cytokine produced by virus-releasing epithelial cells, resulting in rapid recruitment of immune cells and an early first peak in the population of immune cells around 2000 minutes (33 hours, 1 ⅓ days, Fig S5). In these simulations, the increase in the number of immune cells is followed by a rapid increase in the number of dead epithelial cells (Fig S3). The number of immune cells and the number of dead epithelial cells peak around the same time (2000 minutes, 33 hours, 1 ⅓ days), after which the number of immune cells decreases. With no remaining virus-releasing epithelial cells, the total extracellular virus decays until the infection is cleared. Increasing β_{delay} primarily increases the time when all infected epithelial cells have died.

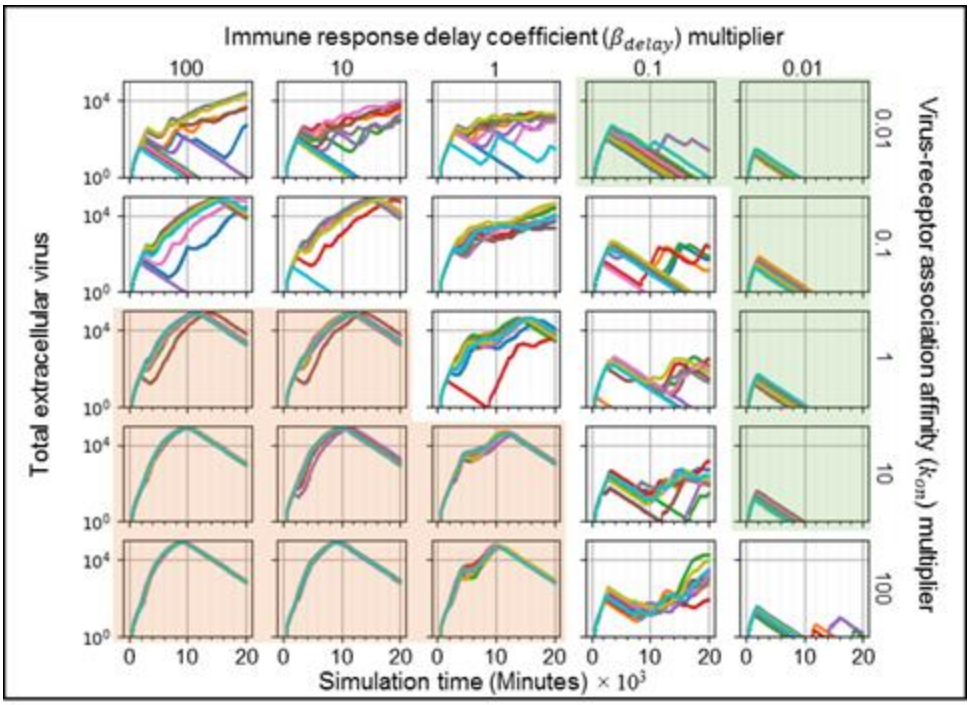


Fig 7. **Sensitivity analysis of the total amount of extracellular virus vs time for variations in virus-receptor association affinity k_{on} and immune response delay coefficient β_{delay}, showing regions with distinct infection dynamics**.

Same parameter sweep as Fig 5. The total amount of extracellular virus for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes. See Fig 4 for the definitions of the classes of infection dynamics.

For moderate to high k_{on} and moderate to low β_{delay} (right unshaded subplots in Figs 5-7), the rate of new infection nearly balances the rate of elimination of infected and virus-releasing epithelial cells, resulting in replicas showing clearance, contaminant, recurrence and slowed infection for the same parameter values, with very few cases of widespread infection. The initial release of virus into the extracellular environment by the first virus-releasing epithelial cell infects a moderate number of uninfected epithelial cells. The resulting cytokine secretion elicits a moderate to high response of the immune recruitment signal (Fig S6) and number of immune cells (Fig S5). The early recruitment

548 of immune cells leads to many epithelial cells dying before 4000 minutes (66 hours, 2 ⅔
549 days) (Fig S3). For a high probability of virus internalization (high k_{on}), even low amounts
550 of extracellular virus are sufficient to cause recurrence. For moderate to low k_{on} and
551 moderate to high β_{delay} (upper left unshaded subplots in Figs 5-7), rate of new infection
552 of epithelial cells is slightly faster than the immune system's response, resulting in a
553 combination of widespread infection, slowed infection and containment, and a few cases
554 of clearance. The immune system is only moderately responsive to the cytokine signal,
555 resulting in a slow to moderate increase in the immune recruitment signal (Fig S6) and in
556 the number of immune cells (Fig S5). Cases of clearance and containment occur for a
557 smaller final number of dead epithelial cells (Fig S3) compared to previously discussed
558 cases.
559
560 **Even moderate inhibition of genomic replication by antiviral therapies significantly**
561 **reduces the spread of infection, but only when initiated soon after infection**
562 Optimal therapeutic use of antiviral drugs requires considering the relationship
563 between molecular *efficacy* (how effectively the drug blocks a particular aspect of the viral
564 life cycle at saturation concentration), *potency of therapy* (the effect of the drug at a
565 molecular level at a given dose) and clinical *effectiveness* (how well the drug reduces the
566 severity or duration of the infection), as well as the tradeoff between side effects and
567 bioavailability. One drug might have moderate efficacy but have few side effects. Another
568 drug might have high efficacy, but have severe side effects at high doses that limit the
569 maximum tolerated dose or use of even moderate doses in prophylaxis. A drug might
570 also have a combination of beneficial and adverse effects (*e.g.*, it might reduce viral

571 replication early in infection, but also be immunosuppressive) [13,37]. Antiviral drugs like
572 Tamiflu retain their ability to block aspects of the viral life cycle (efficacy), but become
573 much less clinically effective as the time before treatment increases: (in adults Tamiflu is
574 most effective when given within 48 hours after exposure and thus is often used
575 prophylactically) [38].

576 In this section we use our model to show the trade-offs between time-of-use and
577 potency of a drug that targets viral genome replication in a host cell. Several antiviral
578 medications for RNA viruses reduce the net viral replication rate by inhibiting synthesis of
579 viral RNA by the viral RNA polymerase. We focus on RNA-synthesis blockers in this paper
580 because viral genome synthesis exponentially increases the production rate of viruses
581 per cell, while the other stages of viral replication have linear amplification effects (see
582 Equations (6)-(9) in *Models and methods*).

583 To simulate the effects of treatment that targets RNA synthesis using different drug
584 efficacies and times of administration, we generated a series of simulations in which we
585 reduced r_{max}, the replication rate of genomic material in the viral replication model
586 (Equation (7) in *Models and methods*), by different amounts and at different times in the
587 simulation. The “viral replication multiplier” represents the potency of the treatment, the
588 factor by which r_{max} is reduced (either a low dose with high efficacy, or a high dose with
589 a less efficacy). The “time delay of application” is the simulation time at which r_{max} is
590 reduced, which corresponds to the time after infection at which the treatment is
591 administered. To characterize therapeutic effectiveness, we distinguished three classes
592 of simulation outcomes:
593

Positive outcomes: effective treatment, where at least 50% of the epithelial cells remain uninfected at the end of the simulation (green-shaded subplots).

Negative outcomes: ineffective treatment, where less than 10% of the epithelial cells remain uninfected at the end of the simulation (orange-shaded subplots).

Intermediate outcomes: partially effective treatment, where between 10-50% of the epithelial cells remain uninfected at the end of the simulation (unshaded or intermediate-shaded subplots).

To characterize how the potency and time of initiation of treatment affect the dynamics of the simulation and treatment effectiveness, we examined the time courses of the number of uninfected epithelial cells (Fig 8), virus-releasing epithelial cells (Fig 9), the total amount of extracellular virus (Fig 10), the number of dead epithelial cells (Fig S7) and the number of immune cells (Fig S9). Intensity of green indicates the percent of simulation replicas that produced positive outcomes for a given set of parameters. Intensity of orange indicates the percent of simulation replicas that produced negative outcomes.

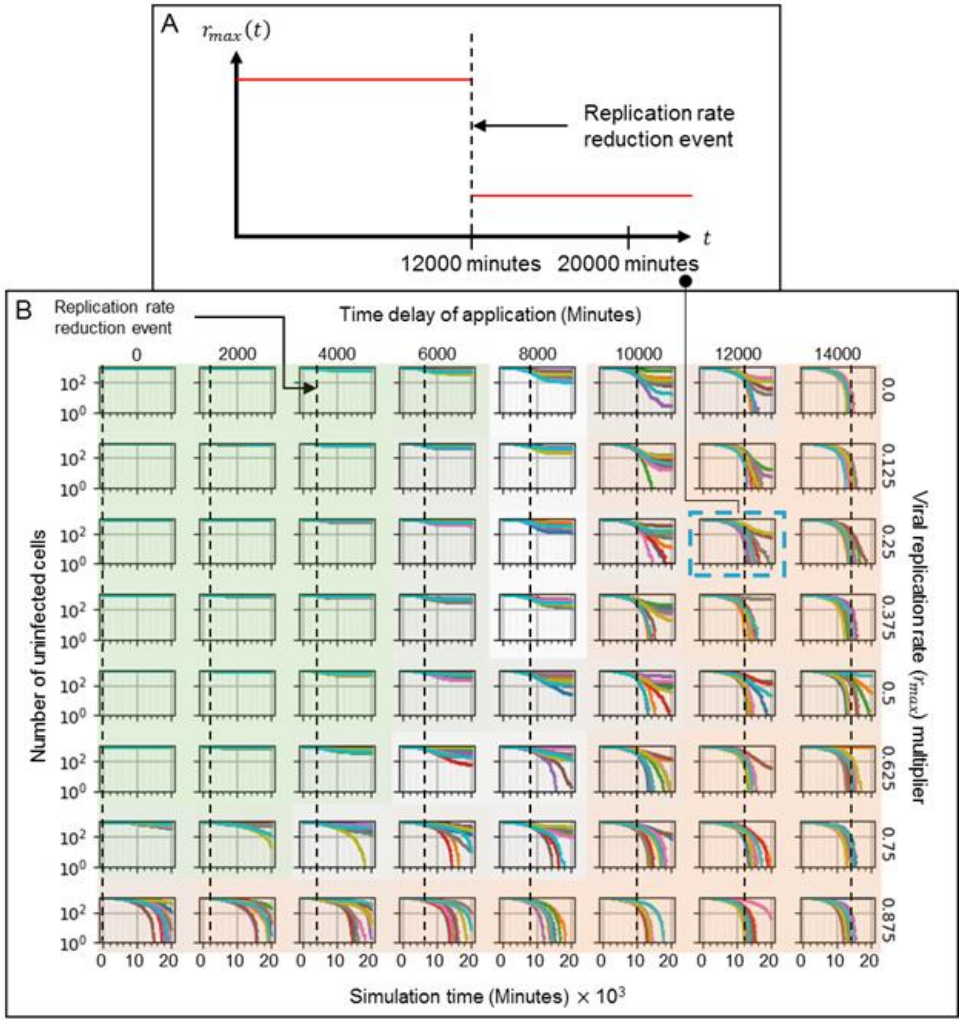


Fig 8. Number of uninfected cells vs time in simulations of a hypothetical drug treatment reducing the viral genome (*e.g.* RNA for SARS-CoV-2) replication rate (r_{max}) as a function of treatment potency and time of initiation of treatment.

Drug therapy is administered at a fixed time after infection and remains activated for the duration of the simulation. (A) Sample treatment, showing the time course of r_{max}. r_{max} is reduced by a multiplier which is one minus the potency of the drug at the given dose, 75% in (A), at a particular time of initiation of treatment (time delay of application), 12000 minutes (200 hours, 8 ⅓ days) in (A). (B) A parameter sweep of the potency of treatment (reduction in baseline viral replication rate r_{max}, vertical) and the time of treatment (dashed lines, horizontal) shows parameter regions where the majority of simulation replicas produce positive outcomes (green-shaded subplots), negative outcomes (orange-shaded subplots) and intermediate cases (intermediate shading or unshaded). Intensity of green and orange indicates the number of positive and negative outcome replicas for each parameter combination (treatment effectiveness). Green regions show that early intervention leads to positive outcomes (is effective) for most ranges of treatment potency, with

high numbers of uninfected epithelial cells at the end of the simulation for almost all simulation replicas. Orange regions show that late interventions result in mostly negative outcomes (ineffective treatment) regardless of the potency, and that outcomes are more variable between replicas, with both positive and negative outcomes for most parameter sets. The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

When the treatment is given early, while the level of extracellular virus is increasing rapidly and exponentially (before 6000 minutes, 100 hours, 4 days) after infection, most of the simulation replica outcomes are positive, showing effective treatment (Figs 9-11, green-shaded subplots). If the drug is administered prophylactically or very soon after infection (at 0 minutes) the treatment potency needs to be only 25% to achieve mostly positive outcomes (effective treatment). Increasing the time to treatment increases the potency required to achieve similar numbers of positive outcomes: the treatment is effective for a potency of at least 37.5% if administered by 4000 minutes (67 hours, 2 ¾ days), and at least 87.5% if administered by 6000 minutes (100 hours, 4 days). For all potencies greater than 12.5%, early intervention prevents significant increase in the number of virus-releasing cells (Fig 9, green-shaded subplots), and a small number of immune cells suffices to stop the spread of infection (Fig S9, green-shaded subplots). In this region, delaying treatment results both in a higher level of extracellular virus (Fig 10, green-shaded subplots) and more dead epithelial cells at the end of simulation (Fig 8, green-shaded subplots). With inhibited viral replication in the infected epithelial cells, the extracellular virus decays until it is mostly cleared by the end of simulation (Fig 10). Variability between simulation replicas for a given parameter set increases with both decreasing potency and increasing time of initiation of treatment.

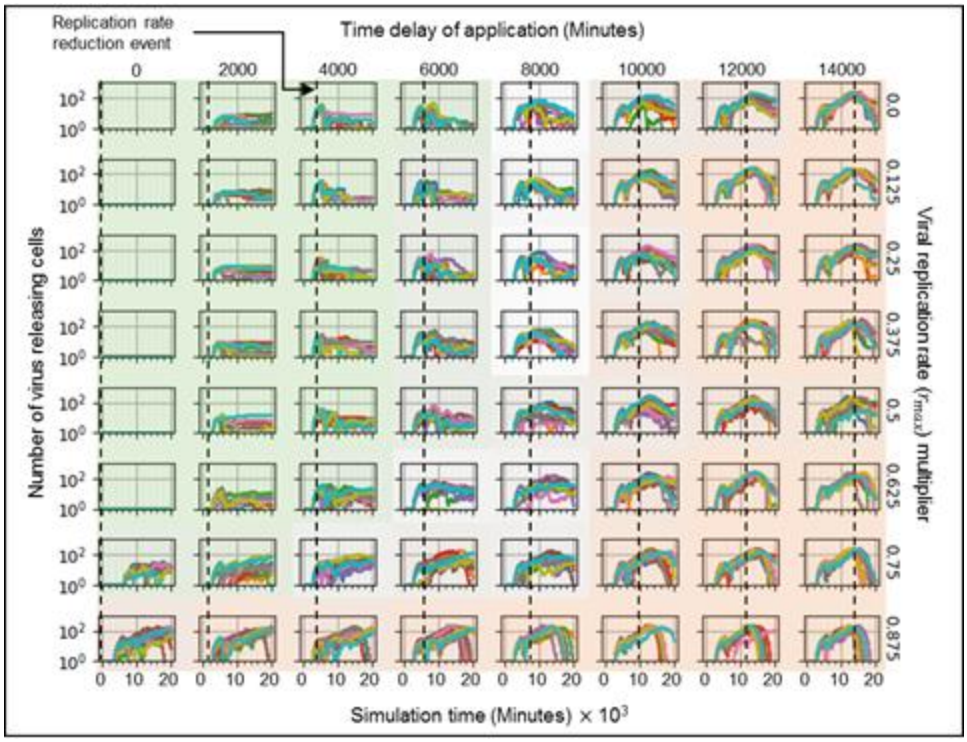


Fig 9. Number of virus releasing cells vs time in simulations of a hypothetical drug treatment reducing the viral genome (*e.g.* RNA for SARS-CoV-2) replication rate (r_{max}) as a function of treatment potency (one minus the viral replication rate multiplier) and time of initiation of treatment.

The number of virus-releasing epithelial cells stays low when the intervention occurs early during infection (when the amount of extracellular virus is increasing rapidly), but continues to increase when the intervention occurs later (when the level of extracellular virus is at or near its maximum in the untreated case). Parameter values, axis types and time-scale and shading as in Fig 8.

If the potency of the treatment is 12.5% (or less), most of the simulation replicas have negative outcomes (low effectiveness), even if the drug is administered prophylactically or soon after infection (at 0 minutes) (Figs 9-11, bottom row). In these cases, the time after infection at which the drug is given makes no significant difference to the treatment effectiveness. When the treatment is given late (time delay of application of at least 10000 minutes, 167 hours, 7 days), regardless of the potency of the drug, most simulation replicas have negative outcomes (Figs 9-11, orange-shaded regions). By the

665 time of treatment, a significant number of epithelial cells have been infected (more than
666 10% in most cases – Fig 9, orange-shaded regions) and a significant amount of virus has
667 been released into the extracellular environment (Fig 10, orange-shaded regions). In
668 addition, a significant number of epithelial cells have died (more than 10% in most cases
669 – Fig S7, orange-shaded regions) and significant recruitment of immune cells has
670 occurred (Fig S9, orange-shaded regions). For higher treatment potency, the level of virus
671 in the extracellular environment starts decreasing immediately after treatment, even when
672 a significant number of virus-releasing epithelial cells remain, indicating that viral
673 replication inside cells has been significantly reduced. Later intervention also increases
674 variability between simulation replicas and, although most simulation replicas have
675 negative outcomes, the same set of parameter values produced two distinct qualitative
676 outcomes (some more and some less favorable) for higher potency (Fig S11, orange-
677 shaded regions). Thus in a few cases, even late treatment can be effective.
678

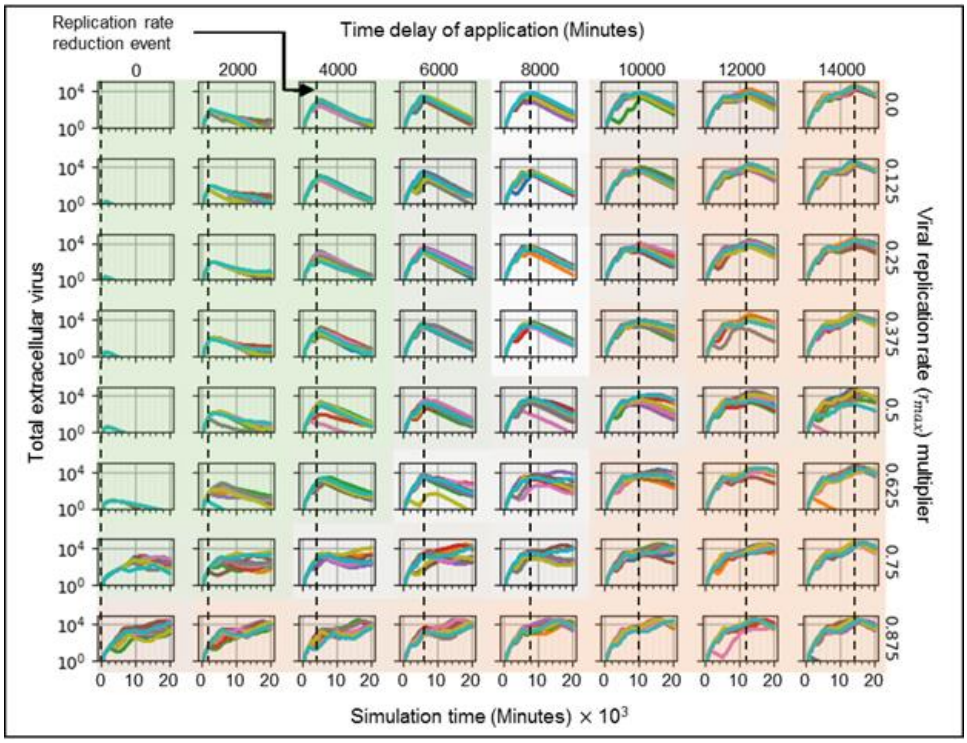


Fig 10. Levels of extracellular virus vs time in simulations of a hypothetical drug treatment reducing the viral genome (*e.g.* RNA for SARS-CoV-2) replication rate (r_{max}) as a function of drug potency (one minus the viral replication rate multiplier) and time of initiation of treatment.

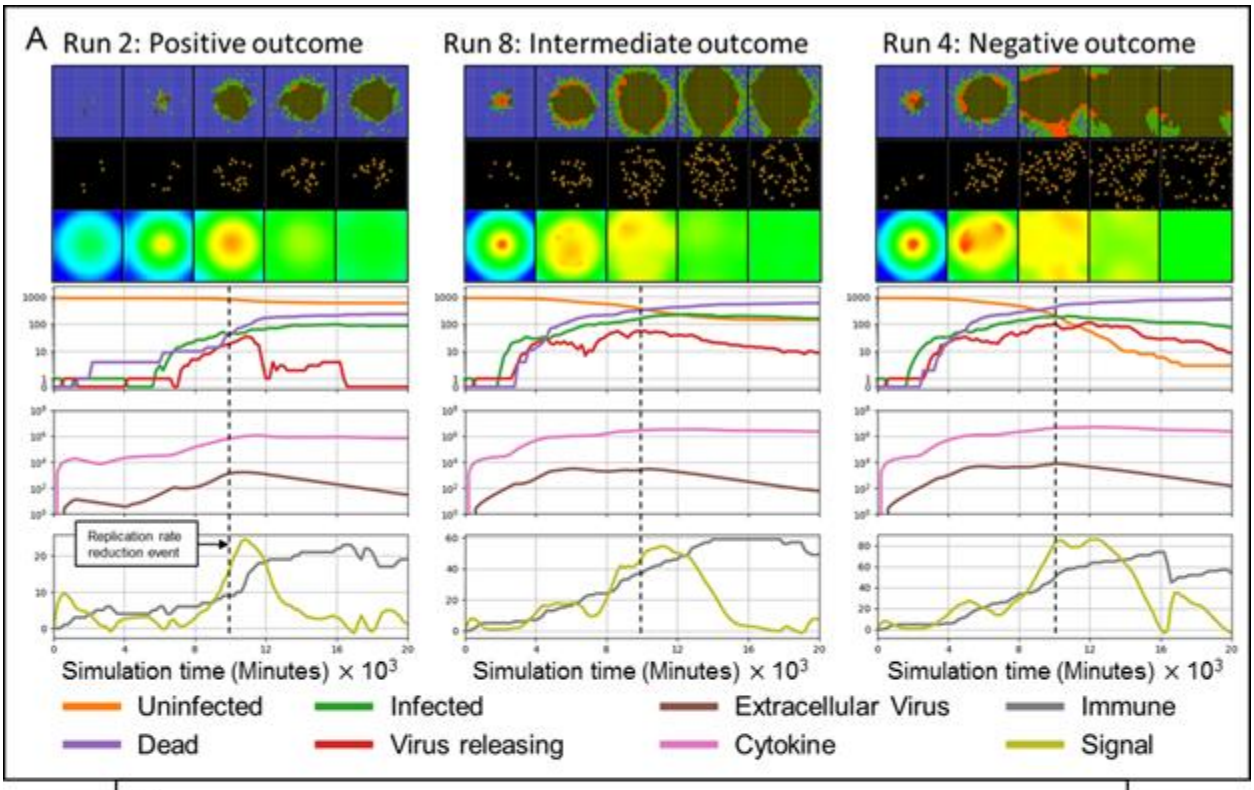
Extracellular virus is cleared or near-cleared when intervention occurs soon after infection. Parameter values, axis types and time-scale and shading as in Fig 8.

When the treatment is given at intermediate times (times between 6000 and 10000 minutes, 100 to 167 hours, 4 to 7 days), most simulation replicas have intermediate outcomes. For potencies above 50%, the fraction of uninfected epithelial cells at the end of simulation is relatively high (around 50%) and the treatment is usually moderately effective (Fig 8). For potencies below 50%, the number of virus-releasing epithelial cells remains approximately constant or continues to increase after treatment (Fig 9) and significant levels of extracellular virus remain at the end of the simulation, and so in most

693 cases the treatment is ineffective (Fig 10). In this regime, variability between outcomes

694 for the same parameter values is higher than for potencies above 50%.

695



696

Fig 11. Difference in treatment effectiveness for different simulation replicas for perfect treatment potency (0 viral replication rate multiplier) near the time when the extracellular virus amount would reach its maximum in the untreated case (10000 minutes, 167 hours, 7 days).

(A) Select simulation replicas for this parameter set showing the variety of possible outcomes (treatment effectiveness). Spatial results show the epithelial and immune cell layers, and the extracellular virus field, at 4000, 8000, 12000, 16000, and 20000 minutes (67, 133, 200, 267 and 333 hours, 2 ¾, 5 ½, 8 ⅓, 11 and 14 days). Cell type colors are the same as in Fig 3A. Virus field values are scaled as in Fig 3A. (B) Time series for all simulation replicas for the selected parameter set: Right column, from top to bottom, number of uninfected epithelial cells, number of infected epithelial cells, number of virus-releasing epithelial cells, number of dead cells. Left column, from top to bottom: total amount of extracellular virus, total amount of cytokine, number of immune cells and immune response state variable. All variables except the immune signal plotted on a logarithmic scale *vs* time.

A particular parameter set (time delay of application of 10000 minutes, 100% potency) produced simulation replicas that had instances of all three outcomes (Fig 11). In a simulation replica with a positive outcome (Run 2, Fig 11A), the first uninfected epithelial cell dies (as well as a few uninfected epithelial cells) before 4000 minutes (67 hours, 2 ¾ days), after which the total extracellular virus gradually decreases. At around 4000 minutes (67 hours, 2 ¾ days), an epithelial cell near the initially-infected cell becomes infected, causing a recurrence of infection, whose spread was stopped by the treatment. In contrast, simulation replicas with intermediate and negative outcomes (Runs 8 and 4, respectively, Fig 11A) have comparable, and significantly more, numbers of infected and virus-releasing epithelial cells at 4000 minutes (67 hours, 2 ¾ days), while total extracellular virus is greater in the replicas with a negative outcome than in the replicas with an intermediate outcome. For the positive outcome replica, after 10000 minutes (167 hours, 7 days), the remaining extracellular virus infects just a few individual epithelial cells throughout the tissue. For the intermediate outcome replica, after 10000

723 minutes (167 hours, 7 days) the number of infected epithelial cells continues to increase
724 until around 12000 minutes (200 hours, 8 ⅓ days) and then declines, while the number
725 of uninfected epithelial cells slightly decreases at the end of the simulation. For the
726 negative outcome replica, after 10000 minutes (167 hours, 7 days) the already depleted
727 number of uninfected epithelial cells continues to rapidly decrease to near zero.
728
729 **Model extensions**
730 In this section we demonstrate the deployment of extensions to the framework
731 described in *Models and methods*, which we will refer to as the "main framework" when
732 discussing extensions in this and subsequent sections, as well as particularization of the
733 framework to specific biological problems like a different virus. We accomplish this
734 through integration of an existing model in the literature of hepatitis C virus (HCV), and
735 elementary examples of adding models of heterogeneous epithelia and tissue recovery,
736 one model of which is constructed from another. We include schematic representations
737 where appropriate of the software implementation according to the architecture we have
738 developed (and will continue to develop) to support broad scientific use through rapid,
739 parallel model development, flexible model integration, and model sharing through our
740 publicly available online repository (see *Collaborative viral infection modeling*
741 *environment* for an overview of basic deployment, implementation, and public distribution
742 of extensions).
743
744 **Heterogeneous susceptibility inhibits spread of infection**. To demonstrate basic
745 extensibility of the framework to model additional complexity associated with viral

infection, we introduced a basic notion of heterogeneity to the epithelium of simulated scenario with the premise that not all epithelial cells can be infected (*i.e.*, some cells are unsusceptible, as in an actual heterogeneous respiratory epithelium) [22]. We implemented heterogeneous susceptibility by randomly selecting a fraction of the epithelial cell population at the beginning of simulation and setting the number of surface receptors of each to zero (*i.e.*, these cells have no surface receptors for internalization of the virus). Otherwise, all mechanisms and model parameter values used in simulations of heterogeneous susceptibility were the same as those used to generate results shown in Fig 3 (*i.e.*, all mechanisms described in *Models and methods* using the baseline parameter set). Implementation of randomly selected heterogeneous susceptibility is available in the add-on modules library and is hosted on our repository for public use with the module name “RandomSusceptibility”. For each set of replicas we simulated ten replicas using the RandomSusceptibility module while varying the fraction of unsusceptible cells (like different locations in the lungs), from 10% to 50% unsusceptible in intervals of 10% (Fig 12).

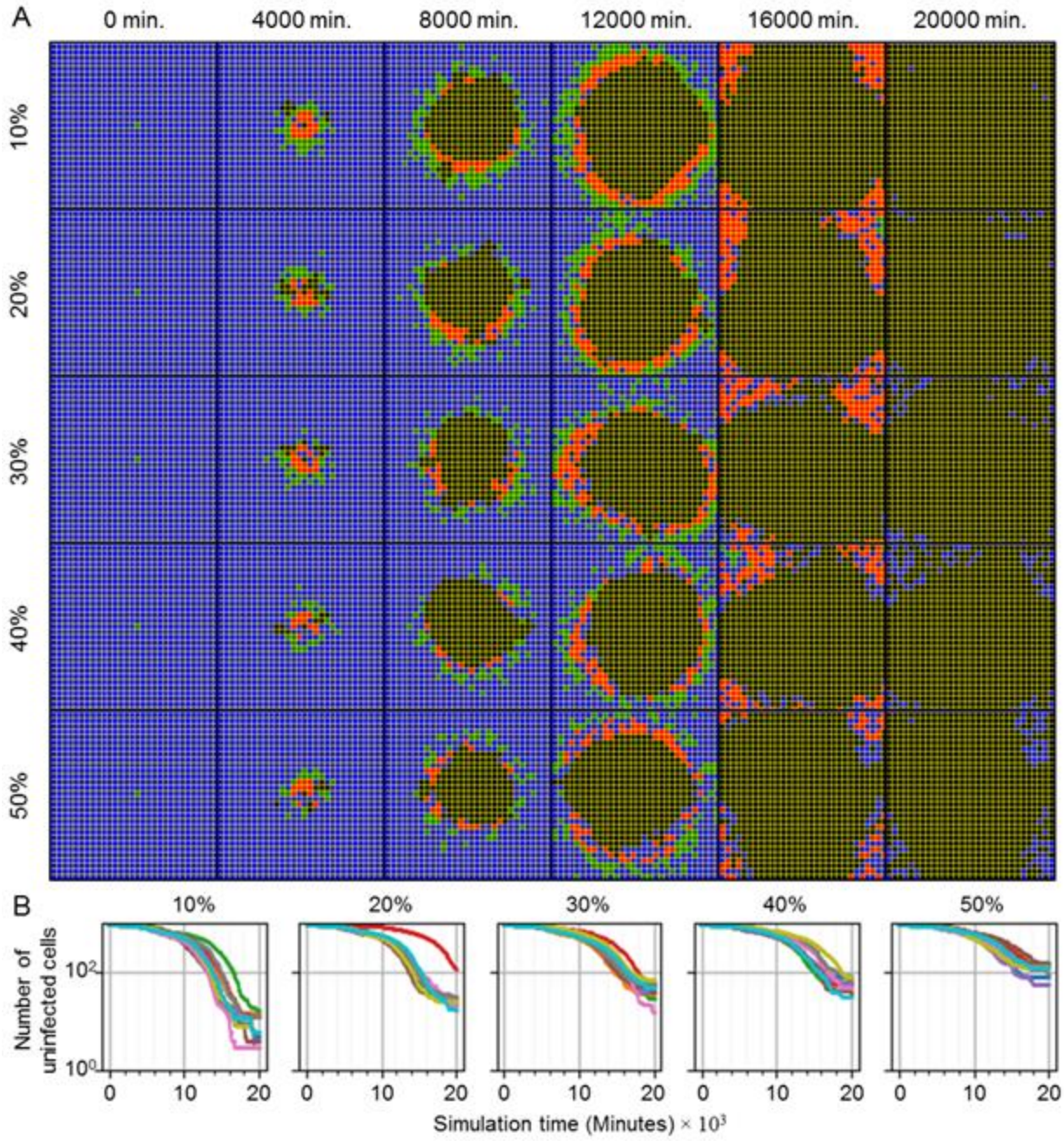


Fig 12. Randomly distributed unsusceptible cells inhibit but do not stop spread of infection throughout the tissue.

(A) Distributions of epithelial cells during simulation time (columns) for varying fraction of unsusceptible cells (rows). From top to bottom, replicas were simulated with 10%, 20%, 30%, 40% and 50% of epithelial cells unsusceptible to viral internalization. Cell type colors are the same as in Fig 3A. (B) Number of uninfected cells during simulation time for ten replicas of each fraction of epithelial cells unsusceptible.

769

770 For all fractions of unsusceptible cells, infection spread throughout the epithelial
771 sheet (Fig 12A). No replica produced the exact definition of widespread infection, since
772 all replicas had at least some remaining uninfected cells at the end of simulation.
773 However, only eight out of fifty total replicas had a few epithelial cells that were not either
774 uninfected or dead at the end of simulation (for infected cells, 1 replica for 10%
775 unsusceptible, and for virus-releasing 1 replica for 10% and 20% unsusceptible, and 2
776 replicas for 30%, 40% and 50% unsusceptible). The distributions of infected and virus-
777 releasing cells were notably different from all previous simulations with appreciable
778 infection in that both subpopulations were noticeably intermixed with uninfected
779 (presumably unsusceptible) cells and increasingly so with increasing fraction of
780 unsusceptible cells. Most of these intermixed, uninfected cells (particularly those nearer
781 to the initial site of infection) also died due to immune response mechanisms (*i.e.*,
782 bystander effect and oxidative killing).

783 The final number of uninfected cells at the end of simulation increased with an
784 increasing fraction of unsusceptible cells (Fig 12B). The rate of spread of infection
785 decreased with an increasing fraction of unsusceptible cells (as observed by inspection
786 in a rightward shift in the total number of uninfected cells). One replica for 20%
787 unsusceptible cells exhibited significantly delayed spread of infection, though this was not
788 observed in other replicas for any other fraction of unsusceptible cells.

789

790 **Integration of an explicit RNA synthesis model allows the spatiotemporal modelling**
791 **of hepatitis C virus infection**. The viral replication model described in *Models and*

methods describes the viral life cycle, from internalization to release, as occurring over four stages and in unitless quantities. As such, it is possible to integrate detailed models of various stages of the viral life cycle into the main framework through appropriate substitution of viral replication model terms and assignment of unit quantities as necessary (*i.e.*, one unit in the original viral replication model corresponds to a physical unit in an integrated model that explicitly describes a lifecycle process). Integration of a detailed model of some viral process then particularizes the virus represented in the main framework to the level of biological information introduced by the integrated model.

As a demonstration of model integration, we particularized viral replication of the framework to hepatitis C virus (HCV) using an existing model of subgenomic HCV RNA synthesis in Huh-7 cells [59]. The model of HCV RNA synthesis describes various aspects of subgenomic HCV replication like translation of HCV polyprotein in the cytoplasm, replication kinetics in vesicular-membrane structures and availability of host ribosomes, and explicitly models quantities of HCV RNA molecules, translation complexes, HCV polyprotein molecules, and necessary enzymes in both the cytoplasm and vesicular-membrane structures.

Integration of the HCV replication model casts the viral genome taking part in genome replication R of the viral replication module described in *Models and methods* as the cytoplasmic plus-strand HCV RNA molecules of the HCV replication model (denoted R_P^{cyt}, see *Integration of an explicit RNA synthesis model*). We assumed that internalized virus from the viral internalization module converts into cytoplasmic RNA, and that some of the decay of cytoplasmic RNA molecules described in the HCV model leads to the production of quantities produced by replicating viral genome from the viral replication

module. To relate the unitless viral replication model described in *E2 - Viral Replication* to the biological quantities of the HCV replication model, we assumed that one unit of replicating viral genome corresponds to 100 HCV RNA molecules, which was found to produce total infection dynamics comparably to the main framework (for comparison with results of this work, rather than for reproduction *in silico* of any specific HCV data) when using both the baseline parameter set demonstrated in Fig 3 and all model parameters reported in [59] when the virus-receptor association affinity coefficient k_{on} was increased by a factor of 100 (Table S2). The integrated HCV model is implemented in the add-on modules library using the aforementioned software architecture and is hosted on our repository for public use with the module name “HCVIntegrated” (Fig 13A).

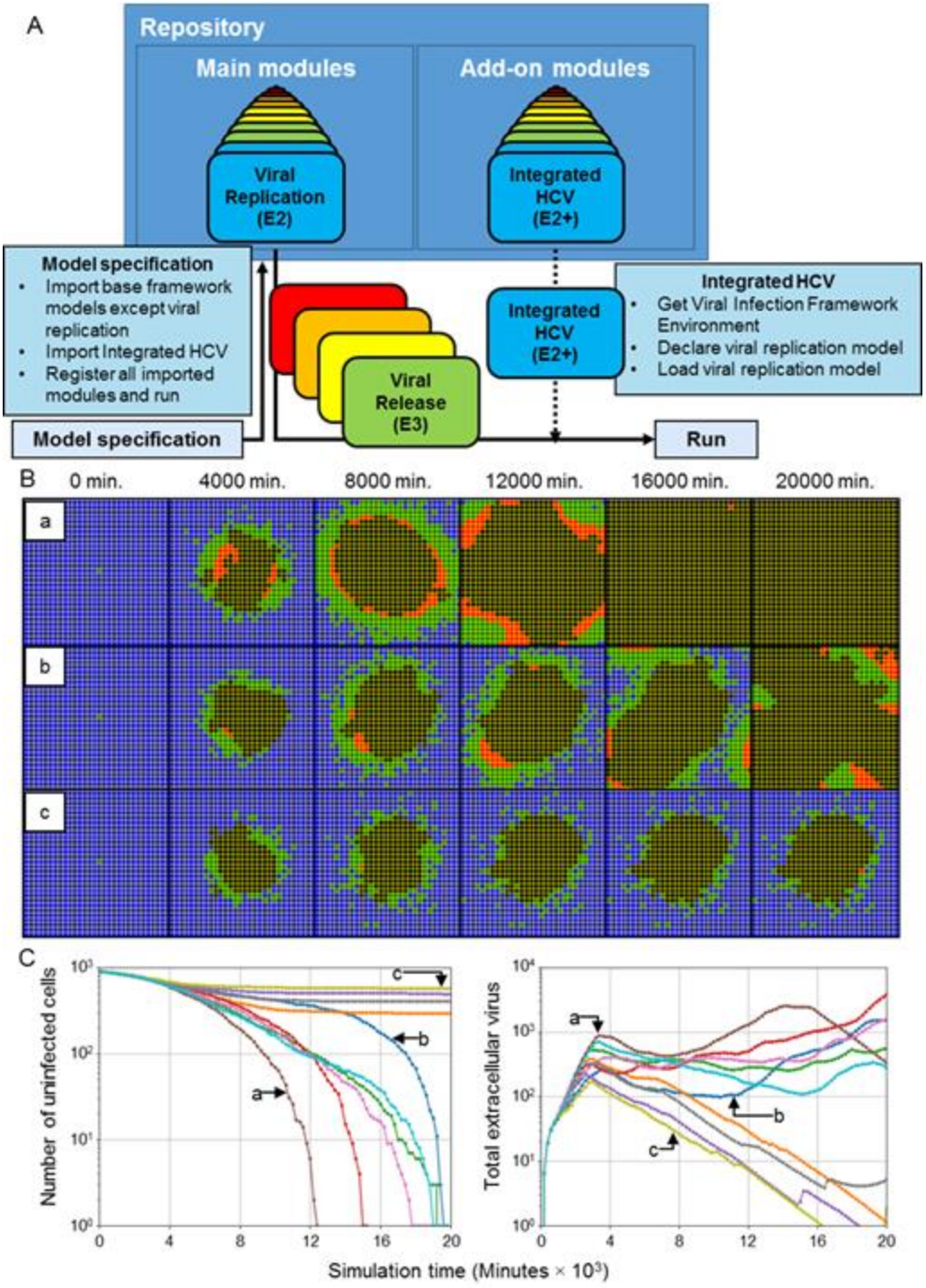


Fig 13. Basic integration of an explicit model of hepatitis C virus subgenomic replication in a spatial context presents stochastic outcomes in ten simulation replicas.

(A) Schematic of implementing the integrated HCV model using the available modules described in *Models and methods* ("Main modules") and add-on modules libraries hosted in the framework public repository. All main modules

are imported except Viral Replication, which is replaced with Integrated HCV. Modules used in a simulation are specified in model specification. The integrated HCV model module is available in the add-on modules library. (B) Distributions of epithelial cells during simulation time for select replicas (labeled “a”, “b” and “c”) using the integrated HCV model. Cell type colors are the same as in Fig 3A. (C) Number of uninfected cells (left) and total extracellular virus (right) during simulation time for ten replicas using the integrated HCV model. Select results shown in (B) are annotated according to replica labels “a”, “b” and “c”.

Ten simulations were executed using the integrated HCV model for a simulation time of two weeks using initial conditions similar to some used in [59], where the initially infected cell was seeded with 500 cytoplasmic viral RNA molecules (*i.e.*, initial $R = 5$, rather than initial $U = 1$ as in all other simulations). Spread of infection with the integrated HCV model produced comparable patterns of spread of infection, where an infection front radially advanced outward from the initial site of infection (Fig 13B). However, one notable difference was that the prominent band of virus-releasing cells during spread of infection only occurred in some simulation replicas (Fig 13B, replica “a”), while in other replicas only small, isolated groups of cells became virus releasing (Fig 13B, replicas “b” and “c”). Variability of outcomes was particularly notable among the ten simulation replicas. Some simulation replicas produced widespread infection at a comparable timescale to that of the baseline parameter set, between one and two weeks (Fig 13C). Such simulation replicas were those that produced comparable spatial distributions of infected and virus-releasing cells (specifically, with a prominent band of virus-releasing cells) to those of the main framework using the baseline parameter set. In other simulation replicas (*e.g.,* Fig 13B, replica “c”), slowed infection occurred due to early elimination of many virus-releasing cells. In such cases, the epithelial sheet had very few virus-releasing cells and scattered infected cells around the region of dead cells.

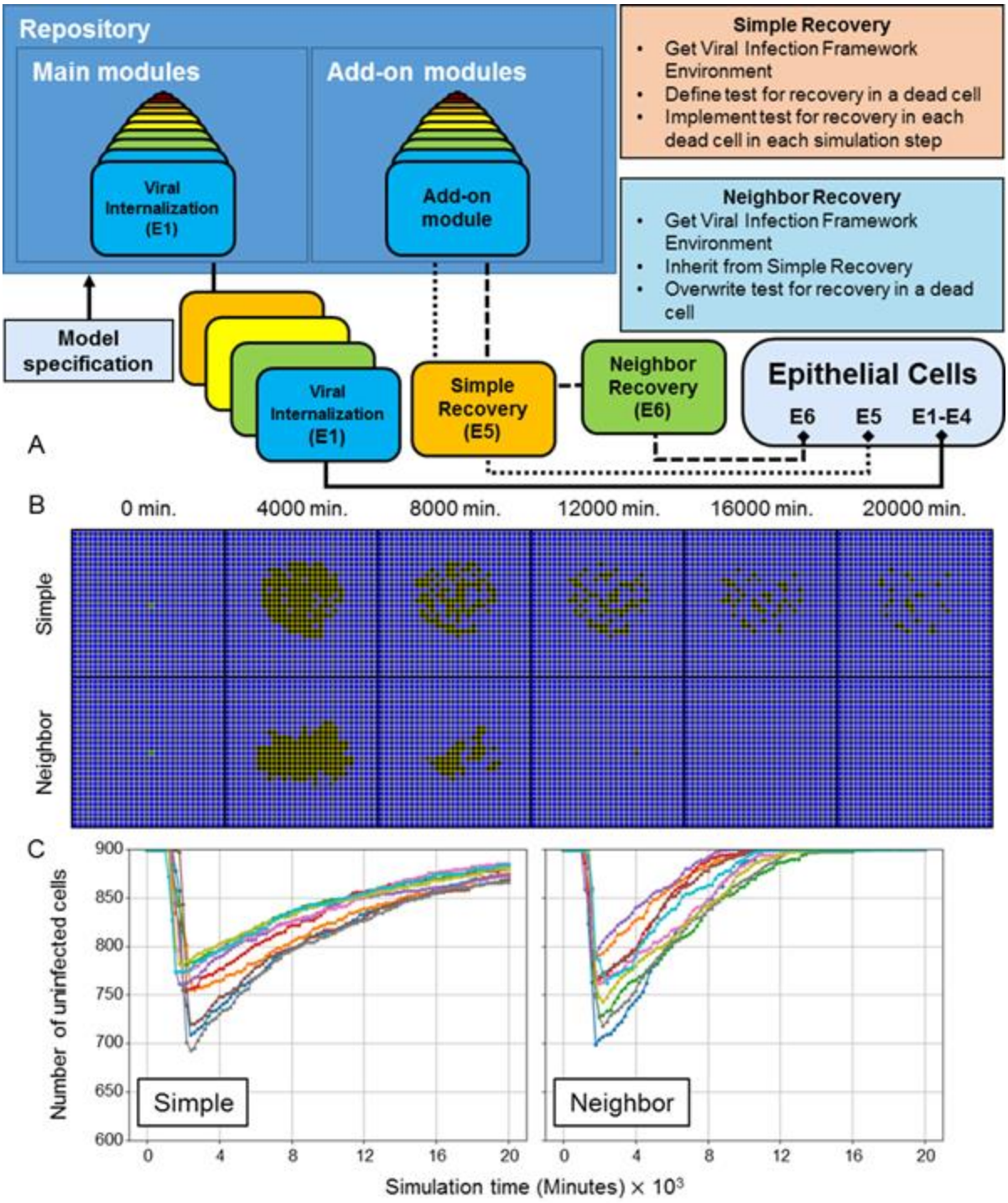
Many infected cells remained infected over a period of over a week, which was not observed using the viral replication model of the main framework. Such infected cells did not contribute to spread of infection, but rather diminished the likelihood of widespread infection by increasing the distance between uninfected and virus-releasing cells. By the end of simulation in replicas that produced slowed infection, total extracellular virus was negligible despite a significant number of infected cells, which presents an outcome not described in *Classification of spatiotemporal infection dynamics* that could be called "benign infection".

An extensible framework architecture enables the inclusion of tissue recovery. As a demonstration of modularity and extensibility, we developed, implemented and tested two models of tissue recovery, where dead cells are replaced over time with uninfected cells. Since epithelial cells in the simulated epithelial sheet are static, removal of dead cells and proliferation of uninfected cells was modeled as the changing of the type of a dead cell to uninfected (rather than explicitly modeling mitosis, [18]). To generate a scenario of viral clearance and significant tissue damage, we simulated the baseline parameter set with the parameter variations in the top-right corner of Figs 5-7, where viral internalization is severely inhibited and the immune response is very strong and fast (*i.e.*, parameters k_{on} and β_{delay} were reduced by a factor of 100 so that the virus is cleared but many uninfected cells die).

In the first model of tissue recovery, called "Simple Recovery", dead cells are replaced by an assumed layer of proliferative cells underneath the simulated epithelial patch, and so each dead cell has a fixed probability of recovering. We approximated the

879 probability of recovery based on an assumed onset of tissue recovery of 7 days, in which
880 case the probability of recovery for each dead cell over a 20-minute simulation step was
881 1.98×10^{-3}. In the second model of tissue recovery, called “Neighbor Recovery”, dead
882 cells are replaced by nearby uninfected cells similarly to wound healing, and so each
883 dead cell has a probability of recovery equal to the number of neighboring uninfected cells
884 multiplied by a coefficient. For comparison of results to those from the Simple Recovery
885 add-on module, we used the same probability coefficient value such that the probability
886 of recovery according to Simple Recovery for each dead cell over a 20-minute simulation
887 step was 1.98×10^{-3} per unit of contact area with neighboring uninfected cell (measured
888 in number of neighboring lattice sites).

889 Both models were implemented in CompuCell3D in the add-on modules library
890 using the aforementioned software architecture and is hosted on our repository for public
891 use. The Simple Recovery model is hosted with the module name “RecoverySimple”, and
892 the Neighbor Recovery model is hosted with the module name “RecoveryNeighbor”.
893 Since the only difference between the two recovery models is the criterion for cell recovery
894 (*i.e.*, whether a fixed probability, or a probability according to the neighborhood of a cell),
895 the implementation of the Neighbor Recovery model inherits all features of the Simple
896 Recovery model implementation using basic Python class inheritance, and required
897 overwriting only one function that implements a criterion of cell recovery during
898 development (Fig 14A, see *Extending a model in CompuCell3D*).
899



900

901 **Fig 14. Even simple model extensions and extensions of extensions can produce notably different**
902 **spatiotemporal emergent features.**

903 (A) Schematic of implementing a modified epithelial cell model using the available modules described in *Models and*
904 *methods* ("Main modules") and add-on modules libraries hosted in the framework public repository. Modules used in a

simulation are specified in model specification. The Simple Recovery and Neighbor Recovery model modules are available in the add-on modules library. (B) Distribution of epithelial cells during simulation time using the Simple Recovery (top) and Neighbor Recovery (bottom) models. Cell type colors are the same as in Fig 3A. (C) Number of uninfected cells during simulation time for ten simulation replicas using the Simple Recovery (left) and Neighbor Recovery (right) models.

All simulation replicas for each recovery model began with tissue insult due to oxidative killing by a strong and fast immune response through the first few days (Fig 14B). All simulation replicas experienced a loss of approximately 100-200 uninfected cells (Fig 14C). An increasing number of uninfected cells clearly demonstrated immediate effects of modeling recovery after oxidative killing subsided (a trend not seen in previous results), where the Simple Recovery model almost completely replenished all killed uninfected cells by the end of simulation, and the Neighbor Recovery model replenished all killed uninfected cells between 8,000 and 13,000 minutes. Neighbor-dependence of the Neighbor Recovery model generated a significantly different distribution of dead cells after oxidative killing, where dead cells were scattered throughout the original region where cells died when using the Simple Recovery model while the region where cells died shrank over time when using the Neighbor Recovery model.

Discussion

Our spatial, multicellular model of primary acute viral infection of an epithelial tissue includes key aspects of viral infection, viral replication and immune response. By investigating sensitivity to model parameters and simulating drug therapies, we identified six distinct spatiotemporal classes of infection dynamics based on the model's transient

929 behaviors and final configurations. Each of our simulation-defined classes corresponds
930 to biologically or clinically observable factors and outcomes. The case of no immune
931 response would be useful for analyzing *in vitro* experiments (*e.g.*, organoids). Widespread
932 infection corresponds to an initial infection that is likely to spread to surrounding tissue
933 and cause major tissue damage. Slowed infection corresponds to an initial infection
934 whose spread is more likely to be eliminated by the adaptive immune response.
935 Containment corresponds to immune-cell elimination of all infected cells but where
936 remaining extracellular virus could result in new sites of infection elsewhere. Recurrence
937 corresponds to the situation when new lesions form within the observed tissue patch.
938 Clearance corresponds to immune-cell-based elimination of all infected cells and
939 extracellular virus (classical viral clearance).

940 We showed that key parameters of the model, such as those affecting viral
941 internalization (*i.e.*, virus-receptor association affinity k_{on}), can lead to both
942 containment/clearance (*e.g.*, small k_{on}, Figs 5-7) and widespread infection (*e.g.*, large
943 k_{on}, Figs 5-7). Multidimensional parameter sweeps showed how the interplay between
944 immune response (*e.g.* immune response delay coefficient β_{delay}) and viral spread could
945 lead to widespread infection (*e.g.*, large β_{delay}, large k_{on}, Figs 5-7), rapidly cleared
946 infection (*e.g.*, small β_{delay}, small k_{on}, Figs 5-7) or containment/clearance after substantial
947 damage (*e.g.*, small β_{delay}, moderate k_{on}, Figs 5-7). Some of these outcomes would be
948 expected biologically (*e.g.*, very fast internalization with a slow immune response is likely
949 to lead to widespread infection; faster and stronger immune responses should control the
950 spread of viral infection within the tissue [Figs 5-7]) and would also occur in deterministic
951 non-spatial models. Others, like the coexistence of replicas with containment/clearance

or failure to control for the same parameter set, are less expected, and could not occur in a deterministic non-spatial model (though they might occur in some stochastic non-spatial models). We have observed this interplay of parameters, as well as the potential for stochastic outcomes, in variations of other parameters of the model, whether related to spatial (*e.g.*, viral and cytokine diffusion coefficients) or deterministic and stochastic cellular aspects (oxidative agent threshold for death and virally-induced apoptosis dissociation coefficient, see *Figs S17-S22*).

We studied the influence of timing and potency of an RNA-polymerization inhibitor like remdesivir [60] on the spread of viral infection within tissue (Figs 9-11). As expected, in our model, drugs with this mode of action can improve viral control in tissue if administered prophylactically at high potency, and their effectiveness decreases the later they are administered. Less obviously, the lower-left region of Figs 9-11 shows how therapies with even reduced potency could control the infection when administered sufficiently early, consistently with predictions from a deterministic, non-spatial ODE model published after submission of this manuscript for publication (though the mode of action is not explicitly described) [61]. While we expect prophylactic or early treatment at the same potency to be more effective than later treatment, our model suggests that, for antivirals, time of treatment is a more significant factor than potency in determining the effectiveness of the therapy. Our model thus suggests that drugs that interfere with virus replication are significantly more effective if used even at very low doses prophylactically or very soon after infection, than they would be if used even at a high dose as a treatment given later after exposure. Specifically, a prophylactic treatment in simulation which reduces the rate of viral RNA synthesis by only 35% (35% potency) is more effective than

a treatment with 100% potency given two and a half days after infection, and has about the same efficacy as a treatment with 50% potency given one day after infection. Our model also showed that because of stochasticity in viral spread, later treatment at moderate to high potency may still be effective in a subset of individuals.

Both parameter sweeps had regions with little variation in outcome between replicas (*e.g.*, the upper-right and lower-left corners of Fig 5). In regions of the parameter space between these extremes (*e.g.*, the unshaded areas in Figs 5 and 9), different replicas showed dramatically different outcomes. One such parameter set in our drug therapy simulations produced three distinct qualitative outcomes (*i.e.*, positive, intermediate and negative outcomes, Fig 11). For these parameters, replica outcomes were particularly sensitive to stochasticity early in infection when only a few cells were infected (Fig 11A), with delayed spread of infection from the first infected cell producing more positive outcomes. Simulation replicas with negative outcomes (Fig 11A, Run 8) had higher extracellular virus levels at earlier times than those with intermediate outcomes (Fig 11A, Run 4), even though the fraction of each cell type was similar. Since the viral replication module is deterministic, the primary cause of this difference is the spatial distribution of cells. Spatial structure (*e.g.*, infection of neighboring cells), stochastic events (*e.g.*, early cell death of infected cells before significant virus release) and cell-to-cell variation (*e.g.*, difference in viral release between cells) all affect the variation between replicas.

Differences in spatiotemporal dynamics and variability of outcomes thus critically depend on the ability of the model to resolve the spread of virus and immune response spatially. The intrinsic stochasticity of many model processes makes the spatial patterns

of the infection front and distribution of tissue damage nontrivial. The spectrum of outcomes in our parameter sweeps (Figs 5-7 and 9-11) depends not only on parameter values and model immune response, but also on the emergent spatial patterns of cytokine and virus fields (*e.g.*, variations within the infection front expose different numbers of uninfected epithelial cells to the immune response). Such stochastic and spatial aspects can also introduce new considerations to ODE models that have been primarily employed in a non-spatial context. For example, as described in the original presentation of the HCV model that was integrated in *Integration of an explicit RNA synthesis model allows the spatiotemporal modelling of hepatitis C virus infection*, the subgenomic kinetics of the HCV model require a minimum number (seven) of cytoplasmic viral RNA molecules to reach a saturated state. When employing the HCV model in a multicellular, heterogeneous context, insufficient internalization of a spatially heterogeneous extracellular viral field for subgenomic replication to produce rampant viral production, leading to insufficient cytokine signaling to invoke further immune response, makes possible the so-called outcome of benign infection.

Future perspectives

Our modeling framework can improve with the inclusion of additional cellular and immune mechanisms discussed in Fig 1. The modularity of model modules and built-in extensibility of the publicly available software implementation enables us, and other interested members of the scientific community, to accomplish such activities collaboratively or independently, concurrently, and even when in theoretical disagreement (see *Collaborative viral infection modeling environment*). Modules

accounting for viral clearance, tissue recovery and persistent adaptive immune response can be added to model later stages of disease progression (as demonstrated in *Model extensions*). The current immune model does not include important signaling factors (*e.g.*, interferon-induced viral resistance in epithelial cells) and the different roles of tissue-local and systemic signals (*e.g.*, various cytokines). It also omits many cell types associated with both innate and adaptive immune response and their roles (*e.g.*, viral scavenging by macrophages, relaying and amplification of immune signals by dendritic cells). Of special interest to results like those presented in this work is the effect of specific roles by individual immune cell phenotypes on emergent dynamics and outcomes, considering that the timing of their activities during progression of events can be quite different (*e.g.*, early neutrophil release of oxidative agent contrasted with later effector T-cell contact-mediated killing). Such details, which we are currently pursuing, are particularly important for using framework to interrogate the spatiotemporal details of the immune response, which are poorly understood. The model does not currently consider the production and role of antibodies in the humoral immune response or tissue recovery after damage (like the demonstration models presented in *An extensible framework architecture enables the inclusion of tissue recovery*). The model also greatly simplifies the structure of the epithelium and its environment, but could be easily generalized to a detailed, three-dimensional geometry, albeit at the cost of computational performance.

Our current results suggest priorities for improving the biological realism within existing modules, and for including modules representing additional biological components and mechanisms. We are currently implementing virus-scavenging by immune cells and local antiviral resistance due to Type 1-IFN paracrine signaling by

epithelial cells. We are calibrating the virus replication module to existing experimental data for SARS-CoV-2 and influenza A. Because different tissues within the body have different responses to local viral infection, developing our framework to support the modeling and simulation of multi-organ disease progression (*e.g.*, by identifying model parameters corresponding to specific tissues and physiological compartments) would allow us to understand the highly variable whole-body progression of many viral diseases.

The immune response to viral infection depends on locus of infection, degree of infection and patient immune state. Understanding the reasons for immune failure to contain infection, or pathological responses like cytokine storms or sepsis, requires models of immune response at multiple locations and scales. The same is true for understanding and predicting the possible protective or adverse effects of coinfection. The number of permutations of infection timing and combination of pathogens is too large to address purely by experiments, but could be addressed by simulations. Spatial modeling is also important because the spatiotemporal dynamics of coinfection within tissues may be important to the outcome (*e.g.*, whether individual cells can be superinfected, whether viral lesions with a tissue are disjoint or overlap, whether the main foci of the pathogens are in the same or different tissues).

We can also study the systemic effects of possible therapies with known molecular modes of action (as seen in *Results*). Evaluating therapies in a simulated context prior to performing animal or human trials could lead to more effective and rapid drug discovery and to optimized dosage and timing of treatments. Understanding the origins of population variability in disease progression is crucial to providing optimal personalized treatment. While the simulations presented here begin with a single infected cell, a

simulation which begins with multiple infected cells might better represent the infection dynamics of patients that have been subject to high level exposure, such as healthcare workers. Factors such as hypertension, immunosuppression and diabetes affect tissue state and immune response and could also be incorporated into our model. More detailed studies of these factors using our model could reveal more about the effects of population variability (due to age, genetic variation, prior drug treatment or immune status) on disease progression. Such computational studies could be accomplished using concomitant, calibrated ODE-based simulations of COVID-19 treatment published as recently as after the initial submission of this manuscript for publication [61].

We are working to implement validated non-spatial models of viral infection and immune response as agent-based spatial models (*e.g.*, viral production, cytokine secretion, tissue damage). By starting with a validated model that uses ordinary differential equations and adding spatial components gradually, we can calibrate our spatial models and validate our results. In ongoing work, we have developed a formal method for spatializing ODE models and employing their parameters such that these analogous spatial models reproduce the ODE results in the limit of large diffusion constants. Using this method, we can combine the ability to do rapid formal parameter identification of ODE-based models and to leverage published ODE model parameters with the flexibility of spatial modeling. In these cases, any differences between the ODE results and the spatial model can be definitively attributed to spatialization (*e.g*. the local spread of virus or cytokine or the limited speed of movement of immune cells), or to additional factors which are difficult to include in an ODE model (*e.g.*, the variability of individual cells or the complex time course of virus release by individual infected cells).

We have developed formal spatializations of a number of interesting ODE models of COVID, such as [61,62], to explore the effects of stochasticity of outcomes, the effects of spatial mechanisms, and infection dynamics at a particular site of infection on the predictions of these models. An additional benefit of our approach is that we can easily and consistently combine and integrate ODE models which focus on different aspects of the complex process of infection, spread and clearance (*e.g.*, combining published models of intracellular INF-induced viral resistance with spatial models of plaque spread *in vitro* [20,63]). We illustrate both of these strengths in *Integration of an explicit RNA synthesis model allows the spatiotemporal modelling of hepatitis C virus infection*, where we integrate a published HCV model of subgenomic replication into our framework. We can also conduct simultaneous, cross-platform validation of spatial models by building multiple implementations of the same conceptual and quantitative models on independent modeling platforms (here Chaste [64,65] and Morpheus [66]).

The COVID-19 crisis has shown that drug discovery and therapy development both require new predictive capabilities that improve their effectiveness and efficiency. We have developed our framework to explore the relationship between molecular, cellular-level and systemic mechanisms and outcomes of acute viral infections like SARS-CoV-2, and to support development of optimal, patient-specific treatments to combat existing and new viruses.

Models and methods

In this section we first present our model as a high-level conceptual model where we list each process included in an implementation-independent manner. We then detail

the quantitative model and its computational implementation, which uses a Cellular Potts representation of cellular dynamics. All quantitative models are implemented in a modular, extensible simulation architecture built using the CompuCell3D simulation environment, which is publicly available for download and further development by interested members of the scientific community (see *Collaborative viral infection modeling environment*).

Conceptual model: biological hypotheses and assumptions

As discussed in *Introduction* (Fig 2A) we consider viral propagation in an epithelial tissue and a lymph node. The tissue contains two interacting spatial components: an epithelium component (consisting of a monolayer of epithelial cells), and an extracellular environment component (containing immune cells, extracellular virus and chemicals). The lymph node component (whose state is affected by signaling from the tissue) adds immune cells to the extracellular space when in a proinflammatory state and removes them when in an anti-inflammatory state. A set of processes and interactions govern how the states of these components evolve in time. We detail these components, processes and interactions in the following subsections and in Fig 2.

Epithelium component. The epithelium component of the model represents the layer of epithelium in the tissue, and is composed of epithelial cells of four types: uninfected, infected, virus-releasing and dead (Fig 2C). We assume that the epithelial cells are immobile. We implicitly model the ECM by considering its influence on all processes in the epithelium component. The epithelial cells contain modules that describe the viral life

cycle and approximate the amount of virus as a continuous quantity (Fig 2B), including: binding and internalization of viral particles from the extracellular environment (**E1**), intracellular replication (**E2**) and release (**E3**) of synthesized virus into the extracellular environment, as well as cell death caused by viral-replication-associated damage, immune-cell killing and oxidative agent killing (**E4**).

E1 - Viral internalization. Module E1 models extracellular virus binding to epithelial cell receptors and internalization (including endocytosis-dependent and -independent routes). Internalization of viral particles involves binding of the viral spike protein to target cell-surface receptors, truncation by surface proteins and receptor-mediated endocytosis or fusion with the host plasma membrane. We assume the dynamics of internalization can be represented by describing the dynamics of virus-surface-receptor binding, determined by the amount of extracellular virus and target surface receptors, and by the binding affinity between them (**T1**→**E1**). We also consider the dynamic depletion of unbound target surface receptors on a cell when it internalizes a virus and superinfection of infected cells. Internalized viral particles initiate the viral replication process (**E1**→**E2** and Fig 2B).

E2 - Viral replication. Module E2 models the viral replication cycle inside a host epithelial cell (Fig 2B). Individual cells infected with many non-lytic viruses show a characteristic three-phasic pattern in their rate of viral release. After infection and during an *eclipse phase*, a cell accumulates but does not yet release newly assembled viruses. In a second phase, the rate of viral release increases exponentially until the virus-releasing cell either dies or, in a third phase, saturates its rate of virus synthesis and release. Viral replication

hijacks host synthesis pathways and may be limited by the availability of resources (amino acids, ATP, *etc.*), synthesis capability (ribosomes, endoplasmic reticulum, *etc.*) or intracellular viral suppression. A quantitative model of viral replication needs to be constructed and parameterized such that it reproduces these three phases.

We model viral replication based on processes associated with positive sense single-stranded RNA (+ssRNA) viruses. +ssRNA viruses initiate replication after unpacking of the viral genetic material and proteins into the cytosol (**E1→E2)**. The viral RNA-dependent RNA polymerase transcribes a negative RNA strand from the positive RNA strand, which is used as a template to produce more RNA strands (denoted by "Viral Genome Replication" in Fig 2B). Replication of the viral genome is the only exponential amplification step in the growth of most viruses within cells. Subgenomic sequences are then translated to produce viral proteins ("Protein Synthesis" Fig 2B). Positive RNA strands and viral proteins are transported to the endoplasmic reticulum (ER) where they are packaged for release. After replication, newly synthesized viral genetic material is translated into new capsid protein and assembled into new viral particles ("Assembly and Packaging" in Fig 2B). These newly assembled viral particles initiate the viral release process (**E2→E3**). We assume the viral replication cycle can be modeled by defining four stages: unpacking, viral genome replication, protein synthesis, and assembly and packaging. Fig 2B illustrates these subprocesses of replication and their relation to viral internalization and release.

E3 - Viral release. Module E3 models intracellular transport of newly assembled virions and release into the extracellular environment (**E3→T1** and Fig 2B "Release"). We

conceptualize the virus being released into the extracellular fluid above the apical surfaces of epithelial cells. Newly assembled virions are packed into vesicles and transported to the cell membrane for release into the extracellular environment (**E2**→**E3**). We assume that no regulation occurs after assembly of new virus particles, and that release into the extracellular environment can be modeled as a single-step process (**E3**→**T1**).

E4 - Cell death. Module E4 models death of epithelial cells due to various mechanisms. Models the combined effect of the many types of virus-induced cell death (*e.g.*, production of viral proteins interferes with the host cell's metabolic, regulatory and delivery pathways) as occurring due to a high number of assembled viral particles in the viral replication cycle (**E2**→**E4**). Models cell death due to contact cytotoxicity (**I3**→**E4**). Models cell death due to oxidizing cytotoxicity (**T3**→**E4**).

Extracellular environment component. The extracellular environment contains the immune cells, extracellular virus, cytokines and oxidative agent, and is the space where transport of viral particles (**T1**), cytokine molecules (**T2**) and the oxidizing agent (**T3**) occurs. We implicitly model the ECM in the extracellular environment by subsuming its geometrical, biochemical and biophysical influences on immune cell motility and virus/cytokine/agent spreading in the chosen rate laws and parameter set. Immune cells are mobile and can be either activated or inactive (**I1**). Inactive immune cells move through random cell motility and activated immune cells chemotax along the cytokine field

(**I2**). The immune cell modules also account for cytotoxic effects of immune cells on contact due to antigen recognition (**I3**) and through the secretion of oxidizing agents (**I4**).

T1 - Viral transport. Module T1 models diffusion of viral particles in the extracellular environment and their decay. Viral particles are transported by different mechanisms (*e.g.*, ciliated active transport, diffusion) and media (*e.g.*, air, mucus) at different physiological locations and through different types of tissue (*e.g.*, nasopharyngeal track, lung bronchi and alveoli). Viral particles are eliminated by a variety of biological mechanisms. We represent these mechanisms by modeling transport of viral particles as a diffusive virus field with decay in the extracellular environment. We model transport in a thin layer above the apical surfaces of epithelial cells. Viral internalization results in the transport of a finite amount of virus from the extracellular environment into a cell and depends on the amount of local extracellular virus and number of cell surface receptors (**T1-E1**). Infected cells release viral particles into the extracellular environment as a result of the viral replication cycle (**E3-T1**).

T2 - Cytokine transport. Module T2 models diffusion and clearance of immune signaling molecules in the extracellular environment. The immune response involves multiple signaling molecules acting upon different signaling pathways. We assume that the complexity of immune signaling can be functionally represented using a single chemical field that diffuses and decays in the extracellular environment. Once infected, epithelial cells secrete signaling molecules to alert the immune system (**E2-T2**). Locally, exposure to cytokine signaling results in activation of immune cells (**T2-I1**). Upon activation,

immune cells migrate towards infection sites guided by cytokine (**T2-I2**). Lastly, activated immune cells amplify the immune signal by secreting additional cytokines into the extracellular environment (**I1-T2**). We model long-range effects by assuming that cytokine exfiltrates tissues and is transported to immune recruitment sites (**T2-L1**).

T3 - Oxidizing agent burst and transport. Module T3 models diffusion and clearance of a general oxidizing agent in the extracellular environment. One of the cytotoxic mechanisms of immune cells is the release of different oxidizing agents, reactive oxygen species like H_2O_2 and nitric oxide. The mechanism of action of such agents varies but we assume that we can generalize such effects by modeling a single diffusive and decaying oxidizing agent field in the extracellular environment. The oxidizing agent is secreted by activated immune cells after persistent exposure to cytokine signals (**I4→T3**). We assume that the range of action of the oxidizing agent is short. Cell death is induced in uninfected, infected and virus-releasing epithelial cells when sufficiently exposed to the oxidizing agent (**T3→E4**).

I1 - Immune cell activation. Module I1 models immune cell maturation due to cytokine signaling. Immune cells mature at the recruitment site before being transported to the infection site as inactive immune cells (**L1→Immune Cells**). After infiltration, immune cells need to be exposed to local cytokine signals before activating (**T2→I1**). Once activated, immune cells chemotax along the cytokine field (**I2**) and amplify immune signaling by releasing cytokine molecules into the extracellular environment (**I1→T2**). Immune cells can also deactivate after a period of activation (**I1** and Fig 2C).

I2 - Immune cell chemotaxis. Module I2 models activated immune cell chemotactic migration towards infection sites. We assume that upon activation (**I1**→**I2**), immune cells move preferentially towards higher concentrations of local cytokine (**T2**→**I2**).

I3 - Immune cell direct cytotoxicity and bystander effect. Module I3 models immune cell cytotoxicity when immune cells (both activated and inactive) identify and induce death in epithelial cells with internal virus. Immune cells identify epithelial cells with internal virus on contact by antigen recognition and induce cell death by activating the caspase cascade (**I3**→**E4**). Uninfected, infected, and virus-releasing epithelial cells in contact with an epithelial cell that is killed by direct cytotoxicity can die through a bystander effect.

I4 - Immune cell oxidizing agent cytotoxicity. Module I4 models activated immune cell killing of target cells through the release of a diffusive and decaying oxidizing agent into the environment. Cell death is induced in uninfected, infected and virus-releasing epithelial cells when sufficiently exposed to the oxidizing agent (**T3**→**E4**).

Lymph node component. The lymph node component models the net pro- or anti-inflammatory state of the immune system. It responds to cytokines received from the tissue and adds or removes immune cells from the tissue (**L1**).

L1 - Immune cell recruitment. Module L1 models immune cell recruitment and infiltration into the tissue in response to cytokine signaling by infected cells and activated immune

cells. Infected cells secrete signaling molecules into the extracellular environment (**E2→T3**), which alerts resident immune cells and recruits new immune cells from the blood, distant lymph nodes and bone marrow. We model the local strength of the cytokine signal as causing an increase in the strength of the signal at the immune recruiting sites. We model long-distance signaling by assuming that cytokine molecules in the extracellular environment exfiltrate the infection site and are transported through the lymphatic system to the lymphatic system to lymph nodes and through the bloodstream to initiate immune-cell recruitment (**T2→L1**). A delay on the order of minutes to hours would represent semi-local recruitment (*e.g.*, at the blood vessels). A delay on the order of days would represent long-range, systemic recruitment (*e.g.*, the time required for a dendritic cell to reach a lymph node and an induced T cell to return). Recruited immune cells are then transported and infiltrate the infection site (**L1→Immune Cell**).

Quantitative model and implementation

For model construction and integration we use the open-source multicellular modeling environment CompuCell3D (www.compucell3d.org) which allows rapid and compact specification of cells, diffusing fields and biochemical networks using Python and the Antimony language [55,67]. CompuCell3D is specifically designed to separate model specification (conceptual and quantitative models) from the details of model implementation as a simulation and to make simulation specification accessible to biologists and others not specializing in software development. In this paper, we specifically designed the Python modules and their cross-scale integration to have clear and stable APIs, allowing modules to be rapidly swapped out by collaborating developers.

CompuCell3D runs on Windows, Mac and Linux platforms without change of model specification, and allows cluster execution for parameter exploration.

Cellular Potts model (CPM). *Cell types*. Cells are divided into two broad groups, epithelial and immune cells, and have a *type* (see Fig 2C) which determines their properties, the processes and interactions in which they participate, and their events and dynamics. Epithelial cells can have one of four types (uninfected, infected, virus releasing and dead). Immune cells can have one of two types (activated and inactive). Cell types can change according to outcomes of various modules, and a module specifying such an event describes both the initial and final cell types of the transition. A cell type in the model is not a phenotype in the biological sense (*e.g.*, *epithelial* cell), but an identifier for the various states that a particular cell can assume (*e.g.*, *dead epithelial* cell). When an epithelial cell changes to the dead type, all epithelial modules are disabled and the cell is generally inactive.

Cellular dynamics. Cellular spatial dynamics is modeled using the Cellular Potts model (also known as *CPM*, or Glazier-Graner-Hogeweg model), which represents generalized cells and medium as occupying a set of sites in a lattice [68]. Random cell motility is modeled as the stochastic exchange of sites at intercellular and cell-medium interfaces. Configurations evolve to minimize the system's effective energy $\mathcal{H}$,

$$\mathcal{H} = \sum_{\sigma} \lambda_{volume} \left(v(\sigma) - V\left(\tau(\sigma)\right) \right)^2 + \sum_{x} \sum_{x' \in N(x)} \left(1 - \delta_{\sigma(x),\sigma(x')}\right) J\left(\tau\left(\sigma(x)\right), \tau\left(\sigma(x')\right)\right) + \mathcal{H}_{chemotaxis}. \quad (1)$$

Here σ is the integer identification of a cell and $\tau(\sigma)$ is the type of cell σ. $v(\sigma)$ and $V(\sigma)$ are the current and target volumes of cell σ, respectively, and λ_{volume} is a volume constraint coefficient. $N(x)$ is the neighborhood of site x, $\delta_{i,j}$ is the Kronecker-delta, and $J(\tau, \tau')$ is the effective contact energy per unit surface area between cells of types τ and τ'. The final term, $\mathcal{H}_{chemotaxis}$, models chemotaxis-directed cell motility, and is prescribed by module I2. The cell configuration evolves through asynchronous lattice-site copy attempts. A lattice-site copy attempt starts by random selection of a site x in the lattice as a target, and a site x' in its neighborhood as a source. A configuration update is then proposed in which the value x' from the source site overwrites the value of x in the target site. The change in total effective energy $\Delta\mathcal{H}$ due to the copy attempt is calculated, and the update is executed with a probability given by a Boltzmann acceptance function,

$$\Pr\left(\sigma(x) \to \sigma(x')\right) = e^{-\max\left\{0, \frac{\Delta\mathcal{H}}{\mathcal{H}^*}\right\}}. \quad (2)$$

Here the intrinsic random motility $\mathcal{H}^*$ controls the stochasticity of accepted copy attempts. Updates that reduce the system's effective energy are always accepted. The unit of simulation time is the Monte Carlo step (*MCS*)–taken to be 20 minutes in this work. One MCS corresponds to considering a number of copy attempts equal to the number of lattice sites.

Epithelial component modules. Processes **E1**-**E4** describe epithelial cell functions as defined below. **E1**, **E2** and **E4** govern the cell-type transitions of epithelial cells (see Fig 15). **E1** transforms an uninfected epithelial cell into an infected epithelial cell. **E2** transforms an infected epithelial cell into a virus-releasing epithelial cell. **E4** transforms a virus-releasing epithelial cell into a dead cell.

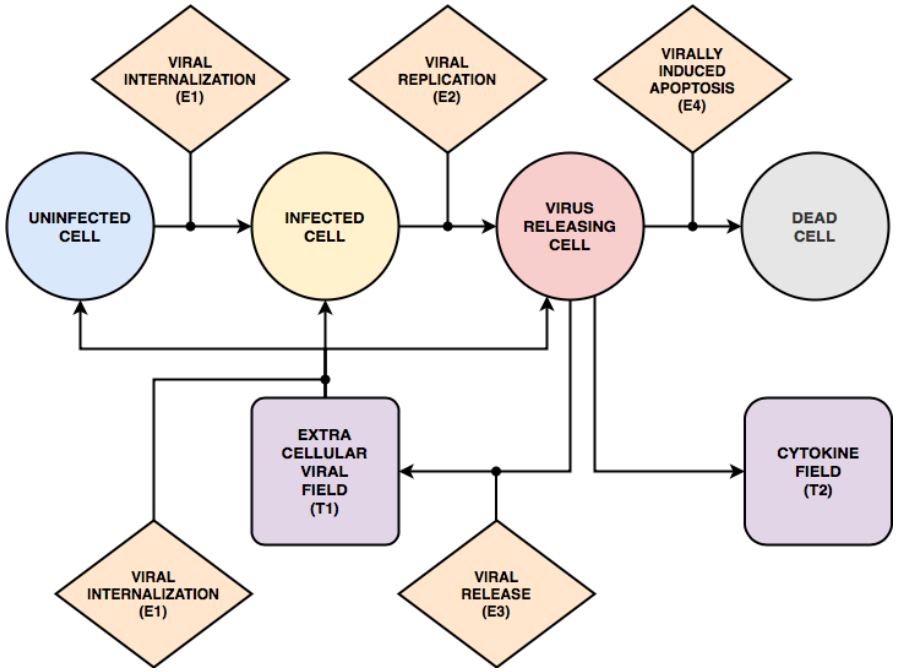


Fig 15. State diagram and interactions of epithelial cells.

Epithelial cells can have one of four "cell types": uninfected, infected, virus-releasing and dead. Uninfected cells become infected cells when the viral uptake model (**E1**) internalizes viruses from the extracellular virus field (**T1**). Infected cells continue internalizing viruses from the extracellular virus field and become virus-releasing cells when the viral replication model (**E2**) produces sufficient newly assembled virions. Virus-releasing cells secrete viruses into the extracellular virus field (**T1**) according to the viral release module (**E3**) and secrete cytokines directly into the extracellular cytokine field (**T2**). Virus-releasing cells can die if the conditions of the virally induced cell-death model (**E4**) are met.

E1 - Viral internalization. To capture the stochasticity associated with internalization of discrete virus particles in terms of discrete binding events, we assign each uninfected, infected and virus-releasing epithelial cell a probability of absorbing diffusive viral particles from the extracellular virus field (**T1**). The uptake probability $\Pr(Uptake(\sigma) > 0)$ for each cell σ is given by a Hill equation of the total amount of diffusive viral particles in the domain of the cell $c_{vir}(\sigma)$, the number of unbound cell surface receptors $SR(\sigma)$ and the binding affinity between them.

$$\Pr(Uptake(\sigma) > 0) = \frac{\Delta t}{\alpha_{upt}} \frac{\left(c_{vir}(\sigma)\right)^{h_{upt}}}{\left(c_{vir}(\sigma)\right)^{h_{upt}} + V_{upt}^{h_{upt}}}, \text{ where } V_{upt} = \frac{R_o k_{off}}{2k_{on} v(\sigma) SR(\sigma)}. \quad (3)$$

Here h_{upt} is a Hill coefficient, R_o is the cell's initial number of unbound receptors, k_{on} is the virus-receptor association affinity, k_{off} is the virus-receptor dissociation affinity, α_{upt} is a characteristic time constant of uptake and Δt is the time represented by one MCS. At each simulation time step, the uptake probability is evaluated against a uniformly-distributed random variable. When uptake occurs, the uptake rate is proportional to the local amount in the virus field (**T1**), and the probability of uptake is used to define the amount ($Uptake$) of virus taken up during the MCS,

$$Uptake(\sigma) = \frac{1}{\Delta t} \Pr(Uptake(\sigma) > 0)\, c_{vir}(\sigma), \quad (4)$$

$$\frac{dSR(\sigma)}{dt} = -Uptake(\sigma). \quad (5)$$

The amount absorbed by each cell ($Uptake$) is uniformly subtracted from the virus field over the cell's domain and the cell's number of cell surface receptors is reduced by the same amount. The amount of virus taken up ($Uptake$) is also passed to the cell's instance

of the viral replication model (**E2**). Infected epithelial cells continue taking up viral particles from the environment until their cell surface receptors are depleted.

E2 - Viral replication. Our viral replication model combines equations and parameters from several sources to represent the replication of a generic virus [7,9,52,53]. The model contains four variables representing viral quantities in different states of the viral replication process: internalized virus U (Equation (6), the process of unpacking), viral genome taking part in genomic replication R (Equation (7), the process of viral genome replication), synthesized protein P (Equation (8), the process of protein synthesis), and assembled and packaged virions A (Equation (9), the process of assembly and packaging). Biologically, the only process which can exponentially increase the rate of virus production by a single cell is viral genome replication, so the equations include the positive feedback by R in Equation (7). Biologically, factors like the cell's metabolism, limited number of ribosomes, maximum rate of endoplasmic reticulum function and activation of intracellular viral suppression pathways all limit production of viral components, so we include a Michaelis-type saturation term for the rate of replication in Equation (7). Each uninfected, infected and virus-releasing cell in the simulation contains an independent copy of the system of ordinary differential equations modeling the viral replication process,

$$\frac{dU}{dt} = Uptake - r_u U, \quad (6)$$

$$\frac{dR}{dt} = r_u U + r_{max} R \frac{r_{half}}{R + r_{half}} - r_t R, \quad (7)$$

$$\frac{dP}{dt} = r_t R - r_p P, \quad (8)$$

$$\frac{dA}{dt} = r_p P - Release. \quad (9)$$

Here r_u is the unpacking rate, r_{max} is the viral replication rate, r_t is the translating rate (rate at which viral genomes turn into RNA templates for protein production) and r_p is the packaging rate. $Uptake$ is defined in **E1** and $Release$ is defined in **E3**. The saturation of the rate of viral genome replication is represented by a Michaelis-Menten function, $\frac{r_{half}}{R + r_{half}}$, where r_{half} is the amount of R at which the viral genome replication rate is reduced to $\frac{r_{max}}{2}$ (and the flux is reduced to $\frac{1}{2} r_{max}\, r_{half}$). The duration of the eclipse phase of single-cell infection (the time between the first entry of the virus into the cell and the first release of newly synthesized virus) is approximately $t_{eclipse} \approx \frac{1}{r_u} + \frac{1}{r_{max}} + \frac{1}{r_t} + \frac{1}{r_p}$ (11.7 hours for the reference parameter set in Table 1), with the additional complication that in our model, an epithelial cell does not release virus until A reaches a threshold value. The timescale for tenfold increase of virus release when viral replication is maximal is $t_{10} \approx \frac{log(10)}{r_{max}}$ (7.7 hours for the reference parameter set in Table 1). The number of newly assembled virions is passed to the cell's instance of the viral release module (**E3**). See Fig 2B for a schematic of the viral replication process and Fig 16 for a representative time series from the viral internalization, replication and release modules.

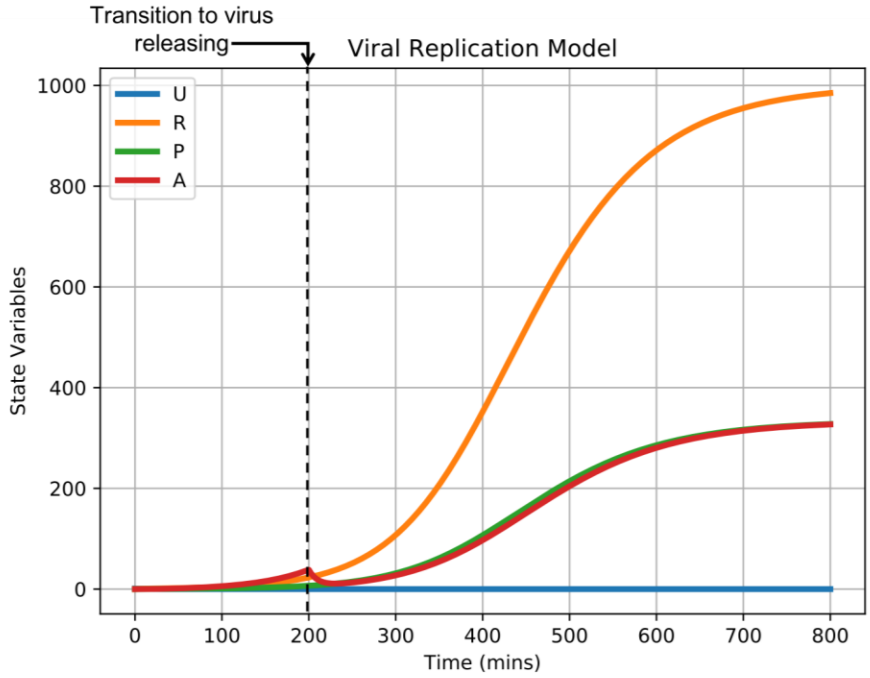


Fig 16. Representative time series of viral internalization, replication and release models from Fig 2B.

A sample simulation of the viral replication model in a single epithelial cell. The model is initialized with one unit of internalized virus (U = 1), and the rest of the state variables set to zero (R = 0, P = 0, A = 0). No additional virus internalization occurs during this sample simulation. Dashed line indicates the time of the cell's transition from the infected to virus-releasing cell type.

E3 - Viral release. Virus releasing cells release viral particles into the extracellular virus field (**T1**). The amount of virus released by a cell per unit time is proportional to the state variable for assembled virions in the viral replication module (**E2**),

$$Release = r_s A. \quad (10)$$

Here r_s is the release rate of viral particles and A is the level of assembled virus in the cell (defined in **E2**). The total amount released by each cell $r_s A \Delta t$ is subtracted from the cell's state variable for assembled virions A and passed to the source term of the extracellular virus field (**T1**) to maintain mass balance.

E4 - Cell death. For cell death due to virally-induced apoptosis, each infected and virus-releasing cell can die due to the amount of intracellular virus. The rate of death is defined as a stochastic function of the state variable for assembled new virions from the viral replication module (**E2**). If a virus releasing cell dies then it changes its cell type to dead and the cell's instances of the viral internalization, replication and release modules are disabled. The probability of virus-induced apoptosis per simulation step is a Hill equation of the current load of assembled virus,

$$\Pr(\tau(\sigma) \to \text{Dead} | \tau(\sigma) = \text{Virus releasing}) = \frac{\Delta t}{\alpha_{apo}} \frac{\left(A(\sigma)\right)^{h_{apo}}}{\left(A(\sigma)\right)^{h_{apo}} + V_{apo}^{h_{apo}}}. \quad (11)$$

where $A(\sigma)$ is the number of assembled virions in cell σ, h_{apo} is a Hill coefficient,V_{apo} is the amount of assembled virions at which the apoptosis probability is 0.5 per unit time and α_{apo} is a characteristic time constant of virally-induced apoptosis. For modeling of cell death due to contact cytotoxicity, see *I3 - Immune cell direct cytotoxicity and bystander effect*. For modeling of cell death due to oxidizing cytotoxicity, see *I4 - Immune cell oxidizing agent cytotoxicity*. Regardless of the death mechanism the internally assembled virions are not released to the environment and do not take action in further infection. We assume that in the process of death the assembled virus and viral particles are either damaged or deactivated.

Lymph node modules. *L1 - Immune cell recruitment*. The total immune cell population is governed by an ordinary differential equation of a dimensionless state variable S that represents immune response due to local conditions and long-distance signaling. Our

convention is that when $S > 0$, immune cells are recruited to the simulation domain; likewise, immune cells are removed from the simulation domain when $S < 0$. Probability functions of S describe the likelihood of immune cell seeding and removal,

$$\Pr(\text{add immune cell}) = \text{erf}(\alpha_{immune} S), \qquad S > 0, \tag{12}$$

$$\Pr(\text{remove immune cell}) = \text{erf}(-\alpha_{immune} S), \qquad S < 0. \tag{13}$$

Here the coefficient α_{immune} is the sensitivity of immune cell addition and removal to the state variable S. The dynamics of S are cast such that, in a homeostatic condition, a typical number of immune cells can be found in the simulation domain, and production of cytokine (**T2**) results in additional recruitment via long-distance signaling (*i.e.*, with some delay). We model this homeostatic feature using the feedback mechanism of the total number of immune cells N_{immune} in the simulation domain. Cytokine signaling is modeled as perturbing the homeostatic state using the term $\alpha_{sig}\delta$. Here δ is the total amount of decayed cytokine in the simulation domain and $\alpha_{sig} > 0$ models signaling by transmission of cytokine to some far-away source of immune cells. We write the rate of change of S as

$$\frac{dS}{dt} = \beta_{add} - \beta_{sub} N_{immune} + \frac{\alpha_{sig}}{\beta_{delay}}\delta - \beta_{decay} S. \tag{14}$$

Here β_{add} and β_{sub} control the number of immune cells in the simulation domain under homeostatic conditions. β_{delay} controls the delay between transmission of the cytokine to the lymph node and corresponding immune response by adjusting the rate of recruitment due to total cytokine (*i.e.*, increasing β_{delay} increases the resulting delay). β_{decay} regulates the return of S to an unperturbed state (*i.e.*, $S = 0$, increasing β_{decay} increases the rate of return to $S = 0$). To determine the seeding location, the simulation space is randomly sampled $n_{seeding}$ times, and an immune cell is seeded at the unoccupied location with the

highest amount of the virus field. If no location is unoccupied, then the immune cell is not seeded. The removal probability is evaluated for each immune cell at each simulation step. Immune cells are removed by setting their volume constraint to zero.

Immune cell modules. The four processes **I1**-**I4** capture immune cell functions which are defined below. These processes control how immune cells are activated, translocate, and kill other cells. Their interactions with epithelial cells and other model components are illustrated in Fig 17.

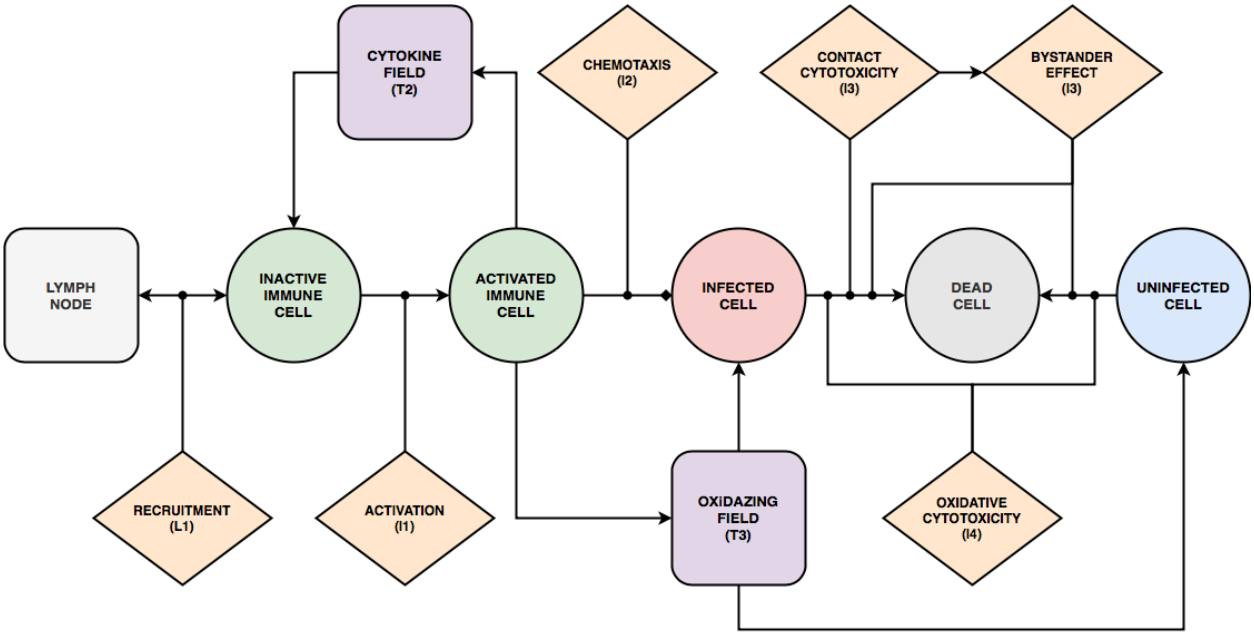


Fig 17. State diagram and interactions of Immune cells.

Immune cells can adopt two different generalized types: inactive and activated. Inactive immune cells are recruited by the cytokine levels according to the immune recruitment module (**L1**).Transition from inactive to activated immune cells is determined by the immune activation module (**I1**) when cells are exposed to cytokines in the tissue. Activated immune cells amplify the cytokine signal by secreting cytokines to the extracellular environment. Activated immune cells chemotax towards virus-releasing cells (**I2**). Immune cells induce death of epithelial cells by direct cytotoxicity when

coming into contact with infected cells (**I3**), bystander effect by killing neighbors of infected cells (**I3**) and through oxidative cytotoxicity (**I4**) by releasing cytotoxic oxidizing agents (**T3**) into the extracellular environment.

I1 - Immune cell activation. Inactive immune cells become activated with a probability according to a Hill equation of the total cytokine bound to the cell $B_{cyt}(\sigma, t)$,

$$\Pr(\tau(\sigma,t) \rightarrow \text{activated immune} | \tau(\sigma,t) = \text{inactive immune}) = \frac{\left(B_{cyt}(\sigma,t)\right)^{h_{act}}}{\left(B_{cyt}(\sigma,t)\right)^{h_{act}} + \left(EC50_{cytact}\right)^{h_{act}}}. \quad (15)$$

After ten hours, an activated immune cell becomes inactive, in which case evaluations of activation (Equation (15)) recommence. The immune cells "forget" a percentage $(1 - \rho_{cyt})$ of the bound cytokine each time step while taking up an amount of cytokine from the environment ($\omega_{cyt}(\tau(\sigma), t)$ defined in **T2**),

$$B_{cyt}(\sigma,t) = \rho_{cyt} B_{cyt}(\sigma, t - \Delta t) + \omega_{cyt}\left(\tau\left(\sigma(x)\right), t\right). \quad (16)$$

I2 - Immune cell chemotaxis. Activated immune cells experience a motile force as a response to a signaling field. Immune cells chemotax along concentration gradients of the cytokine field. The chemotactic effective energy $\mathcal{H}_{chemotaxis}$ associated with the gradient is calculated according to a chemotactic sensitivity parameter $\lambda_{chemotaxis}$ and calculated chemotactic force $F_{chemotaxis}$. The contribution of $\mathcal{H}_{chemotaxis}$ to the change in the system's total effective energy is calculated using $F_{chemotaxis}$ when considering copy attempts. The chemotactic force at a location x is saturated by normalizing the chemotactic sensitivity parameter by the concentration of cytokine at the center of mass of the cell at x, $c_{cyt,CM}\left(\sigma(x)\right)$,

$$F_{chemotaxis}(x) = \frac{\lambda_{chemotaxis}}{1 + c_{cyt,CM}\left(\sigma(x)\right)} \nabla c_{cyt}(x). \quad (17)$$

1490

1491 *I3 - Immune cell direct cytotoxicity and bystander effect*. Immune cells, whether activated
1492 or not, have the ability to kill infected cells by direct contact. At each simulation step,
1493 immune cells trigger cell death in the infected and virus-releasing epithelial cells with
1494 which they come in contact. When an infected cell is killed by direct cytotoxicity, each of
1495 its first-order neighbors is evaluated for cell death by a bystander effect model. Each of
1496 those neighbors $\sigma' \in N(\sigma)$ in the first-order neighborhood $N(\sigma)$ of a cell σ killed by direct
1497 cytotoxicity has a probability $k_{bystander}$ of dying from the bystander effect given by,

$$\Pr(\tau(\sigma', t) \rightarrow \text{Dead} | Direct\ Cytotoxicity(\sigma) = \text{True}) = k_{bystander}\ \forall \sigma' \in N(\sigma). \quad (18)$$

1498

1499 *I4 - Immune cell oxidizing agent cytotoxicity*. Immune cells release a short-range, diffusive
1500 oxidizing agent when exposed to high cytokine concentration (**T3**). The oxidizing agent
1501 kills an epithelial cell of any type when the total amount of oxidizing agent in the domain
1502 of the cell $c_{oxi}(\sigma)$ exceeds a threshold for death τ_{oxi}^{death},

$$\Pr\left(\tau(\sigma, t) \rightarrow \text{Dead} | c_{oxi}(\sigma) > \tau_{oxi}^{death}\right) = 1. \quad (19)$$

1503

1504 **Extracellular environment modules**. *T1 - Viral transport*. The change in concentration
1505 of the virus field c_{vir} is calculated at each location in the simulation domain by solving the
1506 following reaction-diffusion equation,

$$\frac{\partial c_{vir}(x)}{\partial t} = D_{vir} \nabla^2 c_{vir}(x) - \gamma_{vir} c_{vir}(x) + \frac{1}{v\left(\sigma(x)\right)} \left(Release\left(\sigma(x)\right) - Uptake\left(\sigma(x)\right) \right). \quad (20)$$

Here D_{vir} is the diffusion constant of the extracellular virus and γ_{vir} is the decay rate is the decay rate. Uptake and release by a cell at each location are determined using the viral internalization (**E1**) and the viral release (**E3**) modules, and are uniformly applied over all sites of the domain of the cell.

T2 - Cytokine transport. The change in concentration of the cytokine field c_{cyt} is obtained by solving a reaction-diffusion equation of the following general form,

$$\frac{\partial c_{cyt}}{\partial t} = D_{cyt}\nabla^2 c_{cyt} - \gamma_{cyt} c_{cyt} + s_{cyt}. \qquad (21)$$

The decay term $\gamma_{cyt} c_{cyt}$ represents cytokine leaving the simulation domain (*e.g.*, in immune recruitment). To model immune signaling, the rate of cytokine secretion is described by an increasing Hill function of $c_{sig}\big(\sigma(x)\big)$ with Hill exponent $h_{cyt} = 2$. The meaning of $c_{sig}\big(\sigma(x)\big)$ depends on the cell type and the Hill exponent can differ for other cell types and states. The rate of cytokine secretion s_{cyt} is written as,

$$s_{cyt}(x,t) = \sigma_{cyt}\big(\tau(\sigma(x),t)\big)\frac{\Big(c_{sig}(\sigma(x),t)\Big)^{h_{cyt}}}{\Big(c_{sig}(\sigma(x),t)\Big)^{h_{cyt}} + \Big(V_{cyt}\big(\tau(\sigma(x),t)\big)\Big)^{h_{cyt}}} - \omega_{cyt}\big(\tau(\sigma(x),t)\big). \qquad (22)$$

Here $\sigma_{cyt}\big(\tau(\sigma(x),t)\big)$ is the maximum cytokine secretion rate for the cell type at x, $c_{sig}\big(\sigma(x)\big)$ is a quantity that affects the cells cytokine secretion, $\omega_{cyt}\big(\tau(\sigma(x),t)\big)$ is the cytokine uptake rate of the cell type at x and $V_{cyt}\big(\tau(\sigma(x),t)\big)$ is a dissociation coefficient of cytokine secretion for the cell type at x. σ_{cyt} is nonzero for infected epithelial cells, virus-releasing epithelial cells and activated immune cells. For infected and virus-releasing epithelial cells c_{sig} is the amount of assembled virus A in the viral replication module, and for activated immune cells c_{sig} is the total amount of cytokine in the domain

of the cell. Similarly, for epithelial cells V_{cyt} is the amount of assembled virus, and for immune cells V_{cyt} is the amount of cytokine in the domain of the cell. $\omega_{cyt}\left(\tau(\sigma(x),t)\right)$ is constant and nonzero for activated and inactive immune cells.

T3 - Oxidizing agent transport. The oxidizing agent field diffuses according to the reaction-diffusion equation,

$$\frac{\partial c_{oxi}}{\partial t} = D_{oxi}\nabla^2 c_{oxi} - \gamma_{oxi}c_{oxi} + s_{oxi}. \quad (23)$$

Bursts of oxidizing agent are implemented as a source term for one time step according to a rate coefficient σ_{oxi}, which is uniformly mapped onto the source term s_{oxi} over the domain of each activated immune cell. An oxidizing burst occurs in immune cells with an activated state when the total cytokine in the immune cell's domain exceeds a threshold τ_{oxi}^{sec}.

Initial and boundary conditions. The domain of all simulations had dimensions of 90 x 90 x 2 lattice sites. The initial cell configuration consisted of a 30 x 30 sheet of uninfected epithelial cells, each of size 3 x 3, on the lower layer of lattice sites (see Fig S23 for a demonstration of the negligible effects of a non-uniform arrangement of epithelial cells). Epithelial cells were "frozen", in that they were immobile, leaving the remaining 90 x 90 subdomain for occupancy by recruited immune cells. For cellular dynamics and mass transport, periodic boundary conditions were applied in the plane of the epithelial sheet, and Neumann conditions were applied along the direction orthogonal to the epithelial sheet. All field solutions for the diffusive viral, cytokine and oxidizing agent fields were initialized as zero everywhere.

At each first simulation step, the epithelial cell in the center of the sheet was set to infected, and the amount of internalized virus U of the viral replication model was set to a value of one. All epithelial cells were initialized with a number of unbound surface receptors $SR = R_o$. All immune cells, when introduced to the simulation by recruitment, were initialized in an inactive state, and with a bound cytokine value equal to zero ($B_{cyt} = 0$). During transition of an uninfected epithelial cell to the infected type, all state variables of the viral replication model were initialized with a value of zero.

Simulation specifications. Model implementation and all simulations were performed using CompuCell3D, which uses a non-dimensional lattice for CPM-based cellular dynamics and non-dimensional explicit time integration of reaction-diffusion field solutions. As such, a baseline parameter set was constructed for all CPM parameters and modules developed in this work (Table 1). Non-dimensionalization was performed on model parameters for a lattice dimension of 4 μm per pixel along each dimension, at 20 minutes (1/3 hours) per MCS. All replicas were simulated for ten trials, each 1,000 MCS (20000 minutes, 333 hours, 14 days) long. Simulation data was collected at a frequency of 10 MCSs (200 minutes, 3 hours) for all simulations.

Table 1. Parameter values in baseline parameter set.

Conversion Factors	Value	References / Justification
Simulation step Δt	1200.0 s	Selected for approximately 14 days of simulation time in 1k simulation steps
Lattice width	4.0 μm	Selected according to cell diameter
Scale factor for concentration	1×10^{-14} mol	Selected for conversion from mol L^{-1} to mol μm^{-1} (10^{-15}) with 10x adjustment for fewer concentrations less than 1

Simulation parameters	Value	References / Justification
Cell diameter	12.0 μm	Selected according to typical epithelial cell size
Replication rate r_{max}	(1/12)×10^{-3} s^{-1}	Calibrated to timescale of SARS-CoV-2 [56]
Translating rate r_t	(1/18)×10^{-3} s^{-1}	Calibrated to timescale of SARS-CoV-2 [56]
Unpacking rate r_u	(1/6)×10^{-3} s^{-1}	Calibrated to timescale of SARS-CoV-2 [56]
Packaging rate r_p	(1/6)×10^{-3} s^{-1}	Calibrated to timescale of SARS-CoV-2 [56]
Release rate r_s	(1/6)×10^{-3} s^{-1}	Calibrated to timescale of SARS-CoV-2 [56]
Scale factor for number of mRNA per infected cell $mRNA_{avg}$	1000 cell^{-1}	Selected for average production of 2000 virions per cell before death per influenza [75]
Viral dissociation coefficient r_{half}	2000	Derived from $mRNA_{avg}$, r_{max} and r_t
Viral diffusion coefficient D_{vir}	0.01 μm^2 s^{-1}	Selected according to sensitivity analysis and estimated from physiological ranges for lung mucus[1]
Viral diffusion length λ_{vir}	36 μm	Selected according to sensitivity analysis
Viral decay rate γ_{vir}	7.71×10^{-6} s^{-1}	Derived from λ_{vir} and D_{vir}
Cytokine diffusion coefficient D_{cyt}	0.16 μm^2 s^{-1}	[69,70] (IL-2 cytokine)
Cytokine diffusion length λ_{cyt}	100 μm	[69] (IL-2 cytokine)
Cytokine decay rate γ_{cyt}	1.32×10^{-5} s^{-1}	Derived from λ_{cyt} and D_{cyt}
Maximum cytokine immune secretion rate $\sigma_{cyt}(immune\ activated)$	3.5×10^{-4} pM s^{-1}	Estimated as 1/10 of $\sigma_{cyt}(infected)$
Immune secretion midpoint $V_{cyt}(immune\ activated)$	1 pM	[70]
Cytokine immune uptake rate $\omega_{cyt}(immune\ activated)$	3.5×10^{-4} pM s^{-1}	[69]
Maximum cytokine infected cell secretion rate $\sigma_{cyt}(infected)$	3.5×10^{-3} pM s^{-1}	[69]
Infected cell cytokine secretion mid-point $V_{cyt}(infected)$, $V_{cyt}(virus\ releasing)$	0.1	Chosen from typical simulation values of assembled virus. Values stay around 0.1 and increase as infection progresses
Cytokine secretion Hill coefficient h_{cyt}	2	Selected for simplest form with inflection of model response
Immune cell cytokine activation $EC50_{cyt,act}$	10 pM	[69]
Immune cell equilibrium bound cytokine EQ_{ck}	210 pM	Chosen to be 2.1 x $EC50_{cyt,act}$
Immune cell bound cytokine memory ρ_{cyt}	0.99998 s^{-1}	Derived from $\omega_{cyt}(immune\ activated)$ and EQ_{ck}
Immune cell activated time	10 h	[71]

Oxidation Agent diffusion coefficient D_{oxi}	0.64 μm^2 s^{-1}	Selected to be 4 x D_{cyt} to model high diffusivity relative to IL-2
Oxidation Agent diffusion length λ_{oxi}	36 μm	Selected to be 3 cell diameters to model high reactivity
Oxidation Agent decay rate γ_{oxi}	1.32×10^{-5} s^{-1}	Derived from λ_{oxi} and D_{oxi}
Immune cell oxidation agent secretion rate σ_{oxi}	3.5×10^{-3} pM s^{-1}	Selected to be the same as $\sigma_{cyt}(infected)$
Immune cell C_{cyt} threshold for Oxidation Agent release τ_{oxi}^{sec}	10 A.U. = 1.5625 pM	Selected according to sensitivity analysis
Tissue cell C_{oxi}threshold for death τ_{oxi}^{death}	1.5 A.U. = 0.234375 pM	Selected according to sensitivity analysis
Initial density of unbound cell surface receptors R_o	200 cell^{-1}	Selected for potential limiting factor (availability of receptors) from typical simulation extracellular virus field values
Virus-receptor association affinity k_{on}	1.4×10^4 M^{-1}s^{-1}	[72,73]
Virus-receptor dissociation affinity k_{off}	1.4×10^{-4} s^{-1}	[72,73]
Infection threshold	1	Calibrated to timescale of SARS-CoV-2 [56]
Uptake Hill coefficient h_{upt}	2	Selected for simplest form with inflection of model response
Uptake characteristic time constant α_{upt}	20 min	Selected to be the same as Δt
Virally-induced apoptosis Hill coefficient h_{apo}	2	Selected for simplest form with inflection of model response
Virally-induced apoptosis dissociation coefficient V_{apo}	100	Selected according to sensitivity analysis
Virally-induced apoptosis characteristic time constant α_{apo}	20 min	Selected to be the same as Δt
Immune cell activation Hill coefficient h_{act}	2	Selected for simplest form with inflection of model response
Immune response add immune cell coefficient β_{add}	1/1200 s^{-1}	Selected for sensitivity analysis of β_{delay}
Immune response subtract immune cell coefficient β_{sub}	1/6000 cell^{-1} s^{-1}	Selected according to β_{add} for 5 resident immune cells (mean of all immune cell counts per epithelial cell from Control in [22] applied to 900 epithelial cells)
Immune response delay coefficient β_{delay}	1.2×10^6 s	Selected according to sensitivity analysis
Immune response decay coefficient β_{decay}	1/12000 s^{-1}	Selected for sensitivity analysis of β_{delay}
Immune response cytokine transmission coefficient α_{sig}	0.5	Selected for sensitivity analysis of β_{delay}
Immune response probability scaling coefficient α_{immune}	0.01	Selected for sensitivity analysis of β_{delay}
Number of immune cell seeding samples $n_{seeding}$	10	Selected for sensitivity analysis of β_{delay}
Initial target volume	64 μm^3	Derived from cell diameter and lattice width

Lambda volume λ_{volume}	9	Selected for acceptable deformation of immune cells
Initial number of immune cells	0	Selected to demonstrate model feature of resident immune cells
Lambda chemotaxis $\lambda_{chemotaxis}$	1	Selected for appreciable chemotaxis without excessive cell deformation
Intrinsic Random Motility $\mathcal{H}^*$	10	[68]
Contact coefficients J (all interfaces)	10	Selected comparably to [68] for low adhesion immune cell-immune cell and immune cell-medium interfaces

[1] The diffusivity in water for a virus of radius 0.1 microns like SARS-CoV-2 according to Stokes-Einstein is about 3 microns^2/s. The average steady-shear viscosity for lung mucus varies significantly and is shear thinning, but in the more viscous regions is found to vary for frequencies between 10^{-4} and 102 Hz, spanning viscosity values as high as 103 Pa-s and as low as 10^{-2} Pa-s. In general, at low shear rates, the viscosity of human mucus is as high as 104−106 times that of water [74]. Thus the minimal diffusion constant possible would be 3 x 10^{-6} microns^2/s and the maximal rate in water would be 3 microns^2/s. 0.01 microns^2/s is a reasonable geometric interpolation.

References

1. Cao Y-C, Deng Q-X, Dai S-X. Remdesivir for severe acute respiratory syndrome coronavirus 2 causing COVID-19: An evaluation of the evidence. Travel Med Infect Dis. 2020; 101647–101647. doi:10.1016/j.tmaid.2020.101647
2. Wang D, Yin Y, Hu C, Liu X, Zhang X, Zhou S, et al. Clinical course and outcome of 107 patients infected with the novel coronavirus, SARS-CoV-2, discharged from two hospitals in Wuhan, China. Critical Care. 2020;24: 188. doi:10.1186/s13054-020-02895-6
3. Eckhardt M, Hultquist JF, Kaake RM, Hüttenhain R, Krogan NJ. A systems approach to infectious disease. Nature Reviews Genetics. 2020;21: 339–354. doi:10.1038/s41576-020-0212-5
4. Dolley S. Big Data's Role in Precision Public Health. Front Public Health. 2018;6: 68–68. doi:10.3389/fpubh.2018.00068
5. Garira W. A primer on multiscale modelling of infectious disease systems. Infectious Disease Modelling. 2018;3: 176–191. doi:10.1016/j.idm.2018.09.005
6. Koelle K, Farrell AP, Brooke CB, Ke R. Within-host infectious disease models accommodating cellular coinfection, with an application to influenza†. Virus Evolution. 2019;5. doi:10.1093/ve/vez018
7. Wang Y, Heiland R, Craig M, Davis C, Ford Versypt AN, Jenner A, et al. Rapid community-driven development of a SARS-CoV-2 tissue simulator. bioRxiv. 2020; 2020.04.02.019075. doi:10.1101/2020.04.02.019075
8. Hernandez Vargas EA, Velasco-Hernandez JX. In-host Modelling of COVID-19 Kinetics in Humans. medRxiv. 2020; 2020.03.26.20044487. doi:10.1101/2020.03.26.20044487
9. Dee KU, Hammer DA, Shuler ML. A model of the binding, entry, uncoating, and RNA synthesis of Semliki Forest virus in baby hamster kidney (BHK-21) cells. Biotechnology and Bioengineering. 1995;46: 485–496. doi:10.1002/bit.260460513
10. English TJ, Hammer DA. Brownian adhesive dynamics (BRAD) for simulating the receptor-mediated binding of viruses. Biophys J. 2004;86: 3359–3372. doi:10.1529/biophysj.103.027813
11. Boireau S, Maiuri P, Basyuk E, de la Mata M, Knezevich A, Pradet-Balade B, et al. The transcriptional cycle of HIV-1 in real-time and live cells. J Cell Biol. 2007;179: 291–304. doi:10.1083/jcb.200706018
12. Roldão A, Vieira HLA, Charpilienne A, Poncet D, Roy P, Carrondo MJT, et al. Modeling rotavirus-like particles production in a baculovirus expression vector system: Infection kinetics, baculovirus DNA replication, mRNA synthesis and protein production. Journal of Biotechnology. 2007;128: 875–894. doi:10.1016/j.jbiotec.2007.01.003
13. Purohit PK, Inamdar MM, Grayson PD, Squires TM, Kondev J, Phillips R. Forces during bacteriophage DNA packaging and ejection. Biophys J. 2004/11/19 ed. 2005;88: 851–866. doi:10.1529/biophysj.104.047134
14. Yang Q, Catalano CE. A minimal kinetic model for a viral DNA packaging machine. Biochemistry. 2004;43: 289–299. doi:10.1021/bi035532h
15. Graw F, Perelson A. Spatial aspects of HIV infection. 2013. pp. 3–31. doi:10.1007/978-1-4614-4178-6_1
16. Ferl GZ, Theil F-P, Wong H. Physiologically based pharmacokinetic models of

small molecules and therapeutic antibodies: a mini-review on fundamental concepts and applications. Biopharmaceutics & Drug Disposition. 2016;37: 75–92. doi:10.1002/bdd.1994

17. Bauer AL, Beauchemin CAA, Perelson AS. Agent-based modeling of host–pathogen systems: The successes and challenges. Inf Sci (N Y). 2009;179: 1379–1389. doi:10.1016/j.ins.2008.11.012
18. Beauchemin C. Probing the effects of the well-mixed assumption on viral infection dynamics. J Theor Biol. 2006;242: 464–477. doi:10.1016/j.jtbi.2006.03.014
19. Olafuyi O, Badhan RKS. Dose Optimization of Chloroquine by Pharmacokinetic Modeling During Pregnancy for the Treatment of Zika Virus Infection. JPharmSci. 2019;108: 661–673. doi:10.1016/j.xphs.2018.10.056
20. Holder BP, Liao LE, Simon P, Boivin G, Beauchemin CAA. Design considerations in building in silico equivalents of common experimental influenza virus assays. Autoimmunity. 2011;44: 282–293. doi:10.3109/08916934.2011.523267
21. Miao C, Jin M, Miao L, Yang X, Huang P, Xiong H, et al. Early chest computed tomography to diagnose COVID-19 from suspected patients: A multicenter retrospective study. The American Journal of Emergency Medicine. 2020;0. doi:10.1016/j.ajem.2020.04.051
22. Chua RL, Lukassen S, Trump S, Hennig BP, Wendisch D, Pott F, et al. COVID-19 severity correlates with airway epithelium–immune cell interactions identified by single-cell analysis. Nature Biotechnology. 2020;38: 970–979. doi:10.1038/s41587-020-0602-4
23. Zhou S, Zhu T, Wang Y, Xia L. Imaging features and evolution on CT in 100 COVID-19 pneumonia patients in Wuhan, China. Eur Radiol. 2020; 1–9. doi:10.1007/s00330-020-06879-6
24. Taubenberger JK, Morens DM. The pathology of influenza virus infections. Annu Rev Pathol. 2008;3: 499–522. doi:10.1146/annurev.pathmechdis.3.121806.154316
25. Bocharov G, Meyerhans A, Bessonov N, Trofimchuk S, Volpert V. Spatiotemporal Dynamics of Virus Infection Spreading in Tissues. PLOS ONE. 2016;11: e0168576. doi:10.1371/journal.pone.0168576
26. Hattaf K. Spatiotemporal Dynamics of a Generalized Viral Infection Model with Distributed Delays and CTL Immune Response. Computation. 2019;7: 21. doi:10.3390/computation7020021
27. Koullapis PG, Stylianou FS, Sznitman J, Olsson B, Kassinos SC. Towards whole-lung simulations of aerosol deposition: A model of the deep lung. J Aerosol Sci. 2020;144: 105541. doi:10.1016/j.jaerosci.2020.105541
28. Strain MC, Richman DD, Wong JK, Levine H. Spatiotemporal Dynamics of HIV Propagation. Journal of Theoretical Biology. 2002;218: 85–96. doi:10.1006/jtbi.2002.3055
29. Dong X, Foteinou PT, Calvano SE, Lowry SF, Androulakis IP. Agent-Based Modeling of Endotoxin-Induced Acute Inflammatory Response in Human Blood Leukocytes. PLOS ONE. 2010;5: e9249. doi:10.1371/journal.pone.0009249
30. Linderman JJ, Riggs T, Pande M, Miller M, Marino S, Kirschner DE. Characterizing the Dynamics of CD4+ T Cell Priming within a Lymph Node. The Journal of Immunology. 2010;184: 2873–2885. doi:10.4049/jimmunol.0903117
31. Immonen T, Gibson R, Leitner T, Miller M, Arts E, Somersalo E, et al. A hybrid

stochastic-deterministic computational model accurately describes spatial dynamics and virus diffusion in HIV-1 growth competition assay. Journal of theoretical biology. 2012;312C: 120–132. doi:10.1016/j.jtbi.2012.07.005

32. Zorzenon dos Santos RM, Coutinho S. Dynamics of HIV Infection: A cellular automata approach. Phys Rev Lett. 2001;87: 168102. doi:10.1103/PhysRevLett.87.168102
33. Imle A, Kumberger P, Schnellbächer ND, Fehr J, Carrillo-Bustamante P, Ales J, et al. Experimental and computational analyses reveal that environmental restrictions shape HIV-1 spread in 3D cultures. Nat Commun. 2019;10: 2144. doi:10.1038/s41467-019-09879-3
34. Myers MA, Smith AP, Lane LC, Moquin DJ, Vogel P, Woolard S, et al. Dynamically linking influenza virus infection with lung injury to predict disease severity. bioRxiv. 2019; 555276. doi:10.1101/555276
35. Funk GA, Jansen VAA, Bonhoeffer S, Killingback T. Spatial models of virus-immune dynamics. Journal of Theoretical Biology. 2005;233: 221–236. doi:10.1016/j.jtbi.2004.10.004
36. Lippi G, Sanchis-Gomar F, Henry BM. COVID-19: unravelling the clinical progression of nature's virtually perfect biological weapon. Ann Transl Med. 2020;8. doi:10.21037/atm-20-3989
37. Carrasco Pro S, Dafonte Imedio A, Santoso CS, Gan KA, Sewell JA, Martinez M, et al. Global landscape of mouse and human cytokine transcriptional regulation. Nucleic Acids Research. 2018;46: 9321–9337. doi:10.1093/nar/gky787
38. Akira S, Uematsu S, Takeuchi O. Pathogen Recognition and Innate Immunity. Cell. 2006;124: 783–801. doi:10.1016/j.cell.2006.02.015
39. Ahlquist P, Noueiry AO, Lee W-M, Kushner DB, Dye BT. Host Factors in Positive-Strand RNA Virus Genome Replication. J Virol. 2003;77: 8181. doi:10.1128/JVI.77.15.8181-8186.2003
40. Shuster DE, Kehrli ME, Rainard P, Paape M. Complement fragment C5a and inflammatory cytokines in neutrophil recruitment during intramammary infection with Escherichia coli. Infect Immun. 1997;65: 3286.
41. Strieter RM, Lukacs NW, Standiford TJ, Kunkel SL. Cytokines. 2. Cytokines and lung inflammation: mechanisms of neutrophil recruitment to the lung. Thorax. 1993;48: 765. doi:10.1136/thx.48.7.765
42. Chou RC, Kim ND, Sadik CD, Seung E, Lan Y, Byrne MH, et al. Lipid-Cytokine-Chemokine Cascade Drives Neutrophil Recruitment in a Murine Model of Inflammatory Arthritis. Immunity. 2010;33: 266–278. doi:10.1016/j.immuni.2010.07.018
43. Edelblum KL, Turner JR. Chapter 12 - Epithelial cells: structure, transport, and barrier function. In: Mestecky J, Strober W, Russell MW, Kelsall BL, Cheroutre H, Lambrecht BN, editors. Mucosal Immunology (Fourth Edition). Boston: Academic Press; 2015. pp. 187–210. doi:10.1016/B978-0-12-415847-4.00012-4
44. Smith AM. Validated Models of Immune Response to Virus Infection. Curr Opin Syst Biol. 2018/10/31 ed. 2018;12: 46–52. doi:10.1016/j.coisb.2018.10.005
45. Smith AM, Perelson AS. Influenza A virus infection kinetics: quantitative data and models. Wiley Interdiscip Rev Syst Biol Med. 2011;3: 429–445. doi:10.1002/wsbm.129

46. Dusséaux M, Masse-Ranson G, Darche S, Ahodantin J, Li Y, Fiquet O, et al. Viral Load Affects the Immune Response to HBV in Mice With Humanized Immune System and Liver. Gastroenterology. 2017;153: 1647-1661.e9. doi:10.1053/j.gastro.2017.08.034
47. Veiga-Parga T, Sehrawat S, Rouse BT. Role of Regulatory T Cells during Virus Infection. Immunol Rev. 2013;255: 182–196. doi:10.1111/imr.12085
48. Rosendahl Huber S, van Beek J, de Jonge J, Luytjes W, van Baarle D. T cell responses to viral infections - opportunities for Peptide vaccination. Front Immunol. 2014;5: 171–171. doi:10.3389/fimmu.2014.00171
49. Le Page C, Génin P, Baines M, Hiscott J. Interferon activation and innate immunity. Rev Immunogenet. 2000;2: 374–386.
50. Abbas AK, Lichtman AH, Pillai S. Cellular and molecular immunology. 9th ed. Philadelphia, PA: Saunders/Elsevier; 2017.
51. Ploquin A, Zhou Y, Sullivan NJ. Ebola Immunity: Gaining a Winning Position in Lightning Chess. JI. 2018;201: 833–842. doi:10.4049/jimmunol.1700827
52. Regoes RR, Crotty S, Antia R, Tanaka MM. Optimal Replication of Poliovirus within Cells. The American Naturalist. 2005;165: 364–373. doi:10.1086/428295
53. Yin J, Redovich J. Kinetic Modeling of Virus Growth in Cells. Microbiol Mol Biol Rev. 2018;82: e00066-17. doi:10.1128/MMBR.00066-17
54. Yu X, Zeng X, Shi W, Hu Y, Nie W, Chu N, et al. Validation of Cycloserine Efficacy in Treatment of Multidrug-Resistant and Extensively Drug-Resistant Tuberculosis in Beijing, China. Antimicrobial Agents and Chemotherapy. 2018;62. doi:10.1128/AAC.01824-17
55. Swat MH, Thomas GL, Belmonte JM, Shirinifard A, Hmeljak D, Glazier JA. Multi-scale modeling of tissues using CompuCell3D. Methods Cell Biol. 2012;110: 325–366. doi:10.1016/B978-0-12-388403-9.00013-8
56. Harcourt J, Tamin A, Lu X, Kamili S, Sakthivel SK, Murray J, et al. Severe Acute Respiratory Syndrome Coronavirus 2 from Patient with Coronavirus Disease, United States. Emerg Infect Dis. 2020;26: 1266–1273. doi:10.3201/eid2606.200516
57. Cheng AH-D, Cheng DT. Heritage and early history of the boundary element method. Eng Anal Bound Elem. 2005;29: 268–302. doi:10.1016/j.enganabound.2004.12.001
58. Zhao X, Wang W. Fisher waves in an epidemic model. Discrete Cont Dyn-B. 2004;4: 1117–1128. doi:10.3934/dcdsb.2004.4.1117
59. Dahari H, Ribeiro RM, Rice CM, Perelson AS. Mathematical modeling of subgenomic hepatitis C virus replication in Huh-7 cells. J Virol. 2007;81: 750. doi:10.1128/JVI.01304-06
60. Warren TK, Jordan R, Lo MK, Ray AS, Mackman RL, Soloveva V, et al. Therapeutic efficacy of the small molecule GS-5734 against Ebola virus in rhesus monkeys. Nature. 2016/03/02 ed. 2016;531: 381–385. doi:10.1038/nature17180
61. Gonçalves A, Bertrand J, Ke R, Comets E, Lamballerie X de, Malvy D, et al. Timing of Antiviral Treatment Initiation is Critical to Reduce SARS- CoV- 2 Viral Load. In: CPT: Pharmacometrics & Systems Pharmacology [Internet]. John Wiley & Sons, Ltd; 7 Aug 2020 [cited 1 Sep 2020]. Available: https://ascpt.onlinelibrary.wiley.com/doi/abs/10.1002/psp4.12543
62. Goyal A, Cardozo-Ojeda EF, Schiffer JT. Potency and timing of antiviral therapy

as determinants of duration of SARS CoV-2 shedding and intensity of inflammatory response. medRxiv. 2020; 2020.04.10.20061325. doi:10.1101/2020.04.10.20061325

63. Weaver JJA, Shoemaker JE. Mathematical Modeling of RNA Virus Sensing Pathways Reveals Paracrine Signaling as the Primary Factor Regulating Excessive Cytokine Production. Processes. 2020;8: 719. doi:10.3390/pr8060719

64. Mirams GR, Arthurs CJ, Bernabeu MO, Bordas R, Cooper J, Corrias A, et al. Chaste: an open source C++ library for computational physiology and biology. PLoS Comput Biol. 2013;9: e1002970. doi:10.1371/journal.pcbi.1002970

65. Pitt-Francis J, Pathmanathan P, Bernabeu MO, Bordas R, Cooper J, Fletcher AG, et al. Chaste: A test-driven approach to software development for biological modelling. Comput Phys Commun. 2009;180: 2452–2471. doi:10.1016/j.cpc.2009.07.019

66. Starruß J, de Back W, Brusch L, Deutsch A. Morpheus: a user-friendly modeling environment for multiscale and multicellular systems biology. Bioinformatics. 2014;30: 1331–1332. doi:10.1093/bioinformatics/btt772

67. Smith LP, Bergmann FT, Chandran D, Sauro HM. Antimony: a modular model definition language. Bioinformatics. 2009/07/03 ed. 2009;25: 2452–2454. doi:10.1093/bioinformatics/btp401

68. Graner F, Glazier JA. Simulation of biological cell sorting using a two-dimensional extended Potts model. Phys Rev Lett. 1992;69: 2013–2016. doi:10.1103/PhysRevLett.69.2013

69. Oyler-Yaniv A, Oyler-Yaniv J, Whitlock BM, Liu Z, Germain RN, Huse M, et al. A tunable diffusion-consumption mechanism of cytokine propagation enables plasticity in cell-to-cell communication in the immune system. Immunity. 2017;46: 609–620. doi:10.1016/j.immuni.2017.03.011

70. Höfer T, Krichevsky O, Altan-Bonnet G. Competition for IL-2 between regulatory and effector T cells to chisel immune responses. Front Immunol. 2012;3: 268–268. doi:10.3389/fimmu.2012.00268

71. Hugues S, Fetler L, Bonifaz L, Helft J, Amblard F, Amigorena S. Distinct T cell dynamics in lymph nodes during the induction of tolerance and immunity. Nature Immunology. 2004;5: 1235–1242. doi:10.1038/ni1134

72. Walls AC, Park Y-J, Tortorici MA, Wall A, McGuire AT, Veesler D. Structure, Function, and Antigenicity of the SARS-CoV-2 Spike Glycoprotein. Cell. 2020;181: 281-292.e6. doi:10.1016/j.cell.2020.02.058

73. Wrapp D, Wang N, Corbett KS, Goldsmith JA, Hsieh C-L, Abiona O, et al. Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. Science. 2020;367: 1260–1263. doi:10.1126/science.abb2507

74. Lai SK, Wang Y-Y, Wirtz D, Hanes J. Micro- and macrorheology of mucus. Adv Drug Deliv Rev. 2009;61: 86–100. doi:10.1016/j.addr.2008.09.012

75. Stray SJ, Air GM. Apoptosis by influenza viruses correlates with efficiency of viral mRNA synthesis. Virus Res. 2001;77: 3–17. doi:10.1016/s0168-1702(01)00260-x

Acknowledgments

We would like to thank Ramray Bhat, Ryan Danehy, Javier Toledo, Bard Ermentrout, Geoffrey Fox, Yi Jiang, Paul Macklin, Jairaj Mathur, Roeland Merks, Ericka Mochan, Jason Shoemaker and Amber Smith for useful discussions and ideas and for comments on the manuscript. JAG, TJS, JFG and JAS acknowledge funding from National Institutes of Health grants U24 EB028887 and R01 GM122424 and National Science Foundation grant NSF 1720625. LB acknowledges grant 01ZX1707A within the German e:Med initiative (BMBF) and grant 391070520 from the German Research Foundation (DFG). R.K.P. was supported, in part, by National Institutes of Health grants R01 AI071002, R01 AI141222 and R01 HD079327. This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute. The funders had no role in manuscript preparation or the decision to submit the work for publication.

Competing Interests

JAG is the owner/operator of Virtual Tissues for Health, LLC, which develops applications of multiscale tissue models in medical applications and is a shareholder in Gilead Life Sciences.

Supplementary Materials

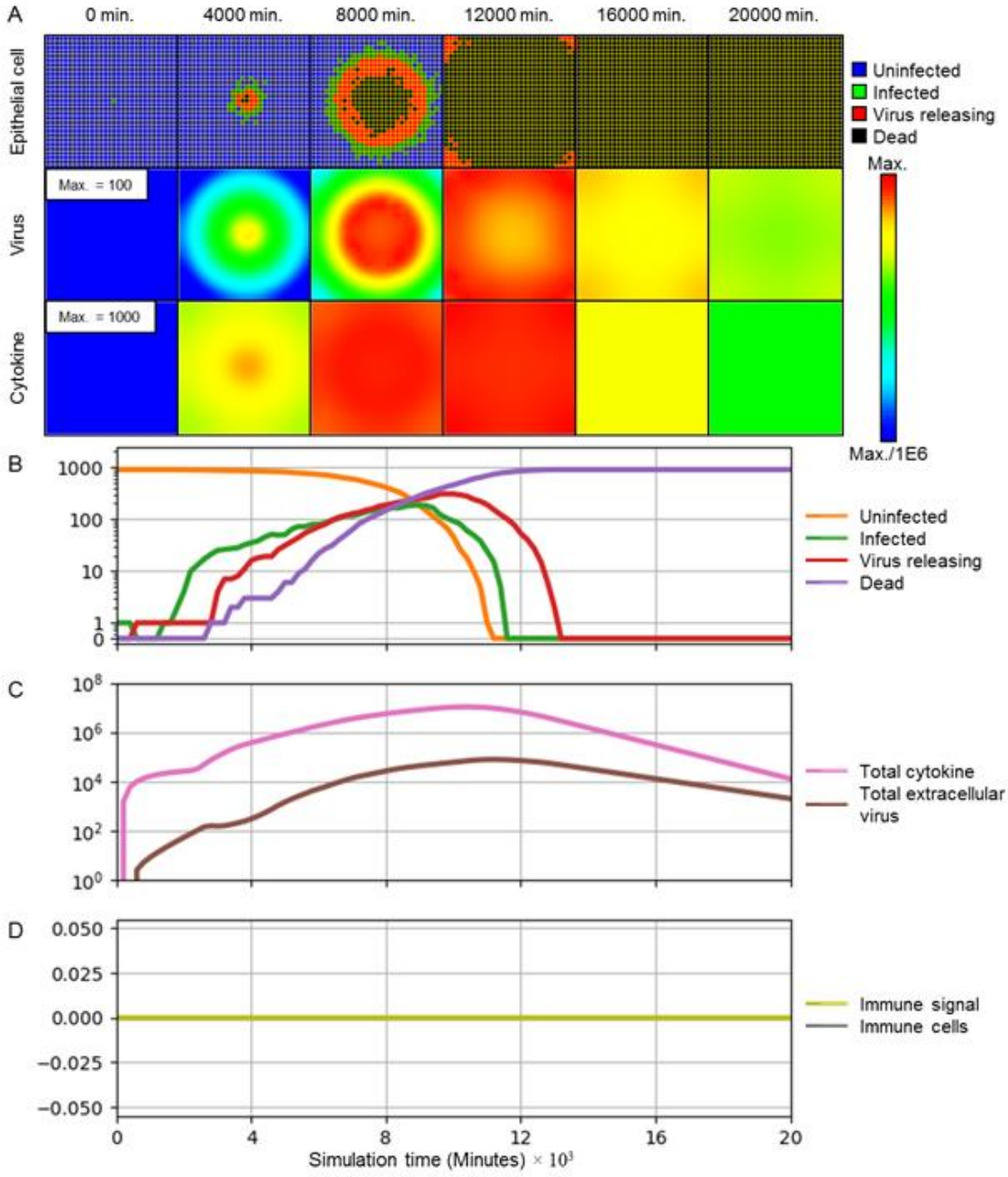


Fig S1. No immune system.

Simulation of the progression of infection in a patch of epithelial tissue, with all parameters as in Fig 3, but with no cellular immune system response corresponding to virus spread in an *in vitro* or organoid culture, or a severely immunosuppressed individual. (A) Snapshots of spatial configuration *vs* time showing progression of simulated

infection. Columns, left to right: 0 minutes (time of initial infection), 4000 minutes (67 hours, 2 ¾ days) after infection, 8000 minutes (133 hours, 5 ½ days), 12000 minutes (200 hours, 8 ⅓ days), 16000 minutes (267 hours, 11 days), and 20000 minutes (333 hours, 14 days). First row: epithelial cell layer composed of uninfected (blue), infected (green), virus releasing (red) cells and dead cells (black). Second row: level of extracellular virus field. Third row: extracellular cytokine field. Fields are color-coded on a logarithmic scale: red corresponds to the chosen maximum value specified in the first panel, blue to six orders of magnitude lower than the maximum value, and values outside this range are colored as their closest border value. (B-D) Simulation time series. (B) Number of uninfected (orange), infected (green), virus releasing (red) and dead (purple) epithelial cells on a logarithmic scale *vs* time *vs* time in minutes. (C) Total extracellular cytokine (magenta) and total extracellular virus (brown) on a logarithmic scale *vs* time in minutes. (D) Value of the immune recruitment signal S (yellow) and number of immune cells (grey) on a linear scale *vs* time in minutes.

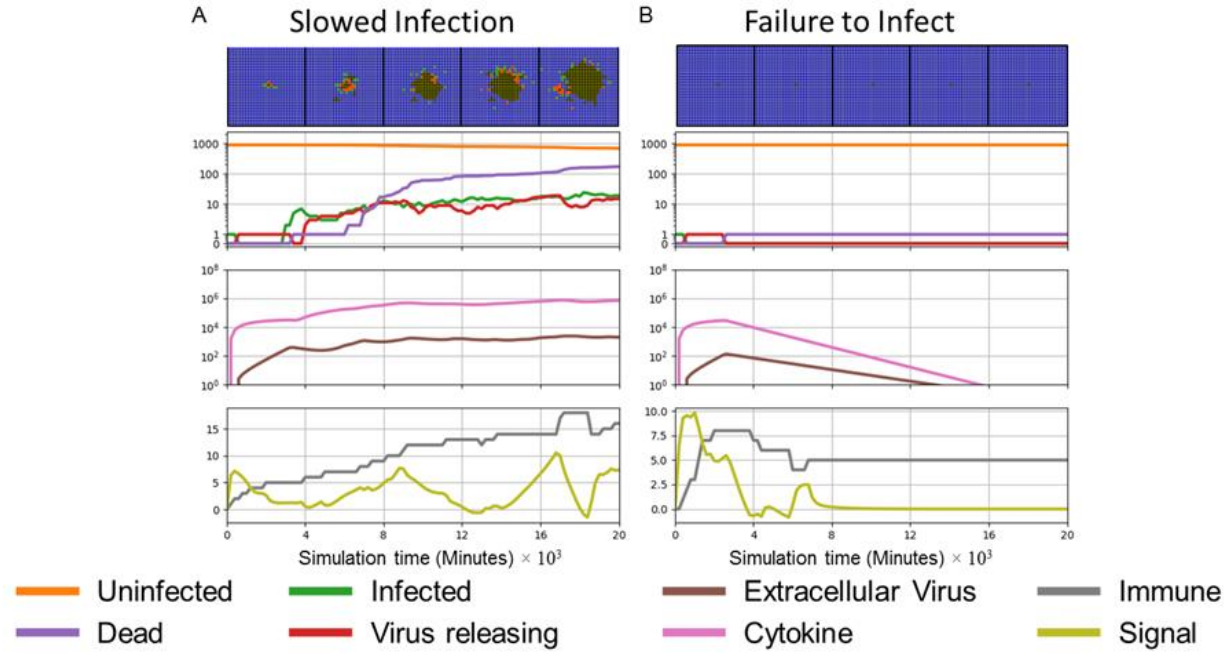


Fig S2. Special cases of spatiotemporal infection dynamics.

(A) A border case of slowed infection towards containment is Slowed Infection with constant virus: when the net effectiveness of viral and immune dynamics are balanced, the number of infected cells and the total extracellular virus fluctuate around steady state levels. (B) A limit case of Clearance is the failure to infect: initially infected cells may replicate and secrete virus, but insufficiently so to infect other cells during simulation time such that any initially infected cells vanish and total extracellular virus drops below a threshold of 10^{-3} per cell area.

Table S1. Varying parameters in simulations shown in Fig 4. Virus-receptor association affinity and immune response delay coefficient shown for no immune response (Fig 4A), widespread infection (Fig 4B), slowed infection (Fig 4C), containment (Fig 4D), recurrence (Fig 4E) and clearance (Fig 4F). All other parameters are as in Table 1.

Parameter	No immune response	Widespread infection	Slowed infection	Containment	Recurrence	Clearance
Virus-receptor association affinity k_{on} (M^{-1} s^{-1})	1.4×10^{4}	1.4×10^{4}	1.4×10^{3}	1.4×10^{2}	1.4×10^{6}	1.4×10^{4}
Immune response delay coefficient β_{delay} (s)	-	1.2×10^{7} s	1.2×10^{8} s	1.2×10^{5} s	1.2×10^{4} s	1.2×10^{5} s

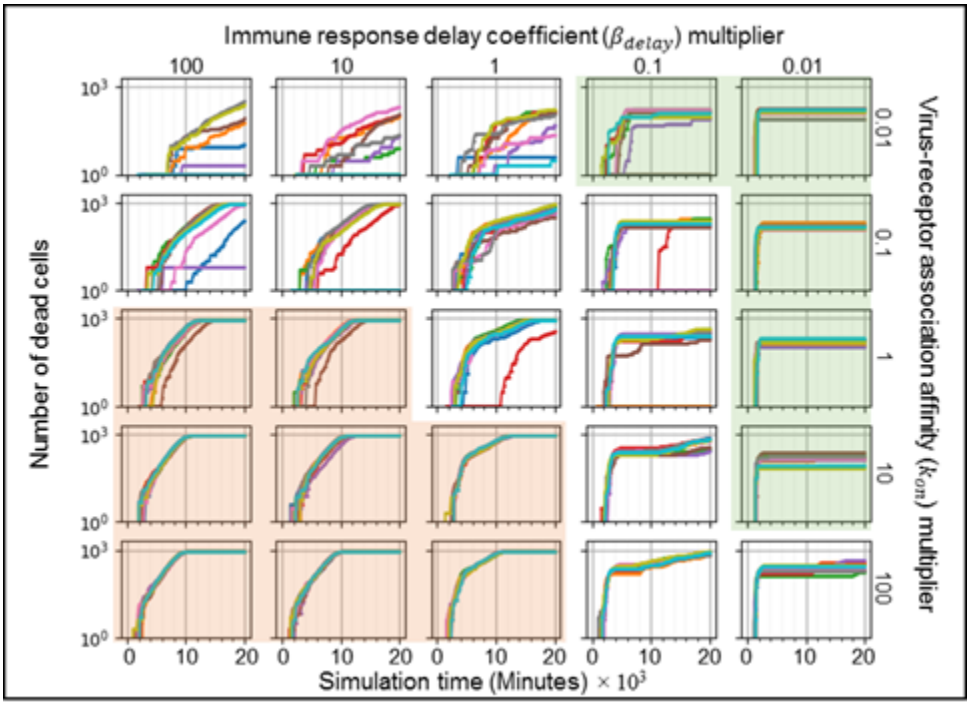


Fig S3. Time series of the number of uninfected cells for simulations in Fig 5.

Logarithmic multidimensional parameter sweep performed by running 10 simulation replicas increasing and decreasing the baseline parameter values 10-fold and 100-fold for all parameter sets and replicas in Fig 5. Results show consistent containment/clearance for small k_{on} and small β_{delay} (upper right, green-shaded subplots), widespread infection for high k_{on} and small β_{delay} (lower left, orange-shaded subplots), and multiple outcomes for the same parameter values (uncolored subplots). Number of cells are shown on a logarithmic scale *vs* time in minutes.

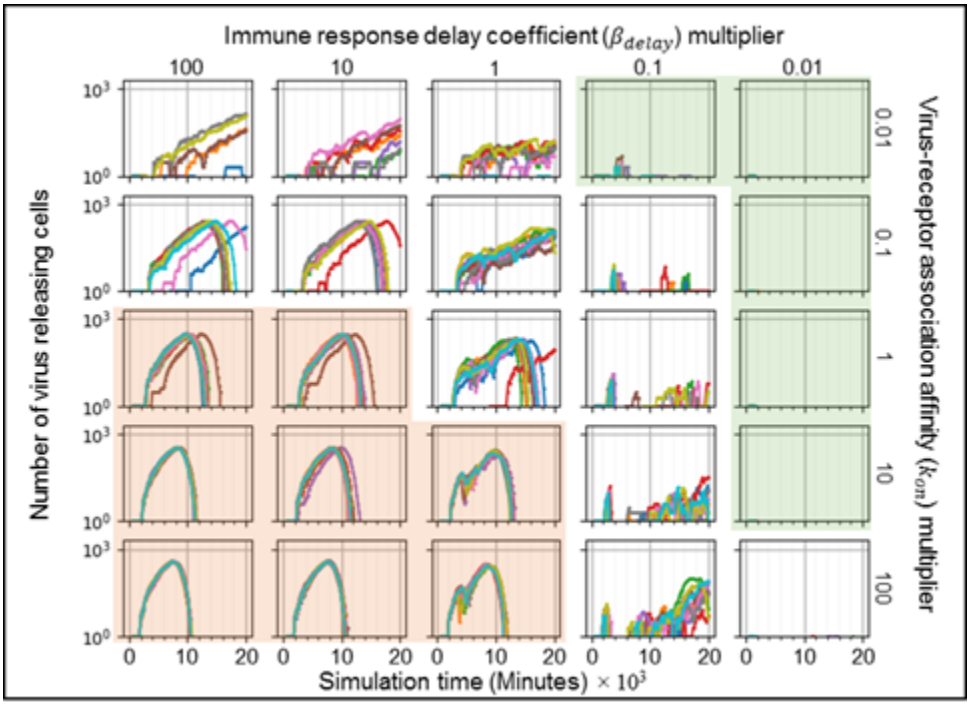


Fig S4. Time series of the number of virus releasing cells for simulations in Fig 5.

Logarithmic multidimensional parameter sweep performed by running 10 simulation replicas increasing and decreasing the baseline parameter values 10-fold and 100-fold for all parameter sets and replicas in Fig 5. Results show consistent containment/clearance for small k_{on} and small β_{delay} (upper right, green-shaded subplots), widespread infection for high k_{on} and small β_{delay} (lower left, orange-shaded subplots), and multiple outcomes for the same parameter values (uncolored subplots). Number of cells are shown on a logarithmic scale *vs* time in minutes.

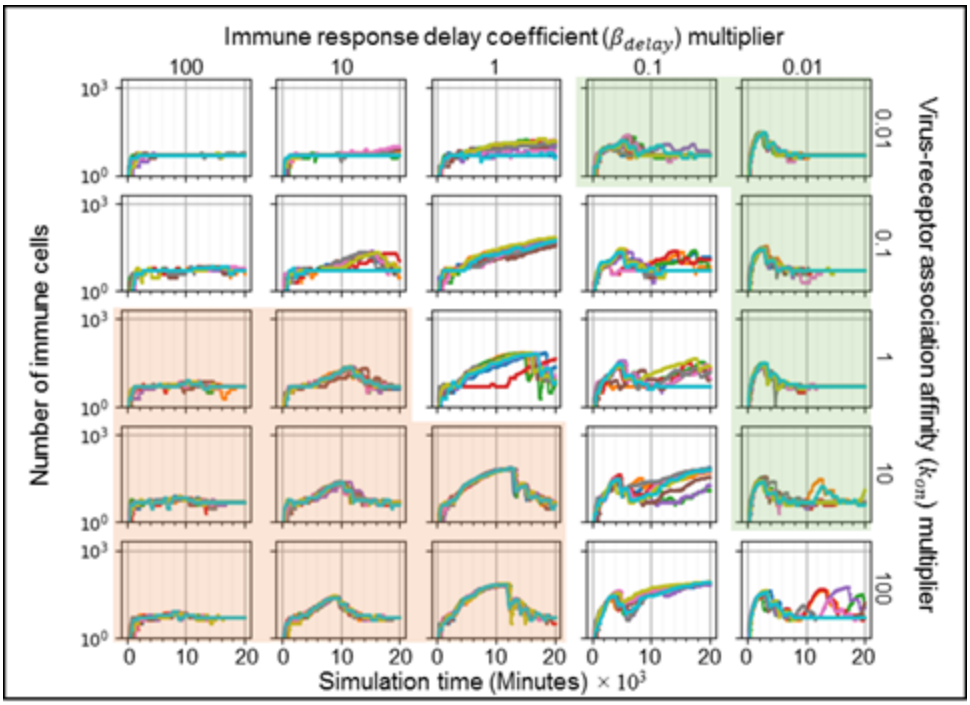


Fig S5. Time series of the number of immune cells for simulations in Fig 5.

Logarithmic multidimensional parameter sweep performed by running 10 simulation replicas increasing and decreasing the baseline parameter values 10-fold and 100-fold for all parameter sets and replicas in Fig 5. Results show consistent containment/clearance for small k_{on} and small β_{delay} (upper right, green-shaded subplots), widespread infection for high k_{on} and small β_{delay} (lower left, orange-shaded subplots), and multiple outcomes for the same parameter values (uncolored subplots). Number of cells are shown on a logarithmic scale *vs* time in minutes.

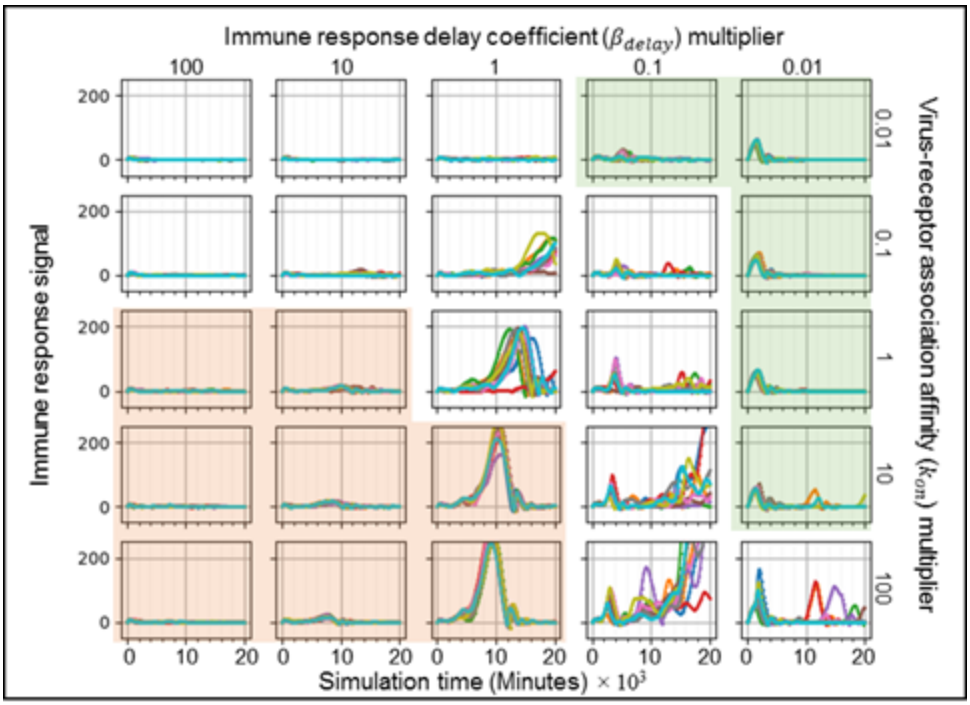


Fig S6. Time series of the immune response state variable S for simulations in Fig 5.

Logarithmic multidimensional parameter sweep performed by running 10 simulation replicas increasing and decreasing the baseline parameter values 10-fold and 100-fold for all parameter sets and replicas in Fig 5. Results show consistent containment/clearance for small k_{on} and small β_{delay} (upper right, green-shaded subplots), widespread infection for high k_{on} and small β_{delay} (lower left, orange-shaded subplots), and multiple outcomes for some parameter sets (unshaded subplots). S is shown on a linear scale *vs* time in minutes.

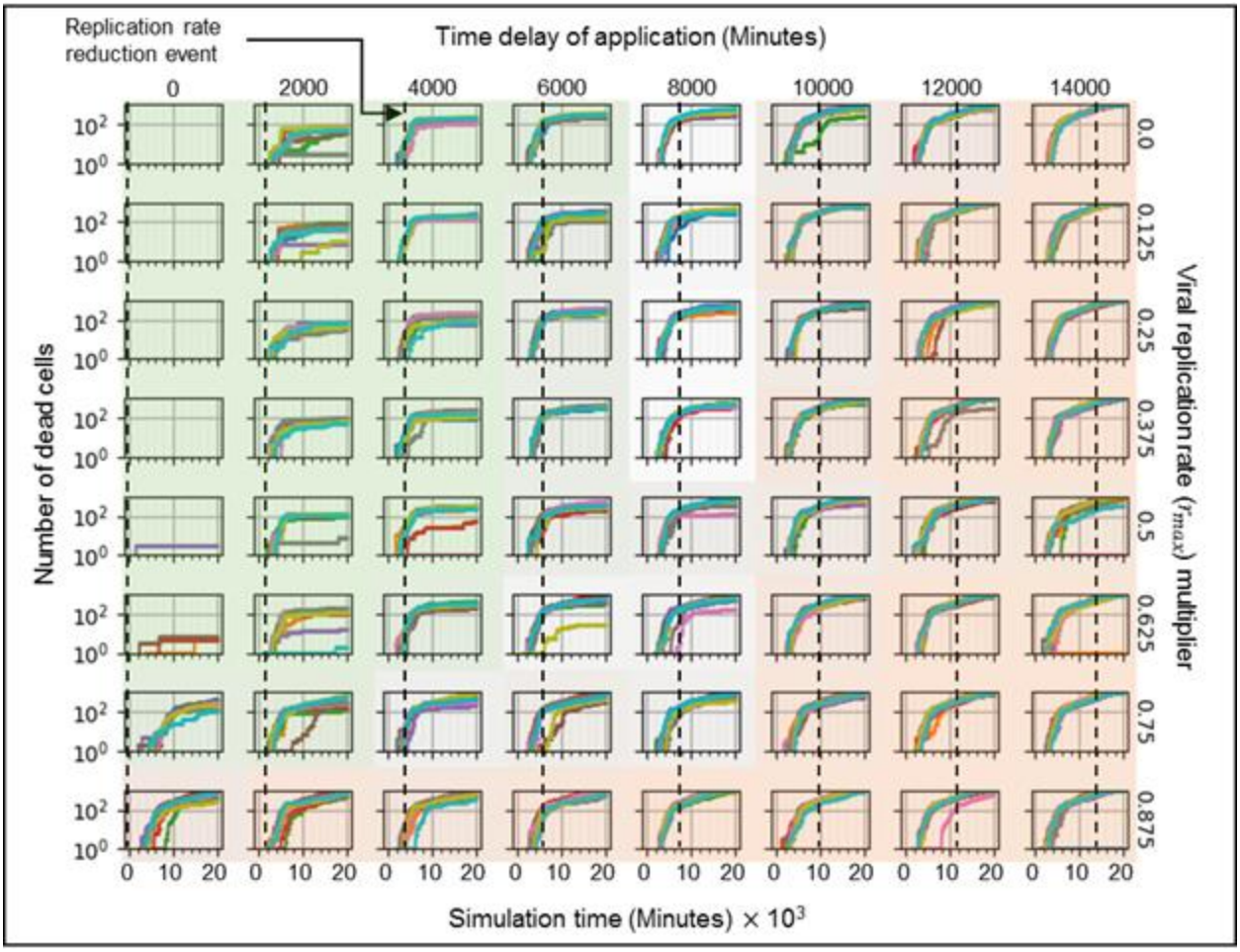


Fig S7. Simulations from Fig 8, showing the number of dead cells.

Time series of the number of dead cells for each simulation replica in Fig 8. Number of cells is shown on a logarithmic scale *vs* time in minutes.

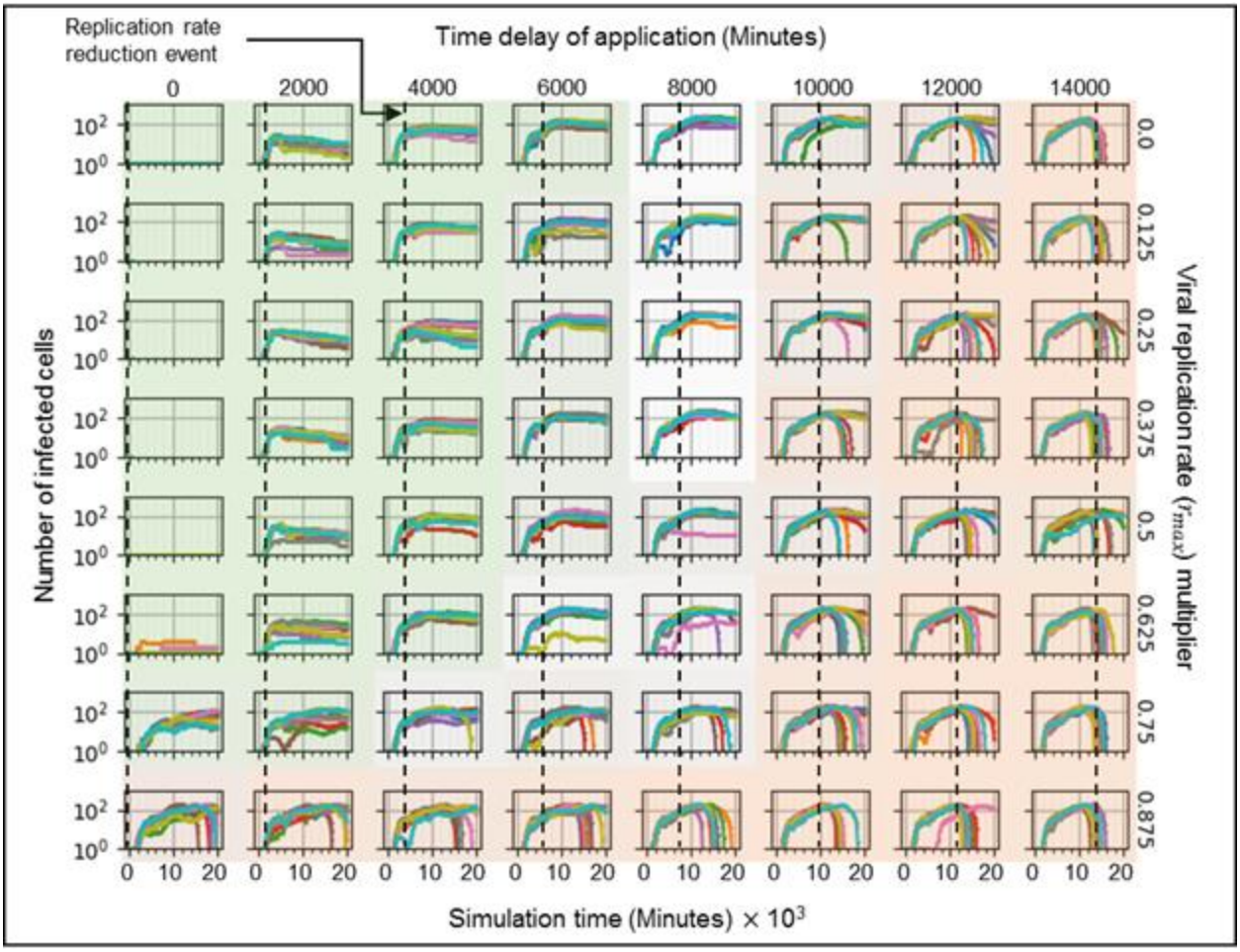


Fig S8. Simulations from Fig 8, showing the number of infected cells.

Time series of the number of infected cells for each simulation replica in Fig 8. Number of cells is shown on a logarithmic scale *vs* time in minutes.

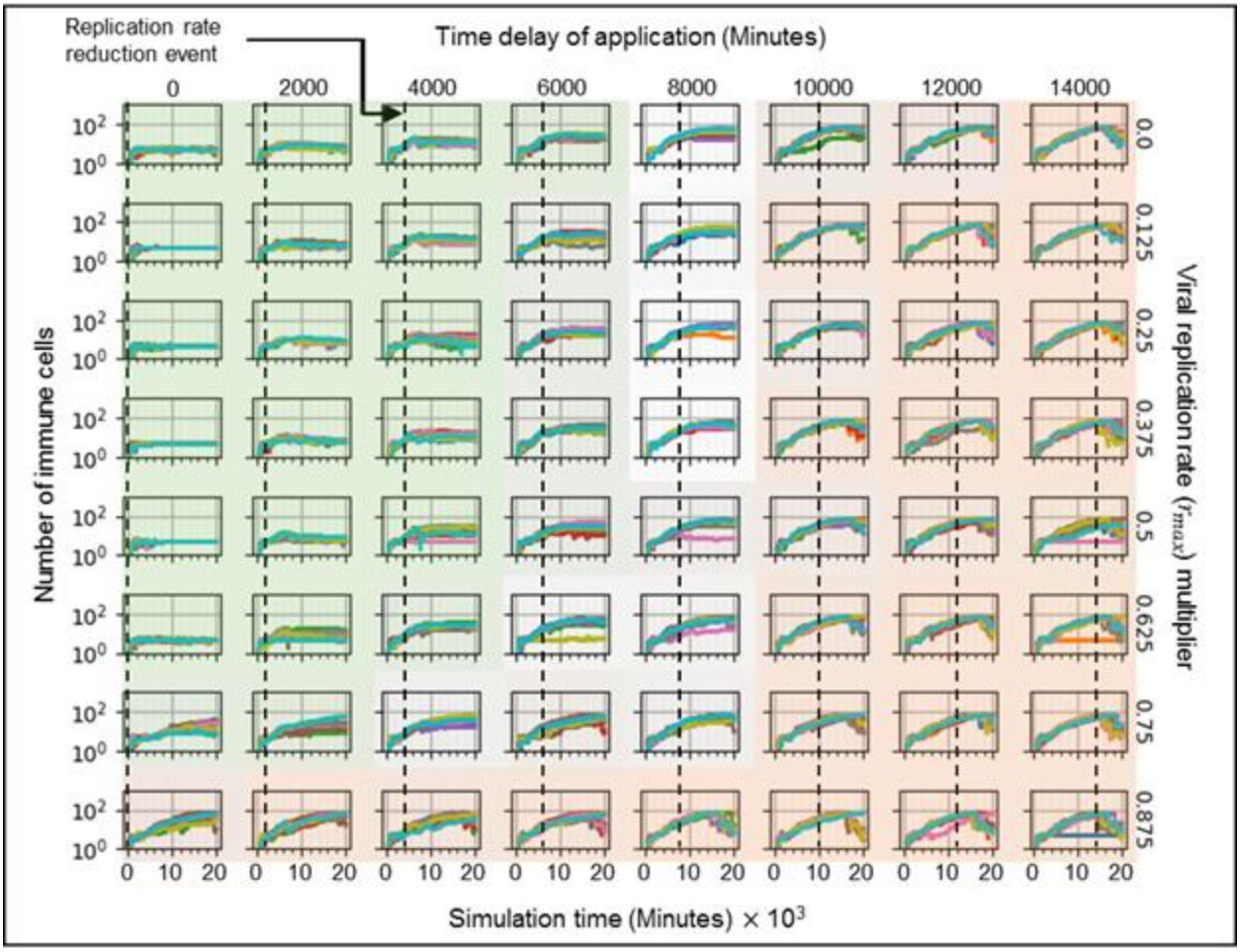


Fig S9. Simulations from Fig 8 showing the number of immune cells.

Time series of the number of immune cells for each simulation replica in Fig 8. Number of cells is shown on a logarithmic scale *vs* time in minutes.

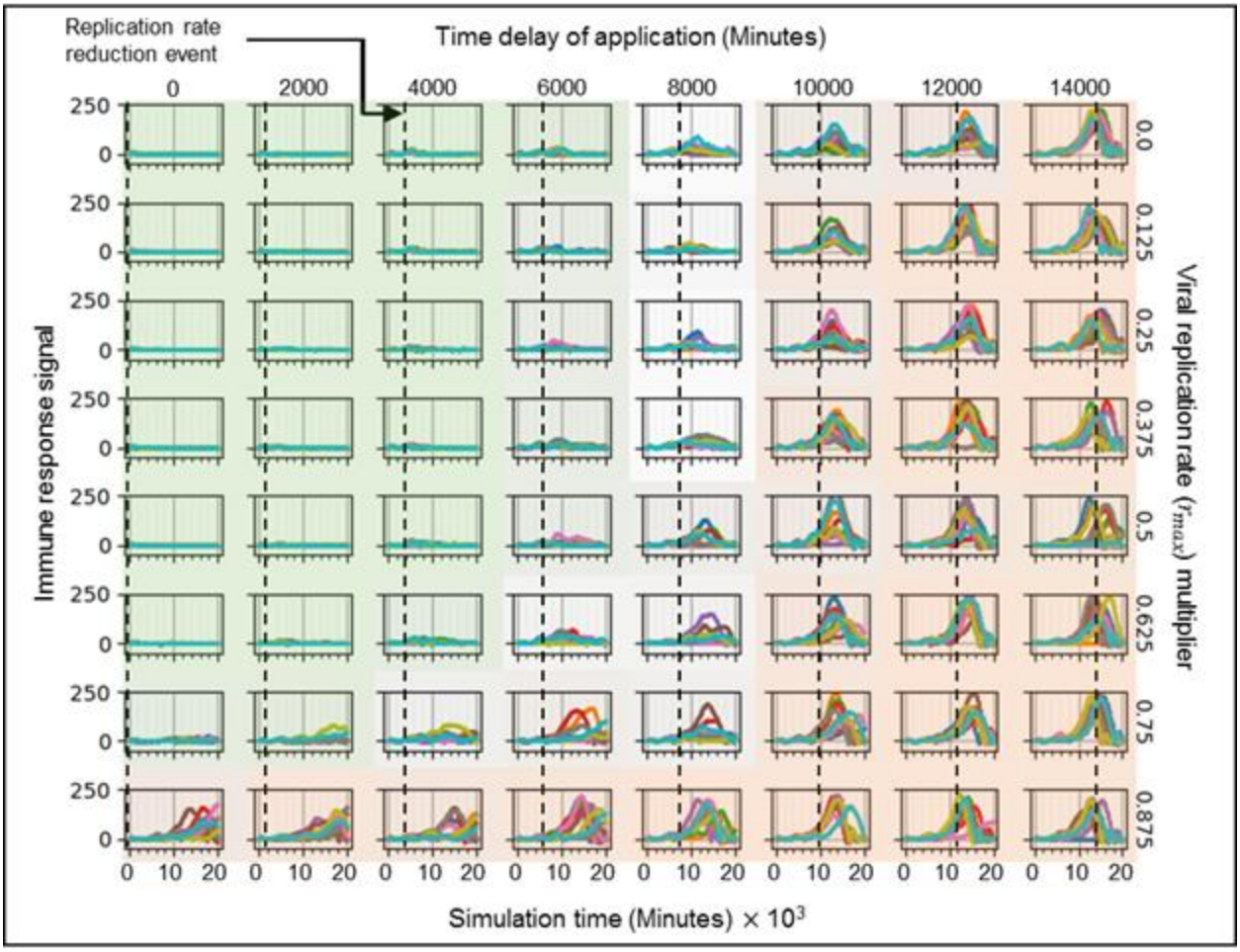


Fig S10. Simulations from Fig 8 showing the immune response state variable.

Time series of the immune response state variable S for each simulation replica in Fig 8. S is shown on a linear scale *vs* time in minutes.

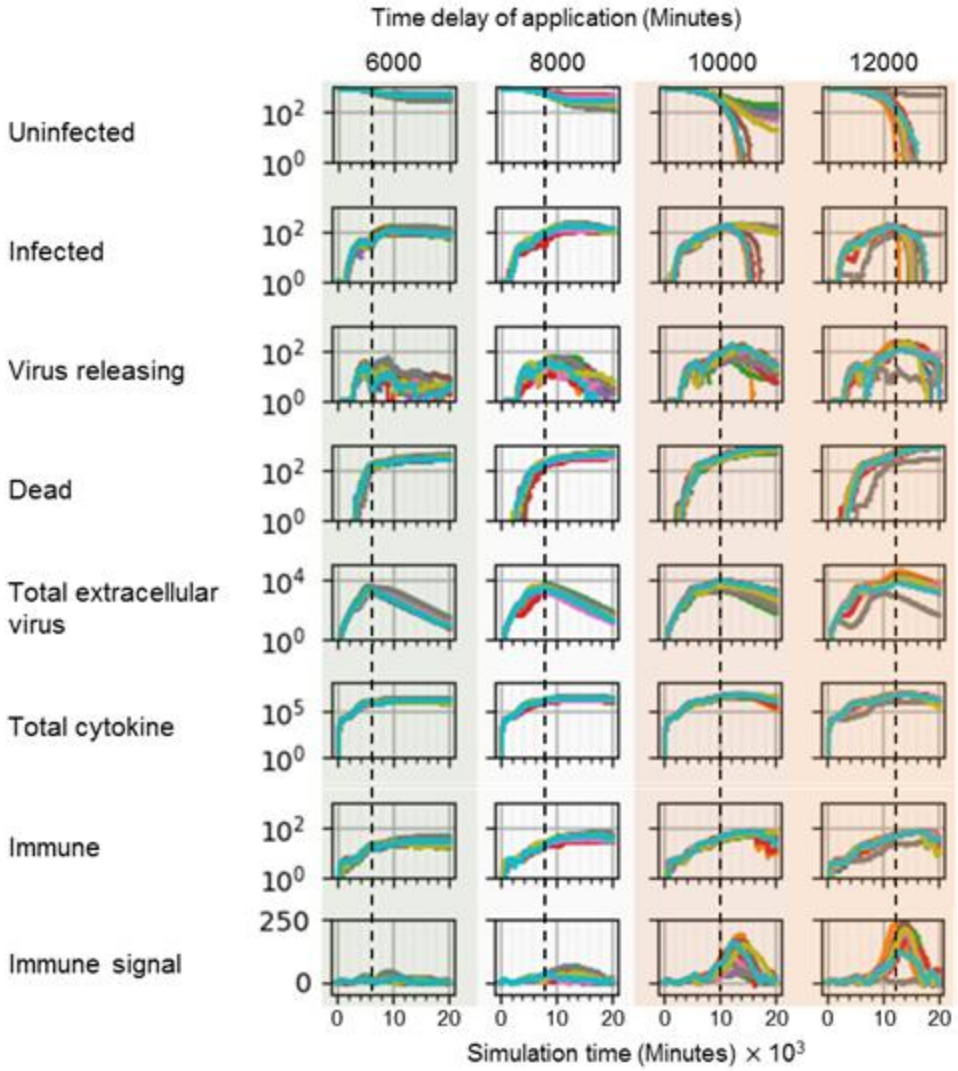


Fig S11. Variation in time of first treatment after infection with a reduced viral RNA replication rate causes a bifurcation in simulation outcomes.

Simulations and parameters are as in Figs 9-11, for a viral replication rate multiplier of 0.375 and, from left to right, time delays of application of 6000, 8000, 10000, and 12000 minutes (100, 133, 167 and 200 hours, 4, 5 ½, 7 and 8 ⅓ days) (dashed lines). Results from all simulation replicas are shown vs time in minutes for, from top to bottom: number of uninfected cells, number of infected cells, number of virus releasing cells, number of dead cells, total extracellular virus, total cytokine, number of immune cells, and immune response state variable S. Parameter set subplots are shaded as in Figs 9-11 according to simulation outcomes.

Downloading and running the simulation

The COVID-19 simulation's source code is available in the GitHub repository https://github.com/covid-tissue-models/covid-tissue-response-models/tree/master/CC3D/Models/BiocIU/SARSCoV2MultiscaleVTM. The simulation is a model specification which runs in the CompuCell3D virtual-tissue simulation environment. To run the simulation requires installation of CompuCell3D version 4.1.1 or later. CompuCell3D is open-source and runs on Windows, Mac and Linux operating systems. It can be downloaded from https://compucell3d.org/SrcBin. Installers are available for Windows operating systems and Mac installation also does not require compilation. CompuCell3D's manuals are available at https://compucell3d.org/Manuals. The COVID-19 simulation can also be run online without requiring any installations or downloads on the nanoHUB servers at https://nanohub.org/tools/cc3dcovid19/. Use of nanoHUB is free but requires user registration. The simulation may take a few moments to load in its nanoHUB deployment; during load the simulation area will be blue.

Twedit++ is a specialized text editor for CompuCell3D simulations which comes packaged with CompuCell3D. Tweedit++ can open *cc3d* file extensions which contain the simulation file structure for CompuCell3D simulations. To open the COVID-19 simulation click "Open CC3D Project" (Fig S12) and select *ViralInfectionVTM.cc3d* in `<repository-folder>/covid-tissue-response-models/CC3D/Models/BiocIU/SARSCoV2MultiscaleVTM/Model`. Once opened, *ViralInfectionVTM.cc3d* will appear in the left- hand panel "CC3D Simulation" (Fig S13). Double click on it to open all simulation files. The main simulation files are: *ViralInfectionVTM.xml*, which defines certain simulation properties (*e.g.*, cell types, lattice

size, energy-constraint plugins, diffusive fields); *ViralInfectionVTMSteppables.py*, which defines the simulation's initial conditions, main interactions and dynamics (*e.g.*, cell initialization, immune-cell recruitment, secretion by cells into fields, uptake by cells from fields); *ViralInfectionVTMSteppableBasePy.py*, where the viral infection Antimony submodel is declared; *ViralInfectionVTMModelInputs.py*, which sets the parameters. The submodels in *ViralInfectionVTMSteppables.py* are organized as python classes, making them easy to modify. Tweedit++ can also copy and rename the simulation project to a new directory by using `CC3D Project; Save Project As`. However the save as does not copy the folder `<...>/Model/nCoVToolkit`, which must be copied into the new simulation directory separately.

CompuCell3D *Player* is a GUI tool which executes CompuCell3D simulations during desktop execution (or on nanoHUB). In order to run the simulation either right click *ViralInfectionVTM.cc3d* in the left hand panel and select "Open In Player" or open *CompuCell3D.exe* to open the CompuCell3D Player and select `File; Open Simulation File` and open *ViralInfectionVTM.cc3d*. Once the simulation is open in Player click play (on the nanoHub deployment the simulation should start automatically). Player will display windows with the cell lattice rendered (Fig S14), set the z-plane to 0 to visualize the epithelial cells and the z-plane to 1 to visualize the immune cells. More windows can be created (menu `Window; New graphics Window,` Fig S15) as needed to visualize the virus, cytokine, oxidative agent fields. Each window has a drop-down menu containing the selection of fields that can be rendered (*i.e.*, the chemical fields and the cell field, Fig S16).

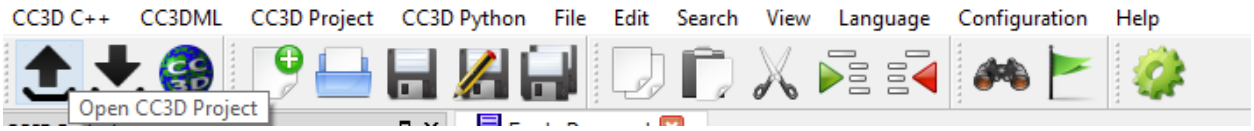


Fig S12. Opening a CompuCell3D project in Tweedit++.

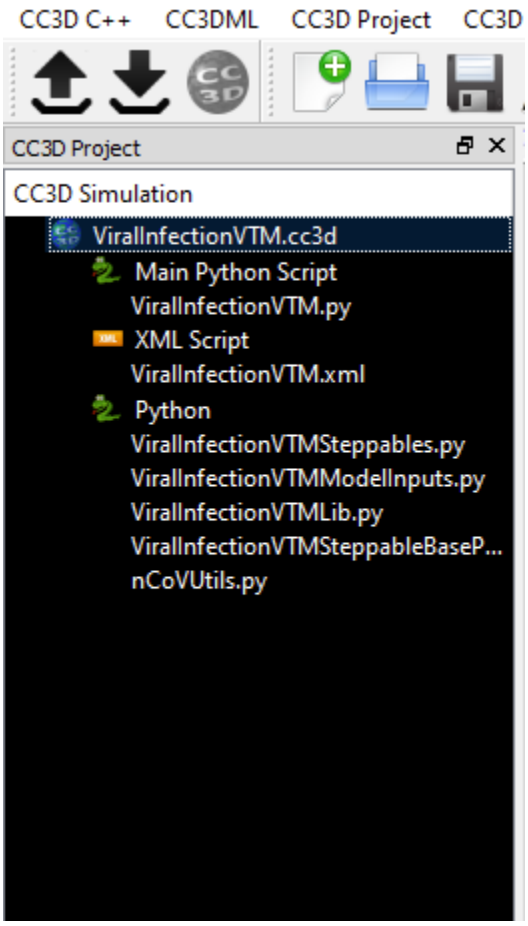


Fig S13. Tweedit++'s left hand panel with simulation project files opened.

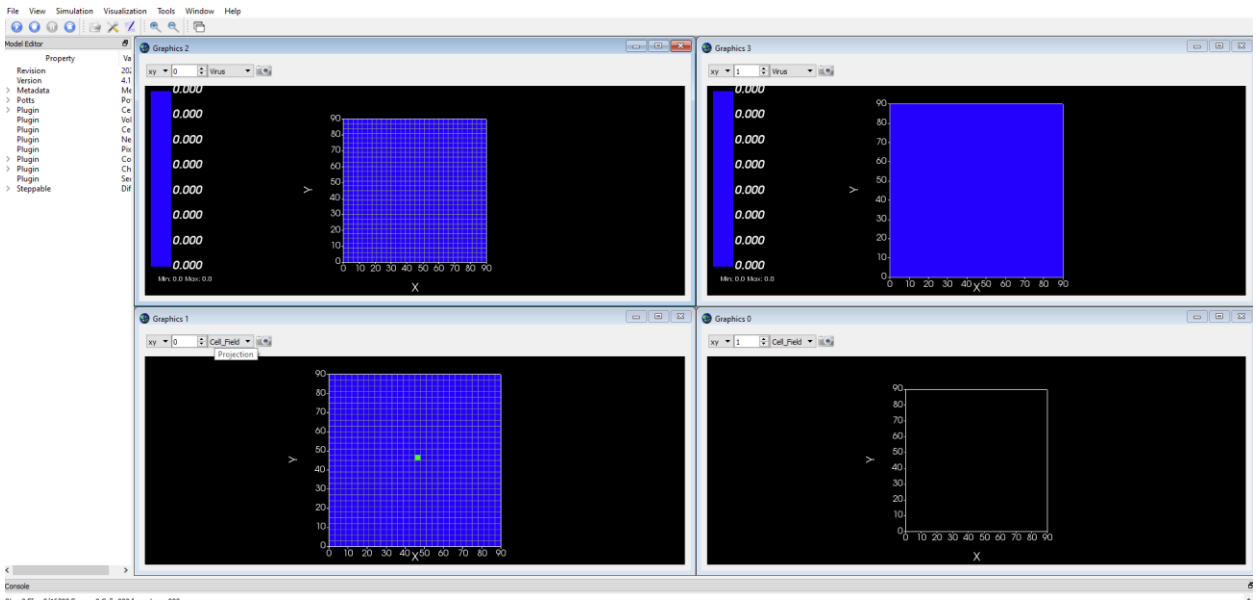


Fig S14. Example of CompuCell3D's player open with the COVID-19 simulation loaded.

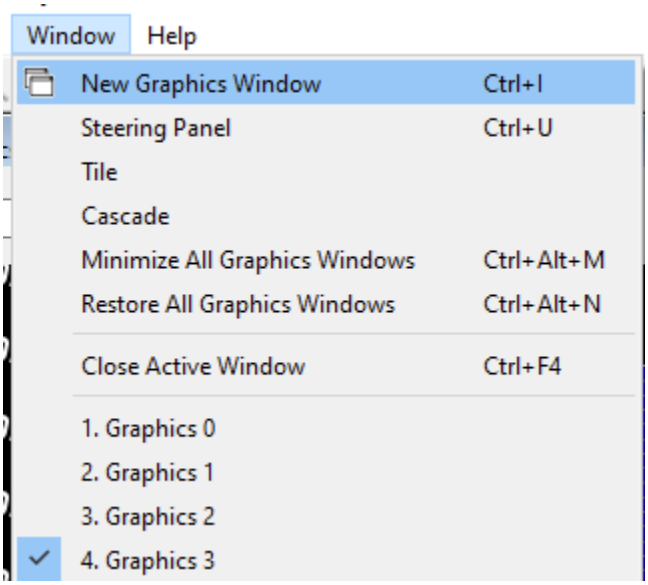


Fig S15. How to open a new simulation render window in CompuCell3D Player.

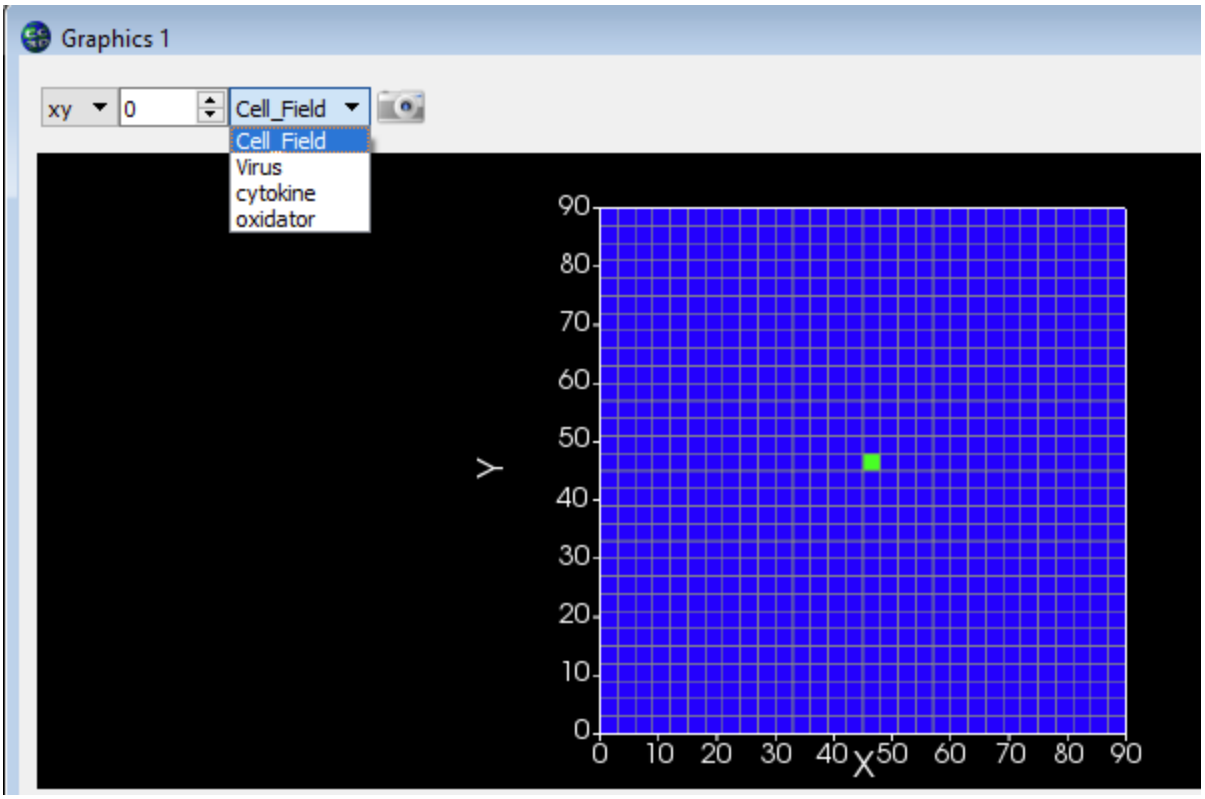


Fig S16. Drop-down menu in simulation render window to select which field to render.

Sensitivity analysis of the baseline parameter set

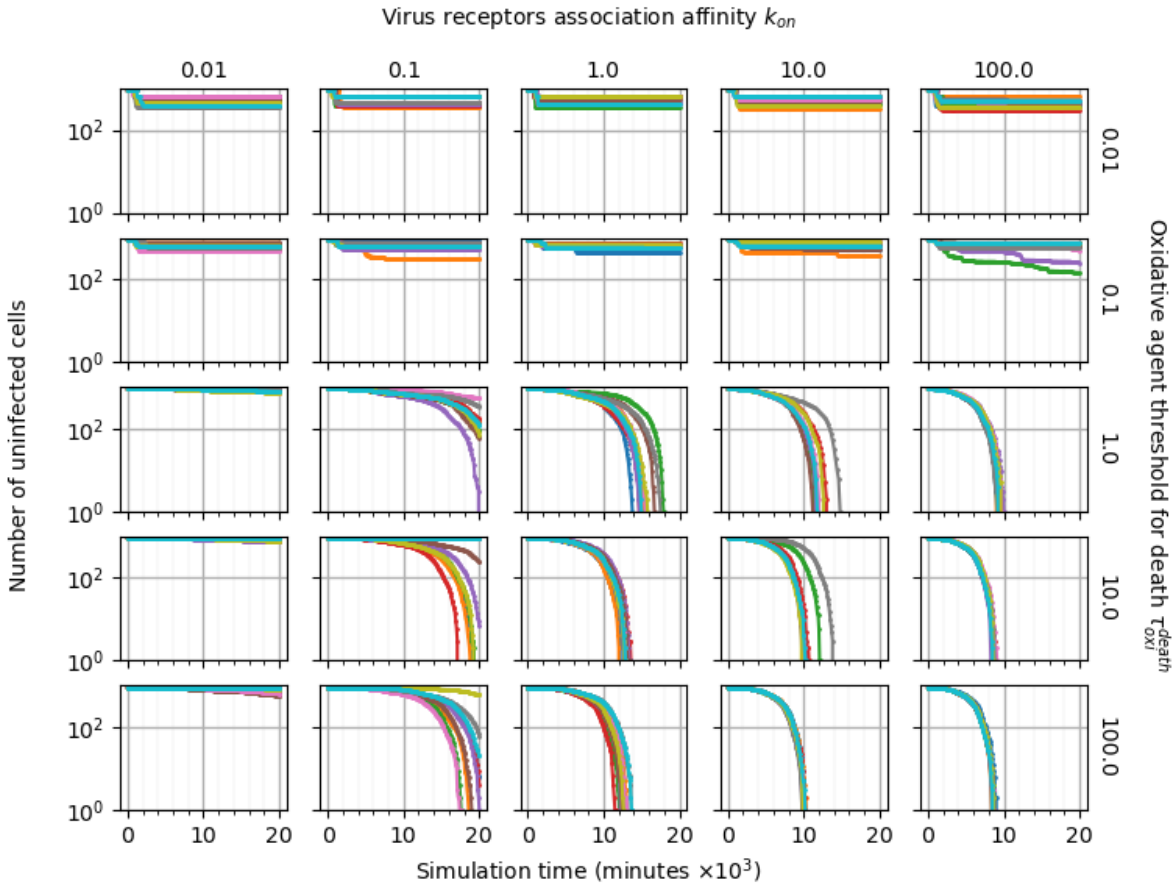


Fig S17. Pairwise parameter sweep of the oxidative agent threshold for death τ_{oxi}^{death} and the virus-receptor association affinity k_{on} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1).

The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

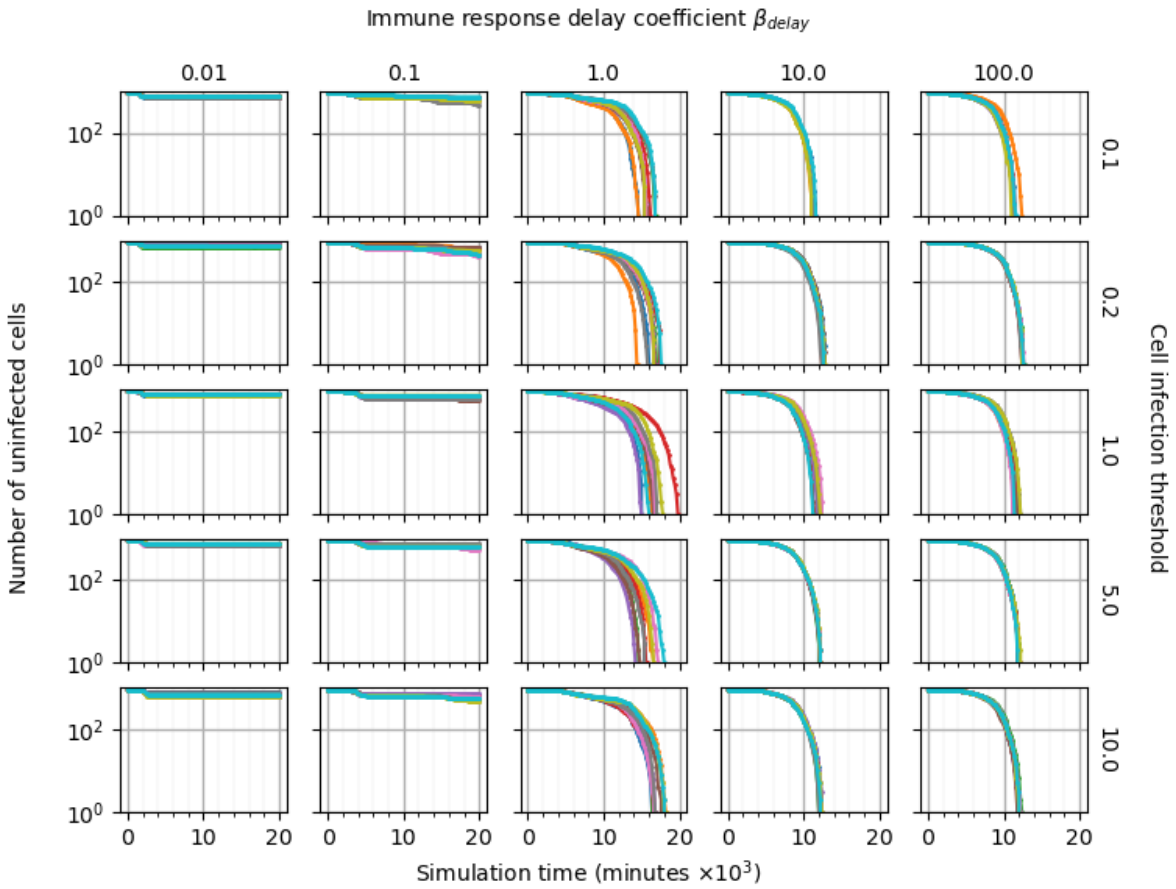


Fig S18. Pairwise parameter sweep of the immune response delay β_{delay} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) and infection threshold ($\times$ 0.1,$\times$ 0.2,$\times$ 1,$\times$ 5,$\times$ 10) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1).

The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

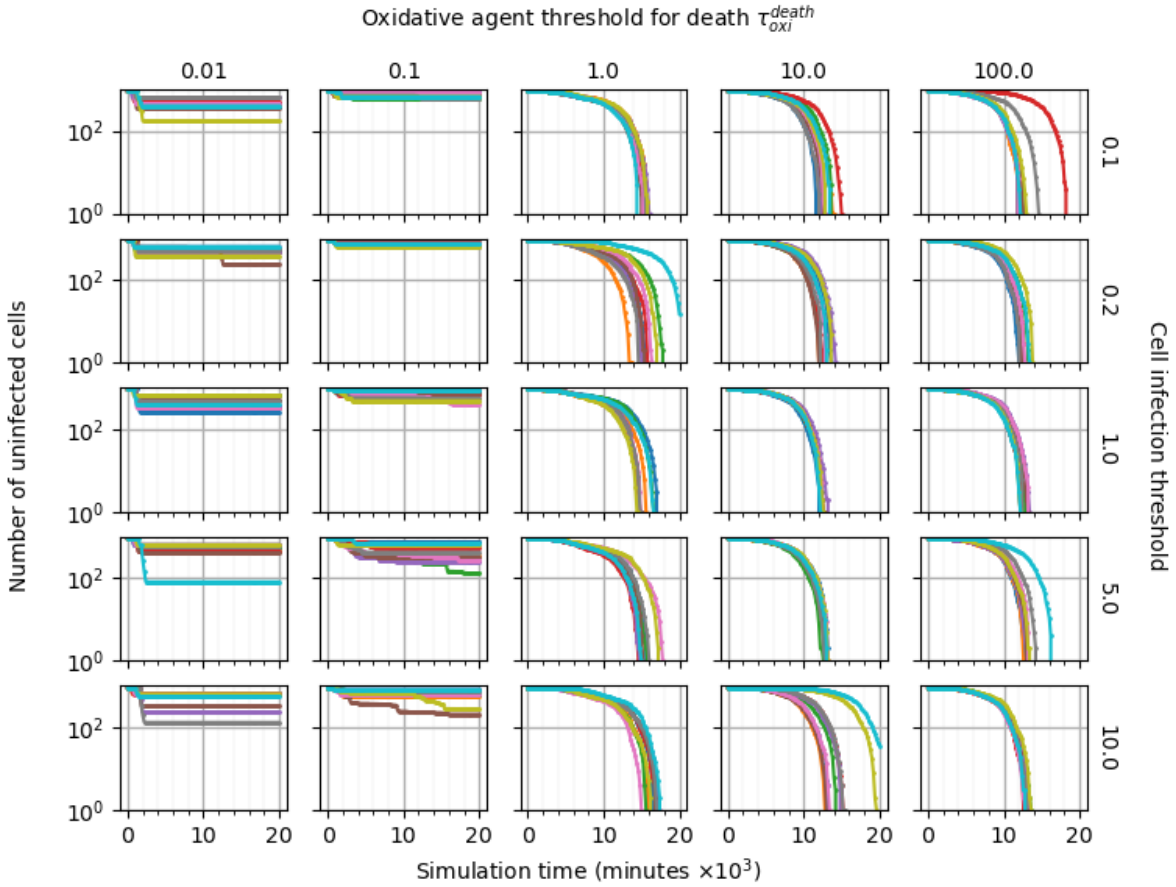


Fig S19. Pairwise parameter sweep of the oxidative agent threshold for death τ_{oxi}^{death} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) and the infection threshold ($\times$ 0.1,$\times$ 0.2,$\times$ 1,$\times$ 5,$\times$ 10) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1).

The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

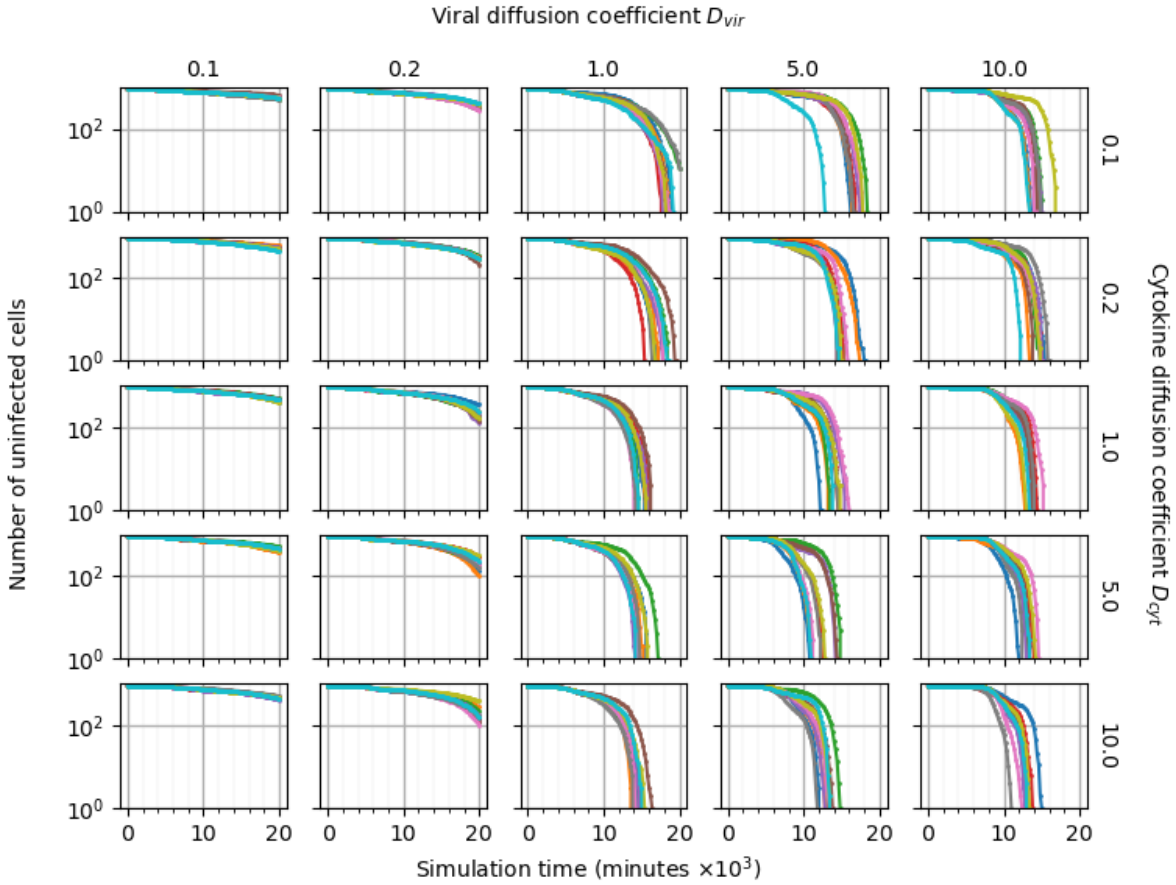


Fig S20. Pairwise parameter sweep of the viral diffusion coefficient D_{vir} and the cytokine diffusion coefficient D_{cyt} ($\times$ 0.1,$\times$ 0.2,$\times$ 1,$\times$ 5,$\times$ 10) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1).

The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

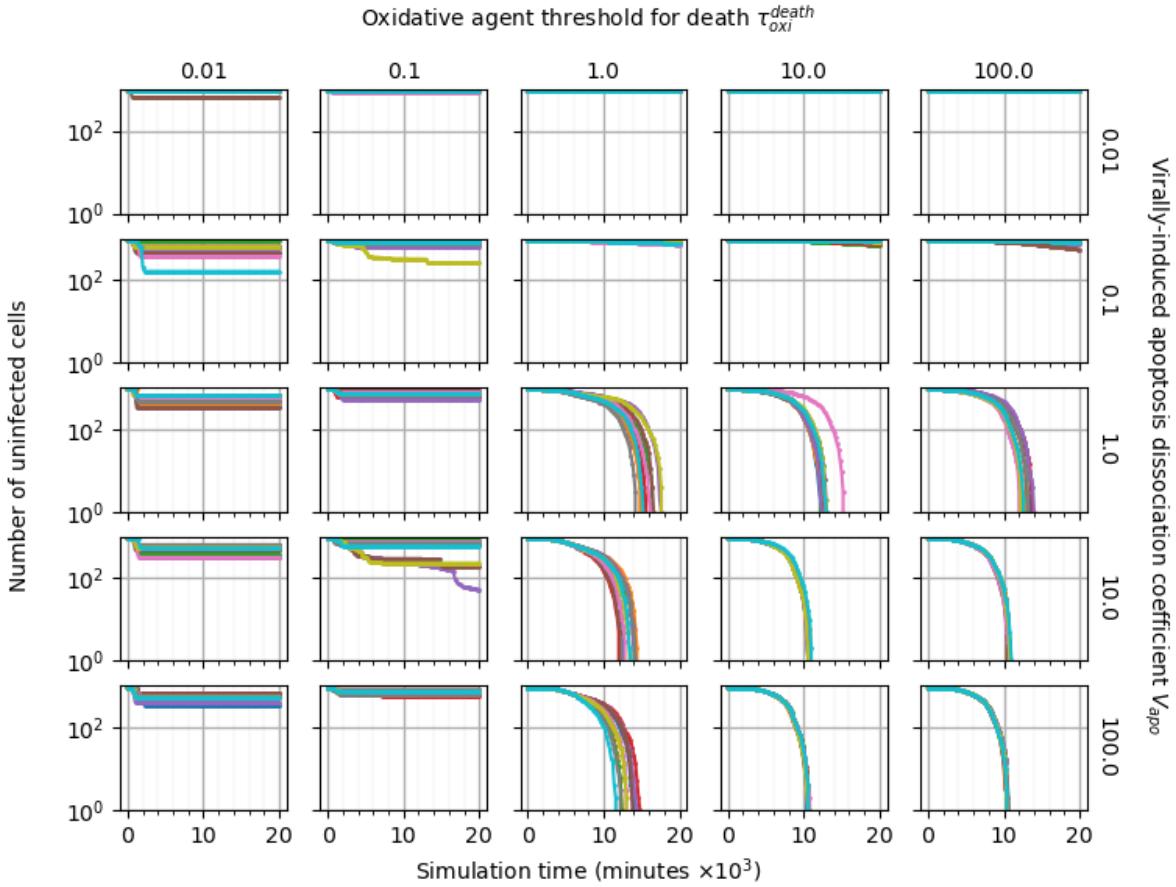


Fig S21. Pairwise parameter sweep of the oxidative agent threshold for death τ_{oxi}^{death} and the virally-induced apoptosis dissociation coefficient V_{apo} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1). The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

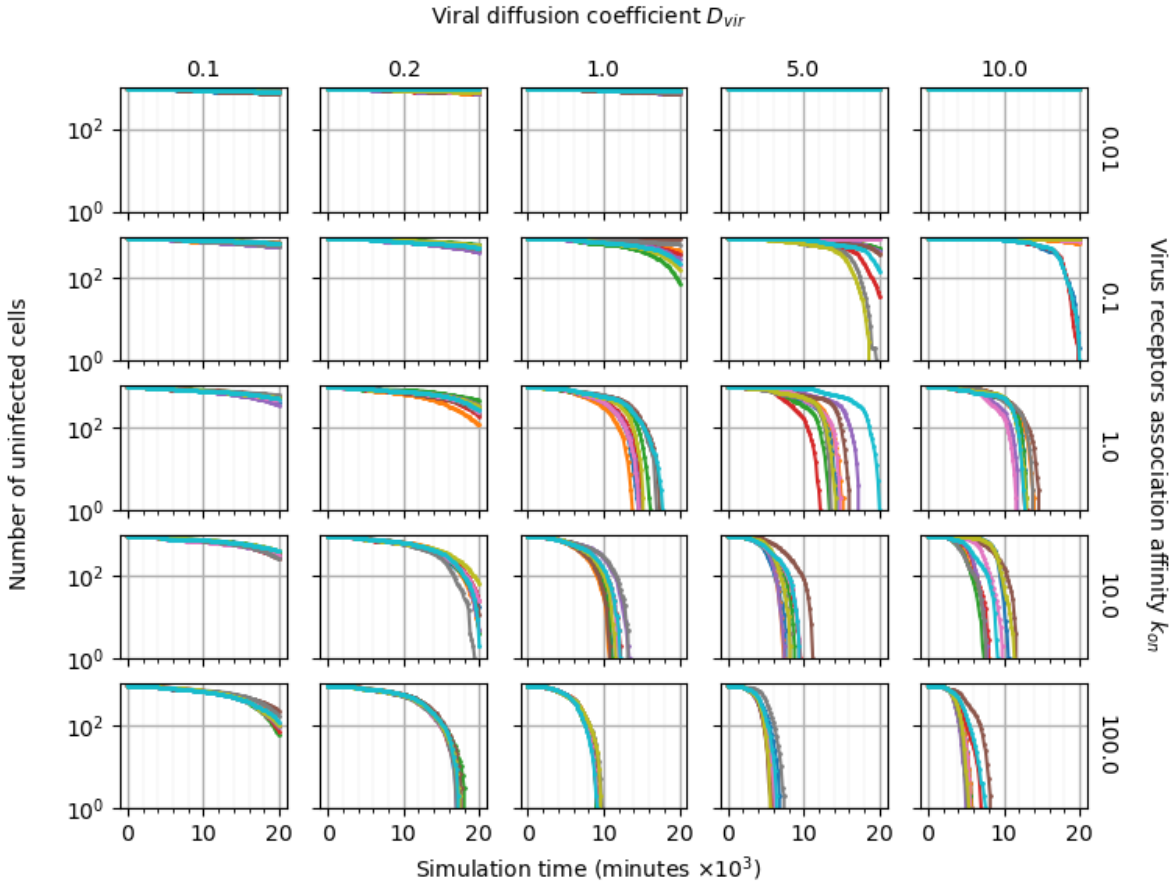


Fig S22. Pairwise parameter sweep of the viral diffusion coefficient D_{vir} ($\times$ 0.1,$\times$ 0.2,$\times$ 1,$\times$ 5,$\times$ 10) and the virus-receptor association affinity k_{on} ($\times$ 0.01,$\times$ 0.1,$\times$ 1,$\times$ 10,$\times$ 100) around their baseline values, with ten simulation replicas per parameter set (all other parameters have their baseline values as given in Table 1). The number of uninfected epithelial cells for each simulation replica for each parameter set, plotted on a logarithmic scale, *vs* time displayed in minutes.

A non-uniform epithelial sheet shows no significant effects in emergent dynamics of simulation outcomes

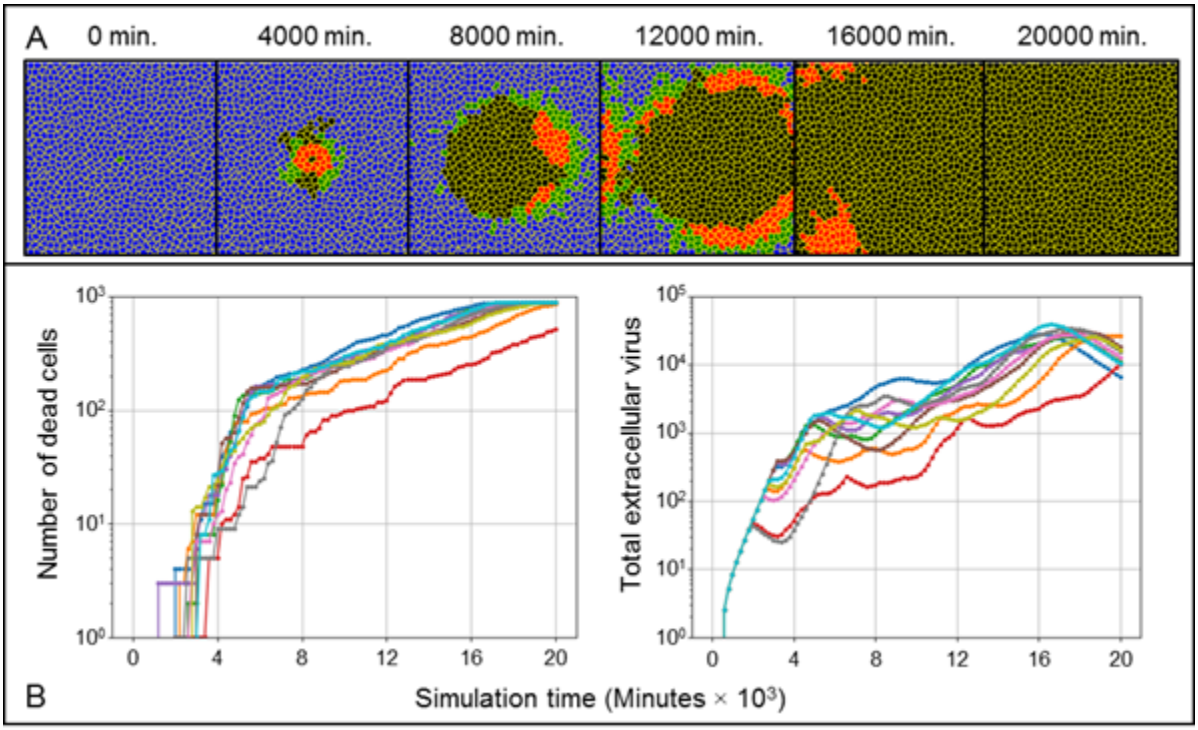


Fig S23. Simulation of viral infection using the baseline parameter set as in Fig 3 but with a non-uniform epithelial sheet.

(A) Widespread infection occurs with the same spatiotemporal features as in Fig 3A in a non-uniform epithelial sheet. (B) Ten simulation replicas with a non-uniform epithelial sheet showed no significant differences in transient metrics compared to simulations with a uniform epithelial sheet.

Integration of an explicit RNA synthesis model

The HCV model in [59] describes subgenomic replication in two compartments, namely the cytoplasm and vesicular membrane structures (VMS). Integration with the viral replication model described in *Quantitative model and implementation* requires the two modifications, one to the HCV model, and one to the viral replication model of the main framework, such that the viral genome taking part in genomic replication from (7) is

a proxy for the cytoplasmic plus-strand RNA molecules of the HCV model. Both modifications are described here.

According to the HCV model, in the cytoplasm,

$$\frac{dR_p^{cyt}}{dt} = k_2 T_c + k_{Pout} R_P - k_I R_{ibo} R_P^{cyt} - k_{Pin} R_P^{cyt} - \mu_P^{cyt} R_P^{cyt} + n_{HCV} r_u U, \quad \text{(S1)}$$

$$\frac{dT_c}{dt} = k_I R_{ibo} R_P^{cyt} - k_2 T_c - \mu_{Tc} T_c, \quad \text{(S2)}$$

$$\frac{dP^{cyt}}{dt} = k_2 T_c - k_c P^{cyt}, \quad \text{(S3)}$$

$$\frac{dE^{cyt}}{dt} = k_c P^{cyt} - k_{Ein} E^{cyt} - \mu_E^{cyt} E^{cyt}, \quad \text{(S4)}$$

where R_P^{cyt} is the number of plus-strand HCV RNA molecules in the cytoplasm, T_c is the number of translation complexes in the cytoplasm, P^{cyt} is the number of HCV polyprotein molecules in the cytoplasm, E^{cyt} is the number of enzyme NS5B and other viral proteins needed for RNA synthesis in the cytoplasm, R_{ibo} is the number of host cell ribosomes ($R_{ibo} = R_{ibo}^{tot} - T_c$ for fixed total available ribosomes R_{ibo}^{tot}), and n_{HCV} relates R_P^{cyt} to unitless R. Simulations of the HCV model were performed as in [59] by initializing R_P^{cyt} with an initial nonzero value in the initially infected cell. In the case of a spatial context, where cells are infected at various times according to progression of infection and subsequent internalization events, subgenomic replication within a particular cell occurs due to internalization of virus by the cell (6). As such, the final term of (S1) was added during integration such that internalized virus acts as a source for R_P^{cyt}.

Likewise, in the VMS,

$$\frac{dR_P}{dt} = -k_3 R_P E + k_{4p} R_{Ids} + k_{Pin} R_P^{cyt} - (k_{Pout} + \mu_P) R_P, \quad \text{(S5)}$$

$$\frac{dR_{ds}}{dt} = k_{4m} R_{Ip} + k_{4p} R_{Ids} - k_5 R_{ds} E - \mu_{ds} R_{ds}, \quad \text{(S6)}$$

$$\frac{dE}{dt} = k_{Ein} E^{cyt} + k_{4m} R_{Ip} + k_{4p} R_{Ids} - k_3 R_P E - k_5 R_{ds} E - \mu_E E, \quad \text{(S7)}$$

$$\frac{dR_{Ip}}{dt} = k_3 R_P E - k_{4m} R_{Ip} - \mu_{Ip} R_{Ip}, \quad \text{(S8)}$$

$$\frac{dR_{Ids}}{dt} = k_5 R_{ds} E - k_{4p} R_{Ids} - \mu_{Ids} R_{Ids}, \quad \text{(S9)}$$

where R_P is the number of plus-strand RNA in the VMS, R_{ds} is the number of dsRNA in the VMS, E is the number of HCV polymerase complexes in the VMS, R_{Ip} is the number of plus-strand RNA replicative intermediate complexes in the VMS and R_{Ids} is the number of plus-strand dsRNA replicative intermediate complexes in the VMS.

Having selected R_P^{cyt} and R as the shared biological object of the two models, mass action $R \rightarrow P$ of the viral replication model of the main framework requires modification. We assume that decay of R_P^{cyt} described in the HCV model leads to production of P through intermediate processes. The viral replication model of the main framework ((6)–(10)) then takes the modified form,

$$\frac{dU}{dt} = Uptake - r_u U, \quad \text{(S10)}$$

$$n_{HCV} R = R_P^{cyt}, \quad \text{(S11)}$$

$$\frac{dP}{dt} = r_t{}'R - r_p P, \quad \text{(S12)}$$

$$\frac{dA}{dt} = r_p P - Release, \quad \text{(S13)}$$

where $r_t{}'$ is the rate of production of P per unit of R associated with decay of R_P^{cyt}. Note that without the introduction of additional decay to the HCV model equations, the integrated form of the HCV model effectively acts the same as its original form within each cell, subject to the cellular and spatial aspects of internalization events. All parameters of the integrated HCV model in simulations shown in *Particularization to hepatitis C virus by integration of an explicit RNA synthesis model* were taken from [59] (Table S2).

Table S2. Parameter values of integrated HCV model.

Simulation parameters	Value
T_c formation rate k_1	4800 molecule^{-1} min. $^{-1}$
Nascent polyprotein cleavage rate k_2	6000 min.$^{-1}$
Viral polyprotein cleavage rate k_c	36 min.$^{-1}$
R_P^{cyt} transport rate into cytoplasm k_{Pin}	12 min.$^{-1}$
R_P transport rate into VMS k_{Pout}	12 min.$^{-1}$
E^{cyt} transport rate in VMS k_{Ein}	7.8×10-4 min.$^{-1}$
R_{Ip} formation rate k_3	1.2 molecule^{-1} min.$^{-1}$
R_P synthesis rate k_{4p}	102 min.$^{-1}$
R_{ds} synthesis rate k_{4m}	102 min.$^{-1}$
R_{Ids} formation rate k_5	240 min.$^{-1}$
R_P^{cyt} degradation rate μ_P^{cyt}	600 min.$^{-1}$
R_P degradation rate μ_P	4.2 min.$^{-1}$
R_{ds} degradation rate μ_{ds}	3.6 min.$^{-1}$

R_{Ip} degradation rate μ_{Ip}	2.4 min.$^{-1}$
R_{Ids} degradation rate μ_{Ids}	7.8 min.$^{-1}$
T_c degradation rate μ_{Tc}	0.9 min.$^{-1}$
E degradation rate μ_E	2.4 min.$^{-1}$
E^{cyt} degradation rate μ_E^{cyt}	3.6 min.$^{-1}$
Total number of available ribosomes R_{ibo}^{tot}	700
RNA conversion factor n_{HCV}	100 molecule R^{-1}
P production rate r_t'	2.5 min.$^{-1}$ R^{-1}

2074

2075 **Collaborative viral infection modeling environment**

2076 Given the immense amount of complexity associated with viral infection,
2077 supporting collaborative, independent, concurrent, and even conflicting, model
2078 development is critical to building an informative and predictive multiscale model of viral
2079 infection. As such, the simulation architecture developed for the CompuCell3D
2080 implementation, as demonstrated in *Model extensions*, supports development,
2081 deployment and distribution of add-on modules following the Python programming
2082 language design principles and practices of extensibility and modularity. This architecture
2083 exploits the architecture of CompuCell3D itself, specifically, that model implementation in
2084 CompuCell3D consists of designing a set of Python classes called “steppables”, each of
2085 which is imported into CompuCell3D and simulated (via exactly two simple lines of Python
2086 code per steppable, see *Deploying a model extension in CompuCell3D*). Each steppable
2087 typically implements a particular model, function, or feature (*e.g.*, viral internalization, data
2088 post-processing and exporting), and provides instructions to CompuCell3D about what to
2089 do during each simulation step along with the core simulation engine (*e.g.*, implementing

the Cellular Potts Model), as well as what to do before and after simulation, through a simple interface (*e.g.*, procedures to perform by a steppable during each simulation step are described in a function “step” in the steppable class definition). This approach is particularly well suited for supporting collaborative, independent, and concurrent model development because model specification of a particular simulation in CompuCell3D consists of selecting and loading a particular set of steppables, each of which can be specified in separate Python scripts and packaged in uniquely named directories, developed by collaborating or independent and otherwise disconnected research groups, and intended to model specific biological phenomena. Furthermore, specification of model implementations using Python classes also enables development of model extensions from existing modules (whether from the main framework or an add-on module) using basic Python class inheritance functionality (where class definitions can be constructed from other class definitions and subsequently modified, see *Extending a model in CompuCell3D*).

We envision a community of modelers much like the community of Python developers, which develops Python packages, called “modules”, that can be publicly distributed and imported into software using a simple, one-line Python command (*e.g.*, import MyModule). As such, we have built into the CompuCell3D implementation used in this work a location for storing a library of add-on modules, as well as supporting architecture to facilitate development and deployment of add-on modules. Furthermore, along with making the simulation framework publicly available online as described in *Downloading and running the simulation*, the online repository also hosts this library of add-on modules as part of the standard download package, which we continue to develop

and maintain, and for which we are currently developing standards (*e.g.*, standard documentation) and supporting tools (*e.g.*, documentation generators). We welcome usage by, and contributions from, all interested groups, and provide a basic overview of deploying and developing model extensions in the remaining discussion of this section.

Deploying a model extension in CompuCell3D. As in any typical CompuCell3D model specification, one script of the simulation files shown in Fig S13, Simulation/ViralInfectionVTM.py, imports all modules of the main framework and loads them into CompuCell3D for simulation. The directory “Simulation” contains all source code of the main framework, while an additional directory “Models” is dedicated to storing source code of the add-on module library. Each add-on module is an importable Python module stored in its own, uniquely named subdirectory (*e.g.*, Models/IUBIOCAddons). Code Snippet S1 shows a section of the contents of Simulation/ViralInfectionVTM.py for a simulation using the Simple Recovery model described in *An extensible framework architecture enables the inclusion of tissue recovery*.

```
from ViralInfectionVTMSteppables import ViralInternalizationSteppable
CompuCellSetup.register_steppable(steppable=ViralInternalizationSteppable(frequency=1))

from Models.RecoverySimple.RecoverySteppables import SimpleRecoverySteppable
CompuCellSetup.register_steppable(steppable=SimpleRecoverySteppable(frequency=1))
```

Code Snippet S1. Select import and load commands from Simulation/ViralInfectionVTM.py.

Lines 1 and 2 in Code Snippet S1 import and load the steppable "ViralInternalizationSteppable" that implements the internalization model described in *E1 - Viral internalization* from Simulation/ViralInfectionVTMSteppables.py. Lines 4 and 5 show that not much is different concerning loading and importing add-on modules. The steppable "SimpleRecoverySteppable" implements the Simple Recovery Model, and is defined in Models/RecoverySimple/RecoverySteppables.py. The only difference between importing and loading a module from the main framework or add-on module library is specifying the location of the Python script containing the steppable to be deployed in a simulation. This way, two model modules can define steppables in Python scripts of the same name without overwriting each other (*e.g.*, Models/GroupX/Steppables.py or Models/GroupY/Steppables.py). The only necessarily unique aspect of a particular model module is the name of its containing directory (*e.g.*, the directory Models/GroupX or Models/GroupY). This scheme isolates model-specific development to the directory in which the add-on model is defined, and modularizes the overall simulation framework into *shareable*, *interchangeable* model components. Furthermore, since development of add-on modules is isolated to a uniquely named directory, the framework promotes concurrent development and implementation of unrelated or even competing models.

Developing a Model Extension in CompuCell3D. Developing a model extension is as simple as the typical CompuCell3D model implementation procedure of developing steppables in Python using the CompuCell3D steppable class "SteppableBasePy" (see

Extending a model in CompuCell3D for discussion of Python class inheritance). Code Snippet S2 shows the application programming interface (API) and select code from the Python script Models/RecoverySimple/RecoverySteppables.py in the add-on module library that implements the Simple Recovery model described in *An extensible framework architecture enables the inclusion of tissue recovery*.

```
1   import sys
2   import os
3   # Import from simulation environment
4   sys.path.append(os.environ["ViralInfectionVTM"])
5   # Import parameter values from RecoveryInputs.py and other stuff
6   from .RecoveryInputs import *
7   from Simulation.ViralInfectionVTMModelInputs import s_to_mcs
8   import random
9   from cc3d.core.PySteppables import *
10
11  class SimpleRecoverySteppable(SteppableBasePy):
12      def __init__(self, frequency=1):
13          """
14          Initialize recovery steppable
15          """
16
17      def start(self):
18          """
19          Share self with framework
20          """
21
22      def step(self, mcs):
23          """
24          Perform recovery test in each dead cell
25          """
26
27      def recover_cell(self, _cell):
28          """
29          Implement recovery
30          """
31
32      def cell_recovers(self, _cell) -> bool:
33          """
34          Test for simple recovery in a cell
35          """
36          return random.random() < recovery_rate * s_to_mcs
```

Code Snippet S2. API for the steppable implementing the Simple Recovery model, derived from Models/RecoverySimple/RecoverySteppables.py.

The exact code of the implementation is shown for the steppable function “cell_recovers” (Lines 32-36).

Lines 1-4 in Code Snippet S2 add the directory containing both the main framework and add-on modules library directories using the environment variable "ViralInfectionVTM", which makes both available to any module loaded into CompuCell3D from Simulation/ViralInfectionVTM.py, whether directly or indirectly (*e.g.*, when Simulation/ViralInfectionVTM.py imports module "A" that imports module "B"). Line 6 imports parameter values of this module defined in Models/RecoverySimple/RecoveryInputs.py, while Line 7 imports a parameter value from the main framework for use in calculations. Line 8 imports the Python standard module "random" for generating random numbers, which, like the rest of the Python standard library and many others, is distributed with CompuCell3D. Line 9 imports everything from the CompuCell3D module "PySteppables", which contains the available Python classes for Python model implementation in CompuCell3D.

Line 11 in Code Snippet S2 begins the definition of the Simple Recovery model steppable, "SimpleRecoverySteppable". SimpleRecoverySteppable inherits from "SteppableBasePy", a steppable class defined in the PySteppables module. Its first three functions, "__init__", "start", and "step", are all functions of the CompuCell3D steppable interface. "__init__" defines the procedures to be performed for initializing the steppable. "start" defines the procedures to be performed after CompuCell3D has initialized but before simulation begins (in the case of this steppable, sharing a reference to itself with the rest of the framework). "step" defines the procedures to be performed during each simulation step (in this case of this steppable, evaluating recovery in each dead cell and

executing recovery when it occurs). The final two functions are specific to this steppable. The first, “recover_cell”, performs the necessary procedures associated with recovery on a cell when given one as an argument (*i.e.*, “_cell”). The second, “cell_recovers”, evaluates whether or not a particular cell is recovered. It should be noted that deployment of the SimpleRecoverySteppable class is not limited to usage directly in CompuCell3D as a simulated steppable. Rather, like the importing of a parameter value from the main framework in Line 7 of Code Snippet S2, the SimpleRecoverySteppable class can be imported into other modules for other purposes, like performing recovery of a dead cell but due to an alternative recovery criterion. The following section describes an example of such functionality.

Extending a model in CompuCell3D. Like any other Python class, steppables (and other code) defined in one model module can be extended by, or integrated into, other modules, such that the components of the overall simulation framework are not only interchangeable and shareable, but also extensible. In the previous section, Code Snippet S2 demonstrated the ability to import a parameter value (*i.e.*, “s_to_mcs”, Line 7) from the main framework for usage in an add-on module. The same can be done for integrating modules (whether from the main framework or add-on library) into other add-on modules, as well as for extending model modules using Python class inheritance. Python class inheritance enables the construction of classes from the definition of other classes, such that functionality and interfaces defined by one class can be employed, selectively adapted, and extended, by subsequent classes that inherit from it. Any inheriting class, called a “derived class”, can replace (*i.e.*, “overwrite”) the executed

code of a function in the definition of an inherited class, called a “base class”, if the derived class defines a function with the same name and arguments. All inherited functions that are not overwritten by a derived class are the same.

Simulation results of the Neighbor Recovery model in *An extensible framework architecture enables the inclusion of tissue recovery* demonstrate deployment using the framework capability of constructing add-on modules from other add-on modules. Computationally, nearly all aspects of the Recovery Model are the same as the Simple Recovery model (*e.g.*, test for recovery in every cell during each simulation step, implement recovery when it occurs). The only difference between the two models is the criterion by which recovery of a dead cell is evaluated, making the implementation of the Neighbor Recovery model a strong candidate for exploiting Python class inheritance, as demonstrated in Code Snippet S3.

```
import sys
import os
# Inherit from Simple Recovery model
sys.path.append(os.environ["ViralInfectionVTM"])
from Models.RecoverySimple.RecoverySteppables import SimpleRecoverySteppable

class NeighborRecoverySteppable(SimpleRecoverySteppable):
    def __init__(self, frequency=1):
        super().__init__(frequency)

    def cell_recovers(self, _cell) -> bool:
        """
        Test for neighbor-dependent recovery in a cell
        """
```

Code Snippet S3. API for the steppable implementing the Simple Recovery model, derived from Models/RecoveryNeighbor/RecoverySteppables.py.

As in Code Snippet S2, Lines 1-4 of Code Snippet S3 makes available the entire framework, while Line 5 imports the Simple Recovery model steppable definition for extension. Line 7 begins the definition of the steppable “NeighborRecoverySteppable” that implements the Neighbor Recovery model by inheriting from the class definition for the Simple Recovery model “SimpleRecoverySteppable”. Since all functionality of the Neighbor Recovery steppable is to be the same as the Simple Recovery steppable except for the recovery criterion, Lines 8-9 initialize the Neighbor Recovery steppable exactly the same as the Simple Recovery steppable. Lines 11-14 begin overwriting the definition of the recovery criterion from Simple Recovery according to the Neighbor Recovery model. Since no other functions of the Simple Recovery steppable are overwritten, they are then exactly the same for the Neighbor Recovery steppable. Furthermore, since the signature of the function that implements the recovery criterion (*i.e.*, “cell_recovers(self, _cell)”) has the exact same name and arguments in both steppables (*i.e.*, the function “cell_recovers” is overwritten by NeighborRecoverySteppable), they can be used in the exact same way by other modules (*e.g.*, a variable “my_recovery_steppable”, whether an instance of SimpleRecoverySteppable or NeighborRecoverySteppable, receives and returns the same type of information). The only difference in behavior between using one or the other is the potential outcome of asking either recovery steppable whether or not a particular dead cell recovers, about which the two models will often disagree.