

ENTERIC IMMUNITY SIMULATOR: A TOOL FOR *IN SILICO* STUDY OF GUT IMMUNOPATHOLOGIES

Katherine Wendelsdorf¹, Josep Bassaganya-Riera, Keith Bisset, Stephen Eubank, Raquel Hontecillas, and Madhav Marathe

Virginia Bioinformatics Institute, Virginia Polytechnic Institute and University, Washington Street, MC 0477 Blacksburg, Virginia 24061 USA

ABSTRACT. Clinical symptoms resulting from microbial infection of the gastrointestinal (GI) tract are often exacerbated by inflammation-induced immunopathogenesis. Identifying novel avenues for treating and preventing such pathologies is necessary and complicated by the complexity of interacting immune pathways in the gut, where inflammatory immune cells are regulated by anti-inflammatory cells. The ENteric Immunity Simulator (ENISI) is a simulator of the GI mucosa created for testing and generating hypothesis of host immune mechanisms in response to the presence of resident commensal bacteria and invading pathogens and the effect on host clinical symptoms. ENISI is an implementation of an agent-based model of individual mucosal immune cells each endowed with a program for movement and differentiation according to their cell-type, *i.e.* epithelial cells, dendritic cells, macrophages, conventional T cells, and natural T-regulatory cells. The internal programs specify movement among the gut lumen, lamina propria, and blood in response to an inflammation-inducing pathogen and tolerance-inducing commensal bacteria. The model focuses on the antagonistic relationship between inflammatory and regulatory (anti-inflammatory) factors whose constant presence characterize mucosal tissue sites.

Through user-manipulation of cell type-specific programs, ENISI allows one to observe the effects of phenotypic changes in individual cell-types, observed *in vitro*, at the tissue level. As such it is a translational research tool that allows one to : *i*) Test plausibility of *in vitro* observed behavior as explanations for observations *in vivo/in situ*, *ii*) Propose behaviors not yet tested *in vitro* that could be plausible explanations for observations at the tissue level. *iii*) Conduct low-cost, preliminary experiments of proposed interventions/ treatments. *iv*) Indicate useful areas of research through identification of missing data necessary to address a specific hypothesis.

An example of such application is presented in which we simulate dysentery resulting from *B. hyodysenteriae* infection and identify aspects of the host immune pathways that lead to continued inflammation-induced tissue damage even after pathogen elimination.

1. INTRODUCTION

1.1. Enteric disease and immune pathways. Enteric diseases are diseases of the gastrointestinal (GI) tract often caused by ingestion of microbes in food and water. **Inflammation** is the immune response by which immune cells eliminate foreign microbes. This response at the site of infection determines the likelihood of persistent infection as well as disease course and severity of clinical symptoms. For this reason, understanding mechanisms of inflammation in the gut and determinants of its strength is a focus of biomedical research that seeks to devise treatments and infection prevention strategies against gut pathogens such as *E.coli* and *H.pylori*.

During inflammation, immune cells detect foreign microbes and secrete cytotoxic factors to eliminate them. Though a necessary function, this occurs at the risk of elimination of bystander host tissue cells that can be the basis of various clinical symptoms including lesions of the epithelial lining and bloody diarrhea. As the GI tract is constantly exposed to foreign antigens, mostly innocuous, this inherent inflammatory

¹Corresponding author: Katherine Wendelsdorf, wkath83@vbi.vt.edu

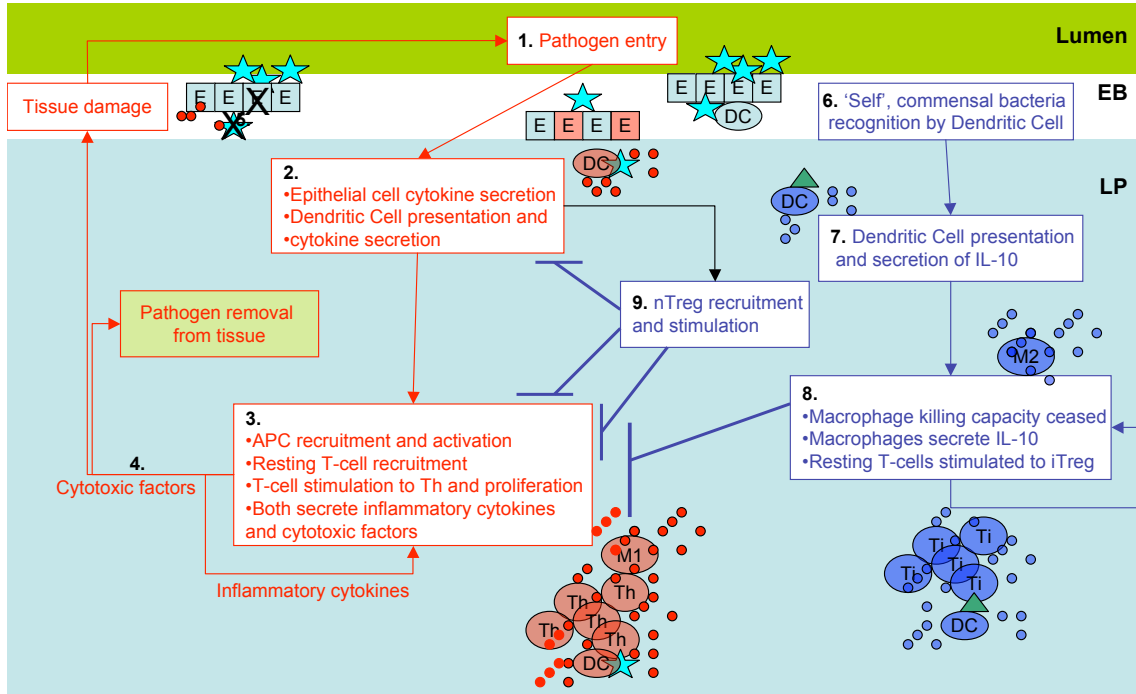


Figure 1: Illustration of inflammatory and regulatory pathways described in section 1.2.

response must be regulated so that the system does not remain in a constant state of tissue-damaging hyper-inflammation. This is carried out by the regulatory immune response triggered by factors such as host tissue damage or **commensal bacteria** of the **gut microflora**. In this parallel response, immune cells are rendered 'tolerogenic' or adopt a 'regulatory' **phenotype**, states in which the cell remains inactive toward a foreign microbe as well as inhibits the inflammatory response in other immune cells. The effect of these tolerogenic immune cells is an environment that requires more stringent conditions for induction of inflammation decreasing the frequency of its occurrence and strength in terms of cytotoxin-mediated tissue damage and pathogen elimination. Indeed, it is due to this **immune regulation** that the beneficial gut microflora, consisting of approximately 10^{14} bacteria, is able to survive and play a critical role in host digestive and metabolic processes. The current picture of the gut mucosa is one in which immune cells of the regulatory and inflammatory responses are in constant competition, with regulatory phenotypes generally predominating [9, 13]. The severity and efficacy of the inflammatory response is, therefore, a complex function of multiple, parallel, competitive processes.

These responses can be defined as immune pathways. Here we define an **immune pathway** as a sequence of events in which the first is the recognition of **antigen** by an immune cell, such that the occurrence of each event in the sequence is necessary for the occurrence of the next event in the sequence. We define an **immunopathological pathway** as an immune pathway that leads to damage of host tissue.

Our research interest is identification of immune mechanisms that determine specific health outcomes following enteric infection, such as full recovery or chronic inflammation. A **health outcome** is defined by a set of clinical symptoms resulting from the acute inflammatory response to pathogen. More specifically,

Definition of Terms

- **Anergic:** Lacking the normal immune response to a particular antigen or allergen
- **Antigen:** Any substance that stimulates an immune response in the body (especially the production of antibodies). These include toxins, signs of tissue damage, and microbial components.
- **Commensal bacteria:** Bacteria that colonize the mammalian gut and carry out processes beneficial to the host.
- **Cytokines:** Signaling molecules secreted by immune cells including interleukins (IL), interferon (IFN), and tumor necrosis factor (TNF) that carry out various functions depending on specific type.
- **Dendritic cells:** An immune cell that recognizes and internalizes foreign microbes. Its primary function is to then present components of the microbe on its surface to resting T cells that may or may not recognize the microbe “antigen”.
- **Epithelial barrier:** Thin monolayer of epithelial cells separating the lumen and lamina propria regions of the gut.
- **Gut microflora:** Population of microorganisms, mostly commensal, that live in the digestive tracts of animals.
- **Immune regulation:** Any process that modulates the frequency, rate, or extent of the inflammatory response.
- **Inflammation:** A localized protective reaction of immune cells in tissue to signs of stress or pathogen presence that is characterized by immune cell recruitment and sometimes tissue damage.
- **Lumen:** The inner open space of a tubular organ such as the stomach or intestine.
- **Macrophages:** An immune cell that recognizes and internalizes foreign microbes. Its primary function is secretion of factors that recruit other immune cells from the blood and cytotoxins that kill microbes and host tissue cells.
- **Phagocytosis:** Internalization of a microbe by an immature dendritic cell or resting macrophage. Generally followed by degradation of the microbe.
- **Phenotype:** The set of observable characteristics, (appearance, behavior, etc.) of an individual resulting from environment-dependent gene expression.
- **Self-antigen:** A component that, upon recognition by an immune cell, induces a tolerogenic or non-reactive state. This is often cellular debris in the system or a by-product of healthy metabolic processes.
- **T-helper cells:** Immune cells that, upon recognition of microbial components (antigen), secrete chemical signals that enhance the activity of surrounding immune cells.
- **Th1:** A T-helper cell phenotype associated with secretion of IL-12 and IFN- γ , which promote inflammatory phenotypes in macrophages.
- **Th17:** A T-helper cell phenotype characterized by secretion of IL-17 and associated with autoimmunity and inflammation-induced tissue damage.
- **T-regulatory cells:** Subpopulation of T cells that act to suppress activation of the immune system and thereby maintain immune system homeostasis and tolerance to self-antigens and commensal bacteria.

Figure 2: Definition of anatomical and immunological terms not given in the text

we seek to identify immune pathways, particularly immunopathological pathways, initiated by pathogen in the gut mucosa. We then seek to manipulate these pathways as forms of treatment.

For this purpose we present ENISI, a simulator of the inflammatory and regulatory immune pathways specifically initiated by microbe-immune cell interaction in the gut. ENISI is a tool for mucosal immunologists to test and generate hypothesized mechanisms for clinical enteric disease outcomes and propose interventions through experimental infection of an *in silico* gut. Simulation outcomes given different experimental conditions inform key mucosal immunity-related questions: *What is the net response to a pathogen given the complex interplay between both regulatory and inflammatory pathways?* and *Which aspects of mucosal immunity allow efficient elimination of a pathogen while keeping immunopathogenic side effects to a minimum?*. Observation of *in silico* behaviors that are not readily seen through *in vitro* and *in vivo* techniques, inform understanding of the system and help in generating novel treatment strategies that can then be tested in the laboratory.

1.2. Mucosal inflammatory and regulatory immune pathways. Here we describe the specific inflammatory immunopathological pathways and the regulatory immune pathways encoded in ENISI as shown in

Figure 1.

The mammalian gut mucosa can be divided into three sites: *i*) The *Lumen*, which has a direct connection to the external environment, houses gut microflora and ingested substances such as food and foreign microbes, *ii*) The *Lamina propria* (LP), tissue separated from the lumen by an epithelial monolayer that is occupied by resting immune cells, *iii*) The epithelial barrier (EB) that divides the lumen and LP.

The inflammatory immune pathways (red lines) are as follows: 1. A pathogenic microbe enters the lumen. 2. Intestinal epithelial cells recognize pathogen and secrete microbicides and various signalling chemicals (**cytokines**) [11]. Resting dendritic cells in the epithelium internalize the microbe and differentiate to an effector phenotype (eDC) that presents components of the pathogen (antigen) on its surface 3. Chemicals secreted by damaged epithelial cells recruit antigen presenting cells (APC), including **macrophages** and additional dendritic cells. Activated, presenting eDC recruits resting T cells to the infection site and secrete cytokines such as IL-12 and IFN- γ that induce T cell differentiation to pro-inflammatory **Th1** and **Th17** phenotypes upon antigen recognition. 4. T cells secrete cytokines that enhance secretion of inflammatory factors by surrounding T cells as well as induce macrophages to a M1 phenotype that secrete cytotoxic proteases and radicals that kill invading bacteria as well as cause indiscriminate tissue damage. 5. Tissue damage results in additional secretion of inflammatory cytokines by epithelial cells. This results in further immune cell recruitment along with openings in the epithelial barrier that can allow direct pathogen entry into the LP at which point recruited immune cells are activated to inflammatory phenotypes completing a positive, inflammatory feedback loop. Inflammation generally dissipates when pathogen is eliminated and direct immune cell stimulation ceases.

The alternate, regulatory pathway is composed of the following events: 6. Dendritic cells contact **self-antigen** or commensal bacteria strains. 7. Upon recognition, dendritic cells differentiate to a *regulatory* phenotype, often termed tolerogenic dendritic cells (tDC), that present components of the tolerance-inducing material and secrete the anti-inflammatory cytokine IL-10. 8. IL-10 induces differentiation of macrophages from an inflammatory M1 phenotype to a regulatory M2 phenotype that also secretes IL-10 and does not secrete cytotoxins [9]. In addition, presenting tDC stimulates resting T cells and induces their differentiation to an induced T regulatory cell (iTreg), also a secretor of IL-10. IL-10 secreted by iTreg and M2 reduces inflammatory cytokine secretion from surrounding immune cells, dampening the inflammatory loop. It also promotes further activation of additional M2 and iTreg and, thereby, closes a positive anti-inflammatory feedback loop.

9. Another regulatory pathway involves *natural* T-regulatory cells (nTreg). These are T cells in the LP that are pre-destined to be regulatory cells independent of the phenotype of the presenting dendritic cell (eDC or tDC). Like iTreg, nTreg secretes IL-10 promoting further M2 creation. In addition, nTreg bind eDC and inhibit their recruitment and stimulation of resting T cells to inflammatory phenotypes [22].

1.3. ENISI. ENISI encodes each immune pathway as an agent-based model representing each individual cell that participates in each component event. Each individual is represented by a finite state automaton that corresponds to an **epithelial cell**, a **commensal bacteria**, a **foreign bacteria**, a **macrophage**, a **dendritic cell**, a **'sampling' dendritic cell** (sDC), a conventional CD4+ T cell (**T cell**), or a natural T-regulatory cell (**nTreg**), where the possible states of the automata are listed in Table 1 and correspond to a cell's *phenotype* and, in some cases, its *Location* or its status as *dead*. A state transition represents differentiation to another phenotype or contact-dependent migration. Cell names written in bold text refer to the corresponding model automaton that represents the cell behavior in the system.

The automata may occupy one of three locations; lumen, LP, and blood. Each automaton is assigned to a specific tissue site upon entry into specific states. Each tissue site is divided into sublocations. The sublocation occupied by an automaton within its assigned tissue site changes randomly at discrete time

intervals. When two individuals are in the same sublocation they are considered in contact and may or may not interact depending on their *state*.

Immune response is modeled as updates in rules for movement and behavior of an individual cell upon interaction with another.

As a spatially explicit, agent-based model ENISI simulations take in to account spatial-temporal heterogeneity across individual cells and allow stochasticity in cell behavior in the form of probabilistic state transitions that are functions of tissue location, cell age, demographics of surrounding immune cells, and duration of contact with other immune cells and antigen. The method allows one to manipulate individual programs with a direct interpretation between changes in model rules and experimental modifications of cells and observe the net effect that arises from localized interactions.

Users may control experimental conditions by using a simple scripting language to specify any of the following features of the system: *i) Infection specifics*: dose and timing of pathogen entry; *ii) Experimental host phenotypes*: parameters governing interactions between specific phenotypes to represent changes in cytokine and cytokine-receptor expression; *iii) Host immunological set-point*: initial immune cell populations present at the time of infection; and *iv) Strain-specific functions of bacteria*: Specifications of interaction conditions and consequences for **commensal bacteria** and **foreign bacteria** that mimick those attributed to experimental strains, *i.e.* the probability that bacteria induce secretion of inflammatory factors by epithelial cells or IL-10 secretion by dendritic cells. The simulator efficiently simulates inflammation at a mucosal site occupied by 10^6 individual cells, a greater number than published to date, within 1 hour.

2. ENISI: THE FORMAL MODEL

The system of immune pathways is represented as a graph dynamical system (GDS). GDS is an abstract representation of a group of entities (cells, bacteria), modeled as nodes, and abstract interactions, modeled as edges. This representation provides a sound basis to develop simulations of diffusion processes in such systems. We present the basic elements of a GDS and then briefly discuss the representation of the system of enteric immune pathways within its framework.

2.1. Graph Dynamical Systems. A graph dynamical system β , is a 4-tuple $\beta = (G, S, F, R)$ consisting of a graph $G(V, E)$ whose node set V represents the collection of agents and whose edge set E represents the set of agent interactions. Let $n = |V|$ denote the number of nodes in G . Each node has a state, a value from a finite set S of all possible state values. Further, there is a family F of functions that describe state transitions. Specifically, each node $v_i \in V$, $1 \leq i \leq n$, has an associated local transition function $f_i \in F$ which determines the next state of the node. In general, f_i may depend on several parameters including the history of the current and previous states of v_i and those of its neighbors in G . For example, a local transition function for the state of v_i at time $(t + 1)$, $s_i(t + 1)$, may depend on parameters over a time window of duration T , $s_i(t + 1) = f_i(s_i(\theta), N[i](\theta), E[i](\theta)), t - T + 1 \leq \theta \leq t$, where $N[i](\theta)$ represents the states of the neighbors of v_i at time θ and $E[i](\theta)$ represents the states of the edges incident on v_i at time θ . Further, each GDS has an associated update scheme R that determines the order in which the local transition functions are computed and states of nodes are updated. For example, a synchronous (*i.e.*, parallel) update scheme is often utilized, where all f_i are executed in parallel, to make the best use of parallel processing. GDS with synchronous update are often called synchronous dynamical systems (SyDS). At any time t , the configuration $C(t)$ of a GDS is a vector $(s_1(t), s_2(t), \dots, s_n(t))$, where $s_i(t)$ represents the state of node v_i at time t . The time evolution of a GDS is represented by the sequence of successive configurations of the GDS.

2.2. System of immune pathways as a GDS. The system of immune pathways in ENISI is represented as a GDS composed of the individual cells that participate in each event within the pathway.

The blood and tissue sites, *Locations*, are divided into discrete patches, the *sublocations*, where a sublocation is defined as the maximum volume at which an individual can be assumed to be in contact with all other individuals in that sublocation. Individuals occupy and migrate between *Locations* according to a *schedule* assigned to each individual by their *state*. Cells occupy a different randomly chosen *sublocation* of the assigned *Location* at short time intervals representing random movement and resulting in a dynamic contact network.

G is the contact graph where nodes are individual cells and edges indicate co-localization in the same sublocation. The cells are represented as a set of finite state automata $\langle c_1 \dots c_n \rangle$ that are considered in contact when in the same *sublocation*. Hence, the model is spatially explicit and the notion of edges is implicit as we assume that, within a *sublocation*, all cells are in contact. The system can thus be considered a Co-evolving Graphical Discrete Dynamical System (CGDDS) where the nodes (cells, bacteria) and edges (contacts) of a graph (the cell-contact network) are updated at discrete time steps representing phenotype change and migration.

Each individual, c_i , occupies a state from the set S composed of states listed in Table 1. Each state $s_i \in S$ corresponds to either a cell's *phenotype*, the *Location* of the cell, or its status as *dead*.

The state transition function f_i depends on the current state of the individual s_i and at least one of the following: *i*) the amount of time i has occupied its current state and *ii*) the states of its contacts in the graph. For each state s there is a set of *Interactor* states I_s such that if a contact j of individual i is in a state $s_j \in I_{s_i}$, i will interact with j and possibly transition states. Whether transition actually occurs upon interaction is probabilistic. Upon transition, the state which i enters depends on s_j as specified by its transition function, $f_i \in F$. The function for the interaction probability may be single contact-dependent, calculated in a pairwise manner, or multicontact-dependent, a function of the configuration C_i of the subnetwork in the specific sublocation occupied by i .

The set of functions, F , is formalized in a set of state-charts, described below in Section 2.4 in greater detail. Upon a change of state, the individual may or may not be assigned to a new *Location*. This contact-dependent transition is an explicit representation of contact-dependent cell differentiation, such as the induction of $memT \rightarrow Th$ upon contact with eDC . T cell activation requires the binding of its surface receptor to antigen that is bound to the dendritic cell surface. Transitions may also implicitly represent differentiation induced by cytokines secreted by surrounding cells, as is the case for a $M1 \rightarrow M2$ transition, which is a function of the cytokine concentrations in the local environment.

Each *event* of the immune pathway is then defined by a specific state transition $s_i \rightarrow s_j$. For example, tissue damage occurs when one cell transitions from the EC state to the $pEcell$ state. A specific *health outcome* is a stable configuration of the system, C_s , following contact between one of the bacteria automata and one of the immune cell automata. For example, immunological tolerance, the outcome in which there is no immune response to the presence of a microbe and the microbe persists in the gut, is defined by a configuration in which no **commensal bacteria** or **foreign bacteria** occupy the B_dead or Bf_dead states and no immune cells occupy a state that corresponds to an inflammatory phenotype; eDC , Th , or $M1$.

2.3. Approximations to the biological model. The contact-dependency of state transitions in the graphical framework as well as the need for computational efficiency require a number of approximations to the biological model. The GDS model stipulates that for a state change in one individual to be induced by another individual, the individuals must be co-located. Hence, the model cannot explicitly include induction of state-transitions across location barriers as may occur when cytokines secreted by a cell in the LP influence migration of cells in the blood. To reduce complexity, individuals are not newly created or removed from the contact network G following the start of the simulation. Rather biological processes that require these functions are either not included or represented in an indirect fashion. For example, the model does not include the constitutive flow of resting immune cells in and out of tissue. Nor do we represent bacterial

replication. The latter approximation can be interpreted as the assumption that each bacterium in contact with the epithelial barrier will be rapidly removed by immune cells before it is able to replicate.

Given these model approximations, we describe how the following biological functions are represented in the ENISI implementation: *i*) bacterial death, *ii*) lymphocyte recruitment, *iii*) T cell proliferation, and *iv*) T cell death. Descriptions of each state referred to in italics is given in Table 1.

i) Bacterial death: As scaling is a constraint, only those bacteria in contact with the epithelial border are represented. Given these simplifications, bacteria in the lumen does not explicitly 'die', but rather it is assumed that when one commensal bacterium is removed by phagocytosis, another bacterium, immediately takes its place due to the high concentration in the outer lumen. Therefore, an individual **commensal bacteria** in the lumen, *i.e.* the *B.lumen* state, is not explicitly removed when it interacts with a **dendritic cell** and phagocytosis occurs. Only the dendritic cell changes states from *iDC.lumen* \rightarrow *eDCL*. In addition, to conserve the number of individuals in the system, we allow individuals in the *B.LP* state that are phagocytosed to transition to the *B.lumen* state, replenishing the lumen population.

ii) Cross-barrier recruitment: A key function of pro-inflammatory epithelial cells, M1, and eDC is secretion of MCP-1, a factor that recruits resting T-cells as well as resting DC and macrophage precursors, called monocytes, from the blood to the inflamed LP tissue. Recruitment, therefore, requires that one individual occupying the *MASource*, *DCSource* or *memTSource* state undergo a state transition to *M0*, *iDC*, or *memT* in the LP, triggered by the transition of another individual from *M0* \rightarrow *M1* or *iDC* \rightarrow *eDC*. The model stipulates that any state transition dependent on the state of another individual be contact-dependent and defined as an explicit interaction. Hence the function of recruitment of monocyte and memory T cells in the blood by M1 and eDC in the LP is represented as follows: Individuals in the *M1* or *eDC* state briefly migrate to a sublocation in the *Blood* location where they contact individuals in the *MASource*, *DCSource*, or *memTSource* states. This induces the contacted monocyte or memory T cell to transition to an *M0*, *iDC*, or *memT* in the LP. Upon this transition the individuals are assigned a new schedule to the LP. The number of memory T cells and monocytes recruited by each M1 and eDC are determined by the parameters ϵ_r and ϵ_t (Table 2), respectively, the average number of monocytes or memory T cells a M1 or eDC contacts when it enters the *Blood*. This is set by the number of individuals assigned to the *MASource*, *DCSource*, and *memTSource* states at the start of simulation. These were initially set to give ϵ_r and ϵ_t the arbitrary value of 10.

iii) T cell proliferation: The current software requires that all individuals in the entire simulation be defined initially by a state and a location. Hence, all nascent T cells that may spawn from a proliferating T cell are anticipated and predefined with the states *ThSource* or *iTregSource* and are assigned to the locations *ThSource* and *iTregSource*. When an individual is in one of the proliferating states, *ThProlif* or *iTregProlif*, it may then migrate to these locations and induce the source cells to its corresponding phenotype. For example, when an individual enters the *ThProlif* state it briefly migrates to a sublocation in *ThSource* where it randomly contacts individuals in the *ThSource* state. Contacted individuals then transition from *ThSource* \rightarrow *Th* and are assigned to the LP. Proliferation by individuals in the *iTregProlif* state is represented in the same manner. Hence, the average number of daughter cells from one proliferating T cell p_T (Table 2), is determined by the average number of *ThSource* a *ThProlif* contacts when it visits the *ThSource* location and is set by the initial number of individuals in the *ThSource* or *iTregSource* states.

iv) T cell death: In the true mucosa, when T cells are no longer active a fraction revert to a resting memory T cell state and the rest undergo programmed cell death. To conserve the number of represented individuals

in the model, when individual T cells undergo programmed cell death they do not enter a *dead* state. Rather, they replenish the *ThSource* and *iTregSource* population pools.

Table 1: Model States

Symbol	State	Initial population size
Phenotypes		
<i>memT</i>	CD4+ Memory T cell	$1 \cdot 10^3$
<i>Th</i>	Active CD4+ T helper cell	0
<i>iTreg</i>	Induced T regulatory cell	0
<i>nTreg</i>	Active natural T regulatory cell	0
<i>mem.nTreg</i>	Resting natural T regulatory cell	0
<i>iDCLumen</i>	Immature sDC in the superficial LP with access to the Lumen	1000
<i>iDCLP</i>	Immature dendritic cell in the LP	1000
<i>eDC</i>	Effector dendritic cell in the LP	0
<i>tDC</i>	Tolerogenic dendritic cell in the LP	0
<i>eDCL</i>	Effector sDC in the lumen	0
<i>tDCL</i>	Tolerogenic sDC dendritic cell in the lumen	0
<i>DCAnergic</i>	Anergic dendritic cell	0
<i>M0</i>	Undifferentiated macrophage	$1 \cdot 10^3$
<i>M1</i>	Activated inflammatory macrophage	0
<i>M2</i>	Activated regulatory macrophage	0
<i>EC</i>	Healthy epithelial cell	$10^4[16]$
<i>pEcell</i>	Damaged or pro-inflammatory epithelial cell	0
<i>MASource</i>	monocytes: MA precursor	10^5
<i>DCSource</i>	monocytes:DC precursor	10^5
<i>memTSource</i>	memory T cell in blood	10^4
<i>ThSource</i>	Potential child cell from a proliferating Th	$5 \cdot 10^5$
<i>iTregSource</i>	Potential child cell from a proliferating iTreg	$5 \cdot 10^5$
Locations		
<i>B.lumen</i>	Commensal bacterium in the lumen	1000
<i>Bf.lumen</i>	Foreign bacterium in the lumen	30
<i>B.LP</i>	Commensal bacterium in the LP	0
<i>Bf.LP</i>	Foreign bacterium in the LP	0
Death		
<i>Edead</i>	Killed epithelial cell	0
<i>B.dead</i>	Killed commensal bacterium	0
<i>Bf.dead</i>	Killed foreign bacterium	0

2.4. State transition functions. Here we describe the state transition functions for each individual. An individual may be one of seven automata corresponding to an **epithelial cell**, a **commensal bacteria**, a **foreign bacteria**, a **macrophage**, a **dendritic cell**, a **'sampling' dendritic cell** (sDC), a conventional CD4+ T cell (**T cell**), or a natural T-regulatory cell (**nTreg**) depending on the state it is initially assigned at the start of the simulation. State transition functions for each automaton are given in a state-chart formalism [8]. In the figures that follow, ovals represent states of the automaton. Solid arrows represent time-dependent transitions labelled with the time in one state before transitioning to another. The dashed arrows represent single contact-dependent transitions, labelled with the set of *Interactor* states necessary to induce state transition and, in parenthesis, the probability of transition upon interaction. The default probability is 1. Dotted arrows represent multicontact-dependent state transitions and are labeled with the function that determines the probability of interaction. Unlabelled solid arrows indicate that transition automatically occurs at the

next update. States outlined in pink indicate the initial state that determines which automaton a cell will be. States are depicted in boxes labelled with blue text that indicate the specific *Location* to which individuals in the state are assigned. Parameters are listed in Table 2 and are referenced in the following descriptions.

Table 2: Parameter values

Symbol	Parameter	Default Value
Birth/death		
μ_E	Turnover time of epithelial cells	12hrs [16]
μ_T	Time a T cell remains active	5days [19]
μ_{M0}	Time a macrophage remains active	75 days
μ_d	Time a dendritic cell remains active	1 day [17]
μ_{dL}	Time a sDC remains active	1 day [17]
μ_{ce}	Probability that <i>pEcell</i> is killed by inflammatory factors	
p_T	Average number of daughter cells produced by a proliferating T cell	500 [23]
p_{Tr}	Average number of daughter cells produced by a proliferating nTreg	0
Migration		
ϵ_r	Average number of monocytes recruited by a single <i>eDC</i> , <i>M1</i> , or <i>pEcell</i>	10
ϵ_t	Average number of resting T cells recruited by a single <i>eDC</i> , <i>M1</i> , or <i>pEcell</i>	10
β_p	Probability that bacteria will enter the lumen upon contact with a <i>pEcell</i>	1
Contact/interactions		
α_T	Probability of memory T cell stimulation	1
α_{Tr}	Probability of memory nTreg stimulation	1
v_T	fraction of active T cells that become memory T cells	0.1 [15]
v_{12}	Probability that <i>M1</i> switches to <i>M2</i>	
v_{21}	Probability that <i>M2</i> switches to <i>M1</i>	
a_1	co-efficient of v_{12} for activators	1
i_1	co-efficient of v_{12} for inhibitors	1
y_1	exponent of v_{12}	4
a_2	co-efficient of v_{21} for activators	1
i_2	co-efficient of v_{21} for inhibitors	4
y_2	exponent of v_{21}	
v_{BM}	probability that microfloral bacteria induces inflammatory phenotype in macrophages	0
v_{BD}	probability that microfloral bacteria induces inflammatory phenotype in LP dendritic cells	0
v_{Bs}	probability that microfloral bacteria induces inflammatory phenotype in 'sampling' dendritic cells	0
v_{EC}	Probability that <i>EC</i> transitions to <i>pEcell</i> upon contact with inflammatory factors	0.05
v_{EB}	Probability that <i>EC</i> is damaged by microbial toxins	1

Epithelial Cells (Figure 3) Epithelial cells are assigned simultaneously to the *LP* and *Lumen* locations representing its status as a barrier. These cells are static and do not change sublocations. Initially in the healthy *EC* state, the cell transitions to a damaged, pro-inflammatory state, *pEcell*, with the probability of v_{EC} upon contact with inflammatory immune cells, individuals in states *Th*, *M1*, or *eDC*. This represents secretion of cytokines, such as IL-6 and IL-17, that induce the NF- κ B pathway in epithelial cells that leads to secretion of various pro-inflammatory mediators [2, 11] as well as cytotoxins secreted by *M1* that damage epithelial cells. This transition also occurs in the presence of foreign, pathogenic bacteria that can induce epithelial damage with a probability of v_{EB} , which will be specific to the bacterial strain that **foreign bacteria** represents.

Upon continued exposure to *Th* and *M1*, the damaged epithelial cell may transition to *Edead* state representing death of the epithelial cell that can occur with continued exposure to toxic factors secreted by

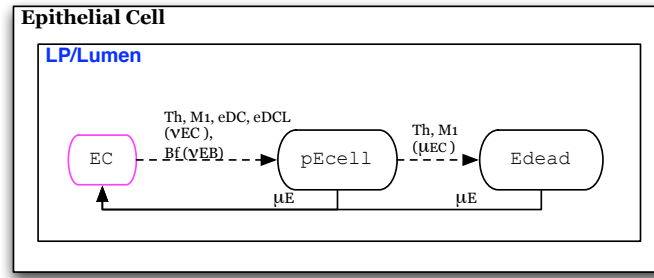


Figure 3: Automata for epithelial cells; individuals initially assigned the EC phenotype

these inflammatory cells. In the *Edead* state epithelial cells no longer secrete pro-inflammatory cytokines. *pEcell* and *Edead* transition to a healthy state after a time period of μ_E representing constitutive turnover that allows replacement of dead and inflammatory epithelial cells with healthy ones [16].

Commensal Bacteria (Figure 4) The model includes individual bacterium of the microbiota that are

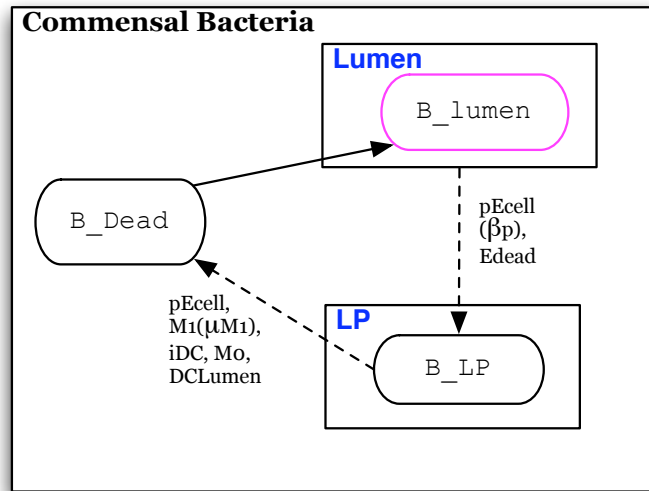


Figure 4: Automata for commensal bacteria; individuals initially assigned the B_lumen phenotype

located in the lumen (*B_lumen*). In the true system this is a diverse population of different strains. Here we endow each individual with behaviors assumed to be shared by most commensal strains. From the lumen, individuals may migrate to the *LP* and transition to *B_LP* when in contact with a *pEcell* with a probability of β_p . This represents cytoskeletal changes induced by inflammatory factors that may render epithelial cells more permeable [2]. This migration may also occur when *B_lumen* is in contact with a dead epithelial cell (*Edead*) representing contact with an opening in the epithelial barrier. Individual bacterium in either location may be killed by microbicidal factors secreted by neighbors in the *pEcell* and *M1* states as well

as phagocytosis by resting macrophage ($M0$) and dendritic cells ($iDCLP$) upon contact. Shortly after death, entry in to the B_Dead state, the individual re-enters the B_lumen state to replenish the pool of commensal bacteria in the lumen which the model assumes to be unlimited.

Foreign Bacteria (Figure 5) Foreign bacteria is assigned to the *Lumen* initially (Bf_lumen) representing

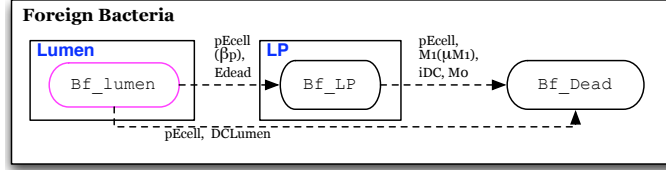


Figure 5: Automata for foreign bacteria (pathogen); individuals initially assigned the Bf_lumen phenotype

entry of a pathogen. Here it may contact $iDCLumen$ at which point it transitions to Bf_death representing internalization by an immature dendritic cell. Alternatively, if it contacts a damaged or dead epithelial cell it may migrate to the LP (Bf_LP) representing a break in the epithelial barrier. Like commensal bacteria, foreign bacteria in the LP may be eliminated upon contact with microbicide-secreting epithelial cells, and cytotoxin secreting Th and $M1$. Alternatively it may transition to Bf_death upon contact with and assumed internalization by $iDCLP$ or $M0$.

Macrophages (Figure 6): Macrophages occupy the LP where they move randomly and are initially in a resting $M0$ state. Upon contact with neighbors in the B_LP state, $M0$ transitions to $M1$ with a probability of v_{BM} , representing the percent of microbiota that is recognized as foreign and $M2$ state with the probability of $1 - v_{BM}$. In the healthy model $v_{BM} = 0$ representing a microflora composed completely of tolerance-inducing commensal bacteria and a macrophage population that recognizes it as such ?? . It is well established that macrophages may switch phenotypes as the cytokine ratio changes in the environment [9]. $M2$ may switch to $M1$ in the presence of inflammatory cytokines such as $IFN\gamma$ and $TNF\alpha$. Conversely, $M1$ may switch to $M2$ when in the presence of the regulatory cytokine IL-10 . In the model, these switches occur with a probability proportional to the number of inflammatory cytokine-secreting cells, N , and IL-10 secreting cells, R , the shared sublocation. $M1$ switch to $M2$ with the probability of v_{12} (Equation 1), where R is the total number of contacts that are of a regulatory phenotype: tDC , $M2$, $iTreg$, $nTreg$ and N is the number of contacts that are of a inflammatory phenotype: eDC , $M1$, Th , $pEcell$, a_1 and i_1 are constants that determine the threshold for the $R : N$ ratio at which v_{12} increases, and y_1 is a constant that determines the rate at which v_{12} increases with the $R : N$ ratio. $M2$ switches to $M1$ with the probability v_{21} (Equation 2), where a_2 , i_2 , and y_2 are constants that may differ from a_1 , i_1 , and y_1 .

$$(1) \quad p(M1 \rightarrow M2) = v_{12} = \left(\frac{a_1 R}{a_1 R + i_1 N} \right)^{y_1}$$

$$(2) \quad p(M2 \rightarrow M1) = v_{21} = \left(\frac{a_2 N}{i_2 R + a_2 N} \right)^{y_2}$$

In the case of no cytokine stimulus, *i.e.* no contact with individuals in either inflammatory or regulatory states, an activated macrophage will revert back to a resting state after a specified period of time, μ_{M0} . To

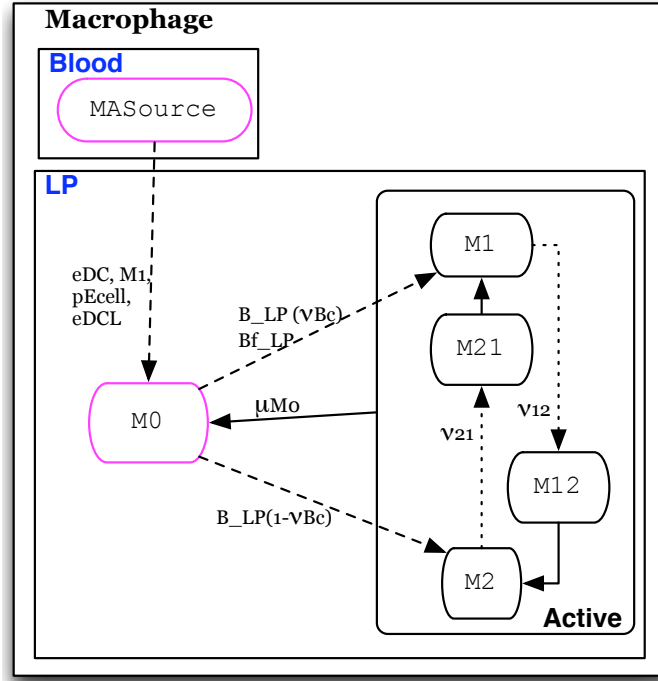


Figure 6: Automata for macrophages; individuals initially assigned the M0 or MASource phenotype

represent monocyte recruitment, individuals initially in the *MASource* state and assigned to the *Blood* transition to *M0* when in contact with individuals in the *pEcell*, *M1*, or *eDC* states. Upon entry in to the *M0* state the individual is assigned to the *LP*.

Dendritic Cells (Figure 7): Dendritic cells are initially in a resting, immature state in the *LP* (*iDCLP*). These may be referred to as *LP* dendritic cells and are distinct from the 'sampling' dendritic cells associated with the *EB*. The state transition model is similar to that of macrophages. Upon contact with bacteria an individual in the *iDCLP* state transitions to an inflammatory effector DC (*eDC*) with the probability of v_{BD} and remains in the *LP*. With the probability of $1 - v_{BD}$ an individual in the *iDCLP* state will transition to a tolerogenic DC state (*tDC*). The individual remains in one of these active states for a time period μ_d before migrating or dying [17]. The model represents this removal by reversion to the *iDCLP* state, recycling the individual to replenish the immature dendritic cell pool from an assumed unlimited monocyte pool. Upon contact with neighbors in the active *nTreg* state, *eDC* is rendered **anergic** at a probability of k_T , transitioning to the *eDCanergic* state, where it is incapable of stimulating T cells [9]. *DCSource* are assigned to the blood and transition to *iDC* when in contact with individuals in the *pEcell*, *M1*, or *eDC* states. At this point the *iDC* is assigned to the *LP*.

'Sampling' Dendritic Cells (Figure 8): *sDC* represents a dendritic cell that resides in the superficial *LP*, in association with the *EB*, where extensions of its cellular body breach the *EB* to contact and 'sample' microbes in the lumen. These dendritic cells are believed to be a distinct phenotype from *LP* dendritic cells

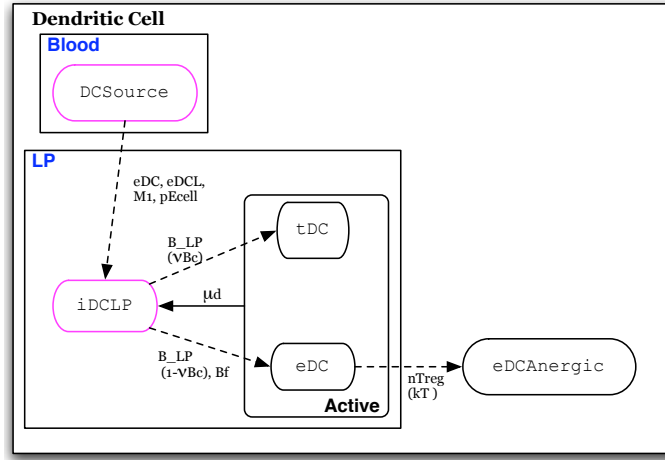


Figure 7: Automata for dendritic cells; individuals initially assigned the iDCLP or DCSource phenotype

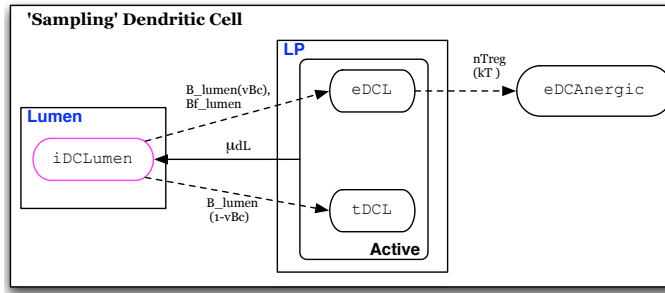


Figure 8: Automata for dendritic cells; individuals initially assigned the iDCLumen phenotype

described above [11, 21]. This is represented by assigning resting sDC to the lumen represented by the *iDCLumen* state. It can be seen that the transition path from *iDCLumen* is similar to LP dendritic cells in that *tDCL* and *eDCL* are assigned to the LP and carry out the same functions and behaviors as *eDC* and *tDC*. However, the parameters governing lifespan and predisposition towards effector or tolerogenic phenotypes following antigen recognition may differ [21]. As a simplification, the default model assumes the same lifespan and probabilities of differentiation upon antigen recognition as dendritic cells deeper in the LP. These assumptions may be revisited at a later date or modified by the user.

Conventional CD4+ T cells (Figure 9): To conserve the number of individuals, the model only represents T cells that specifically recognize and are stimulated by products of the commensal microflora or the pathogen represented by **foreign bacteria**. A T cell is initially in a resting, memory state in the LP (*memT*) or blood (*memTSource*). An individual in the *memT* state will transition to an active inflammatory Th, a Th1 or Th17, when in contact with neighbors in the *eDC* state. Alternatively, it may transition to an active

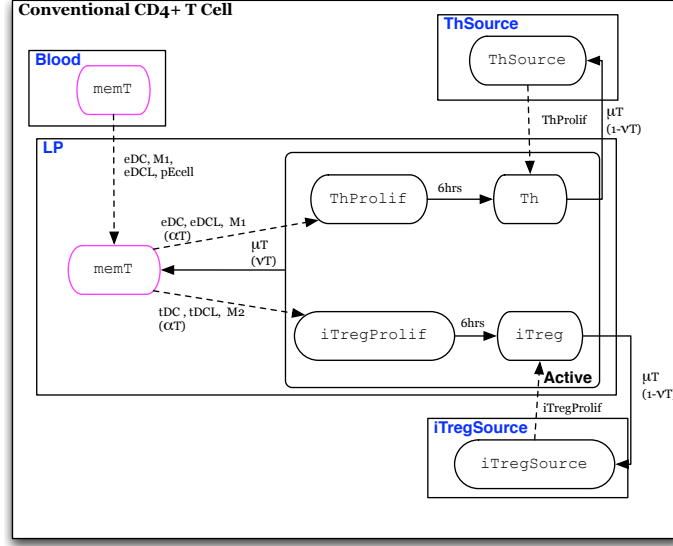


Figure 9: Automata for conventional CD4+ T cells; individuals initially assigned the memT or memTSource phenotype

iTreg when in contact with neighbors in the tDC state. This rule represents the fact that T cell phenotype depends on the cytokines secreted by the antigen-presenting cell (APC) during antigen recognition by the T cell receptor [18]. In either case, activation occurs with a probability of α_T . The value of α_T represents the probability that the antigen presented by a specific APC is recognized by the receptor of the contacted T cell. Upon stimulation the T cell enters a proliferation state, $ThProlif$ or $iTregProlif$, for approximately 6 hours [26]. In this state the cell can induce transition of source cells to a Th or $iTreg$ state giving rise to a new population of active T cells of its same phenotype. The value p_T is the average number of children T cells produced by one proliferating T cell. From this state, the individual T cell transitions to a non-proliferating active state, Th or $iTreg$. T cells remain in an active state for a period μ_T after which a fraction, ν_T , become memory T cells and may be re-stimulated by APC, *i.e.* individuals in the $M1$, $M2$, tDC , or eDC states. The rest undergo programmed cell death represented by reversion to its associated source state, $ThSource$ or $iTregSource$.

Natural T-regulatory Cells (nTreg) (Figure 10): Natural T-regulatory cells follow a very similar path to conventional T cells. The primary difference is that an activated nTreg has only one, regulatory phenotype regardless of the state of the antigen presenting macrophage or dendritic cell. nTreg proliferate upon stimulation giving rise to p_{Tr} daughter cells. Whether memory nTreg proliferate upon stimulation *in vivo* is still not clear [24]. Hence, the value of p_{Tr} may be assigned according to the assumption one wishes to make regarding nTreg proliferation capacity following antigen recognition. Upon contact with neighbors in the $M1$, $M2$, eDC , or tDC state an individual in the mem_nTreg state transitions to the active $nTreg$ state with a probability of α_{Tr} , potentially different from that of conventional CD4+ T cells. However, in the default model $\alpha_{Tr} = \alpha_T$.

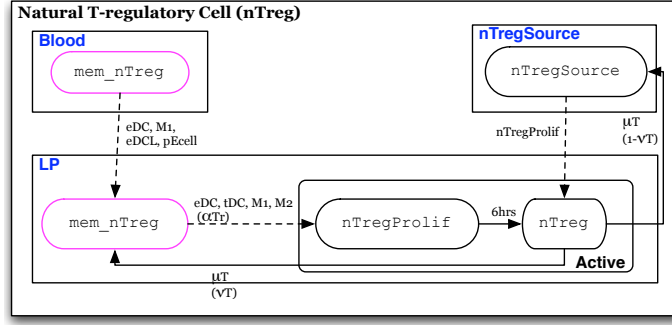


Figure 10: Automata for nTreg cells; individuals initially assigned the mem_nTreg or mem_nTregSource phenotype

3. IMPLEMENTATION

3.1. Rule Specification. Simulation specifications are set by the **configuration** file, the interaction **manifestation**, and the **scenario**. The configuration file sets the length of the simulation and which state transitions are multicontact-dependent (discussed below). The interaction manifestation calculates whether an interaction results in a state transition and the path of state transitions for each automaton. Each manifestation is encoded as a probabilistic timed transition system (PTTS), an extension of the finite state machine (FSM), with the following components: (1) Each individual that interacts with another will progress through a series of states. Each state is assigned to an *Interactor Set*, which determines with which other states it will interact. (2) An individual stays in a state for a period of time (dwell time) and then transitions to a succeeding state. Both the dwell time and the state transition are probabilistic. Once the cell has interacted, the progression of states and dwell times is purely a local calculation and is not affected by any other individual.

The **scenario** specifies state-dependent schedule assignment and state transition conditions that can override the transition pathway encoded in the **manifestation**. This is done in the form of *interventions* that may be implemented throughout the simulation. Each file is discussed in further detail in Appendix A.

3.2. Computation structure. The computation structure of this implementation consists of three main components: cells, locations, and message brokers. We assume a parallel system consisting of N cores, or processing elements (PEs). Processing proceeds in the following manner:

- (1) **Partitioning:** Cells and locations are partitioned into N groups denoted by C_1, C_2, \dots, C_N and L_1, L_2, \dots, L_N respectively. Currently the distribution is done in a round-robin fashion to allow even load balancing and simpler data management. Each PE also creates a copy of the message broker, denoted by MB_1, MB_2, \dots, MB_N . Each PE then executes the ENISI algorithm on its local data set (C_i, L_i) .
- (2) **Computing Visit Data:** The first step consists of computing a set of visits for each individual, C_i for the cycle. This also involves computing any state changes and applying events such as interactions, interventions, etc. A light-weight “copy” of each cell (called a *visit message*) is then sent to each location (which may be on a different PE) via the local message broker.
- (3) **Computing Interactions:** Each location receives the visit messages and forms a serial discrete event simulation (DES) by collecting the messages into a time-ordered list of arrive and depart events. Using this data, each location computes interactions for each individual at that location. Outcomes of these computations are then sent back to the “home” PEs of each cell via the local message broker.

- (4) Collecting Interaction Messages: Interaction messages for each cell on a PE are merged, processed and the resulting state of each affected cell is updated.

All the PEs in the system are synchronized after each simulation phase above. This guarantees that each location has received all the data required to form a DES and each cell has all the data needed to compute its new state. Below is a pseudocode description of this parallel version of the ENISI algorithm:

```

initialize;
partition data across PEs partition;
for  $t = 0$  to  $T$  increasing by  $\Delta t$  do
    foreach cell  $c_j \in C_i$  do
        send visits to location PEs;
        computeVisits( $j, t$  to  $t + \Delta t$ );
        sendVisits( $MB_i$ );
    Visits  $\leftarrow MB_i$ .retrieveMessages();
    synchronize();
    foreach location  $l_k \in L_i$  do
        compose a serial DES;
        makeEvents( $k$ , Visits);
        turn visit data into events;
        computeInteractions( $k$ );
        Process Events;
        sendOutcomes( $MB_i$ );
     $MB_i$ .retrieveMessages();
    synchronize();
    foreach  $c_j \in C_i$  do
        combine outcomes of multiple interactions;
        updateState( $c_j$ );

```

The ENISI framework is implemented in C++ and uses the Message Passing Interface (MPI) for distributed processing. An instance of the ENISI algorithm is run on each PE.

3.3. Algorithm. Here we describe the ENISI algorithm in further detail. This is an agent-based interaction algorithm that is an extension of one previously described in [3]. In short, the global contact graph and state of individual cells is updated in each cycle through 3 phases of computation. Each cycle represents six hours of simulation time. These phases are described below using the example shown in Figure 11 that depicts the movement and interaction of a set of individuals $[i\ j\ k\ q\ u]$ among a set of sublocations $[y\ z]$.

Phase 1: Each individual is assigned a schedule (Fig. 11A)

At the beginning of each cycle, each cell is provided a schedule according to its initial *phenotype*. This is a structure that specifies the tissue site (Location) and sublocation that will be occupied by each individual i and the times at which i arrives (StartTime) and i departs (EndTime) the sublocation.

Phase 2: Sublocation builds network and calculates interactions (Fig. 11B)

Each sublocation receives the list of individuals that will occupy it throughout the cycle along with the arrival and departure times of each. Each time an individual k arrives in sublocation y , it is added to the current contact network in the sublocation y , g_y , which is a subnetwork of the global network G . Let g_y^t be the set of individuals in the subnetwork g_y at time t . Each time an individual i departs sublocation y at time t ,

A. Phase 1: Each cell gets a scheduleSchedule for individual i on day 1:

Location	Sublocation	StartTime	EndTime
LP	y	0	30
LP	z	31	60

B. Phase 2: Each sublocation accepts list of visitssublocation y :

ID	StartTime	EndTime
i	0	30
j	25	35
k	29	35

sublocation z :

ID	StartTime	EndTime
q	0	40
u	10	45
i	31	60

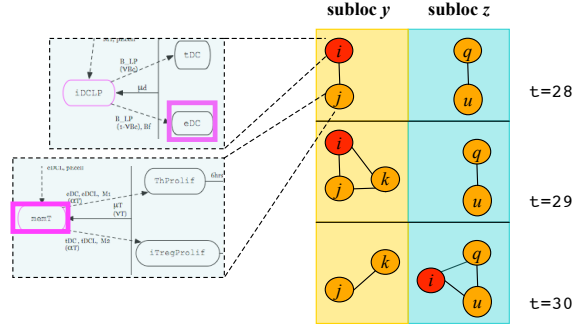
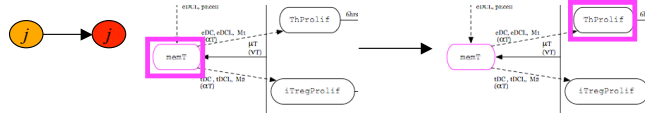
**C. Phase 3: Each cell assesses messages**

Figure 11: Individual i is in the eDC state and is able to induce a state transition in j , which is of a memT phenotype. At time $t = 28$, i and j are in the same sublocation according to the schedules assigned in phase 1. At time $t=29$ k enters sublocation y and is now in contact with i and j . At time $t = 30$, i departs sublocation y to sublocation z . Sublocation y then calculates whether a state transition occurred for i , j , and k . It is determined that a state transition occurs in j and sublocation y sends a transition message to j . In phase 3 j receives the transition message and transitions to a new state determined by the T cell-specific automata probability of $p_{s_j x} = \alpha_T$.

the sublocation performs the following processes: *i*) i is removed from g_y^t , *ii*) sublocation y calculates if i interacts; a function of the amount of time i was in contact with each of the other cells in g_y^t , its state s_i , and the configuration of g_y^t . *iii*) sublocation y carries out the same calculation on the remaining individuals j and k upon departure of i .

Whether an individual interacts with others in g_y^t is determined by a probability p calculated by one of two functions:

i) A *single contact-dependent* function given in Equation 3 where τ is the duration of contact, and p is a constant. This is a pairwise calculation. Hence, if an interaction occurs while performing the calculation on j , only j will receive the message and potentially change states. In other words, the contact graph is directed in that i points to j , when i is able to induce a transition in j . If j also induces a transition in i upon interaction, that is $s_j \in I_{si}$, then a separate interaction is calculated on i .

ii) A *multicontact-dependent* function given in Equation 4 where A is the total number of neighbors that

induce a state change and I is the total number of neighbors that inhibit the state change. The variables a , i , and y are constants.

$$(3) \quad p = 1 - \exp(\tau \ln(1 - \rho))$$

$$(4) \quad p = \left(\frac{aA}{aA + iI} \right)^y$$

The state of the individual i which of the two functions is used. If a state-transition occurs for an individual j then sublocation y sends an *interaction message* to j . However, the state of j will not immediately change.

Phase 3: Individual receives interaction message and determines new state (Fig. 11C)

At the end of each 6 hour cycle, if an individual j received a message it then transitions according to the specifications in the **manifestation**. In the *single contact-dependent* transitions, the individual determines to which state it transitions according to its current state and the phenotype of the interactor as this specifies the **manifestation** used to determine the conditions of the state transition. In the *multicontact – dependent* transition, the new state is determined by the current state alone. The state transition to the determined state then occurs at the probability p_{s_jx} , the probability of transition of individual in state s_j to the next state x in the transition pathway of its automaton specified in the **manifestation**.

This implementation employs two simplifications that require several model approximations; synchronous state update and the assumption of full connection of each subgraph. Here we discuss each:

Synchronous update scheme: All states are updated at the end of an cycle according to the conditions of the first interaction message during the cycle. Hence, any changes in behavior that result from the state transition do not take place until the next cycle. The model, therefore, assumes a 6 hour delay between a cell receiving the signal to differentiate and actual expression of cytokines or movement-mediating factors, such as integrins, that will affect subsequent movement, contacts, and effects on contacted cells. Another effect is an introduction of error when a bi-directional interaction is meant to result in the immediate removal of one of the individuals of the interacting pair. This is the case for interaction between a **dendritic cell** and a **foreign bacteria**. This interaction induces transitions $iDCLumen \rightarrow eDCL$ in the **dendritic cell** and $Bf_Lumen \rightarrow Bf_dead$ in the **foreign bacteria**. In the true system, the foreign bacteria would be removed as internalization is required for dendritic cell activation. However, in the model, bacteria will remain in the Bf_Lumen state for the duration of the cycle and be free to interact and induce $iDCLumen \rightarrow eDCL$ transition in other dendritic cells that it contacts in that period. This could be interpreted as each model bacterium being a representative of a population of bacteria such that when one interacts with an immune cell and dies, the others are able to continue on in the system. Hence, one bacterium can be considered a group of b bacteria where b is the average number of individuals in the $M0$, $iDCLumen$ and $iDCLP$ states it is expected to contact in one cycle.

Pairwise contact in complete graph: Each subnetwork g in a sublocation is a complete, fully connected graph. In the case that an interaction calls for a single contact-dependent transition there arise situations in which i may simultaneously be in contact with individuals of its interactor set I_{si} , that lead to different state transitions. An example is a resting T cell in contact with both a tDC and eDC. In such cases, the individual “chooses” which individual to interact with probabilistically. In the default settings, this probability is 0.5.

3.4. Parameterization. Table 2 list the parameters of the ENISI model along with the default values assigned. Values are assigned from literature when direct measurements are available. Others were given with basic assumptions of the model that are described throughout the text of Section 2. For example, ENISI only simulates those cells that recognize and react to commensal bacteria and the pathogen represented by **foreign bacteria**. Hence, the parameter α_T is set at 1, all T cells represented have receptors specific to the bacteria present. In addition, it is assumed that damaged, pro-inflammatory epithelial cells are completely permeable to bacteria by setting $\beta_p = 1$. Macrophages are assumed long lived and remain in the active state by setting μ_{M0} to be the length of the simulation. Where transition probabilities could not be logically reasoned from the knowledge of the system, we initially assume that each contact that may result in a state change does resulted in the state change.

Certain parameters will have to be estimated to represent specific experimental conditions. These include parameters governing state transition functions that involve interaction with **foreign bacteria** where **foreign bacteria** is meant to represent a specific pathogen. Such parameters will have to be fit to experimental infection data for the specific pathogen to be represented. Such infection studies generally report data in the form of qualitative measurements of symptom severity such as epithelial damage and duration of illness that can be mapped to the number of individuals occupying a certain state in the model, *i.e.* the configuration of the system on a given day d . There are three types of parameters that govern whether an individual will occupy a specific state s on day d post-infection given it is in state x on day $d - 1$. These are *i*) the dwell time in state s , *ii*) the probability p_{sx} that an individual in state s will transition to state x , and *iii*) The constants of the interaction equations (Equation 3 and Equation 4). Most dwell times are available in the literature leaving the need to estimate p_{xs} .

To reproduce dynamics seen experimentally, we identify cell population of a state s whose count does not fit the configuration mapped to the experimentally observed health outcome. We then assign ranges to the transition probability that yields the desired count with Equation 5 which gives the expected number of individuals in state s on day d , κ_s^d , where x is the preceding state to s in the manifestation, p_{xs} is the probability of the transition $x \rightarrow s$ upon interaction, $\kappa_{I_x}^{d-1}$ is the number of individuals in a state from the set I_x of states that interact with x , $subloc_y$ is the constant number of sublocations in Location y , and ϵ_x is the constant number of times individuals in state x change sublocations in one cycle. The estimated value of p_{sx} , \hat{p}_{sx} , is then calculated from Equation 6 holding the variable $\kappa_{I_x}^{d-1}$ at its value with some initial value for p_{sx} and setting κ_s^d to the expected value determined from experimental data. During calibration, values for p_{xs} are then sampled from a tight range around \hat{p}_{xs} .

$$(5) \quad \kappa_s^d = \frac{p_{xs} \kappa_{I_x}^{d-1} \kappa_x^{d-1} \epsilon_x}{subloc_y}$$

$$(6) \quad \hat{p}_{xs} = \frac{subloc_y \kappa_s^d}{\kappa_{I_x}^{d-1} \kappa_x^{d-1} \epsilon_x}$$

The number of sublocations in each Location affects the rate at which cells contact eachother. Though current *in vivo* visualization techniques allow one to estimate the frequency with which certain cells contact others, this data was not found for the cell types represented in the LP. Equation 7 determines the number of sublocations in Location y , $subloc_y$, necessary for a single cell i to contact r cells in state s in one cycle, where κ_s is the number of individuals in the state s , and ϵ_i is the number of sublocations individual i visits in one cycle. As more imaging data for the LP in inflammatory conditions becomes available and contact frequencies observed, the simulated contact frequency can be systematically set by solving this equation.

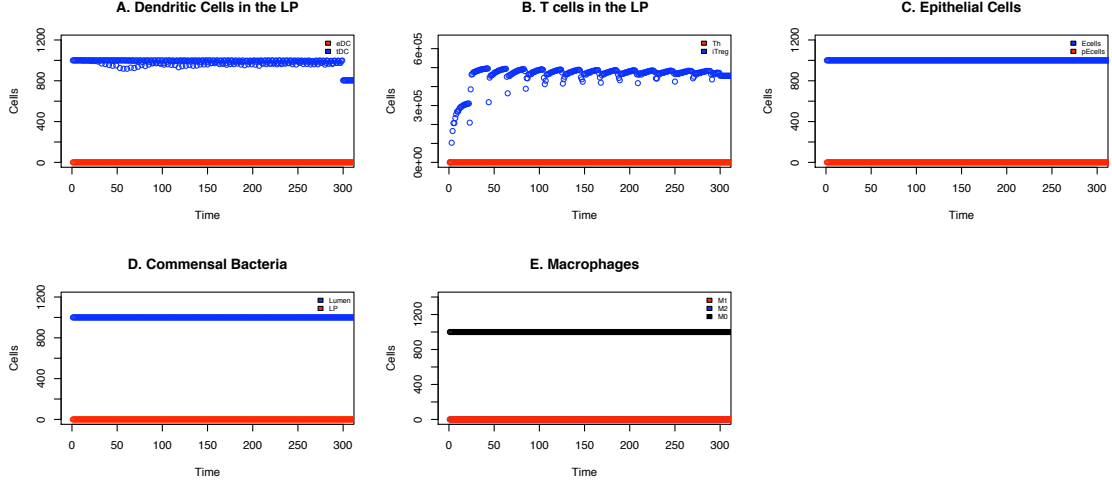


Figure 12: Dynamics of cell populations over a period of 75 days with only commensal bacteria in the lumen and no pathogen present. The x-axis is labelled in time units of 6 hours.

$$(7) \quad subloc_y = \frac{\kappa_s \epsilon_i}{r}$$

4. SIMULATION OUTPUT AND APPLICATION

ENISI allows users to test hypotheses and identify pathways to tissue-level phenomena by specifying conditions in the *in silico* mucosal environment and observing resulting effects. Simulations with ENISI provide visual outputs in two formats; *i*) a plot of the total number of individuals in each state in each location over time and *ii*) a report of the number of individuals in each state that interacts with an individual in a user-specified state $s_i \in S$ and induce the state change $s_i \rightarrow s_j$ over user-specified time periods during the simulation. These counts may then be represented in a number of graphical formats.

Here we demonstrate an application of ENISI by simulating a typical inflammatory response to bacteria of the *B. hyodysenteriae*, an experimental model for chronic immunopathological colon inflammation. We then turn to examples of simulation output to identify a key pathway by which chronic inflammation persists and epithelial cell damage occurs following *B. hyodysenteriae* elimination.

4.1. *B. hyodysenteriae* induced colitis: calibration and model validation. *B. hyodysenteriae* infection is characterized by severe, transient dysentery in which epithelial lining is damaged and bacteria is detected in the LP resulting in fever and bloody diarrhea. These symptoms last for 1 week [14] followed by low-level persistence of inflammatory factors and continued epithelial damage even after pathogen has been eliminated.

We seek to identify pathways that lead to two different health outcomes following infection: *i*) *complete recovery*: All pathogen is removed, there is no bacteria in the LP and all epithelial cells are alive after pathogen elimination and the number of immune cells in a state corresponding to an inflammatory phenotype is below a threshold x . In other words, a configuration in which $\kappa_{[B_{LP}, B_{fLP}, pE_{cell}, E_{dead}]} = 0$ and $\kappa_{[Th, M1, eDC, eDCL]}^t < x$, where t_{Bf_dead} is the time at which the last cell enters the *Bf_dead* state and $t > t_{Bf_dead}$.

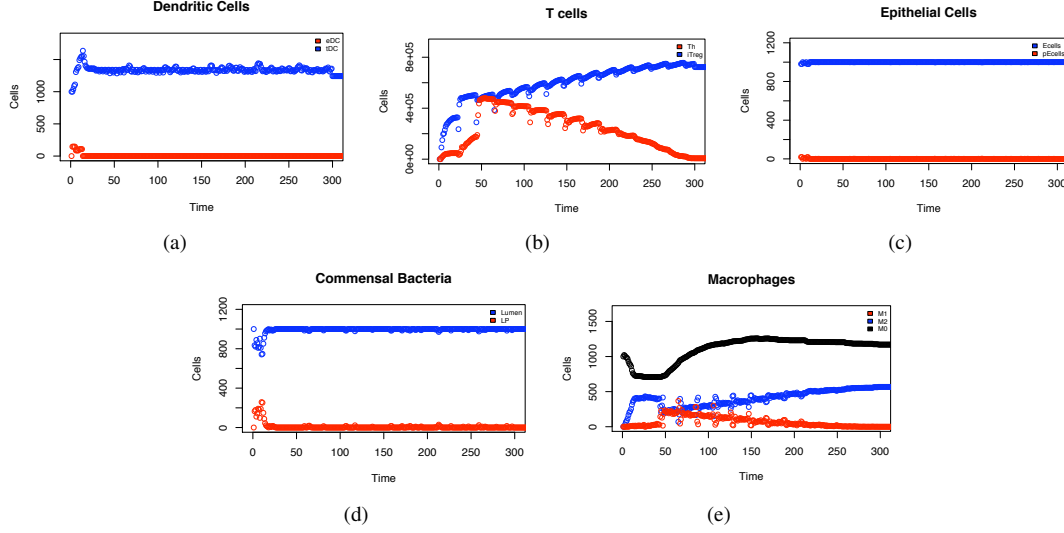


Figure 13: Dynamics of cell populations over a period of 75 days following infection with *B. hyodysenteriae*. The x-axis is labelled in time units of 6 hours.

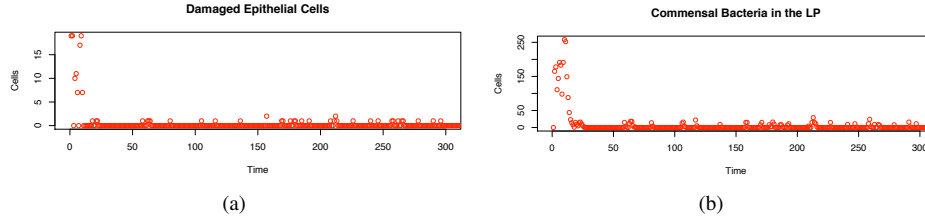


Figure 14: Number of of damaged epithelial cells and commensal bacteria in the LP over a period of 75 days following infection with *B. hyodysenteriae*. The x-axis is labelled in time units of 6 hours.

ii) *Chronic inflammation*: A configuration in which $\kappa_{[B_LP, pEcell, Edead]}^t \geq 1$ and $\kappa_{[Th, M1, eDC, eDCL]}^t \geq x$, where $x = 1$.

First, we demonstrated that the system behaves as expected in the absence of pathogen. Figure 12 shows results from a simulation of a healthy, pathogen-free mucosa over 75 days following population of the lumen by commensal bacteria. As is expected to occur in a healthy steady-state system [21, 18], there is immune activation by commensal bacteria, shown by elevated κ_{iDC} (Figure 12A) and κ_{iTreg} (Figure 12B), however there is no inflammatory response allowing all **epithelial cells** to remain in a healthy *EC* state (Figure 12C). With no epithelial damage, commensal bacteria remain in the *B_Lumen* state with no invasion in to the LP (Figure 12D) and **macrophages** remain unstimulated (Figure 12E).

To represent *B. hyodysenteriae* we assigned values to parameters governing **foreign bacteria** state transitions and functions according to observations of interactions between bacteria of the *Brachyspira* genus and immune cells reported in the literature. For example, bacteria of the *Brachyspira* genus has been shown

to cause cell death and inflammatory cytokine secretion in epithelial cells upon contact [20]. Therefore, the parameter v_{EB} was assigned a value of 1. We then fit parameters governing adoption and transition of inflammatory states to the early, acute stage dynamics of *B. hyodysenteriae* infection in pigs [14] as well as unpublished observations by collaborators. Specifically, we compared model predictions to qualitative measurements of immune cell levels and clinical symptoms of pigs over a period of two weeks following *B. hyodysenteriae* infection.

During calibration, the parameters that needed to be changed from default values to fit qualitative experimental dynamics were v_{EC} as well as the constants for the functions v_{12} and v_{21} (Table 2). The constants y_1 and y_2 of functions v_{12} and v_{21} , respectively, were increased incrementally by 1 until we observed the expected dynamics of a transient M1 response in the presence of pathogen that is then overcome by the M2 population. The parameter v_{EC} , which can also be considered $p_{EC_{cell}p_{EC_{cell}}}$, was calculated with Equation 6 such that, on average, at least 1 cell enters the pEC_{cell} state when Th1 and M1 are at their peak values on day 8 post-infection with pathogen. Note that this peak time coincides with cycle 50 on the x-axis of plots in Figure 13 as each cycle represents 6 hours.

Experimental infection was simulated by adding 30 individuals in the *Bf_Lumen* state on days 1, 2, and 3 [14, 12]. Table 1 gives the number of individuals initially assigned to each of the states to represent an immunologically inactive system at the time of infection. We then follow the epithelial cell, immune cell, and bacteria populations over a period of 75 days. Here we define symptomatic dysentery as > 1 individuals in the pEC_{cell} state and > 1 bacteria in the *BLP* state.

Following infection, cell population dynamics accurately reflect those seen in experimental infections of pigs [14, 12] (Figure 13) with three distinct phases: acute inflammation (days 1-6), decline of inflammation (days 7-50), and the recovery/chronic phase (days 51-75).

Acute inflammation is marked by an increase in κ_{eDC} (Figure 13A) and κ_{Th} (Figure 13B), followed by epithelial damage (Figure 13C) that is shortly followed by bacterial invasion in to the LP (Figure 13D). At this time macrophages are stimulated and we see κ_{M1} (Figure 13E) rise in conjunction with κ_{Th} (Figure 13B) along with increased monocyte recruitment and transient reduction in M2 (Figure 13E).

As validation, the simulated infection reproduces dynamics to which parameters were not fitted. Specifically, epithelial damage and bacterial invasion to the LP that are seen with in the first week of infection last for a period of 1 week (Figure 14), the number of inflammatory T cells rise for 10 days, longer than the clinical symptoms, and extension of the simulated infection past 2 weeks show a low level of epithelial damage along with low level bacterial presence in the LP (Figure 14).

4.2. Mechanism of chronic inflammation. To identify the source of this continued epithelial damage we observed the states of those neighbors that induce the transition $EC \rightarrow pEC_{cell}$ for all **epithelial cells** that undergo this transition during days 1-6, days 7-50, and day 51-75 corresponding to the three phases of infection.

The histogram in Figure 15 shows the number of individuals of each phenotype whose interaction with an individual in the *EC* state results in the transition of $EC \rightarrow pEC_{cell}$. It can be seen that at all stages of infection, it is individuals occupying the *Th* state that are inducing the most epithelial damage. We then report the states of neighbors that induce the transition $memT \rightarrow Th$ (Figure 15). In this case, it is clear that in the initial stage of infection, ‘sampling’ eDC in the lumen plays a significant role. During the peak of inflammation eDC in the lumen is no longer as significant and macrophages dominate as key drivers of Th stimulation with M1 playing a significant role. However, in the chronic phase, it is individuals in the transient state *M21*, an intermediate state between M2 and M1, that are solely responsible for Th1 stimulation. To further demonstrate this point, simulated infections were replicated in the absence of the ability of M1 to induce state change in neighbors of the *memT* state, allowing T cell stimulation to occur only through contact with eDC. The result is the dynamics shown in Figures 16 and 17. In this scenario there is a weaker, more

rapidly subsiding inflammatory response as ‘sampling’ dendritic cells stimulate T cells. However, dendritic cell activation is short lived and once the pathogen is removed *eDCL* is quickly removed. The reduced κ_{Th} during the acute inflammation phase results in less epithelial damage as well as lower inflammatory cytokine concentrations reducing the frequency of $M2 \rightarrow M1$ transitions allowing a tolerogenic environment to persist.

The conclusion of this simple demonstration is that residual tissue damage occurs through Th1/Th17-mediated cytotoxicity which is stimulated by M1 that has recently transitioned from the M2 phenotype. Hence the presence of M1 is due to two parallel consequences of pathogen presence *i*) the rise of Th1/Th17 and damaged epithelial cells which leads to increase inflammatory cytokine concentration, and *ii*) increased damage of epithelial layer which allows greater invasion of commensal bacteria. The implication is that commensal bacteria, which directly induces a tolerogenic response, is indirectly responsible for maintaining immunopathological chronic inflammation response via macrophage stimulation once the environmental concentration of inflammatory cytokines reaches above a certain threshold.

Indeed, when simulated infection was repeated in the absence of commensal migration, $\beta_p = 0$, chronic inflammation and epithelial damage is not seen following elimination of *B. hyodysenteriae* (not shown). This is despite the fact that **foreign bacteria** is still able to invade the LP.

5. RELEVANCE OF ENISI AND FUTURE DIRECTIONS

Aspects of the presented inflammatory and regulatory immune pathways have been represented in previous models of mucosal infection [1, 30, 5, 6] that have provided insight on mechanisms of clinical symptoms as well as pathogen persistence. The ENISI model is unique in its scope and approach. The model incorporates regulatory mechanisms of both adaptive and innate immunity, multi-location migration of cells, and cross talk between antigen presenting cells and T-cells. In addition, it is mechanism-based explicitly representing each participating cell of the immune pathway. This facilitates mapping of model parameter specifications and predictions to laboratory techniques that manipulate specific cell populations.

We previously implemented a larger scale version of the model, encompassing these aspects, as a system of differential equations. Simulations based on this initial version identified a relationship between the Th and M1 concentrations in the LP and chronic epithelial damage [29]. However, differential equations (ODEs) can only capture the dynamics of each cell population as a whole. Hence, this work identified a relationship between M1 and Th levels and epithelial damage, but the ODE representation did not allow us to identify the specific pathways in which T cells induce epithelial damage after being stimulated by M1 macrophages. An additional drawback of the ODE representation is that it assumes deterministic, average behavior by each individual cell. However, biological systems are known to act stochastically due to attributes, such as cytokine secretion and association time with stimulating factors, that vary widely across individual cells in a population. Additionally, the randomness introduced by cell movement leads to non-uniform distribution across single tissue sites. Due to these assumptions of determinism and homogeneity, that are surely violated by the system in reality, dynamics predicted by an ODE model may not accurately reflect those seen in nature.

The ENISI model can be viewed as an extension of the interacting state machine models or agent-based models. A key aspect of these models is a procedural and interactive (a.k.a. mechanistic, algorithmic, executable) view of the underlying systems. In this view components of the system interact locally with other components and the behavior of individual objects is described procedurally as a function of the internal state and the local interactions. This agent-based approach allows incorporation of spatial effects and randomness of cell-cell and cell-bacteria contact. In the case of colonic inflammation spawned by a small number of pathogen, such randomness is believed to significantly affect the outcome of the system and, therefore, an

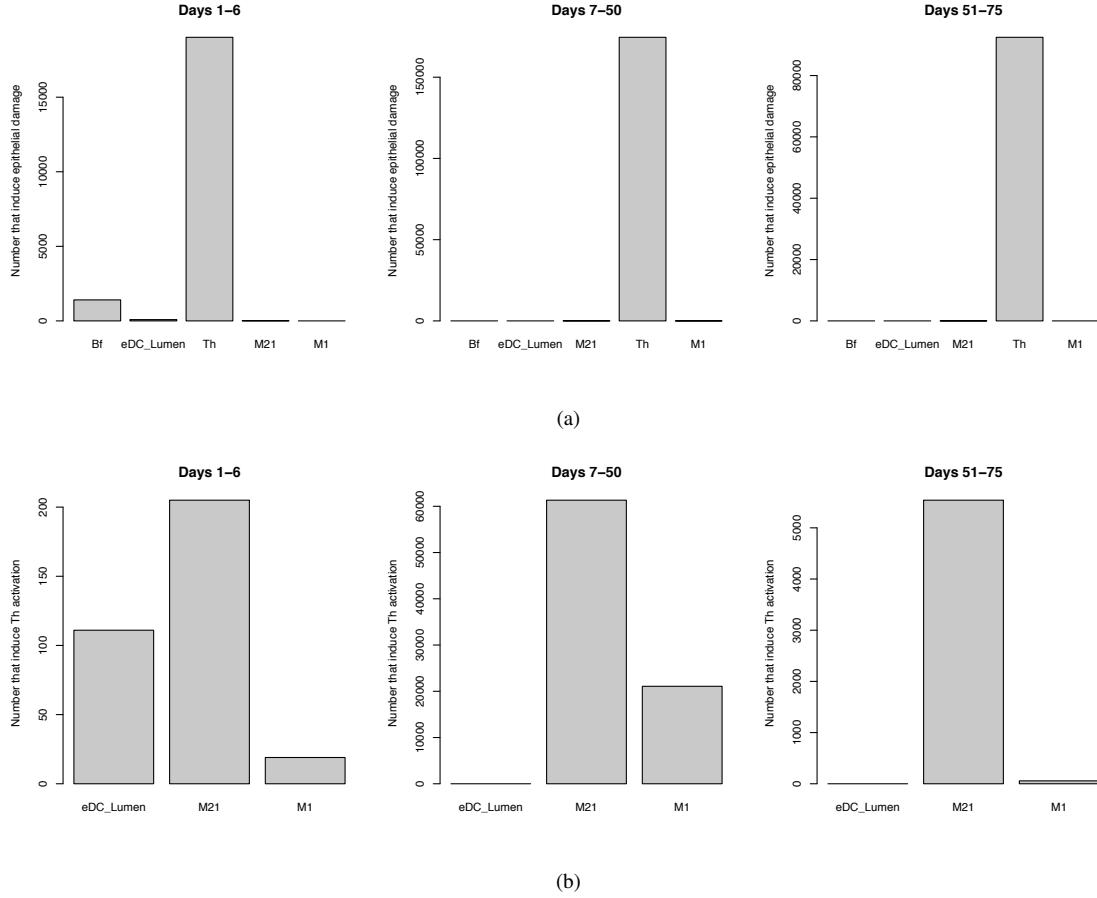


Figure 15: (a) Histogram of the number of individuals in each state that interact with an **epithelial cell** and induce the transition $EC \rightarrow pE_{cell}$. (b) Histogram of the number of individuals in each state that interact with a **T cell** and induce the transition $memT \rightarrow Th$

agent-based model is an appropriate representation [4]. This also creates a foundation for encompassing emergent properties such as bacterial strain evolution and changes in microflora demographics as the model is elaborated and the simulator extended. However, the drawback to such methods is that they are often not scalable due to limitations of computation power.

ENISI is implemented using an algorithm that is an extension of that used to simulate epidemic spread across large social networks [3]. The ENISI model was implemented on this software platform because the algorithm is known to scale to large numbers approaching those found in the true system. Scalability is highly relevant when seeking to reproduce emergent tissue-level phenomena by simulating individual cell interactions. Larger scale models are necessary as the purpose of immune simulators is to reproduce dynamics in a true *in vivo* system where immune cell concentrations can reach $10^8/mL$ [10]. It may not be sufficient to simulate the dynamics of a small sample and extrapolate results to the entire organ. To do so is

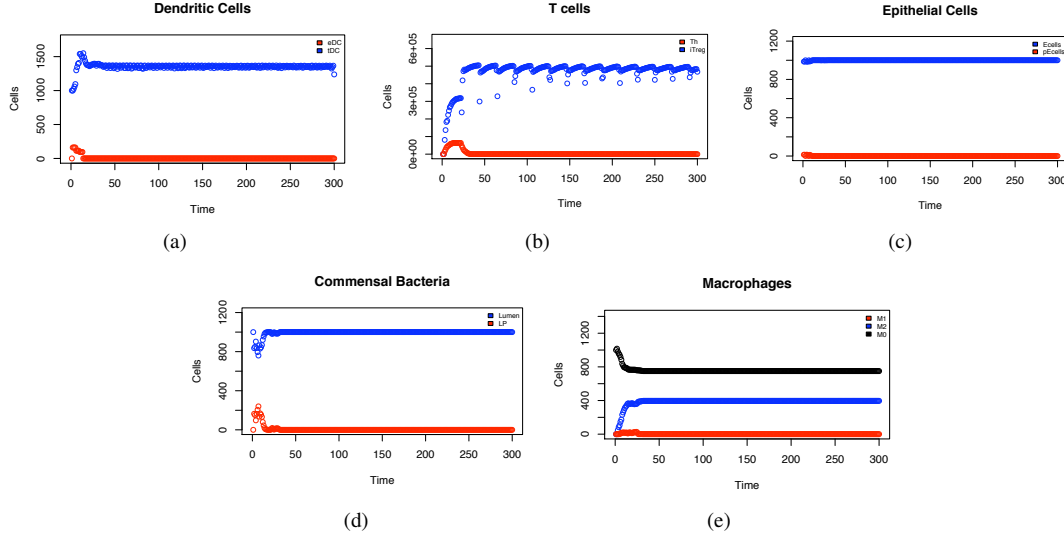


Figure 16: Dynamics of cell populations over a period of 75 days following infection with *B. hyodysenteriae* without T cell stimulation by M1. The x-axis is labelled in time units of 6 hours.

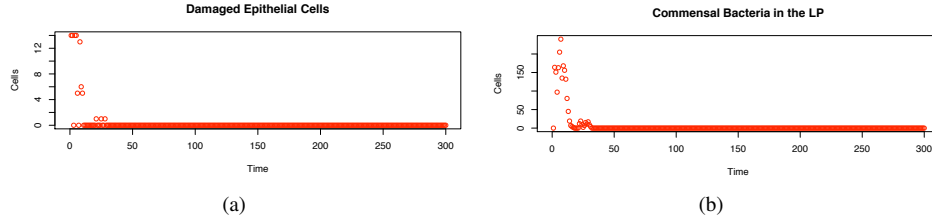


Figure 17: Number of of damaged epithelial cells and commensal bacteria in the LP over a period of 75 days following infection with *B. hyodysenteriae* with out stimulation of T cells by M1. The x-axis is labelled in time units of 6 hours.

to ignore non-linear and complex nature of the cell interactions and dynamics and make the assumption of uniform mixing which defeats the purpose of an agent-based approach.

There are various general, agent-based biological simulator tools publicly available including Rhapsody [7, 28], NFSim [25], and that developed by [27] that translate graphical models in to executable code to run simulations. These simulators place an emphasis on rules governing cell-cell contacts and signaling interactions allowing one to enter complicated functions for these mechanisms. They, therefore, provide the useful capability of incorporating complex mathematical models for receptor-ligand interactions and phenotype differentiation in to cell contact networks. However, the scalability of these implementation algorithms in term of system complexity and the number of individuals in a network is unclear. For example, Rhapsody has been shown to simulate up to 10^4 individuals efficiently [7, 28] .

ENISI is a unique contribution to the field of immunological tools as an agent-based model of an unprecedented scale, simulating complex interaction and migration of 10^6 individuals over a simulated 3 month

period with in 1 hour. Though it currently requires scripting to create simulation specifications, a graphical user interface will be publicly available in September 2011 at <http://www.modelingimmunity.org>.

The ENISI model presented here may be implemented with other biological simulators [7, 28, 25, 27] at a later date to provide a complementary tool with which immunologists can potentially conduct smaller scale simulations that include more complex rules for cell-cell interactions and phenotype differentiation.

5.1. Future steps. ENISI is an evolving *in silico* system. It is currently being extended to include distal lymphoid tissues, as well as automata to represent specific bacterial species such as *Helicobacter pylori* and *Escherichia coli*. In addition T cell populations will be further refined in to separate Th1 and Th17 types.

6. APPENDIX A: SIMULATION FILES AND SPECIFICATIONS

All individuals that will be present throughout the simulation must be declared in the **population file** where they are assigned a unique identifier and an initial state. Locations must be declared in the **location files** where they are given unique identifiers and divided in to sublocations. Movement of each declared individual among the declared locations is then specified in the **schedule files**.

Simulation specifications are set by the **configuration file**, the interaction **manifestation**, and the **scenario**. The configuration file sets the length of the simulation, which state transitions are multicontact-dependent (discussed below), and points to all input and output files. Each automaton is encoded as a set of probabilistic times transition systems (PTTSs) in the manifestation files, which calculate whether an interaction results in a state transition and the path of state transitions. The **scenario** specifies state-dependent schedule assignment and state transition conditions that can override the transition pathway encoded in the **manifestation**. This is done in the form of *interventions* that may be implemented throughout the simulation.

In the current code, time specifications correspond to a single cycle of the ENISI algorithm which is 24 simulation hours (86400 simulation seconds). With the current parameter set in ENISI, each cycle represents 6 real world hours.

6.1. Input Files.

(1) Population file: cells.txt

Cells.txt specifies the complete population of cells in the following format:

```
< CellId > < location > < ActiveState > < BirthDeathState > < RegState > < InflammState >
< Type >
```

It lists each individual (< CellId >) along with their initial location (< location >), and their initial states in each of the four manifestations; activation.mnf (< ActiveState >), birth_death.mnf (< BirthDeathState >), Regulatory.mnf (< RegState >), and Inflamm.mnf (< InflammState >). Manifestations are explained below. The < Type > column is not relevant and is simply filled with a '1' for each individual.

(2) Location files: Default.cfg sites.txt, sublocation.txt

Each tissue site, or *Location*, corresponds to a *LOCATION_TYPE*. Each *LOCATION_TYPE* is composed of multiple *sublocations* that may be further divided in to *compartments*. The configuration file (Default.cfg) defines each *LOCATION_TYPE*. sites.txt defines the sublocations, and sublocation.txt defines the compartments. Each is described here:

- (a) **Default.cfg:** The number of sublocations for each *LOCATION_TYPE* and their corresponding identifiers is set in the configuration file with the following syntax:

```
LOCATION_TYPE_RANGE_1 1 5000
LOCATION_TYPE_RANGE_2 5001 6000
```

```

LOCATION_TYPE_RANGE_3 6001 16000
LOCATION_TYPE_RANGE_4 16001 16110
LOCATION_TYPE_RANGE_5 16111 16111
LOCATION_TYPE_RANGE_6 16112 16112
LOCATION_TYPE_RANGE_7 16113 16113
LOCATION_TYPE_RANGE_8 16114 16114
LOCATION_TYPE_RANGE_9 16115 16115
LOCATION_TYPE_RANGE_10 16116 16116
LOCATION_TYPE_RANGE_11 16117 16117

```

In this version there are 11 Locations. LOCATION_TYPE1 is the LP, LOCATION_TYPE2 is the LN and LOCATION_TYPE4 is the EB. All others are currently not occupied by cells and are simply place holders for tissue sites that may be added in the future. These are arbitrarily assigned one sublocation. In this example, the LP, LOCATION_TYPE1, is composed of sublocations 1 through 5000 and the LN, LOCATION_TYPE2, is sublocations 5001 through 6000. There are a total of 16117 sublocations that cells may occupy.

- (b) **sites.txt:** This file lists all the sublocations in the simulation in the following format:

```
< siteId > < type1 >
```

The < siteId > column simply lists all sublocation identifiers, 1 through 16117. The < type1 > column is irrelevant and, at this point, is filled with the arbitrary number '1'.

- (c) **sublocation.txt:** In the case that one wishes to further divide sublocations in to compartments, this is done in sublocations.txt. This file specifies, for each LOCATION_TYPE (< type >), the range of compartments, specified by < start_id > and < end_id >, for each sublocation in LOCATION_TYPE< type >. This is done in the following format:

```
< type > < start_id > < name > < end_id >
```

The < name > column is not relevant and may be any arbitrary, alphanumeric label. One range is specified for each LOCATION_TYPE and all sublocations will be divided in to the same number of compartments. Only individuals in the same compartment are considered in contact.

- (3) **Schedule files:** Located in the directory Schedules/

Each schedule file specifies the movement of each individual with in one simulation cycle in the following format:

```
< CellId > < Location > < SubLocation > < StartTime > < EndTime > 1
```

Example:

```

i -x -x 0 7200 1
i -x -x 7201 14400 1
i -x -x 14401 21600 1
i -x -x 21601 28800 1
i -x -x 28801 36000 1
i -x -x 36001 43200 1
i -x -x 43201 50400 1
i -x -x 50401 57600 1
i -x -x 57601 64800 1
i -x -x 64801 72000 1

```

```
i -x -x 72001 79200 1
i -x -x 79201 86400 1
```

For each individual c_i it lists the individual identifier ($\langle CellId \rangle$) and each location ($\langle Location \rangle$ and $\langle SubLocation \rangle$) that c_i visits during the cycle as well as the amount of time spent in each sublocation, $StartTime$ to $EndTime$, in simulation seconds. In the above example, $-x$ in the $\langle Location \rangle$ and $\langle SubLocation \rangle$ columns specifies a randomly chosen sublocation of LOCATION_TYPE x and a randomly chosen compartment within that sublocation from the range specified in the sublocations.txt file. Hence, individual c_i occupies a different sublocation of Location x every 7200 simulation seconds over a single cycle corresponding to 24 simulation hours (86400 simulation seconds).

- (4) **Manifestation files:** As described in section 3, a cell c_i is in contact with a neighbor c_j when they are in the same sublocation. Each individual that interacts with another will progress through a series of states. The interaction **manifestation** calculates whether c_i interacts with c_j and, if it does, whether c_i will transition states from its current state s_i to the next state of the automaton. The manifestations encode the path of state transitions for each automaton as a probabilistic timed transition system (PTTS), an extension of the finite state machine (FSM), with the following components: (1) Each state ($\langle state \rangle$) is occupied for a period of time ($\langle DwellTime \rangle$). (2) Each state is assigned an *InteractorState*. This value determines to which *interactor set*, I , the state belongs and, subsequently, with which other states it will interact. To clarify, let the state of individual c_i be s_i . For each state s there is an interactor set I_s of states such that if a contact c_j of individual c_i is in a state $s_j \in I_{s_i}$, then c_i will interact with c_j and probabilistically transition states. (3) Each state has an $\langle Infectiousness \rangle$ and $\langle Susceptibility \rangle$ value between 0 and 1. The $\langle Susceptibility \rangle$ of a state determines how likely the individual c_i is to interact with a neighbor c_j if $s_j \in I_{s_i}$. The $\langle Infectiousness \rangle$ of s_i determines how likely c_j is to transition if $s_i \in I_{s_j}$. The $\langle TransitionType \rangle$ value of a state indicates whether transition to the next state is time-dependent or contact-dependent. Specifically, $\langle TransitionType \rangle = 0$ indicates transition from the *state* to the *nextstate* is time-dependent and $\langle TransitionType \rangle = -1$ indicates it is contact-dependent. Once it is determined that a state transition will occur based on these properties of s_i , c_i will transition to the next state ($\langle next_state \rangle$) with a specified probability ($\langle probability \rangle$). This is encoded in the manifestation files in the following syntax:

```
DISEASE_STATE <state> <DwellTime> <InteractorSet> 0 <Infectiousness> <Susceptibility>
<TransitionType>
DISEASE_LINK "Untreated" <probability> <next\_state>
```

With in one manifestation an individual's state may transition along multiple paths distinguished by probabilities $p_1 \dots p_n$ assigned to each of n paths such that $\sum_{i=1}^n p_i = 1$. A simple example is shown in Figure 18A that shows a sample automaton of individual i . In this example, i undergoes the transition $memT \rightarrow ThProlif$ when it interacts with individuals that occupy the eDC , $eDCL$, or $M1$ state and is activated with a probability of α_T . An individual with whom i interacts is termed its *interactor*. Alternatively, i is not activated with a probability of $1 - \alpha_T$ and remains in the $memT$ state. After a dwell time of 6 hours, the individual undergoes the transition $ThProlif \rightarrow Th$. After a dwell time of μ_T , it undergoes the transition $Th \rightarrow memT$ with a probability of v_T and $Th \rightarrow ThSource$ with a probability of $1 - v_T$.

When multiple paths are not distinguished by a probability but rather by the state of the interactor, as is the case in Figure 18B, then that state is not reachable in the same manifestation. In this

example, i will transition from a $memT$ state to either a $ThProlif$ or an $iTregProlif$ state depending on the state of its interactor. In the case that the $memT$ contacts an individual j whose state $s_j \in [eDC, eDCL, or M1]$ then it enters $ThProlif$, however, if $s_j \in [tDC, tDCL, M2]$, then i takes a different path to $iTregProlif$. This requires a **secondary manifestation** in which the alternate interaction ($memT \rightarrow iTreg$) is calculated and then used to transition the state in the **main manifestation** as demonstrated in Figure 18C. This synchronization of multiple manifestation for the same automaton is controlled through the **scenario**.

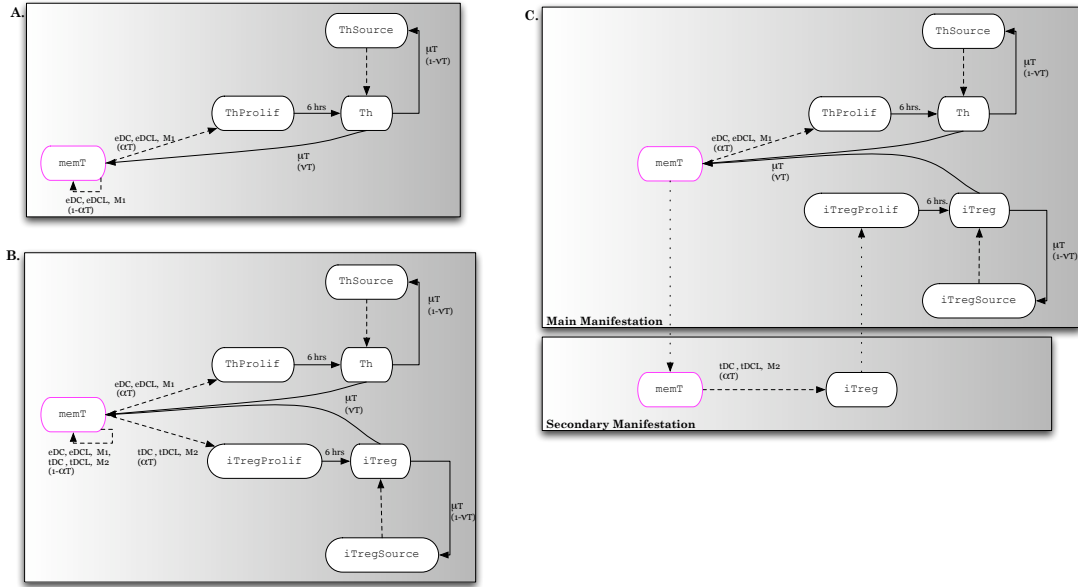


Figure 18: Ovals are states of the automaton and pink highlights the initial state. Dashed arrows represent contact-dependent transitions, solid arrows represent time-dependent transitions, and dotted arrows point from states in one manifestation to the state which is subsequently adopted in the other. A. One manifestation that composes the T cell automaton. This contains 2 pathways leaving from $memT$. Which path is taken is probabilistic. B. An extension in which an additional pathway may be taken from the $memT$ state and which path is taken is now determined by the state of contacts. C. Example of a separation of a single automaton in to two manifestations. Contact with individuals with a state from the set $[tDC, tDCL, M2]$ induce transition to $iTreg$ in the secondary manifestation, which causes the main manifestation to adopt the $iTregProlif$ state.

- (5) **Scenario file: Default.scn** In the **scenario** one specifies *i*) the day on which bacterial doses are given, *ii*) schedule assignment conditions, and *iii*) state transition conditions that can override the transition pathway encoded in the **manifestation**. This is done in the form of *interventions* that may be implemented throughout the simulation. The scenario file uses a specific syntax shown below:
- ```
bnf
```

The scenario is a series of triggers and actions. A trigger is a conditional statement that is applied to each interactor individually. It can be a function of the individual's state, the simulation 'day', or cycle, or an attribute assigned to an individual such as 'age'. The grammar for the scenario file language can be seen in Figure 19. Each line of the scenario is read for each person at the beginning of each cycle.

If a trigger evaluates to true, one or more actions are executed. These actions can modify the interactor by changing its attributes or schedule, or explicitly transitioning its state in one of the

```

⟨scenario⟩ → version ⟨maojr⟩.⟨minor⟩
 (⟨intervention⟩ | ⟨trigger⟩ | ⟨comment⟩)*;
⟨intervention⟩ → intervention ⟨intervention_name⟩ ⟨action⟩+;
⟨trigger⟩ → trigger [repeatable] [single] [with prob = ⟨real⟩] ⟨condition⟩ ⟨action⟩ |
 state (on entry | on exit) ⟨state_name⟩ [with prob = ⟨real⟩] ⟨condition⟩ ⟨action⟩;
⟨action⟩ →
 apply ⟨intervention_name⟩ [with prob= (⟨real⟩ | ⟨real_var⟩)] |
 treat ⟨fsm_name⟩ ⟨treatment_name⟩ |
 untreat ⟨fsm_name⟩ ⟨treatment_name⟩ |
 schedule ⟨sched_name⟩ ⟨priority⟩ |
 unschedule ⟨priority⟩ |
 infect ⟨fsm_name⟩ |
 transition ⟨fsm_name⟩[:⟨state_name⟩] [keep time | normal] |
 remove |
 endsim |
 message ⟨string⟩ |
 set (⟨var_name⟩ | person.⟨person_attribute⟩) (= ⟨integer⟩ | ++ | -- | += ⟨integer⟩ | -=
 ⟨integer⟩) |
 add (⟨int_var⟩ | ⟨integer⟩ to ⟨set_name⟩ |
 delete (⟨int_var⟩ | ⟨integer⟩ from ⟨set_name⟩);
⟨condition⟩ → ⟨or_expr⟩;
⟨or_expr⟩ → ⟨and_expr⟩ | or ⟨and_expr⟩;
⟨and_expr⟩ → ⟨not_expt⟩ | and ⟨not_expt⟩;
⟨not_expt⟩ → not ⟨or_expr⟩ | (⟨or_expr⟩) | ⟨base_expr⟩;
⟨base_expr⟩ → ⟨binary_cond⟩ | ⟨set_cond⟩ | true | false;
⟨binary_cond⟩ →
 ⟨int_var⟩ ⟨binary_op⟩ ⟨integer⟩ |
 ⟨real_var⟩ ⟨binary_op⟩ ⟨real⟩ |
 ⟨string_var⟩ ⟨binary_op⟩ ⟨string⟩;
⟨set_cond⟩ → ⟨set_name⟩ intersect ⟨set_name⟩ is not null |
 ⟨set_name⟩ contains (⟨int_var⟩ | ⟨integer⟩);
⟨binary_op⟩ → < | <= | = | != | >= | >;
⟨var⟩ → ⟨int_var⟩ | ⟨real_var⟩ | ⟨string_var⟩;
⟨int_var⟩ →
 day |
 time |
 person.id |
 person.removed |
 person.⟨person_attribute⟩ |
 ⟨fsm_name⟩.infected |
 ⟨fsm_name⟩.⟨fsm_attribute⟩ |
 ⟨var_name⟩;
⟨real_var⟩ → ⟨fsm_name⟩.infectivity | ⟨fsm_name⟩.susceptibility;
⟨string_var⟩ → ⟨fsm_name⟩.state;
⟨XXX_name⟩ → [a-zA-Z0-9]+;
⟨string⟩ → “[a-zA-Z0-9]+”;
⟨comment⟩ → #.*⟨EOL⟩;

```

Figure 19: Grammar of EpiSimdemics scenario file.

PTTSs. Actions can also modify scenario variables. A trigger can only directly specify a single action to execute. Multiple actions can be combined into an *intervention*, which can be executed using the *apply* action. Interventions are also necessary to probabilistically execute one or more actions, as described below.

- (a) **Triggers:** A trigger condition is series of comparisons linked by boolean operators. Values that can be compared included: an interactor's demographics and attributes, attributes of the current state of any of the PTTSs, and the value of scenario variables, described in more detail below.

There are two types of triggers. The first is evaluated at the start of each simulation day by every interactor, and each after each transition of a PTTS by the transitioning interactor. The second is evaluated whenever a particular PTTS state is entered, or alternatively, exited. A trigger may have one or more modifiers, single, repeatable, and probabilistic evaluation.

**repeatable:** Normally, a trigger can only succeed once per interactor. A repeatable trigger can succeed multiple times.

**single:** Normally, a trigger is evaluated independently against each interactor. A single trigger is evaluated only once at the start of each day, and is independent of any interactor. A single trigger can also be repeatable and/or probabilistic.

**with prob:** When a probability is specified, it is added as another check in the condition. If the condition evaluates as true, a Bernoulli trial is performed. If it succeeds, the action is executed as normal. If it fails, the trigger is reevaluated at the next opportunity.

- (b) **Scenario Variables:** Scenario variables can be written (assigned, incremented, and decremented) and read in the scenario file. All scenario variables are initialized to zero (scalar) or empty (set). The read value is always the value at the end of the previous simulation day. Any writes are accumulated locally, and synchronized among processors at the end of each simulated day (cycle). Currently, scenario variables are limited to integer values.

There are two types of variables: scalar and sets. Scalar variables hold a single integer value while sets can hold multiple values. Various set operations are supported such as interscetion and union, as well as the adding and removing of individual values.

There are two read-only scenario variables: *day* and *time*. These variables hold the simulation day and simulation time at which the action was executed.

- (c) **Interactor and PTTS Attributes:** Each person has a set of attributes and demographics. Demographics are input to the simulation at initialization, while attributes are initialized to 0. Otherwise, they are treated identically and both are considered attributes. Each PTTS has a set of attribute names associated with it, with unique values in each state. The attribute values are fixed for the duration of the simulation. Both person and PTTS attributes can be used in trigger condition expressions. The value of the PTTS state attribute contains the full name of the state, in the form "*PTTS name:manifestation name:state name*", where the *PTTS name* comes from the configuration file, while the other two parts come from the PTTS input file. In this way, an interactor can have multiple copies of the same PTTS that will be treated independently.
- (d) **Actions:** Actions that can be performed by a trigger are detailed below.

**apply:** Apply an intervention, which is a group of one or more actions from this list. An intervention can be applied probabilisticly, in a way similar to triggers. The difference is that if the Bernoulli trial fails when applying an intervention, the trigger itself is still considered to have succeeded, and the trigger will not be evaluated again for this interactor (unless it is repeatable). In this way, a distinction can be made between a decision that is repeatedly evaluated, such as "On each day that a person is symptomatic, Go to

the doctor with a 30% probability”, versus one that should only be evaluated once “30% of the people who are symptomatic will be diagnosed.”

**treat, untreat:** Add or remove a label to the set of labels to be considered when choosing a set of edges to use for a transition in a PTTS. The set of labels must match the edge labels exactly for an edge to be considered. We are considering relaxing this restriction in the future.

**schedule, unschedule:** Add or remove a schedule from an interactors priority list. When building a daily schedule, the schedule type with the highest priority is used.

**infect, transition:** Cause a forced transition in a PTTS, as opposed to a timed transition. Infect is the same as a normal transition, with the added check that the PTTS is currently in an uninfected state. When a PTTS is forcibly transitioned, the new state can either be explicitly specified, or chosen as part of the normal transition process. The normal transition process is to select from the weighted edges that are part of the transition set with the correct label. The dwell time in the new state can be one of four possibilities:

**normal:** Pick the dwell time from the distribution in the new state. This is the default.

**keeptime:** Keep the transition time from the old state. The transition time from this state will be the same as from the old state, had the forced transition not occurred.

**fixed:** Pick the dwell time from the distribution in the new state, and subtract the amount of time already spent in the old state. If this results in a dwell time that is equal to or less than zero, perform another transition according to the normal transition rules (not yet implemented).

**proportional:** Pick the dwell time from the distribution in the new state, but reduce it by the percentage of the dwell time spent in the old state. So if the old dwell time was 48 hours, and the individual has already spent 36 hours in that state (75% of the total), and the new dwell time was 72 hours, the value of 18 hours is used (not yet implemented).

**remove:** Remove an interactor from the simulation. No further interactions will be evaluated for this interactor. This action takes effect at the beginning of the next simulation day.

**endsim:** End the simulation early. Useful for dynamically evaluating stopping criteria. This action takes effect at the beginning of the next simulation day. (currently broken and should not be used).

**message:** Write a message to the log file, tagged with the simulation time the action was taken, and the associated interactor.

**set:** Alter the value of a scalar scenario variable or interactor attribute. The value read from a scenario variable is always the value at the end of the previous simulation day. Any writes are accumulated locally, and synchronized among processors at the end of each simulated day.

**add, delete:** Modify the contents of a set scenario variable. Values added to or deleted from a set can either be constants, or the value of other scalar scenario variables.

**6.2. Automatically generated output files.** Upon running a simulation, certain outputs will be generated automatically in the directory Outfiles/ by each processor. Each output file is described below.

- (1) **disease.txt.A:** This file lists each time an individual changes states, the state which it enters, and the state which it exited in the following format:



< cell\_id > < time > < state\_entered > < state\_exited >

*time* is given in simulation seconds. The numbers representing each state is given at the top of the document.

- (2) **dendogram.txt.A:** This file lists each time an individual changes states, the individual's id (< *person* >), the time at which the transition occurred (< *time* >), the manifestation dictating the transition function (< *fsm* >), the state of the individual at the time of change (< *infecteeState* >), the id of the contact that induced the change (< *InfectedBy* >), the state of the contact that induced the change (< *InfecteeState* >), the location where the transition occurred (< *actloc* >), as well as information about the location that is not currently relevant (< *actsubloc* > < *acttype* > < *roomtype* >). The columns are labeled as follows:  
 < *person* > < *homeloc* > < *time* > < *fsm* > < *infecteeState* > < *infectedBy* > < *infecteeState* >  
 < *actloc* > < *actsubloc* > < *acttype* > < *roomtype* >

The numbers representing states correspond to those in the *disease.txt.A* file and the numbers representing the different fsm's are given at the top of the document.

- (3) **stat.txt:** The stat file lists, for each day (< *day* >), the number of individuals (< *count* >) in a specific state (< *value* >) in each Location (< *locationType* >) in the following format:  
 < *day* > < *locationType* > < *value* > < *count* >

The corresponding *value* for each state is given at the top the *disease.txt.A* file. A stat file is generated for each processor, hence each *stat.txt.\** file contains count for each state in a location as calculated over the sublocations on a single processor. As sublocations for a single Location are distributed among the processor the total count in a Location is a sum over all the *stat.txt.\** files generated.

- (4) **Trans files:** As described in section 3, certain individuals will transition states according to the *multicontact-dependent* function (Equation 4) depending on their current state. Different *multicontact-dependent* functions can be created in the configuration file (discussed below) to be applied to different sets of 'target' states. The Trans files report information about each multicontact-dependent transition that occurs during the simulation.

- (a) **Trans.txt:** This file reports, on each day (< *day* >), for each specified function (< *name* >), the number of individuals in states that induce the transition (< *activators* >), the number of individuals that inhibit the transition (< *inhibitors* >), as well as the number of individuals in one of the target states (< *targets* >) and whether the transition occurred (< *infected* >). The columns are labelled as follows:

< *day* > < *name* > < *start* > < *end* > < *activators* > < *inhibitors* > < *targets* > < *infected* >

- (b) **Transdetail.txt:** This file reports the same information as *Trans.txt* as well as the unique identifier of each individual in the target state < *id* > and the calculated probability of infection given the count of *activators* and *inhibitors* (< *probability* >):

< *day* > < *name* > < *start* > < *end* > < *id* > < *activator* > < *inhibitor* > < *target* >  
 < *prob* > < *infected* >

## REFERENCES

- [1] J. C. Arciero, G. B. Ermentrout, J. S. Upperman, Y. Vodovotz, and J. E. Rubin. Using a mathematical model to analyze the role of probiotics and inflammation in necrotizing enterocolitis. *PLoS One*, 5(4):e10066, 2010.
- [2] D. Artis. Epithelial-cell recognition of commensal bacteria and maintenance of immune homeostasis in the gut. *Nat Rev Immunol*, 8(6):411–20, 2008.
- [3] C. Barrett, K. Bisset, S. Eubank, X. Feng, and M. Marathe. Episimdemics: an efficient algorithm for simulating the spread of infectious disease over large realistic social networks, 2008.
- [4] Perelson A. Bauer A., Beauchemin C. Agent-based modeling in host-pathogen systems: The successes and challenges. *Information Sciences*, 179:1379–1389, 2009.
- [5] M. J. Blaser and D. Kirschner. Dynamics of helicobacter pylori colonization in relation to the host response. *Proc Natl Acad Sci U S A*, 96(15):8359–64, 1999.
- [6] M. J. Blaser and D. Kirschner. The equilibria that allow bacterial persistence in human hosts. *Nature*, 449(7164):843–9, 2007.
- [7] Cohen I.R. Efroni S., Harel D. Reactive animation: Realistic modeling of complex dynamic systems. *Computer*, pages 38–47, 2005.
- [8] D. Harel Gery and E. Executable object modeling with statecharts. *Computer*, 30(7):31–42, 1997.
- [9] S. Gordon and P. R. Taylor. Monocyte and macrophage heterogeneity. *Nat Rev Immunol*, 5(12):953–64, 2005.
- [10] A. T. Haase. Population biology of hiv-1 infection: Viral and cd4(+) t cell demographics and dynamics in lymphatic tissues. *Annual Review of Immunology*, 17:625–656, 1999.
- [11] D. A. Hill and D. Artis. Intestinal bacteria and the regulation of immune cell homeostasis. *Annu Rev Immunol*, 28:623–67, 2010.
- [12] R. Hontecillas and J. Bassaganya-Riera. Peroxisome proliferator-activated receptor gamma is required for regulatory cd4+ t cell-mediated protection against colitis. *J Immunol*, 178(5):2940–9, 2007.
- [13] A. Iwasaki. Mucosal dendritic cells. *Annual Review of Immunology*, 25:381–418, 2007.
- [14] R. Jonasson, M. Andersson, T. Rasback, A. Johannisson, and M. Jensen-Waern. Immunological alterations during the clinical and recovery phases of experimental swine dysentery. *J Med Microbiol*, 55(Pt 7):845–55, 2006.
- [15] S. M. Kaech and R. Ahmed. Immunology: CD8 T cells remember with a little help. *Science*, 300(5617):263–5, 2003.
- [16] D. Kelly, J. I. Campbell, T. P. King, G. Grant, E. A. Jansson, A. G. Coutts, S. Pettersson, and S. Conway. Commensal anaerobic gut bacteria attenuate inflammation by regulating nuclear-cytoplasmic shuttling of ppar-gamma and rela. *Nat Immunol*, 5(1):104–12, 2004.
- [17] A. Lanzavecchia and F. Sallusto. Regulation of t cell immunity by dendritic cells. *Cell*, 106(3):263–6, 2001.
- [18] D. R. Littman and A. Y. Rudensky. Th17 and regulatory t cells in mediating and restraining inflammation. *Cell*, 140(6):845–58, 2010.
- [19] A. L. Marzo, K. D. Klonowski, A. Le Bon, P. Borrow, D. F. Tough, and L. Lefrancois. Initial t cell frequency dictates memory cd8+ t cell lineage commitment. *Nat Immunol*, 6(8):793–9, 2005.
- [20] R. Naresh, Y. Song, and D. J. Hampson. The intestinal spirochete brachyspira pilosicoli attaches to cultured caco-2 cells and induces pathological changes. *PLoS One*, 4(12):e8352, 2009.
- [21] S. C. Ng, M. A. Kamm, A. J. Stagg, and S. C. Knight. Intestinal dendritic cells: their role in bacterial recognition, lymphocyte homing, and intestinal inflammation. *Inflamm Bowel Dis*, 16(10):1787–807, 2010.
- [22] Y. Onishi, Z. Fehervari, T. Yamaguchi, and S. Sakaguchi. Foxp3(+) natural regulatory t cells preferentially form aggregates on dendritic cells in vitro and actively inhibit their maturation. *Proceedings of the National Academy of Sciences of the United States of America*, 105(29):10113–10118, 2008. 330BB Times Cited:23 Cited References Count:39.
- [23] F. Sallusto, J. Geginat, and A. Lanzavecchia. Central memory and effector memory t cell subsets: function, generation, and maintenance. *Annu Rev Immunol*, 22:745–63, 2004.
- [24] E. M. Shevach, R. A. DiPaolo, J. Andersson, D. M. Zhao, G. L. Stephens, and A. M. Thornton. The lifestyle of naturally occurring cd4+ cd25+ foxp3+ regulatory t cells. *Immunol Rev*, 212:60–73, 2006.
- [25] M. W. Sneddon, J. R. Faeder, and T. Emonet. Efficient modeling, simulation and coarse-graining of biological complexity with nfsim. *Nature Methods*, 8(2):177–U112, 2011.
- [26] S. Stoll, J. Delon, T. M. Brotz, and R. N. Germain. Dynamic imaging of t cell-dendritic cell interactions in lymph nodes. *Science*, 296(5574):1873–6, 2002.
- [27] T. Sutterlin, S. Huber, H. Dickhaus, and N. Grabe. Modeling multi-cellular behavior in epidermal tissue homeostasis via finite state machines in multi-agent systems. *Bioinformatics*, 25(16):2057–2063, 2009.
- [28] N. Swerdlin, I. R. Cohen, and D. Harel. The lymph node b cell immune response: Dynamic analysis in-silico. *Proceedings of the IEEE*, 96(8):1421–1443, 2008. 329IH Times Cited:2 Cited References Count:56.

- [29] K.V. Wendelsdorf, J. Bassaganya-Riera, R. Hontecillas, and Eubank S. Model of colonic inflammation: Immune modulatory mechanisms in inflammatory bowel disease. *Journal of Theoretical Biology*, 2010. Article in Press, doi: 10.1016/j.jtbi.2010.03.027.
- [30] J. E. Wigginton and D. Kirschner. A model to predict cell-mediated immune regulatory mechanisms during human infection with mycobacterium tuberculosis. *J Immunol*, 166(3):1951–67, 2001.