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

Katherine Wendelsdorf[1], 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 $H.pylori$.

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

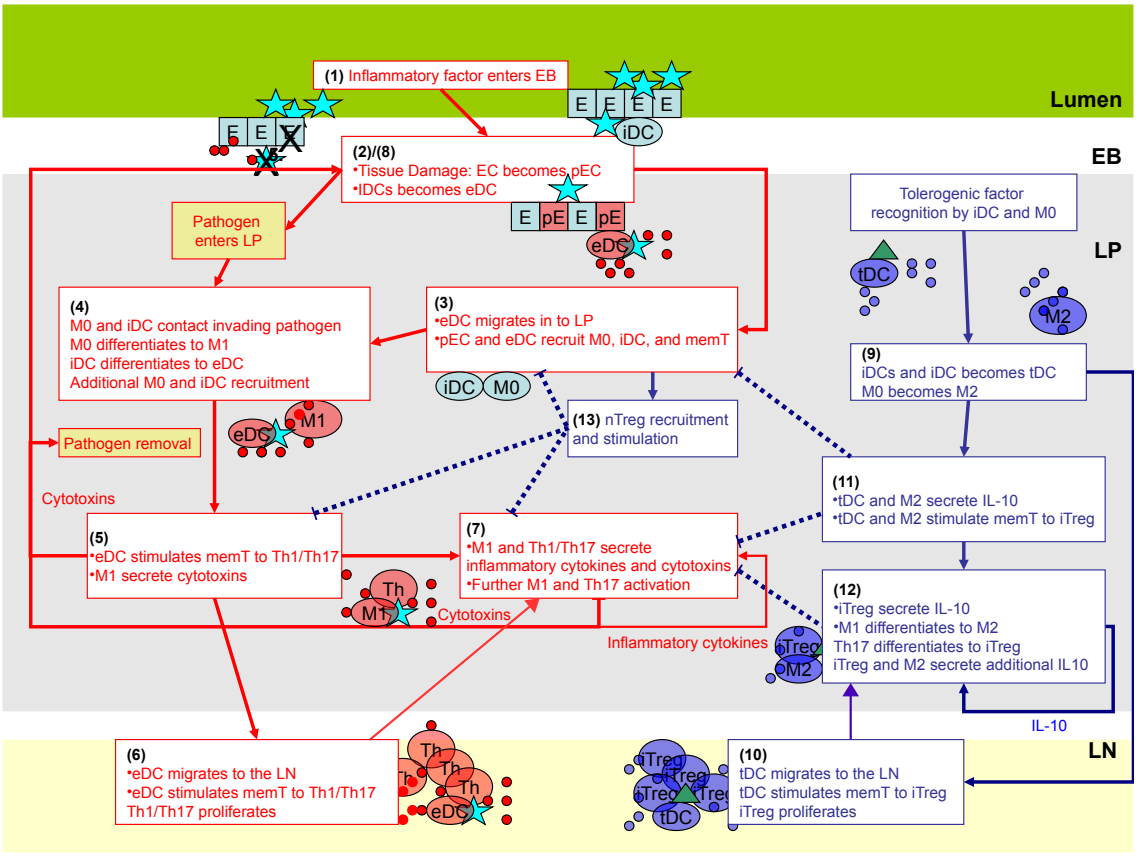


Figure 1: Illustration of inflammatory and regulatory pathways described in section 1.2.

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 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].

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

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 occurence 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, 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 and 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, is the entry site for ingested food and foreign microbes, and houses gut microflora, *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), a monolayer of epithelial cells, that divides the lumen and LP, *iv*) the gastric or mesenteric lymph node (LN), the primary site of T cell activation.

The events of inflammatory immune pathways (red lines) are as follows with numbers corresponding to those labeling events depicted in Figure 1: **(1)** A pathogen, such as foreign bacteria, enters the lumen and contacts the EB. **(2)** Intestinal epithelial cells (EC) differentiate to a pro-inflammatory phenotype (pEC) either in response to damage caused by the pathogen or by mere recognition of the pathogen. The pEC phenotype secretes microbicides and various signalling chemicals (cytokines) and may be permeable allowing pathogen entry in to the LP [11]. At the same time, immature 'sampling' dendritic cells (iDCs), which reside in the EB and contact microbes in the lumen, internalize the pathogen and mature to an effector phenotype (eDC) that migrates in to the LP and presents components of the pathogen (antigen) on its surface **(3)** Chemicals secreted by damaged epithelial cells and eDC recruit resting macrophages (M0) and dendritic cells (iDC). **(4)** These macrophages and dendritic cells may then contact and internalize pathogen that has entered the LP. This leads to maturation of iDC to an eDC phenotype and differentiation of macrophages to an inflammatory M1 phenotype. **(5)** Mature, antigen-presenting eDC go on to recruit resting CD4+ T cells (restingT) to the LP and secrete cytokines such as IL-12 and IFN-γ that induce T cell differentiation to a pro-inflammatory Th1 or Th17 phenotype (Th) upon antigen recognition [13]. These stimulated T cells enter a transient state of proliferation, each giving rise to approximately 500 daughter cells of the same phenotype. M1 go on to secrete secrete cytotoxic proteases and radicals that kill surrounding microbes as well host epithelial cells. **(6)** Presenting effector dendritic cells then migrate to the LN and to contact resting memory and neive T cells, stimulating them to a Th1 or Th17 phenotype. The inflammatory T cells mature here before migrating to the LP site to contribute to the inflammatory response at the infection site .**(7)** At the infection site in the LP Th1/Th17 cells secrete cytotoxins and cytokines that enhance secretion of inflammatory factors by surrounding T cells as well as induce additional macrophages to a M1 phenotype and T cells to a Th17 phenotype **(8)** The epithelial cells damaged by Th1/Th17 and M1 secreted factors then respond by secreting additional inflammatory cytokines resulting in more immune cell recruitment along with openings in the epithelial barrier that can allow direct pathogen entry into the LP at which point recruited M0 and iDC are activated to inflammatory phenotypes. This completes a positive, inflammatory feedback loop. Inflammation generally dissipates when pathogen is eliminated and direct immune cell stimulation ceases.

The anti-inflammatory, regulatory pathway is composed of the following events: Dendritic cells and macrophages contact a tolerogenic factor such as self-antigen or commensal bacteria that reside in the lumen or has invaded the LP through the damaged epithelium. **(9)** Upon internalization of the commensal or self-antigen,

dendritic cells differentiate to a tolerogenic phenotype (tDC) and macrophages differentiate to an M2 phenotype. **(10)** Like eDC, tDC migrates to the LN where it contacts and stimulates T cells to an iTreg phenotype, which go on to the infection site in the LP . **(11)**M2 and tDC go on secrete the anti-inflammatory cytokine IL-10. IL-10 reduces inflammatory cytokine and cytoxin production in surrounding immune cells, dampening the inflammatory loop. They also present the tolerance-inducing antigen to CD4+ T cells inducing their differentiation to T regulatory cells (iTreg). **(12)** iTreg goes on to secrete additional IL-10. The increase in environmental IL-10 induces differentiation of macrophages already of an inflammatory M1 phenotype to the regulatory M2 phenotype and T cells of the Th17 phenotype to an iTreg phenotype [9]. This promotes further IL-10 production and stimulation of T cells to iTreg closing a positive anti-inflammatory feedback loop.

(13) 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) or macrophage. nTreg may have a reduced proliferation capacity compared to conventional T cells. 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].

Certain genetic predispositions or immune dysfunctions can result in an inflammatory pathway being initiated by commensal bacteria strains [21].

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 **tolerogenic bacteria**, a **commensal bacteria**, an **inflammatory 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 and migrate among three locations; lumen, LP, and LN creating a contact network in which neighbors may or may not interact depending on their *state*. Immune response is modeled as state changes resulting from interactions and subsequent update of rules for individual movement and behavior.

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 **tolerogenic bacteria**, **commensal bacteria**, and **inflammatory bacteria** that mimick those attributed to experimental strains. 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. 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 $\xi(t)$ of a GDS is a vector $(s_1(t), s_2(t), ..., s_n(t))$, 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 with in the pathway. The cells are represented as a set of finite state automata $< c_1 ... c_n >$ corresponding to one of eight cell-types (epithelial cell, tolerogenic bacteria, commensal bacteria, foreign bacteria, macrophage, dendritic cell, 'sampling' dendritic cell, T cell, or nTreg). Individual cells move among tissue sites, *Locations*, that are divided in to discrete patches, the *sublocations*. 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. 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 as well as the amount of time c_i has occupied its current state and 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 c_j of individual c_i is in a state $s_j \in I_{si}$, c_i may interact with c_j and possibly transition states. Whether transition actually occurs upon interaction is probabilistic. Upon transition, the state which c_i enters depends on s_j as specified by the transition function of c_i, $f_i \in F$. Here, F is represented as the

set of cell-specific automata that are probabilistic timed transition systems. The function for the interaction probability may be single contact-dependent, calculated in a pairwise manner, or multicontact-dependent, a function of the configuration ξ_g of the subnetwork g in the specific sublocation occupied by c_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 $restingT \rightarrow Th1$ 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 $s_i \rightarrow x$, where $x \in S$. For example, tissue damage occurs when one cell represented by an **epithelial cell** automaton makes the transition $EC \rightarrow pEcell$. A specific *health outcome* is a stable configuration of the system, ξ_s, following contact between one of the bacteria automata and one of the immune cell automata.

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 **tolerogenic bacteria** or **commensal bacteria** in the lumen, *i.e.* the *TolB_lumen* or *CommB_lumen* states, 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 *TolB_LP* state that are phagocytosed to transition to the *TolB_lumen* state, replenishing the lumen population.

ii) Cell 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 *TSource* state undergo a state transition to *M*0, *iDC*, or *restingT* 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: *MASource*, *DCSource* and *TSource* states represent cell that are initially in the blood and do not interact with cells in the LP until they are 'recruited' to the tissue site by individuals in the *eDC*, $M1$, *eDCLumen*, and *pECell* states. In the model, cells occupying *MASource*, *DCSource* or *TSource* states are present in the LP, but do not interact with other cells. Upon contact with cells in one of the recruiting states *eDC*, $M1$, *eDCLumen*, and *pECell* the contacted monocyte or memory T cell transitions to an $M0$, *iDC*, or *restingT* state. 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 in each sublocation of the LP. This is set by the number of individuals assigned to the *MASource*, *DCSource*, and *TSource* states at the start of simulation and is, by default, $100x$ the initial number of resting $M0$, *iDCLP*, and $restingT$.

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 inactive states *ThSource* or *iTregSource* and are assigned to the LP and LN locations. When an individual is in one of the proliferating states, $Th1Prolif$, $Th17Prolif$ or $iTregProlif$, it may interact with and induce the source cells to its corresponding phenotype. For example, when an individual enters the $Th1Prolif$ state it contacts individuals in the $Th1Source$ state. Contacted individuals then transition from $Th1Source \rightarrow Th1$. Proliferation by individuals in the $iTregProlif$ and $Th17Prolif$ states are represented in the same manner. Hence, the average number of daughter cells from one proliferating T cell in the LP p_T or the number from a T cell in the LN p_t (Table 2), is determined by the average number of $Th1Source$ a $Th1Prolif$ contacts and is set by the initial number of individuals in the $Th1Source$, $Th17Source$ or *iTregSource* states. This is, by default $500x$ the initial number in the resting T cell state $restingT$.

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 $Th1Source$, $Th17Source$, and $iTregSource$ population pools.

2.4. **State transition functions.** In this section we describe the state transition functions and conditions for schedule changes (inter-tissue migration). State transition conditions of each of the eight cell-type specific automata are depicted in a state chart like formalism in Figures 3-9(a) [8]. In these depictions, red arrows indicate transitions that represent events in the inflammatory pathway and blue arrows indicate transitions that represent those of the regulatory pathway. 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 initially assigned. Parameters are listed in Table 2. Parameters are listed in Table 2 and are referenced in the following descriptions.

Table 1: Model States

State	Description	Initial Number Individuals
	Phenotypes	
$restingT$	Resting conventional CD4+ T cell	$1 \cdot 10^3$
$Th1$	Active CD4+ T helper cell 1	0
$Th17$	Active CD4+ T helper cell 17	0
$iTreg$	Induced T regulatory cell	0
$nTreg$	Active natural T regulatory cell	0
$resting\_nTreg$	Resting natural T regulatory cell	0
$iDCs$	Immature 'sampling' DC in the superficial LP with access to the Lumen	1000
$tDCL$	Tolerogenic sDC in the lumen	0
$eDCL$	Effector sDC in the lumen	0
$iDCLP$	Immature dendritic cell in the LP	1000
eDC	Effector dendritic cell in the LP	0
tDC	Tolerogenic dendritic cell in the LP	0
$DCAnergic$	Anergic dendritic cell	0
$M0$	Undifferentiated macrophage	$1 \cdot 10^3$
$M1$	Activated inflammatory macrophage	0
$M2$	Activated regulatory macrophage	0
EC	Healthy epithelial cell	10^5[17]
$pEcell$	Damaged or pro-inflammatory epithelial cell	0
$pEcell\_noR$	Dysfunctional pro-inflammatory epithelial cell that does not recruit	0
$MASource$	monocytes: MA precursor	10^5
$DCSource$	monocytes:DC precursor	10^5
$TSource$	resting T cell in blood	10^4
$Th1Source$	Potential child cell from a proliferating Th1	$5 \cdot 10^5$
$Th17Source$	Potential child cell from a proliferating Th17	$5 \cdot 10^5$
$iTregSource$	Potential child cell from a proliferating iTreg	$5 \cdot 10^5$
	Locations	
$CommB\_lumen$	Commensal bacterium in the lumen	50
$InfB\_lumen$	Inflammatory bacterium in the lumen	30
$TolB\_lumen$	Tolerogenic bacterium in the lumen	1000
$CommB\_LP$	Commensal bacterium in the LP	0
$InfB\_LP$	Inflammatory bacterium in the LP	0
$TolB\_LP$	Tolerogenic bacterium in the LP	0
	Death	
$Edead$	Killed epithelial cell	0
$CommB\_dead$	Killed commensal bacterium	0
$InfB\_dead$	Killed inflammatory bacterium	0
$TolB\_dead$	Killed tolerogenic bacterium	0

Schedule changes are depicted for cell-types that migrate in Figures 4(b), 4(d), 4(f), 6(b), 7(b), 8(b), and 9(b). In these figures dashed arrows represent schedule reassignment triggered by entry in to a specific state and are labelled with the specific state that triggers the reassignment. Solid arrows represent schedule reassignments that occur after specified time period in a state. These arrows are labelled by the state which must be occupied and, in parenthesis, the amount of time in the state at which reassignment occurs.

Epithelial Cells (Figure 3): This population composes the epithelial barrier (EB). Each epithelial cell is assigned simultaneously to the *LP* and *Lumen* locations representing its status as a barrier. These cells are

Table 2: Parameter values

Symbol	Parameter	DefaultValue
	Birth/death	
μ_E	Turnover time of epithelial cells	$12hrs$[17]
μ_T	Time a T cell remains active	7days [20]
μ_{M0}	Time a macrophage remains active	indefinite
μ_d	Time a dendritic cell remains active	1 day [18]
μ_{ce}	Probability that $pEcell$ is killed by inflammatory factors	
p_T	Average number of daughter cells produced by a proliferating T cell in the LP	500 [23]
p_t	Average number of daughter cells produced by a proliferating T cell in the LN	500 [23]
	Migration	
ε_r	Average number of monocytes recruited by a single eDC, $M1$, or $pEcell$	10
ε_t	Average number of resting T cells recruited by a single eDC, $M1$, or $pEcell$	10
β_p	Probability that bacteria will enter the lumen upon contact with a $pEcell$	1
ε_{IE}	Time active T cell remains in the LN before migrating to LP	2 days
ε_{EI}	Time presenting DC remains in the LP before migrating to LN	1 day
	Contact/interactions	
α_T	Probability of resting T cell stimulation	1
p_{17}	Probability of resting T cell stimulation to $Th17$ vs. $Th1$ by eDC or $M1$	0.5
α_{nTreg}	Probability of resting nTreg stimulation	1
ν_T	fraction of active T cells that become memory T cells	0.1 [16]
ν_{12}	Probability that M_1 switches to M_2	$\frac{a_1R}{a_1R+i_1N})^{y_1}$
ν_{21}	Probability that M_2 switches to M_1	$\frac{a_2N}{i_2Ra_2N})^{y_2}$
a_1	co-efficient of ν_{12} for activators	1
i_1	co-efficient of ν_{12} for inhibitors	1
y_1	exponent of ν_{12}	4
a_2	co-efficient of ν_{21} for activators	1
i_2	co-efficient of ν_{21} for inhibitors	4
y_2	exponent of ν_{21}	
ν_{r17}	Probability $iTreg \rightarrow Th17$	$\frac{a_rN}{i_rR+a_rN})^{y_r}$
ν_{17r}	Probability $Th17 \rightarrow iTreg$	$\frac{a_{17}R}{a_{17}R+i_{17}N})^{y_{17}}$
a_r	co-efficient of ν_{r17} for activators	1
i_r	co-efficient of ν_{r17} for inhibitors	1
y_r	exponent of ν_{r17}	4
a_{17}	co-efficient of ν_{17r} for activators	1
i_{17}	co-efficient of ν_{17r} for inhibitors	4
y_{17}	exponent of ν_{17r}	
ν_{BM}	probability that commensal bacteria induces inflammatory phenotype in macrophages	0
ν_{BD}	probability that commensal bacteria induces inflammatory phenotype in dendritic cells	0
ν_{Bs}	probability that commensal bacteria induces inflammatory phenotype in 'sampling' dendritic cells	0
ν_{EC}	Probability that EC transitions to $pEcell$ upon contact with inflammatory factors	
ν_{EB}	Probability that EC is damaged by microbial toxins	0.05
β_r	Ability of commensal or inflammatory bacteria to induce chemoattractant expression in epithelial cells	1
β_c	Ability of pEcell to secrete cytokines that induce $M2 \rightarrow M1$ and $iTreg \rightarrow Th17$	1
β_d	Amount of microbicide secreted by $pECell$, pEC_noR in response to commensal or inflammatory bacteria	
μ_{M1}	Ability of M1 to eliminate bacteria	

static and do not change sublocations nor migrate to other tissues. Initially in the healthy EC state, the cell transitions to a damaged, pro-inflammatory state, $pEcell$, with the probability of ν_{EC} upon contact with

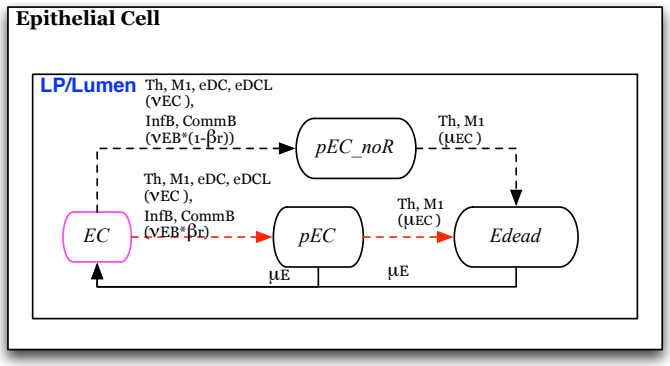


Figure 3: Automata for epithelial cells; individuals initially assigned the EC phenotype

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-κB 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 inflammatory or commensal bacteria that can induce epithelial damage with a probability of ν_{EB}, which will be specific to the bacterial strains that **inflammatory bacteria** and **commensal bacteria** are meant to represent. 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 these inflammatory cells. In the $Edead$ state epithelial cells no longer secrete pro-inflammatory cytokines. From the $pEcell$ and $Edead$ states, the epithelial cell transitions 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 [17].

Upon contact with bacteria, epithelial cells have been shown to secrete chemoattractants that recruit other immune cells along with inflammatory cytokines [11, 2]. However, it has been shown that some bacterial strains, such as *H. pylori*, are able to inhibit this function. The parameter β_r is the probability that **inflammatory bacteria** and **commensal bacteria** are strains that induce chemoattractant secretion. Alternatively, $1-\beta_r$ is the probability that they are strains which inhibit chemokine secretion, yet may still induce inflammaotry cytokine secretion or permeability in the epithelial barrier. This is represented by a transition to the impaired pEC_noR state as opposed to the fully functioning $pECell$ state, where cells in the pEC_noR state do not recruit resting immune cells to the infection site.

Bacteria (Figure 4): The model only represents individual bacterium that are in direct contact with the epithelial barrier (EB). There are three general types of bacteria represented: A **tolerogenic bacteria** that induces regulatory tDC and M2 phenotypes in antigen-presenting cells, an **inflammatory bacteria** that represents a general pathogen and induces an inflammatory M1 or eDC phenotype in antigen-presenting cells, a general **commensal bacteria** that may have inflammatory and tolerogenic properties depending on user specifications. These may represent different strains present in the microbiota and are described below. Each may represent a single species of interest or a composite population with intrapopulation characteristics implicitly represented by parameter values. In general, all bacteria are initially present in the lumen. From the lumen, individuals may migrate to the *LP* represented by a state transition upon contact with *Edead* or *pEcell* that triggers a schedule reassignment to the LP (Figures 4(b) 4(d) 4(f)). These transitions represent contact

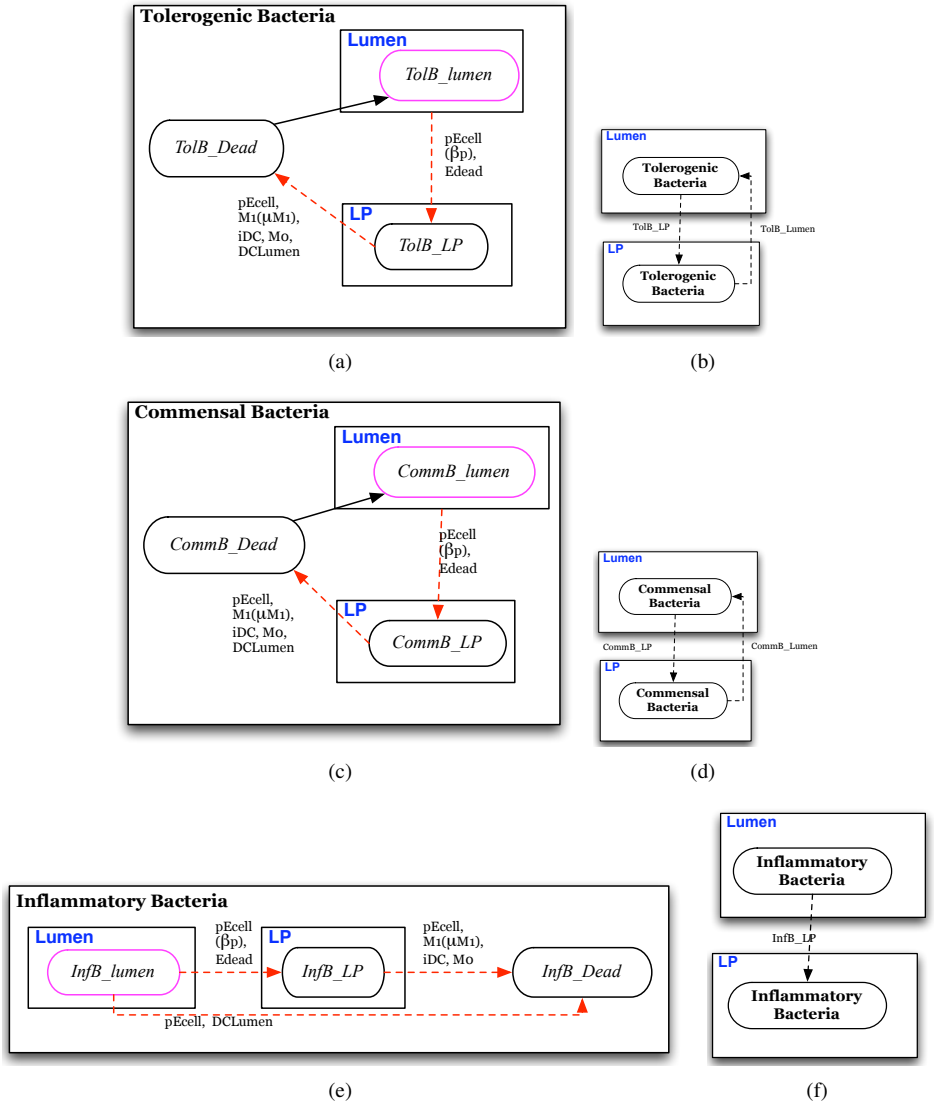


Figure 4: **??** Automata for commensal bacteria; individuals initially assigned the TolB_lumen phenotype (b) Conditions for schedule reassignment of **tolerogenic bacteria** (c) Automata for commensal bacteria; individuals initially assigned the B_lumen phenotype (d) Conditions for schedule reassignment of **Commensal Bacteria** (e) Automata for foreign bacteria (pathogen); individuals initially assigned the Bf_lumen phenotype (f) Conditions for schedule reassignment of **foreign bacteria**

with either an epithelial cell made permeable by cytokine-induced damage [2] or a hole in the epithelial barrier due to epithelial death. 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*, *iDCs*) upon contact.

(1) **Tolerogenic Bacteria (Figure 4(a))** Tolerogenic Bacteria represents constitutively present tolerance inducing factors including some commensal bacterial strains in the microbiota. As it represents a strain of the microbiota, we assume that the true lumen population is much larger than the subpopulation localized to the epithelial barrier represented here. Operating under this assumption, shortly after entry in to the *TolB_Dead* state (representing death), the individual re-enters the *TolB_lumen* state to replenish the pool of commensal bacteria in the lumen which the model assumes to be unlimited.

(2) **Inflammatory Bacteria (Figure 4(e))** Inflammatory bacteria represents invading pathogen that is not assumed to be constitutively present in the microbiota. These individuals carry out the following functions: *i*) induce an inflammatory phenotype in dendritic cells and macrophages, *ii*) damage epithelial cells leading to permeability of EB *iii*) induce epithelial cells to secrete recruiting chemokines in response to damage, *iv*) induce epithelial cells to secrete inflammatory factors that drive macrophage and T cell differentiation to M1 and Th17 phenotypes respectively, *v*) induce epithelial cells to secrete microbicides. Like tolerogenic bacteria and commensal bacteria of the microbiota, inflammatory bacteria in the *LP* may be eliminated upon contact with microbicide-secreting epithelial cells, and cytotoxin secreting Th and $M1$ and internalization by *iDCLP* or $M0$. However, it is not replaced once killed.

(3) **Commensal Bacteria (Figure 4(c))** This automaton represents constitutively present strains of the microbiota that may trigger tolerogenic as well as inflammatory response in different cell-types. As it represents a strain of the microbiota, we assume that, like for **tolerogenic bacteria**, the true lumen population of commensal bacteria is much larger than the subpopulation localized to the epithelial barrier represented here. Hence, like **tolerogenic bacteria**, shortly after entry in to the *CommB_Dead* state, the individual re-enters the *CommB_lumen* state to replenish the pool of commensal bacteria in the lumen which the model assumes to be unlimited.

Macrophages (Figure 5): Macrophages occupy the *LP* where they move randomly and are initially in a resting $M0$ state. Individuals transition from the $M0$ state to $M1$ when in contact with inflammatory bacteria in the LP and to $M2$ when in contact with tolerogenic bacteria. Upon contact with neighbors that are **commensal bacteria**, $M0$ transitions to $M1$ with a probability of ν_{BM}, representing the percent of microbiota that is recognized as foreign and $M2$ state with the probability of $1-\nu_{BM}$. 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 ν_{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 ν_{12} increases, and y_1 is a constant that determines the rate at which ν_{12} increases with the $R:N$ ratio. M2 switches to M1 with the probability ν_{21} (Equation 2), where a_2, i_2, and y_2 are constants that may differ from a_1, i_1, and y_1.

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

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

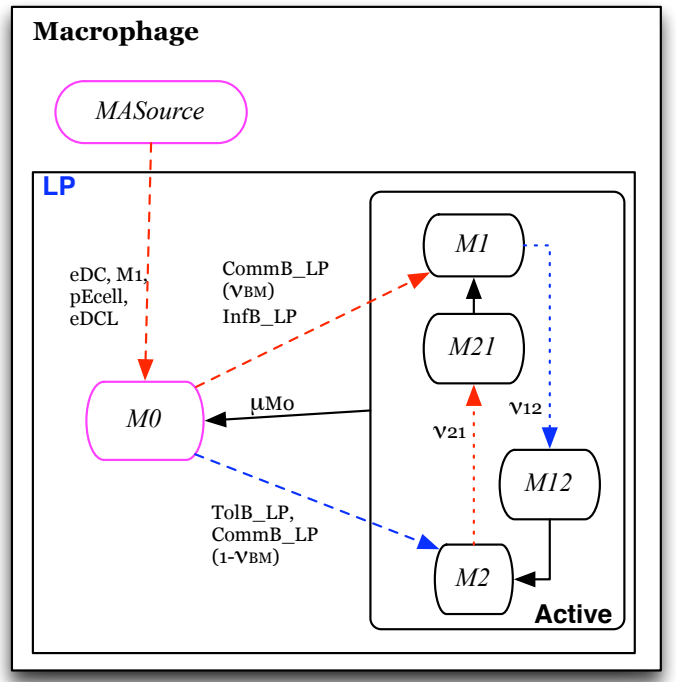
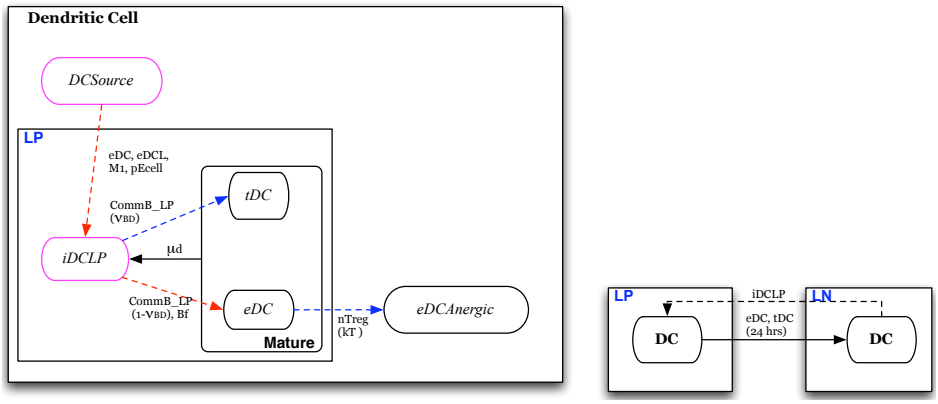


Figure 5: Automata for macrophages; individuals initially assigned the M0 or MASource phenotype

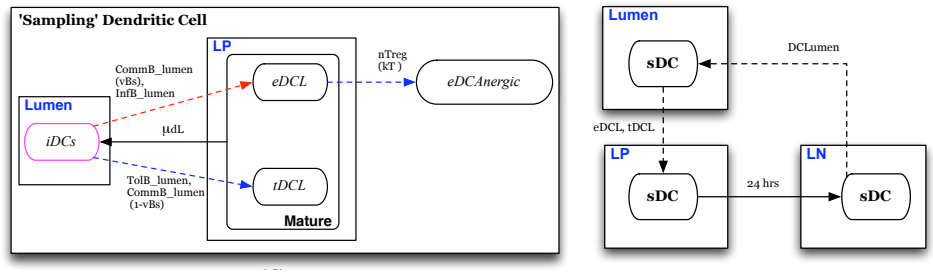
In the case of no cytokine stimulus, *i.e.* no contact with individuals in either inflammatory or regulatory states, an activated macrophage will revert back to a resting state after a specified period of time, μ_{M0}. To represent monocyte recruitment, individuals initially in the inactive *MASource* state transitions to $M0$ when in contact with individuals in the $pEcell$, $M1$, or eDC states.

Dendritic Cells (Figure 6): Dendritic cells are initially in a resting, immature state in the LP (*iDCLP*) (Figure 6(a)). These may be referred to as LP dendritic cells and are distinct from the 'sampling' dendritic cells associated with the EB [21, 13]. The state transition model is similar to that of macrophages. An individual in the *iDCLP* state transitions to an inflammatory effector DC (*eDC*) upon contact with commensal bacteria with the probability of ν_{BD} or when in contact with inflammatory bacteria. Alternatively, it transitions to a tolerogenic DC state (*tDC*) when in contact with commensal bacteria with a probability of $1 - \nu_{BD}$ or when in contact with tolerogenic bacteria. In the mature states, *tDC* or *eDC*, the dendritic cell remains in the *LP* for 24 hour before migrating to the *LN* (Figure 6(b)). The individual remains in one of these mature states for a time period μ_d before dying [18]. 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 (Figure 6(a)). Upon transition to the *iDCLP* state, it is reassigned back to the *LP* (Figure 6(b)). 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]. To represent recruitment, *DCSource* transitions to *iDC* when in contact with individuals in the $pEcell$, $M1$, or eDC states.



(a) (b)

Figure 6: (a) Automata for dendritic cells; individuals initially assigned the iDC or DCSource phenotype. (b) Conditions for schedule reassignment of **dendritic cells**

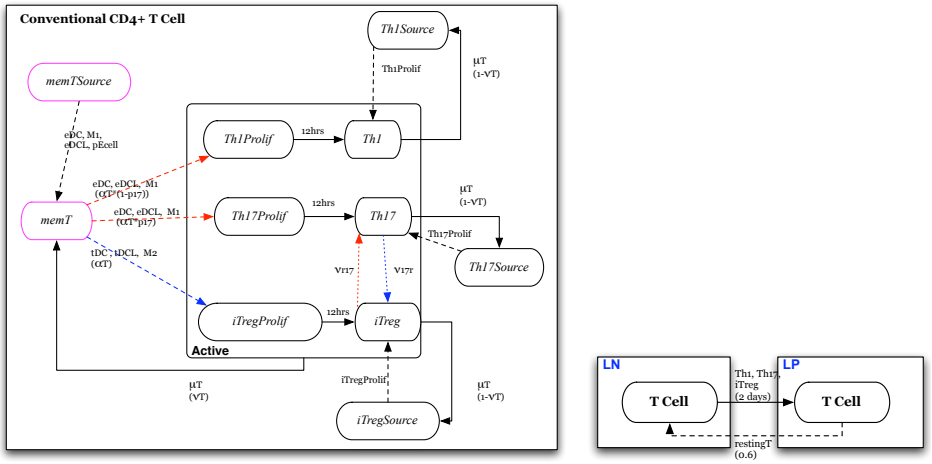


(a) (b)

Figure 7: (a) Automata for commensal bacteria; individuals initially assigned the DCLumen phenotype (b) Conditions for schedule reassignment of **'sampling dendritic cells'**

'Sampling' Dendritic Cells (Figure 7): 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 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 state transition path from *iDCLumen* is similar to LP dendritic cells. Those in the mature states *tDCL* and *eDCL* are assigned to the *LP* (Figure 7(b)) 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. As with LP dendritic cells, 'sampling' dendritic cells in the mature state are assigned the *LN* after 24 hours. Upon re-entry in to the *DCLumen* state, they are reassigned to the lumen.

Conventional CD4+ T cells (Figure 8): To conserve the number of individuals, the model only repre-



(a) (b)

Figure 8: (a) Automata for commensal bacteria; individuals initially assigned the restingT or TSource phenotype (b) Conditions for schedule reassignment of **conventional CD4+ T cells**

sents T cells that specifically recognize and are stimulated by products of the strains represented by the bacteria automata in the model. A T cell is initially in a *restingT* state that represents either a naive or memory T cell (Figure 8(a)). This resting T cell population is divided among the LP and LN Locations with the majority, 60%, in the LN. The state *TSource* are resting T cell precursors and transition the to *restingT* state upon contact with chemoattractant secreting individuals in the *eDC*, *eDCL*, *M*1, and *pECell* states. An individual in the *restingT* state will transition to an active inflammatory Th1 or Th17, when in contact with neighbors in the *eDC*, *eDCL*, or *M*1 state. Whether it becomes a Th1 or Th17 is determined by the parameter p_{17} which is specified by the user according to the bacterial strain they wish to represent with **inflammatory bacteria** and **commensal bacteria**. Alternatively, it may transition to an active iTreg when in contact with neighbors in the *tDC*, *tDCL* or *M*2 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 [19]. 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, $Th1Prolif$, $Th17Prolif$ or $iTregProlif$, for approximately 12 hours [26]. In this state the cell can induce transition of source cells to a $Th1$, $Th17$ or $iTreg$ state giving rise to a new population of active T cells of its same phenotype as described in section 2.3. The value p_T is the average number of children T cells produced by one proliferating T cell in the LP, presumably from a stimulated effector memory T cell, and p_t is the average number of children T cells produced by one proliferating T cell in the LN, presumably from a stimulated central memory T cell or naive T cell . From this state, the individual T cell transitions to a non-proliferating active state, $Th1$, $Th17$ or $iTreg$. T cells that are activated in the LN are reassigned to the LP after a maturation period of 2 days [14]. Th17 has been shown to differentiate in to iTreg and iTreg in to Th17 depending on the cytokine environment [19]. In the mode, $Th17$ may differentiate to $iTreg$ with the probability of v_{17r} that is a function of the infammatory and

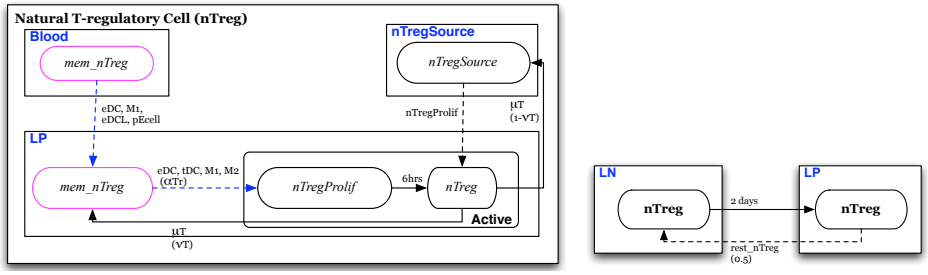
regulatory cells in the local environment as shown in Equation 3, where R is the total number of contacts that are of a regulatory phenotype: tDC, $M2$, $iTreg$, $nTreg$ and N is the number of contacts that are of a inflammatory phenotype: eDC, $M1$, Th, $pEcell$, a_{17} and i_{17} are constants that determine the threshold for the $R:N$ ratio at which ν_{17r} increases, and y_{17} is a constant that determines the rate at which ν_{17r} increases with the $N:R$ ratio. iTreg switches to Th17 with the probability ν_{r17} (Equation 4), where a_r, i_r, and y_r are constants that may differ from a_{17}, i_{17}, and y_{17}.

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. Memory T cells remain the LP with a probability of 0.4, representing effector memory T cells, or the LN with a probability of 0.6, representing central memory T cell. This fraction is based on a study in mice that showed 60% of CD8+ memory T-cells created in response to lymphocytic choriomeningitis virus were $CD62L$+, a marker for central memory versus effector memory T-cells [20]. Such *in vivo* data was not found for CD4+ T-cells specifically. The rest undergo programmed cell death represented by reversion to its associated source state, $Th1Source$, $Th17Source$ or $iTregSource$, which are equally distributed among the proliferation sites, the LP and LN (Figure 8(b)).

$$p(Th17 \rightarrow iTreg) = \nu_{17r} = \left(\frac{a_{17}R}{a_{17}R + i_{17}N}\right)^{y_{17}} \tag{3}$$

$$p(iTreg \rightarrow Th17) = \nu_{r17} = \left(\frac{a_rN}{i_rR + a_rN}\right)^{y_r} \tag{4}$$

Natural T-regulatory Cells (nTreg) (Figure 9): Natural T-regulatory cells follow a very similar path to



(a) (b)

Figure 9: (a) Automata for commensal bacteria; individuals initially assigned the restingT_nTreg or restingT_nTregSource phenotype (b) Conditions for schedule reassignment of **nTreg**

conventional T cells (Figure 9(a)). 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 *rest_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 $\alpha_{Tr} = \alpha_T$. Migration of nTreg is similar to that of conventional T cells in that nTreg actived in the LN migrate to the LP after a maturation period of 2 days and memory nTreg are distributed between the LP and LN (Figure 9(b)).

3. IMPLEMENTATION

3.1. **Algorithm.** The computation structure of 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:

Partitioning: Cells and locations are partitioned into N groups denoted by $C_1, C_2, \ldots, C_N$ and $L_1, L_2, \ldots, L_N$ 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 $MB_1, MB_2, \ldots, MB_N$. Each PE then executes the ENISI algorithm (described below) on its local data set (C_i, L_i).
Computing Visit Data: The first phase of the algorithm consists of computing a set of visits for each individual, c_i for the cycle according to the assigned schedule. 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.
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.

Whether an individual c_j interacts with others in the sublocation is determined by a probability p calculated by one of two functions:
i) *Single contact-dependent* function:

$$p = 1 - exp(\tau \ln(1-\rho)) \tag{5}$$

where τ is the duration of contact, and ρ is a constant. This is a pairwise calculation between each individual c_j and each neighbor c_k such that $s_k \in I_{sj}$. Note that if $s_k \notin I_{sj}$, then there is no chance of interaction and p is not calculated. Hence, if an interaction occurs while performing the calculation on an individual c_j in contact with a neighbor c_k, only c_j will receive the interaction message and potentially change states. A separate interaction probability is then calculated on c_k.
ii) *Multicontact-dependent* function:

$$p = \left(\frac{aA}{aA + iI}\right)^y \tag{6}$$

where A is the total number of neighbors in a state that induces a state change and I is the total number of neighbors in a state that inhibits a state change in c_j. The variables a, i, and y are constants.

The function used is determined by the state of individual c_j. For each individual that interacts, an *interaction message* is then sent back to the "home" PEs of each cell via the local message broker.
Collecting Interaction Messages: At the end of each cycle, interaction messages for each cell on a PE are merged, processed and the resulting state of each individual automaton is updated according to its type-specific transition function. If an individual c_j received a message it then transitions from its current state s_j to the next state in its automaton, x, with the probability $p_{s_j x}$.

This synchronous update at the end of each cycle assumes that any changes in behavior that result from the state transition do not take place until the next cycle. As each cycle represents six simulation hours, the synchronous update assumes a six 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 neighboring cells.

A pseudocode version of the algorithm is show below as **Algorithm 1**. All the PEs in the system are

Algorithm 1: A pseudocode description of the parallel version of the ENISI algorithm

```
initialize;
partition data across PEs;
for t = 0 to T increasing by △t do
    foreach cell c_j ∈ C_i do
        send visits to location PEs;
        computeVisits(j, t to t + Δt);
        sendVisits(MB_i);
    Visits ← MB_i.retrieveMessages();
    synchronize();
    foreach location l_k ∈ 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 ∈ C_i do
        combine outcomes of multiple interactions;
        updateState(c_j);
```

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.
The ENISI framework is implemented in C++ and uses the Message Passing Interface (MPI) for distributed processing.

3.2. **Example.** Here we describe the ENISI algorithm with an example of the immunological scenario of T cell stimulation shown in Figure 10. This example depicts the movement and interaction of a set of individuals $[c_i\ c_j\ c_k\ c_q\ c_u]$ among a set of sublocations $[y\ z]$. In this example c_i is a **dendritic cell** in the *eDC* state and c_j is a **T cell** that is initially in the $restingT$ state. When c_i and c_j come in contact, they interact and c_j undergoes the state transition $restingT \rightarrow Th1Prolif$.
Phase 1: Each individual is assigned a schedule (Fig. 10(a))
At the beginning of each cycle, each cell is provided a schedule, according to its initial state, that specifies the tissue site (Location) and sublocation that will be occupied by each individual i and the times at which it arrives (StartTime) and departs (EndTime) the sublocation. In this example, the schedule of individual c_i changes from sublocation y to sublocation z at time $t = 30$.
Phase 2: Sublocation builds network and calculates interactions (Fig. 10(b))

(a) Phase 1: Each cell gets a schedule

Schedule for individual c_i on day 1:

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

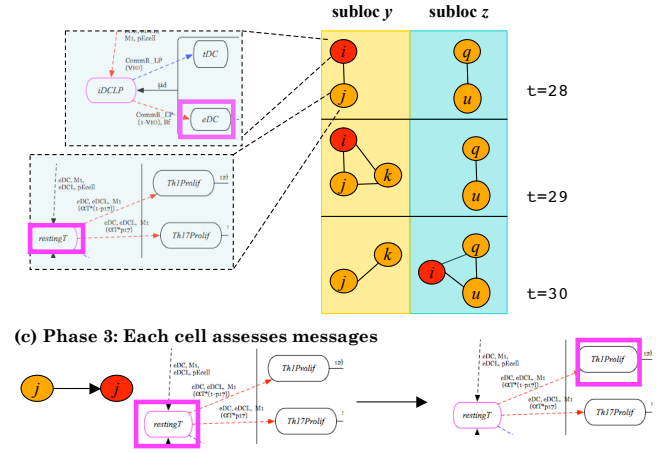
(b) Phase 2: Each sublocation accepts list of visits

sublocation *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 10: Individual c_i is in the *eDC* state and is able to induce a state transition in c_j in the *restingT* state. (a) At time t = 28, c_i and c_j are in the same sublocation according to the schedules assigned in phase 1. (b) At time t=29 c_k enters sublocation y and is now in contact with c_i and c_j. At time t = 30, c_i departs sublocation y to sublocation z. Sublocation y then calculates whether a state transition occurred for c_i, c_j, and c_k. It is determined that a state transition occurs for c_j and sublocation y sends a transition message to c_j. (c) In phase 3 c_j receives the transition message and transitions to a new state determined by the T cell-specific automata probability of $p_{restingTTh1Prolif} = \alpha_T$.

Each sublocation receives the list of individuals that will occupy it throughout the cycle along with the arrival and departure times of each. When individual c_k arrives in sublocation y at time $t = 29$, 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 $t = 30$. When individual c_i departs sublocation y at time t, the sublocation performs the following processes: $i)$ c_i is removed from g_y^t, $ii)$ sublocation y calculates if c_i interacts; a function of the amount of time c_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 c_j and c_k upon departure of c_i. In Figure 10 individual c_i is in the state $s_i = eDC$, c_j is in the state $s_j = restingT$ and $eDC \in I_{restingT}$. Therefore, the interaction probability p is calculated for the pair c_i, c_j using the *single contact-dependent* function (Equation 5). In this example, the probability is such that a state-transition occurs for individual c_j and the sublocation y subsequently sends an *interaction message* to c_j. However, the state of c_j will not immediately change from $restingT \rightarrow Th1Prolif$.

Phase 3: Individual receives interaction message and determines new state (Fig. 10(c))

At the end of each simulation cycle (6 real world hours), individual c_j assesses the interaction messages received and then transitions from $restingT \rightarrow Th1Prolif$ with the probability α_T, the probability of T cell

stimulation representing the antigen specificity of the specific TcR of T cell c_j.

3.3. **Effect of Model Approximations.** This implementation employs two simplifications that require 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. As each cycle represents 6 real world hours, the model 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 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 **inflammatory bacteria**. This interaction induces transitions *iDCLumen* $\rightarrow$ *eDCL* in the **dendritic cell** and $InfB\_lumen \rightarrow InfB\_dead$ 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 $InfB\_lumen$ 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 an individual c_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. 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.

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 5 and Equation 6). 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 7 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 **??** holding the variable $\kappa_{I_d}^{d-1}$ 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 $\hat{p}_{xs}$.

$$\kappa_s^d = \frac{p_{xs}\kappa_{I_x}^{d-1}\kappa_x^{d-1}\varepsilon_x}{subloc_y} \tag{7}$$

$$\hat{p}_{xs} = \frac{subloc_y\kappa_s^d}{\kappa_{I_x}^{d-1}\kappa_x^{d-1}\varepsilon_x} \tag{8}$$

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 9 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.

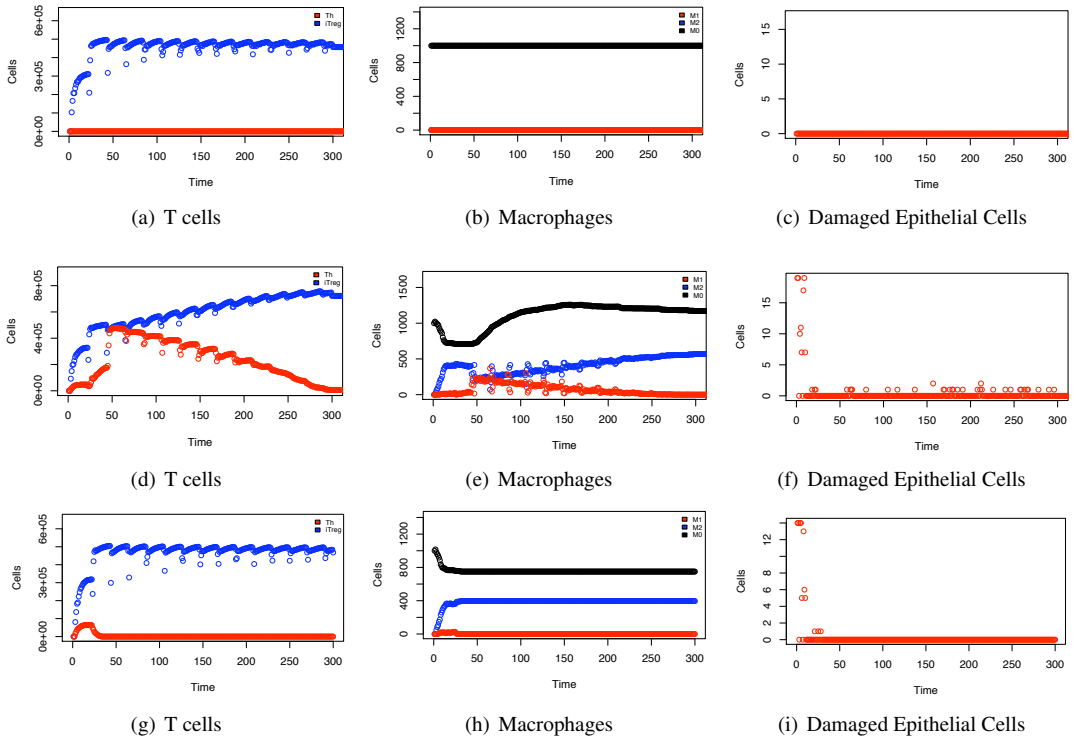
$$subloc_y = \frac{\kappa_s\varepsilon_i}{r} \tag{9}$$

4. Example Application: Explanation of *B. hyodysenteriae* associated pathogenesis

B. hyodysenteriae infection is characterized by severe dysentery in which epithelial lining is damaged and bacteria is detected in the LP resulting in fever and bloody diarrhea. Associated with infection is the phenomenon that, following *B. hyodysenteriae* -induced epithelial damage, normally tolerogenic commensal bacteria is able to induce an inflammatory response allowing persistent tissue damage [12].

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 interact with an individual in a user-specified state $s_i \in S$ and induce the state change $s_i \rightarrow x$ over a user-specified time period 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 *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 may persist with continued epithelial cell damage following *B. hyododysenteriae* elimination.



(a) T cells (b) Macrophages (c) Damaged Epithelial Cells

(d) T cells (e) Macrophages (f) Damaged Epithelial Cells

(g) T cells (h) Macrophages (i) Damaged Epithelial Cells

Figure 11: Dynamics of cell populations over a period of 75 days. 'Th' refers to Th1. Top: no pathogen present, Middle: following infection with *B. hyodysenteriae*, Bottom: following infection with *B. hyodysenteriae* without M1 mediated T cell stimulation. The x-axis is labelled in time units of 6 hours.

Let κ_s^t be the number of individuals in state s at time t and $\kappa_{[s,u]}^t$ be the number of individuals in state s or u at time t.

In this demonstration, we seek to identify pathways that lead to two different health outcomes following infection, complete recovery and chronic inflammation. *Complete recovery* is a configuration in which $\kappa_{[B_{LP},Bf_{LP},pEcell,Edead]}^t = 0$ and $\kappa_{[Th1,M1,eDC,eDCL]}^t < r$, where r is a threshold value, $t > t_{Bf\_dead}$, and t_{Bf_dead} is the time at which the last cell enters the Bf_dead state. *Chronic inflammation* is a configuration in which $\kappa_{[B\_LP,pEcell,Edead]}^t \geq 1$ and $\kappa_{[Th1,M1,eDC,eDCL]}^t \geq r$, where $r = 1$.

Parameter values for this demonstration are listed in Table 2 and were assigned according to published observations of interactions between bacteria of the *Brachyspira* genus and immune cells or were fit to cell population dynamics reported in animal infection models [15, 12]. Parameter space was reduced by simplifying assumptions. For example, commensal strains represented do not induce an inflammatory response

leading to the assignment $\nu_{BM} = \nu_{BD} = \nu_{Bs} = 0$. In this example Th17 is not included as we focus on the competition between the Th1 and iTreg responses, specifically. In other words $p_{17} = 0$.

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.

Figure 11 shows simulated dynamics of certain cell populations over a period of 300 cycles (75 days) under different infection scenarios. The top panel shows expected behavior in a pathogen-free mucosa following population of the lumen by commensal bacteria [21, 11]. Specifically, immune activation occurs as shown by elevated κ_{tDC} (not shown) and κ_{iTreg} (Figure 11(a)). However there is no inflammatory response nor damaged epithelial cells (Figure 11(c)). This inhibits bacterial invasion in to the LP and macrophages remain unstimulated (Figure 11(b)). The middle panel shows the system response to the addition of 30 individuals in the $InfB\_lumen$ state, representing *B. hyodysenteriae*, on days 1, 2, and 3. The dynamics observed are in agreement with those seen experimentally in pig infections [15, 12] and generally expected with three distinct phases; *i) the acute inflammation (days 1-6)* marked by an increase in κ_{eDC} (not shown) and κ_{Th1} (Figure 11(d)), followed by epithelial damage (Figure 11(f)) that is shortly followed by bacterial invasion in to the LP (not shown). At this time macrophages are stimulated and we see κ_{M1} (Figure 11(e)) rise in conjunction with κ_{Th} (Figure 11(d)) along with increased monocyte recruitment and transient reduction in M2 (Figure 11(e)); *ii) the decline of inflammation (days 7-50)* in which clinical symptoms of epithelial damage and bacterial invasion to the LP subside after 1 week (Figure 11(f)) as the number of inflammatory T cells continues to rise before declining, along with M1, after 2 weeks. This decline occurs in conjunction with pathogen elimination; *iii) the chronic phase (days 51-75)* is marked by continued low level epithelial damage (Figure 11(f)) along with bacterial presence in the LP (not shown). In this phase foreign bacteria has already been removed.

To identify the source of this continued epithelial damage we observed the states of those neighbors that induce the transition $EC \rightarrow pEcell$ for all **epithelial cells** that undergo this transition during the three phases of infection. In Figure 12(a), it can be seen that at all stages of infection, it is individuals occupying the $Th1$ state that are inducing the most epithelial damage. In Figure 12(b) we report the states of neighbors that induce the transition $restingT \rightarrow Th1$ showing clearly that, 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. In this phase, only commensal bacteria is present to activate macrophages. To further demonstrate that it is stimulation of Th1 by macrophages that drives tissue damage, simulated infections were repeated in the absence of the ability of M1 to induce state change in neighbors of the $restingT$ state, allowing T cell stimulation to occur only through contact with eDC. The result is the dynamics shown in the bottom panel of Figure 11. 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_{Th1} 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. This allows a tolerogenic environment to persist.

The conclusion of this simple demonstration is that residual tissue damage occurs through Th1-mediated cytotoxicity which is stimulated by M1 that has recently transitioned from the M2 phenotype. The presence of M1 is due to two parallel consequences of pathogen presence i) the increase of the Th1 population that leads to increase of M1-inducing inflammatory cytokine in the tissue, and ii) increased damage of the epithelial layer which allows invasion of commensal bacteria. The implication is that commensal bacteria, which directly induces a tolerogenic response, is indirectly responsible for maintaining immunopathological

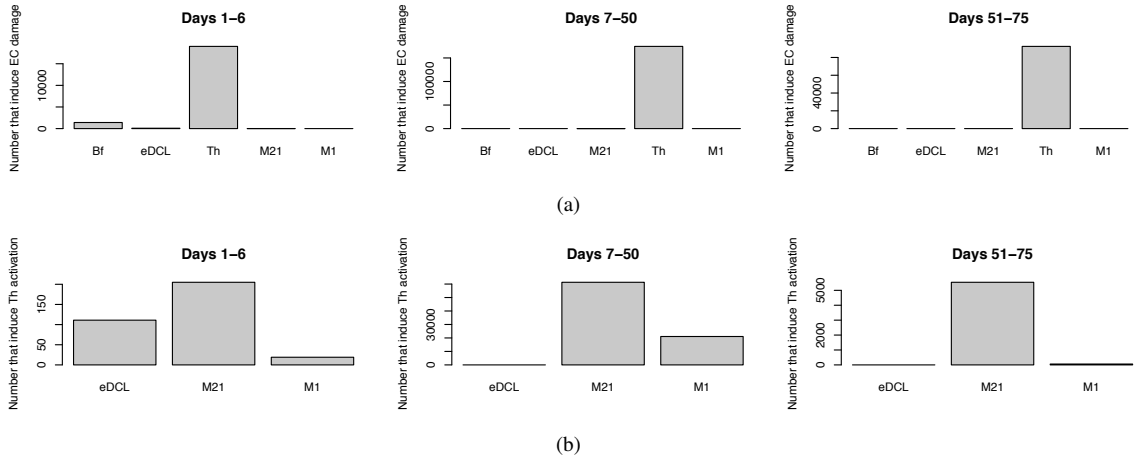


Figure 12: (a) Histogram of the number of individuals in each state that interact with an **epithelial cell** and induce the transition $EC \rightarrow pEcell$. 'Th' refers to Th1. (b) Histogram of the number of individuals in each state that interact with a **T cell** and induce the transition $restingT \rightarrow Th$

chronic inflammation via macrophage stimulation once the environmental concentration of inflammatory cytokines reaches above a certain threshold.

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 previoulsy 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 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 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.

ENISI is an evolving *in silico* system. Future versions will include dynamic creation and removal of individuals from the social network. In addition, a model is being developed to incorporate models of intracellular signaling pathways in to state transition probabilities. This will allow explicit representation of cytokine and receptor-mediated gene regulation.

6. Appendix A: Details for constructing and running a simulation

6.1. **Rule Specification.** 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.

All files necessary to run ENISI are located on svn at the url https://svn.vbi.vt.edu/svn/simfrastructure/HIV/EpiSimdemics2/broker

6.2. **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 are 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, alpanumeric 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 ($< CellId >$) and each location ($< Location >$ and $< SubLocation >$) 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 $<$ *Location* $>$ and $<$ *SubLocation* $>$ columns specifies a randomly chosen subocation of LOCATION_TYPEx and a randomly chosen compartment with in 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).

There are four schedules to which each individual may be assigned.

(a) **base:** The file base_schedule.txt Specifies the default schedule for each individual i, which it follows until a new schedule is specified in the **scenario**.
(b) **LN:** GLN.txt specifies movement with in the lymph node.
(c) **LP:** LP.txt specifies movement with in the Lamina Propria tissue site.
(d) **Epi:** Epi_Barrier.txt specifies movement with in the lumen side of the epithelial barrier.

(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 ($<$ *state* $>$) is occupied for a period of time ($<$ *DwellTime* $>$). (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 $<$ *Infectiousness* $>$ and $<$ *Susceptibility* $>$ value between 0 and 1. The $<$ *Susceptibility* $>$ 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 $<$ *Infectiousness* $>$ of s_i determines how likely c_j is to transition if $s_i \in I_{s_j}$. The $<$ *TransitionType* $>$ value of a state indicates whether transition to the next state is time-dependent or contac-dependent. Specifically, $<$ *TransitionType* $>= 0$ indicates transition from the *state* to the *nextstate* is time-dependent and $<$ *TransitionType* $>= -1$ 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 ($<$ *next_state* $>$) with a specified probability ($<$ *probability* $>$). 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* $>$

A simple example is shown in Figure 13(a) that shows a subset of the T cell automaton of an individual c_i. In this example, c_i is in the *restingT* state and transitions when it interacts with individuals that occupy a state in the set [*eDC eDCL M*1]. Upon interaction it may transition along two possible paths, either to the $Th1Prolif$ state or the $Th17Prolif$ state, chosen with the probability of 0.5 for each. After a dwell time of 48 simulation hours (12 real world hours), the individual undergoes the transition $Th1Prolif \rightarrow Th1$ or $Th17Prolif \rightarrow Th17$. A description of this path in the manifestation syntax is shown in Figure 13(b).

Note that with in one PTTS an individual's state may transition along multiple paths distinguished by probabilities $p_1 \ldots p_n$ assigned to each of n paths such that $\sum_{i=1}^{n} p_i = 1$. The PTTS does not allow

multiple paths that are not distinguished by a probability but rather by the state of the interacting neighbor, as is the case in Figure 13(c). In this example, c_i will transition from a *restingT* state along different paths depending on the state of its interacting neighbor. In the case that c_i in the *restingT* state contacts an individual whose state is in the set $[eDC\ eDCL\ M1]$ it will enter the $Th1Prolif$ or $Th17Prolif$ state. However, if the state of the contact is of a different set $[tDC\ tDCL\ M2]$, then c_i transitions along a different path to $iTregProlif$. This third path requires a **secondary PTTS**, in a separate manifestation file, in which the alternate interaction ($restingT \rightarrow iTreg$) is calculated and then used to transition the state in the **main PTTS** as demonstrated in Figure 13(d). This synchronization of multiple PTTSs, in different manifestations, for the same automaton is controlled through the **scenario**.

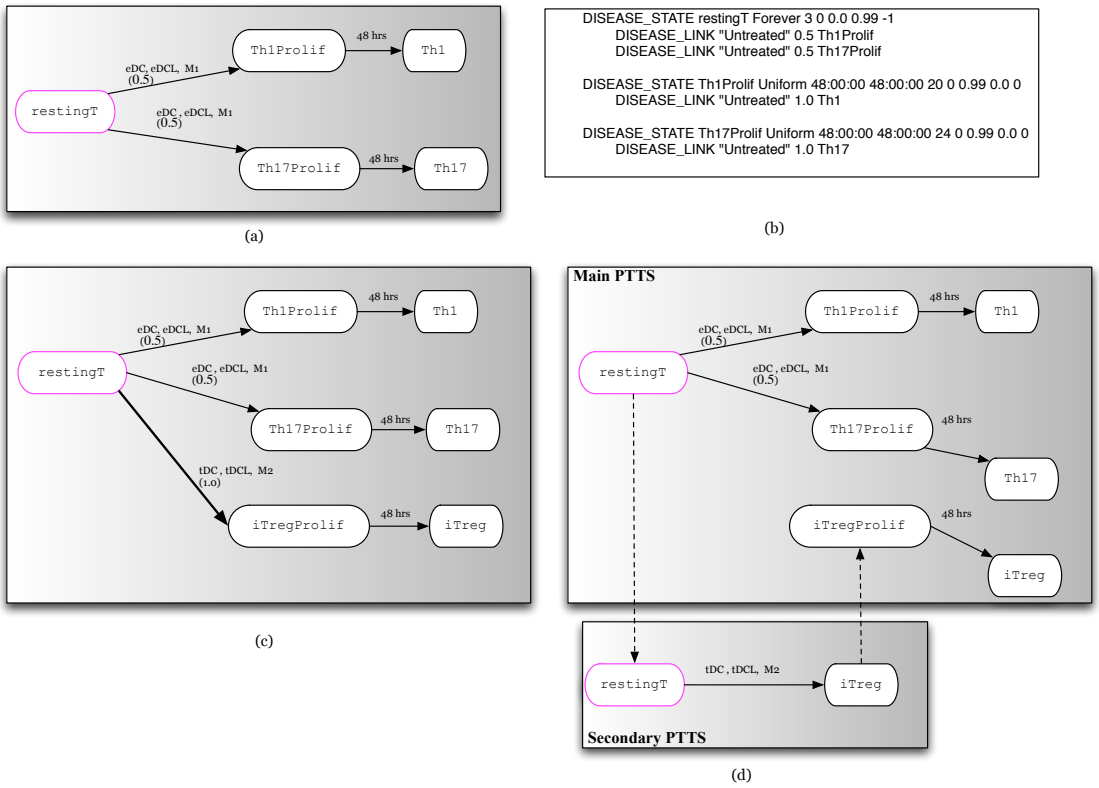


Figure 13: A subset of the **T cell** automaton encoded as two PTTSs: (a) The PTTS that is a subset of pathways in the **T cell** automaton in which the transition of *restingT* to $Th1Prolif$ or $Th17Prolif$ occurs when the T cell is in contact with individuals in states in the set $[eDC\ eDCL\ M1]$. (b) The PTTS depicted in (a) as encoded in the manifestation file. (c) An extension in which an additional pathway may be taken from the *restingT* state to the $iTregProlif$ state (bold arrow) when in contact with individuals in states from a different set $[tDC\ tDCL\ M2]$. (d) Example of a separation of the single automaton in to two PTTSs. Contact with individuals with a state from the set $[tDC\ tDCL\ M2]$ induce transition to *iTreg* in the secondary PTTS, which causes the main PTTS to adopt the $iTregProlif$ state.

To include these neighbor state-dependent transition paths in the automata, ENISI has four manifestation files, each encoding a different PTTS, among which state transitions of each cell-type

specific automaton are divided. These are: activation.mnf, birth_death.mnf, Regulatory.mnf, and Inflamm.mnf. There is a fifth manifestation, empty.mnf, which must be included but does not encode a PTTS.

The specific state transitions and synchronization schemes for each automaton are depicted in Figures 14 - 22. In these figures, the blue outline represents the PTTS encoded in Regulatory.mnf, black the PTTS in activation.mnf, red the PTTS in Inflamm.mnf, and green the PTTS in birth_death.mnf. Squares are states of the automaton where the thicker lined squares indicate inactive states in each PTTS. Solid arrows represent contact-dependent and time-dependent transitions in a single PTTS. These are labelled, respectively, with the set of states with whom contact is necessary for transition and the dwell time at which transition occurs. In parenthesis is the probability that transition occurs when these conditions are met. The default probability is 1.0. Dashed arrows point from states in one manifestation to the state which is subsequently adopted in the other.

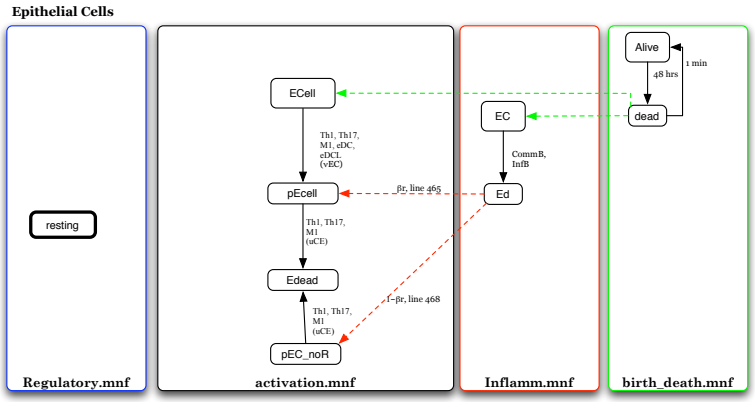


Figure 14: The PTTSs that encode the **epithelial cell** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

(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 23. Each line of the scenario is read for each person at the begining 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 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

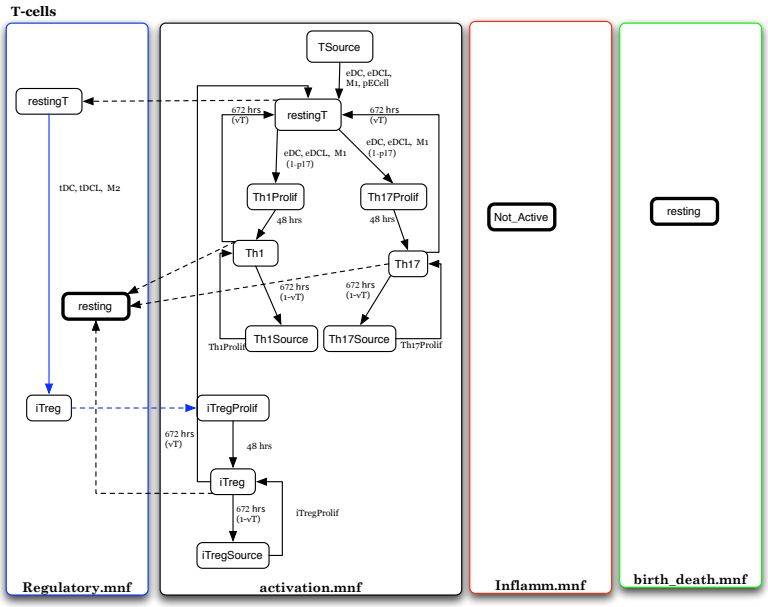


Figure 15: The PTTSs that encode the **T cell** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

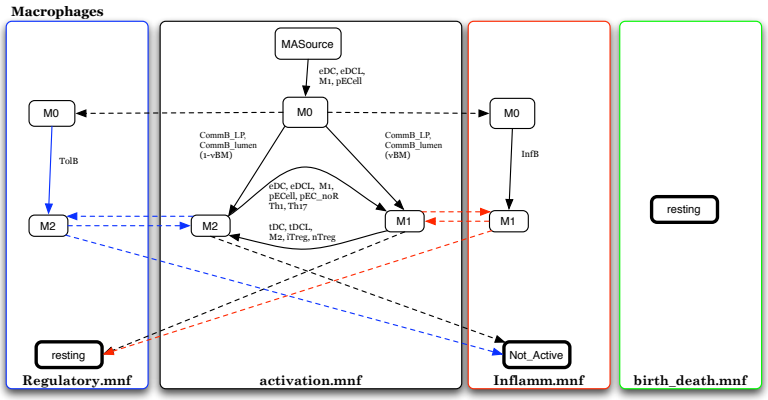


Figure 16: The PTTSs that encode the **macrophage** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

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

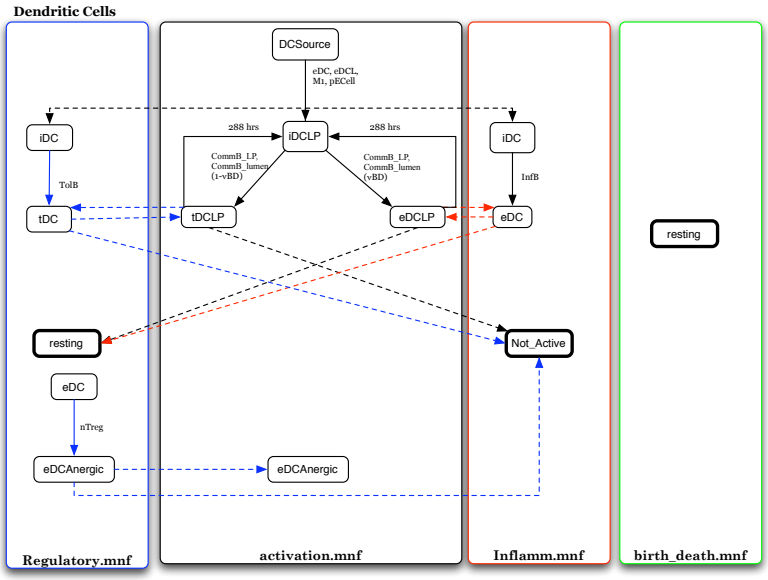


Figure 17: The PTTSs that encode the **dendritic cell** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

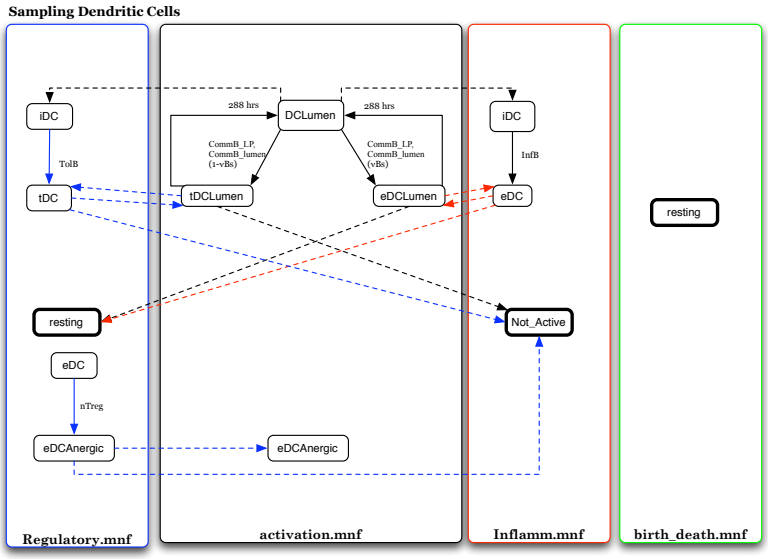


Figure 18: The PTTSs that encode the **'sampling' dendritic cell** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

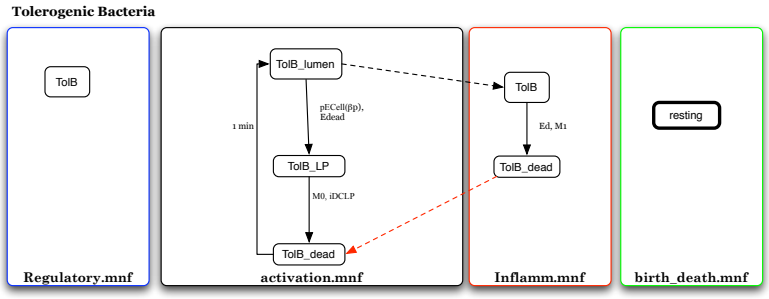


Figure 19: The PTTSs that encode the **tolerogenic bacteria** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

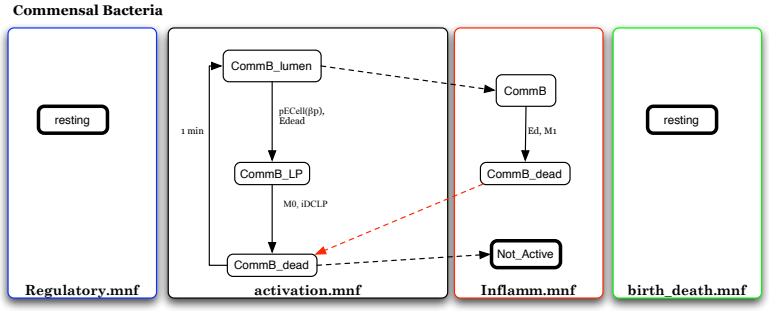


Figure 20: The PTTSs that encode the **commensal bacteria** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

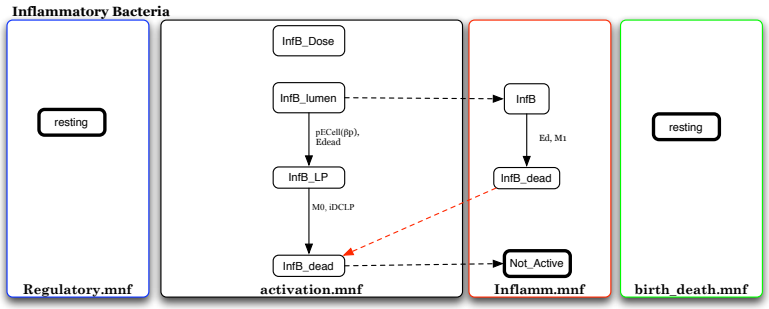


Figure 21: The PTTSs that encode the **inflammatory bacteria** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

current state of any of the PTTSs, and the value of scenario variables, described in more detail below.

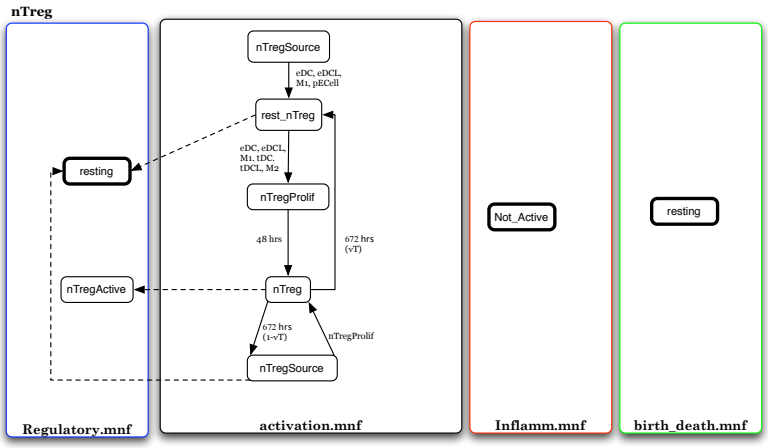


Figure 22: The PTTSs that encode the **nTreg** automaton. Dashed arrows point from states in one PTTS that trigger transition to a state in another PTTS.

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

⟨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⟩] [**keeptime** | **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_expr⟩;
⟨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 23: Grammar of EpiSimdemics scenario file.

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 in 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 selecte 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 form 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 begining of the next simulation day.

endsim: End the simulation early. Useful for dynamically evaluating stopping criteria. This action takes effect at the begining 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.

6.3. **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 occured ($< actloc >$), as well as information about the location that is not currently relevant ($< actsubloc > < acttype > < rooomtype >$). The columns are labeled as follows:
$< person > < homeloc > < time > < fsm > < infecteeState > < infectedBy > < infectorState >$
$< actloc > < actsubloc > < acttype > < rooomtype >$

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:** A described in section 3, certain individuals will transition states according to the *multicontact-dependent* function (Equation 6) 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* ($< probability >$):
$< day > < name > < start > < end > < id > < activator > < inhibitor > < target > < prob > < infected >$

(5) **Gen_Input.pl**. This perl script automatically generates cells.txt, sites.txt, and all schedule files. Initial population sizes are specifed by assigning values to the following variables at lines 8- 28:
$iDC = 1000;
$DCSource = 10000;
$TolB_LP = 0;
$TolB_Lumen = 1000;
$EC = 1000000;
$restingT = 1000;
$TSource = 10000;
$ThMLNSource = 0;
$ThSource = 500000;
$Th17Source= 500000;
$iTregSource= 500000;
$M0 = 1000;
$MASource = 10000;
$nTregNaive = 0;
$nTregMLNSource= 0;
$nTregSource= 0;
$DCLumen = 1000;
$mem_nTregSource = 0 ;
$InfB_Lumen = 30;
$InfB_Doses = 60;
$CommB_Dose = 10;
$Th1_LP = 30;
$Th17_LP = 60;
$iTreg_LP = 10;
$Th1_LN = 30;
$Th17_LN = 60;
$iTreg_LN = 10;

Locations are divided in to sublocations by assigning values to the variables listed below. These are currently functions of the number of cells in the system:
$loc_1 = 50*($restingT+$iDC+$M0);

$loc_2 = 0.2*$loc_1;
$loc_3 = 100000;
$loc_4 = $EC;
$loc_5 = 1;
$loc_6 = 1;
$loc_7 = 1;
$loc_8 = 1;
$loc_9 = 1;
$loc_10 = 1;
$loc_11 = 1;

$loc_1 is the number of sublocations in to which LOCATION_TYPE1 (the LP) will be divided, $loc_2 is the number of sublocations dividing LOCATION_TYPE2 (the LN), and $loc_4 is the number of sublocations dividing LOCATION_TYPE4 (the lumen).

7. How to Run A Simulation manually

(1) Check out files from svn:
> svn co https://svn.vbi.vt.edu/svn/simfrastructure/HIV/EpiSimdemics2/broker working_directory_name
> cd working_directory_name
(2) In default.xml specify the initial population, intervention day and dose, simulation length.
(3) Run Gen_Input.pl to generate sites.txt, cells.txt, identities.txt and all schedule files:
> Gen_Input.pl
(4) Run a simulation with default parameter values (Table **??**) by submitting the simulation using qsub:
> qsub default.qsub

7.1. **Note on dose and timing of infection.** Infection of the system with bacteria occurs through two interventions in the scenario file, 'inf_with_CommB' and 'inf_with_InfB' which transitions bacteria from the inactive *CommB_Dose* and $InfB\_Dose$ states to the *CommB_lumen* and $InfB\_lumen$ states, respectively, on the specified day. By setting the probability of these interventions to 1.0 and the values of '$InfB_Dose' and '$CommB_Dose' in Gen_Input.pl to the desired one-time infection dose, one can control the exact number of bacteria administered on a specific day. However, this scheme does not allow for a specific number of bacteria to be administered on different days. Currently, multiple infections on different days are administered by applying the intervention with a probability x where x is the fraction of the total dose one wishes to administer on the first day of inoculation. The intervention is then applied on a later day with a probability of 1.0 which allows the rest of the individuals in the inactive 'dose' states to transition to active bacterial states. The syntax for this is shown below:

trigger day = 4 and activated.state = "activated:bacterium:InfB_Dose"
    apply inf_with_InfB with prob = 0.5

trigger day = 8 and activated.state = "activated:bacterium:InfB_Dose"
    apply inf_with_InfB with prob = 1.0

intervention inf_with_InfB
    transition inflamm:Bacteria:InfB normal
    transition activated:bacterium:InfB_lumen normal

7.2. **Parameter adjustment and setting experimental conditions.** In the case that one wishes to change parameters from their default values, Table 3 provides each parameter in the model and where it is specified in the code. These may be mapped to specific experimental conditions. For example, Experimental host phenotypes may be represented by varying the susceptibility of each immune cell population to specific cytokines or the amount of various cytokines produced by different immune cell populations during infection (ν_{21}, ν_{12}, ν_{17r}, ν_{r17}, ν_{EC}, etc). *Host immunological set-point* can be represented by the initial cell and tolerogenic bacteria populations in the mucosa as well as the fraction of microflora **commensal bacteria** that can induce an inflammatory phenotype (ν_{BM}, ν_{BD}, and ν_{Bs}). Infection and colonization by specific bacteria strains may be represented by varying parameters that govern the immune cell response to bacteria (ν_{BM}, ν_{BD},ν_{Bs}, and p_{17}) as well as the parameters governing the effect of bacteria on epithelial cells (β_p, β_r, β_c, and β_d).

7.3. **Specifying contact-dependent state changes.** Each state in each PTTS is assigned an 'InteractorState' (discussed in section 6.2(4)) specified in the column after the dwell time (The 4th or 5th column). Table 5 gives the InteractorState for each state in the manifestation files. Note that states are written in the format $< manifestation >:< group >:< state >$, where $< manifestation >$ refers to one of the four manifestations to which the state belongs, $< group >$ is an optional label for organizing he states and $< state >$ refers to the specific state occupied by the individual. The interactor set I_s for each state s is specified in the section of code beginning at line 112 in EpiSimdemics/src/InfectiousDisease/SlotDiseaseProp2.C. Let s_i induce a state change in individuals occupying state s_j, hence, $s_i \in I_{s_j}$. We term s_i the 'infector' state and s_j the 'susceptible' state. In other words, individuals in state s_j are susceptible to individuals in the state s_i. The table in InfectionRelationships0.9.xls specifies, for each state, which other states it is susceptible to. Columns correspond to infector states and rows are labelled in with the susceptible states. '1' indicates that contact between the two states results in potential state change, where as '0' indicates that there is no state change upon contact.To permit one state to induce state change in another, specify the 'infectioustype' and 'susceptibletype' at line 112 in src/InfectiousDisease/SlotDiseaseProp2.C. Then rebuild EpiSimdemics from the build directory with the command:

    >>$EpiSimdemics/build> make EpiSimdemics.

For example, to remove that ability of M1 macrophages to stimulate resting T cells, comment out the lines 199,200:

```
if (infectioustype == 12 && susceptibletype == 3)
    return true;
```

where 12 is the InteractorState of M1 and 3 is the InteractorState of resting T cells (Table 5).

7.4. **Multi-contact dependent state transitions.** To create a multi-contact dependent function and indicate to which 'target' states it is to be applied, one must give the specifications in the configuration file. With the syntax given below, one must specify the name of the function (*name*), the InteractorSet to which the target state belongs (T_1 T_2 ...). There can be multiple target states. As well as the InteractorSet to which the activators (A_1 A_2...) and inhibitors belong (I_1 I_2...). Then one must specify the manifestation ($manifestation$)

to which the target, activator, and inhibitor states are encoded and the constants for the function (a, i, y).

set AI *name* active
{ TARGET_ATTRIBUTE $manifestation$:prodrome
TARGET_VALUES T_1 T_2 ...

ACTIVATOR_ATTRIBUTE $manifestation$:prodrome
ACTIVATOR_VALUES A_1 A_2 ...

INHIBITOR_ATTRIBUTE $manifestation$:prodrome
INHIBITOR_VALUES I_1 I_2 ...

INFECTION_FSM $manifestation$

FUNC_RATE_A a
FUNC_RATE_I i
FUNC_RATE_Y y
}

7.5. **Processing output files.** The automatically generated outfiles can be processed to produce visualizations 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 $s_i \rightarrow x$ over user-specified time periods during the simulation. These counts may then be represented in a number of graphical formats. The procedures for generating these files are described below.

(1) Graphing population counts by Location: This involves running two scripts: **PopsByLoc.py** and **Graph_PopCount.r** following the steps below:

(a) Once the simulation is complete and automatic output files are generated in the Outfiles/ directory, create a list of all the stat files:
>> ls Outfiles/stat.txt* >> stat_list.txt This is the input for PopsByLoc.py which aggregates the data over all the stat files to give one file **PopsByLocation.txt** that gives state counts per day per location in the format:
$< day > < location > < state > < count >$

(b) Run PopsByLoc.py with the command:
>> python PopsByLoc.py stat_list.txt
The generated PopsByLocation.txt is then the input for the R file, **Graph_PopCount.r**, that generates a plot of the population counts for each location over time.

(c) Run Graph_PopCount.r by first entering R and then providing the path to the R script:
>> R
$R>> source("working_directory_name/Graph_PopCount.r")
The resulting graphs are then generated in a pdf format in the directory Outfiles/PopsByLocation/

(2) Compute the count of transition-inducing neighbors: This involves two scripts; **infector_states.py** and **Graph_infector_states.r**. The script **infector_states.py** reads the dendogram.txt.A file, and for each individual that transitions from a specific state s_i, it identifies the state of the neighbor s_n that induced the state change ($< Infector\_state >$). It then sums the number of times an individual in state s_i transitions due to contact with a neighbor in the state s_n ($< Number >$) with in a specified time period ($< Phase >$). This information is then reported in the output file **Inducer_states**_$< s_i >$**.txt** in the following format:
$< Phase > < Infector\_state > < Number > < Infector\_name >$

To generate these files do the following:

(a) In infector_states.py specify the state of interest s_i, the time intervals for each phase, and the name of the output file at the specified lines. To specify s_i enter the value to which it corresponds in the disease.txt.A file.

(b) Run infector_states.py to generate Inducer_states_$< s_i >$.txt.
>> python infection_states.py

(c) Graph the results in Inducer_states_$< s_i >$.txt with Graph_infector_states.r. First, specify the name of the input and output files at the specified lines. Then enter R and give the path to the script Graph_infector_states.r.
>>R
$R >> source("working_directory_name/Graph_infector_states.r")
This will create the pdf file **Inducer_states**_$< s_i >$**.pdf**, a histogram of the number of individuals in each state that induced a transition in individuals in the state s_i over different time periods of the simulation.

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. Gery and E. Harel. 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] C.A Janeway, P. Travers, M. Walport, and M. Shlomchik. *Immunobiology: The immune system in health and disease*. Garland Publishing, 5 edition, 2001.

[15] 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.

[16] S. M. Kaech and R. Ahmed. Immunology: CD8 T cells remember with a little help. *Science*, 300(5617):263–5, 2003.

[17] 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.

[18] A. Lanzavecchia and F. Sallusto. Regulation of t cell immunity by dendritic cells. *Cell*, 106(3):263–6, 2001.

[19] D. R. Littman and A. Y. Rudensky. Th17 and regulatory t cells in mediating and restraining inflammation. *Cell*, 140(6):845–58, 2010.

[20] 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.

[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.

Table 3: Parameter values

Symbol	Parameter	IntheCode
	Birth/death	
μ_E	Turnover time of epithelial cells	birth_death.mnf: Dwell time of birth_death:Epithelium:alive
μ_T	Time a T cell remains active	activation.mnf: dwell time in activated:activation:Th1, Th17, iTreg
μ_{M0}	Time a macrophage remains active	Default.scn: Age at which intervention M1rests or M2rests is triggered, lines 445 and 443.
μ_d	Time a dendritic cell remains active	activation.mnf: dwell time in activated:activation:tDC_LP,tDCL,eDCLP,eDCL
μ_{ce}	Probability that $pEcell$ is killed by inflammatory factors	activation.mnf: $p(pECell \rightarrow Edead)$
p_T	Average number of daughter cells produced by a proliferating T cell in the LP	Gen_Input.pl: $0.5\frac{\$Th1Source+\$Th17Source+\$iTregSource}{\$loc\_1}$
p_t	Average number of daughter cells produced by a proliferating T cell in the LN	Gen_Input.pl: $0.5\frac{\$Th1Source+\$Th17Source+\$iTregSource}{\$loc\_2}$
	Migration	
ε_r	Average number of monocytes recruited by a single eDC, $M1$, or $pEcell$	Gen_Input.pl: $\frac{\$MASource+\$DCSource}{\$loc\_1}$
ε_t	Average number of resting T cells recruited by a single eDC, $M1$, or $pEcell$	Gen_Input.pl: $0.5\frac{\$TSource}{\$loc\_1}$, $0.5\frac{\$TSource}{\$loc\_2}$
β_p	Probability that bacteria will enter the lumen upon contact with a $pEcell$	activation.mnf: $p(TolB\_lumen \rightarrow TolB\_LP)$, $p(CommB\_lumen \rightarrow CommB\_LP)$, $p(InfB\_lumen \rightarrow InfB\_LP)$
ε_{IE}	Time active T cell in GLN before migrating to LP	Default.scn: Age at which T cells change schedule, lines 341-350
ε_{EI}	Time presenting DC in LP before migrating to GLN	Default.scn: Age at which dendritic cells change schedule, lines 197-206
	Contact/interactions	
α_T	Probability of memory T cell stimulation	activation.mnf: $p(memoryTiTregProlif)$, $p(memoryT \rightarrow Th1Prolif) + p(memoryT \rightarrow Th17Prolif)$
p_{17}	Probability of memory T cell stimulation to Th17 vs. Th1	activation.mnf:$p(memoryT \rightarrow Th17Prolif)$ vs. $p(memoryT \rightarrow Th1Prolif)$
α_{nTreg}	Probability of memory nTreg stimulation	activation.mnf: $p(mem\_nTreg- > nTregProlif)$
ν_T	fraction of active T cells that become memory T cells	activation.mnf: $p(Th1 \rightarrow memoryT)$, $p(Th17 \rightarrow memoryT)$, $p(iTreg \rightarrow memoryT)$
a_1	co-efficient of ν_{12} for activators	Default.cfg: line 162
i_1	co-efficient of ν_{12} for inhibitors	Default.cfg: line 163
y_1	exponent of ν_{12}	Default.cfg: line 164
a_2	co-efficient of ν_{21} for activators	Default.cfg: line 177
i_2	co-efficient of ν_{21} for inhibitors	Default.cfg: line 178
y_2	exponent of ν_{21}	Default.cfg: line 179
a_r	co-efficient of ν_{r17} for activators	Default.cfg: line 192
i_r	co-efficient of ν_{r17} for inhibitors	Default.cfg: line 193
y_r	exponent of ν_{r17}	Default.cfg: line 194
a_{17}	co-efficient of ν_{17r} for activators	Default.cfg: line 207
i_{17}	co-efficient of ν_{17r} for inhibitors	Default.cfg: line 208
y_{17}	exponent of ν_{17r}	Default.cfg: line 209
ν_{BM}	probability that commensal bacteria induces inflammatory phenotype in macrophages	activation.mnf: $p(M0 \rightarrow newM1)$
ν_{BD}	probability that commensal bacteria induces inflammatory phenotype in dendritic cells	activation.mnf: $p(iDC \rightarrow eDCLP)$
ν_{Bs}	probability that commensal bacteria induces inflammatory phenotype in 'sampling' dendritic cells	activation.mnf: $p(DCLumen \rightarrow eDCLumen)$
ν_{EC}	Probability that EC transitions to $pEcell$ upon contact with inflammatory factors	activation.mnf: $p(ECell \rightarrow pEcell)$
ν_{EB}	Probability that EC is damaged by microbial toxins upon contact with commensal or inflammatory bacteria	Inflamm.mnf: $p(EC \rightarrow Ed)$
β_r	Ability of commensal or inflammatory bacteria to induce chemoattractant expression in epithelial cells	Default.scn: The probability of intervention NormalpEC (line 464) vs. intervention pEC_noR (line 467).
β_c	Ability of pEcell to secrete cytokines that induce $M2 \rightarrow M1$ and $iTreg \rightarrow Th17$	Default.cfg: Turned on/off by inclusion or exclusion from lines 157, 159, 172, 174, 187, 189, 202, and 204.
β_d	Amount of microbicide secreted by epithelial cell in response to contact with commensal or inflammatory bacteria	Inflamm.mnf: Infectiousness of Ed.
μ_{M1}	Ability of M1 to eliminate bacteria	Inflamm.mnf: infectiousness of M1.

Table 4: Interactor States

Interactor State	States
1	inflamm:APC:iDC, activated:activation:iDC, activated:Macrophage:M0
2	regulatory:APC:eDCLP,activated:activation:eDCLP,activated:activation:eDCLumen
3	activated:activation:Tresting, activated:activation:restingT
4	activated:activation:Th1
5	activated:bacterium:TolB_LP, activated:bacterium:CommB_LP
6	activated:Epithelium:pEcell, inflamm:ECell:Ed
7	activated:activation:APCSource,activated:activation:TSource,activated:Macrophage:MASource
8	activated:Epithelium:Edead
9	activated:bacterium:TolB_lumen,birth_death:bacterium:TolB_lumen,birth_death:bacterium:dead, activated:bacterium:CommB_lumen
10	activated:nTreg:nTregNaive
11	activated:nTreg:nTregLP,regulatory:nTreg:nTregActive
12	activated:Macrophage:M0
13	regulatory:APC:M2, activated:Macropahge:M2
14	regulatory:APC:tDC, activated:activation:tDC_LP
15	inflamm:Ecell:Ecell, activated:Epithelium:Ecell
16	activated:activation:iTregLP
17	regulatory:Tcell:Tresting, regulatory:Tcell:restingT
18	MR*
19	M1_I*
20	activated:activation:Th1Prolif
21	inflamm:Bacterium:InfB
22	activated:bacterium:InfB_lumen, activated:bacterium:InfB_LP
23	activated:activation:Th1Source
24	activated:activation:Th17Prolif
25	activated:Epithelium:pEcell_noR
26	activated:activation:Th17Source
27	activated:activation:iTregSource
28	activated:activation:iTregProlif
29	activated:nTreg:nTregProlif
30	activated:nTreg:nTregSource
31	activated:activation:Th17
32	inflamm:bacterium:CommB
33	inflamm:Macrophage:M1

Table 5: *MR and M1_I are states that are currently not implemented. To use the values 18 and 19 to create new InteractorSets, these must be removed from the EpiSimdemics code file src/InfectiousDisease/SlotDiseaseProp2.C