

An Implementation Of The King - Kesson Flavivirus Decoy Model

James Peterson* Steven Crawford Alexander Engau
Joseph Johnson Andreas Tsolakis
Zachary Voller Leonard Wilkins Jr.
Department of Mathematical Sciences
Clemson University, Clemson, SC 29634

November 26, 2007

Abstract

This paper concentrates on the development of a model of the broadly tuned adaptive immune response seen in flavivirus infections. The model specifically addresses flavivirus pathogen virulence based on differential $NK - \kappa B$ mediated MHC-I upregulation in G_0 vs. G_1 cell states. The model is agent based although a mathematical description is also given. The MHC-I upregulation appears to lead to the development of broadly tuned T cells for flavivirus peptide antigens. This broad tuning leads to immune system attacks on healthy cells. Hence, this type of model is referred to as a *decoy* model. The up-regulation mechanism may be based on changes in antigen processing within the cytoplasm prior to endoplasmic reticulum assembly of the antigen to the MHC-I molecule, specificity issues due to Langerhans cells antigen processing or other means. We show that irregardless of the particular mechanism, a model which includes the G_0/G_1 differential activation leads to collateral damage to healthy uninfected cells. We discuss our model choices carefully so that our model can be modified as needed by other researchers.

1 Introduction:

Flavivirus is a family of viruses transmitted by mosquitoes and ticks that cause sociologically and economically important diseases, including dengue, yellow fever, tick-borne encephalitis virus, and West Nile fever. They are widely distributed throughout the world with the exception of the polar region, although a specific flavivirus may be geographically restricted to a continent or a particular part. With global warming, these single-stranded RNA viruses are entering the radars of more regions of the world than ever. West Nile virus, for example, has emerged in recent years in temperate regions of Europe and North America, presenting a threat to public, equine, and animal health. The most serious manifestation of the West Nile virus infection is fatal encephalitis (inflammation of the brain) in humans and horses, as well as mortality in certain domestic and wild birds. The virus is maintained in nature through a transmission cycle involving mosquitoes and birds. Children will usually experience

*petersj@clemson.edu

an apparent or a mild febrile illness. Adults may experience a dengue-like illness whilst the elderly may develop an encephalitis which is sometimes fatal. The diagnosis is usually made by serology although the virus can be isolated from the blood in tissue culture. No vaccine from the virus is available and there is no specific therapy.

This work will focus on building an interesting model of the host - pathogen interactions for viruses from the model *flaviridae*. There are many different viruses in this family such as dengue fever, yellow fever, Japanese encephalitis and West Nile fever. We focus on the specific virus within *flaviridae* known as the West Nile fever virus (Brinton, 2002, [4]). This virus exhibits important behaviors within the host and pathogen and in the disease dynamics that are sufficiently general to be applicable to both the *flaviridae* family and other virus families as well. The flavivirus infections feature a substantial up-regulation of cell surface molecules of a variety of types on cells which are in the G_0 resting state of cell division. Curiously, cells which are dividing (i.e. in the G_1 state) do not have this up-regulation. Although many cell surface molecules are expressed at a much higher rate in the quiescent cells, we will focus in this paper primarily on the increase in the MHCI complex. We build an agent-based model as closely tied to the epidemiological and biological literature as possible (although there is a mathematical framework as well) and we show that the model exhibits the kind of collateral damage seen in flavivirus infections that lead to host death in a small but troubling 3% to 8% of the infected population.

All modeling of this nature requires many compromises and many levels of abstraction. We model the infection within a single *abstract* host. We choose initial cell population sizes and immune system components so that our *abstract* version of a host exhibits reasonable responses closely aligned to what we find in the literature. The paper begins with a discussion of relevant flavivirus background material in Section 2 so that we can put our model within the proper context. Then, in Section 3, we discuss at some length a full host - pathogen multi-scale model of such infections which requires system components at many times scales. Although we do not build a model of such generality and power in this paper, we felt it is important to see how the model we do build fits within the full host - pathogen system modeling framework. The one - host model is then discussed in detail in Section 4. This section sets out the basic agent based model we use and its interpretation as a standard iterated discrete time system. In addition, we discuss thoroughly its implementation in MatLab. Simulation results and our conclusions are presented in Section 5 and Section 6, respectively.

2 Background and Significance:

The flaviviruses attempt to exploit the immune response in a novel way using what is called the $NK - \kappa B$ pathway. For a general review of this pathway and its relation to virus infection, see (Santoro, et. al. 2003, [16]). We provide some pertinent details in our discussion below as much of this is relevant for our discussion of the flavivirus exploitation of the immune system response. Further, this background is also required to explain specific features of our modeling process.

The nuclear factor family $NK - \kappa B$ is a family of proteins such as $p50$, $p52$ and $relA$. These factors bind into dimers such as a $p50 - relA$ among other possibilities. They are very important as many genes possess $NK - \kappa B$ binding sites within their promoter regions and hence free $NK - \kappa B$ can have an enormous impact on the function of a cell. Within a cell, the $p50 - relA$ dimers are bound with inhibitory proteins of the $I\kappa B$ family to create an inactive complex. This inactive complex must be activated to split the $NK - \kappa B$ from the inhibitory protein $I\kappa B$ so that the $NK - \kappa B$ is free to modulate gene activation within the nucleus. The $NK - \kappa B$ pathway can be activated via a large

variety of membrane based mechanisms such as growth factors, inflammatory cytokines, bacterial products and stress signals. The second messenger pathways initiated by these primary signals can be extraordinarily complex in nature, but all such resulting cascades essentially separate the $p50-relA$ dimers from the inactive complex. The inhibitory proteins of the $I\kappa B$ family then degrade and the $NK - \kappa B$ dimers translocate into the nucleus. The $NK - \kappa B$ binds to a κB element of DNA having the general form $5' - GGGACTTTC - 3'$. There are hundreds of genes that have a κB element in their promoter region and hence the potential impact of freed $NK - \kappa B$ is enormous. The targets of these activated genes include cytokines, chemokines, receptors for various kinds of adhesions, receptors facilitating migration of substances across blood vessel walls and various parts of the adaptive immune system. Hence, $NK - \kappa B$ promotes inflammatory response. In particular, $NK - \kappa B$ activates the Major Histocompatibility Class I (MHC-I) receptors. In addition, $NK - \kappa B$ can enhance cell survival by switching on genes that dampen or suppress cell kill signals.

Since $NK - \kappa B$ is present in the cell cytoplasm in a bound complex, no protein synthesis is required to create it. Hence action of this pathway is quite rapid (\sim minutes) and the resulting activation can have a profound effect on cell life.

Due to this, the activation of the $NK - \kappa B$ pathway is an attractive target for a virus. Many viruses have evolved strategies to modulate the $NK - \kappa B$ pathway. Vaccinia interferes with $NK - \kappa B$ induction, cow pox and raccoon pox inhibit $I\kappa B$. The above viral strategies attempt to block $NK - \kappa B$ activation to enhance the survival of an infected cell. However, viruses can also attempt to exploit $NK - \kappa B$ for a purpose of their own. HIV and MECV (encephalomyocarditis) exploit $NK - \kappa B$ activation that suppresses cell kill signals. This enables these viruses to block apoptosis and prolong the survival of the host cell to gain time for replication and increase viral load. Some viruses possess $NK - \kappa B$ binding sites which when activated by $NK - \kappa B$ that is present, enhance viral replication. Hence, the equilibrium between the host's and the pathogen's advantage for $NK - \kappa B$ activation is critical to understanding the dynamics of the infection.

The flaviviruses upregulate the production of MHC-I receptors by a factor of 5 – 10. Upregulation of MHC-I receptors by flavivirus is discussed in (Kesson et. al, 2001, [9]) in which experimental evidence is presented to prove that increases in a variety of cell surface recognition molecules actually occur. These experiments showed MHC-I increases in mouse embryo fibroblasts due to West Nile Fever infection. Further pathway details are provided in (Cheng et. al., 2004, [5]) and in (Arnold et. al., 2004, [1]). Normally, this would cause a large increase in the visibility of the flavivirus infected cell to the adaptive immune system. This would seemingly guarantee faster clearing of the virus infected cells potentially interfering with the pathogen's ability to replicate enough to transmit itself to its mosquito host during blood feedings. However, this is not the case and the strategy induces two effects. First, there is enhanced infected cell survival in a time window long enough to facilitate transmission to the mosquito host. This clearly benefits the pathogen. Second, there is a large increase in the number of uninfected cells that are targeted for lysis by the T cells that recognize the flavivirus MHC-I peptide receptor. This collateral damage may be responsible for the percentage of human and animal hosts that are killed by the infection.

The mechanism by peptides are extracted from the pathogen for presentation on the cell surface with a MHC-I + peptide complex are important to our model and hence are reviewed briefly here (see, e.g. Shastri et. al., 2002, [17]). $CD8+$ T cells are a key mechanism for detecting and eliminating abnormal cells. $CD8+$ T cells probe the MHC-I + peptide complexes on a target cells surface for peptides that are classified as abnormal or foreign. There are two essential steps in the generation of an MHC-I + peptide complex. The first step is the generation of the short 9 – 10 long amino acid protein called a peptide that is associated with some protein product of the flavivirus. Some portion of

newly synthesized proteins are degraded immediately after creation. The proteasome in the cytoplasm is a complicated heterogeneous structure with several catalytic activities. Its function in the generation of the 8 – 10 long peptides called antigens associated with a viral protein is as follows. The proteasome cleaves the viral protein at a highly specific place in its polypeptide chain to generate the *C* end of the antigenic peptide. Thus, the *C* end of the peptide is made in the cytoplasm and the fragment created by the proteasome is then further processed to create the *N* end. Once this is done, the peptide has been generated and it can be incorporated into the MHC-I molecule. The cleaving at the *N* end can take place in either the cytoplasm or the endoplasmic reticulum (ER). The antigenic processing transporter (TAP) moves either the unfinished peptide (only the *C* end has been completed) or the completed peptide into the ER. Inside the ER, MHC-I molecules have been created for assembly with the viral peptides that have been processed. The completed peptide - MHC-I complexes then migrate to the cell surface for presentation to T cells.

Each human has about 6 different types of MHC-I molecules while there are approximately 800 MHC-I molecules in the human population as a whole. A given MHC-I molecule is created from four protein subunits: three α helices α_1 , α_2 and α_3 and a β microglobulin labeled β_2 . The α_3 and β anchor the MHC-I molecule to the cell surface and the α_1 and α_2 units create a structure on top of the anchor which provides a cradle which can hold a short peptide of length 8 – 10 amino acids. The cradle is about 25 angstroms long and 10 angstroms high in general and most of the amino acids that make up the cradle are highly conserved. However, there are 10 residues that are very polymorphic whose exact values change in different MHC-I alleles. There are 2 – 7 residues that affect T cell binding in various ways. Each MHC-I molecule tends to effectively bind peptides that share conserved residues at perhaps three positions. The remaining 6 – 7 positions are highly variable allowing each type of MHC-I molecule to present approximately 20^6 to 20^7 molecules; i.e. 6.4×10^7 to 1.28×10^9 . Each of these peptide - MHC-I bound complexes (pMHC-I complexes) then has associated with it a specific T cell which will recognize the pMHC-I complex. Since an individual human can have 6 MHC-I types, each human should be able to recognize about 3.8×10^8 to 7.8×10^9 antigens presented via the MHC-I molecules.

In (King, et. al., 2003, [10]), a model is presented that attempts to explain why upregulation of MHC-I molecules is beneficial to the pathogen. Their model has the following characteristics. In an adaptive immune response, viral peptides are specifically recognized by T cells in the context of the cell surface molecules such as MHC-I via the T cell receptor, *TcR*. This response is further refined by *CD8* and *CD4* molecules on the T cell which further specify p-MHC-I recognition on the infected cell. The success of this recognition is influenced by the *CD8*, *CD4* and an entire array of intercellular adhesion molecules (*ICAM – 1* and *CD54*) and vascular adhesion molecules (*VCAM – 1*, *CD106*). These are recognized by receptors *LFA – 1* (*CD18* and *CD11a*) and *VLA – 4* (*CD49d* and *CD29*) which are found on the T cells.

Interleukins such as *IFN – 8* (type 2 *IFN*) are produced by T cells and other directly produced antiviral cytokines like *IFN – α* and *IFN – β* (both type 1 *IFN*) and tumor necrosis factor *TNF* produced by virus infected cells strongly upregulate the production of MHC, *ICAM – 1* and *VCAM – 1* proteins.

Any increase in the cell surface concentration of cell surface molecules such as MHC and other adhesion molecules, significantly increases the probability of their adhesive interaction with T lymphocytes. Such cells thus become the target of immune attention and are more efficiently recognized by T cells that target the pMHC-I complex that is specific for a viral peptide. Thus, increased MHC-I production on flavivirus infected cells via the *NK – κB* pathway results in increased lysis by virus specific cytotoxic lymphocytes (CTLs).

It is also known that the upregulation of MHC-I due to flavivirus occurs much more in cells that are in their resting phase, the G_0 state (see Kesson, et. al., 2001, [9]). In addition, the flaviviruses upregulate the cell adhesion molecules ($ICAM - 1$ and $CD54$) and ($VCAM - 1$, $CD106$) among others. The concentration of these adhesion molecules increase 2 – 6 times within four hours of infection. Hence infected G_0 cells have a very high immune system profile.

Langerhans cells (LC) occur in a cellular network within the epidermis. They form an extensive network via direct dendritic contact. Infected LC upregulates MHC-I and the adhesion molecules also prior to migrating to the local lymph nodes to present captured antigen. These antigens are not created via the viral peptide creation by the proteosome and insertion into a MHC-I molecule. The pMHC-I complex is very specific to the virus protein used to create the peptide. However, the antigen presented to the lymph node via the Langerhans cells migration is not as specific. These antigens would not normally be recognized by the immune system as readily because they are not as specifically tuned as the pMHC-I complexes to T cell lysis. However, the flavivirus enhanced the creation of a wide range of cell surface and adhesion molecules even in the Langerhans cells thereby creating populations of cytotoxic lymphocytes which are active below the normal pMHC-I recognition threshold. Note that without the upregulation of the cell surface molecules due to the flavivirus, this would occur at a much lower statistical rate.

The CTL thus formed would have a statistically high probability of recognizing bound peptides within the pMHC-I complex. Langerhans formed CTL's would only have to match some of the 6 – 7 residues captured in the MHC-I cradle. In addition, these broadly tuned CTLs could effectively recognize many pMHC-I complexes, even those not associated with viral peptides. Initially, MHC would only increase on G_0 state infected cells. However, secretion of type 1 IFN by the infected cells would then increase MHC-I on all cells in a neighborhood of the infected cells. These cells might not even be infected yet they would be a high risk of destruction by the broadly tuned CTLs. This situation then undergoes a positive feedback loop as activated T cells secrete $IFN - \gamma$ which upregulates cell surface molecule further.

Thus, the upregulation of the cell surface molecules by flavivirus creates a situation where there is significant risk of collateral damage. Since many flavivirus infect certain types of neurological cells preferentially, there is a larger chance of brain infection in a statistically significant portion of the population.

The flavivirus does not significantly upregulate cells in their active states of cell replication such as G_1 and so forth. Most cells are in state G_0 but a percentage are in active states. Thus, the virus can replicate in relative safety with the G_1 population which expresses pMHC-I complexes at a much lower rate. The virus can thus grow in safety and increase the probability that its population will be high enough to successfully transfer to the mosquito host population during a blood feeding.

3 The Full Host - Pathogen Immunological Model:

The life expectancy of the flavivirus infected G_0 cells is reduced in a normal fashion by CTLs. A simple model is a modified version of the cytotoxic immunity model from (Iwasa, et. al., 2004, [8]):

$$\begin{aligned}
x'_a &= \lambda - d x_a - \sum_{i=1}^n \beta_i y_{ia} \\
y'_{ia} &= (\beta_i x - a_i - p_{ia} z_i) y_{ia}, \quad 1 \leq i \leq n \\
z'_i &= c_i y_{ia} - b_i z_i, \quad 1 \leq i \leq n \\
x_a + x_b &= x
\end{aligned}$$

We use x_a to denote the abundance of uninfected G_0 cells; x_b , the number of uninfected G_1 cells and x , total abundance of uninfected cells. The variable y_{ia} denotes the abundance of infected G_0 cells for virus strain i . For West Nile virus, there is only one strain and so $i = 1$, but for Dengue fever virus, $i = 4$. Similarly, y_{ib} denotes the abundance of infected G_1 cells for virus strain i . The abundance of CTLs specific to virus strain i is given by z_i . We let the birth rate of the uninfected cells be designated by λ . Hence, λx is the number of cells in state G_1 . The uninfected cells die at a rate d implying that the death of G_0 and G_1 cells is given by $d x_a$ and $d x_b$ respectively. The G_0 infection rate is proportional to the abundance of uninfected cells and infected cells $\beta_i x y_{ia}$. Infected cells die at rate $a_i y_{ia}$ due to CTL action. The efficacy of the immune response in killing infected cells in the G_0 state is given by p_{ia} . We also assume immune activity increases at a rate proportional to the abundance of pathogen, $c_i y_{ia}$ and decreases at rate $b_i z_i$. Note pathogen locked into G_1 cells do not affect these rates.

The cells in G_1 cycle are not highly visible to the immune system and hence are free to increase viral load via replication in the relatively safe environment of the cycling cell. This is modeled by a modification of the non-cytotoxic immunity model of (Iwasa, et. al. 2004, [8]). The system is as follows:

$$\begin{aligned}
x'_b &= \lambda - d x_b - \sum_{i=1}^n \frac{\beta_i x_b y_{ia}}{1 + \eta_i z_i} \\
y'_{ib} &= \frac{\beta_i x y_{ib}}{1 + \eta_i z_i} - a_i y_{ib}, \quad 1 \leq i \leq n
\end{aligned}$$

The combined system models the G_0 and G_1 interactions as a dynamical system. Since the flavivirus infections cause significant collateral damage in a significant percentage of human hosts, the model above could be extended to model the peptide - MHC-I complexes which are suspected of being broadly tuned. Some of this can be modeled by the numerical choices of critical parameters in the models, but a more sophisticated model should be based on epitopes. Since each human expresses six different MHC-I alleles and the human population as a whole expresses approximately 800 MHC-I alleles, there is considerable variability in the viral antigens or epitopes expressed on cell surfaces. Assume there are M_i epitopes for virus strain i . Let z_{ij} is the abundance of CTLs specific to epitope j for virus strain i . The epitope version of the model is then

$$\begin{aligned}
x'_a &= \lambda - d x_a - \sum_{i=1}^n \beta_i y_{ia} \\
y'_{ia} &= (\beta_i x - a_i - p_{ia} \sum_{j=1}^{M_i} c_{ij} z_j) y_{ia}, \quad 1 \leq i \leq n \\
z'_{ij} &= \sum_{j=1}^{M_i} c_{ij} y_{ia} - b_j z_j, \quad 1 \leq i \leq n \\
x'_b &= \lambda - d x_b - \sum_{i=1}^n \frac{\beta_i x_b y_{ia}}{1 + \sum_{j=1}^{M_i} \eta_{ij} z_j} \\
y'_{ib} &= \frac{\beta_i x y_{ib}}{1 + \sum_{j=1}^{M_i} \eta_{ij} z_j} - a_i y_{ib}, \quad 1 \leq i \leq n \\
x_a + x_b &= x
\end{aligned}$$

The model above could then be recast into a nonlinear operator equation on appropriate function spaces. Letting \mathcal{X} denote the state vector and \mathcal{P} denote the parameter vector, we could write

$$\mathcal{X} = \begin{bmatrix} x_a, x_b \\ y_{ia}, 1 \leq i \leq n \\ y_{ib}, 1 \leq i \leq n \\ z_{ij}, 1 \leq j \leq M_i, 1 \leq i \leq n \end{bmatrix}, \quad \mathcal{P} = \begin{bmatrix} \lambda, d \\ x_a, x_b \\ \beta_i, a_i, 1 \leq i \leq n \\ p_{ia}, 1 \leq i \leq n \\ c_{ij}, \eta_{ij}, 1 \leq j \leq M_i, 1 \leq i \leq n \end{bmatrix}.$$

Then, letting F be the state dynamics operator, the system becomes

$$\mathcal{X}' = F(\mathcal{X}, \mathcal{P}) \quad (1)$$

$$\mathcal{X}(0) = \mathcal{X}_0 \quad (2)$$

for appropriate initial conditions \mathcal{X}_0 . Equation 2 has an additional dependency on the particular MHC-I alleles that are present in the human host. There are six MHC-I alleles present in the human host out of the 800 possible alleles in the human population. Hence, a binary string of size 800 can encode the allele set in a given human host. Define the MHC-I identifier, h , as a binary string whose value at position i is 1 if allele i is present in the host and whose value is 0 otherwise. System 2 can then be rewritten to add this allele dependency as

$$\mathcal{X}' = F(\mathcal{X}, \mathcal{P}, h) \quad (3)$$

$$\mathcal{X}(0) = \mathcal{X}_0 \quad (4)$$

It is clear that the parameter vector \mathcal{P} contains scalar values which are related to the way the flavivirus upregulation of MHC-I and other cell surface molecules constructs a broadly tuned set of CTLs. As

discussed earlier, the broad tuning of the CTLs is probably responsible for the virulence seen in a significant percentage of the human host population. Hence, a model is used to set these important parameters. Hence, the CTLs for a given host allele set h is a nonlinear function, $\Phi(h)$. Based on the host allele set, h , the $\Phi(h)$ model is dependent on several factors.

- The flavivirus may interfere with the cleaving of viral proteins by the proteasome and create a population of peptides. It is known from studies of antigen presentation and CTL function (Brigl et. al., 2004, [3]), (Natarajan, et. al., 2002, [13]), (Kumänovics etl. al., 2003, [11]) and (Shastri et. al., 2002, [17]) that the viral proteins are cleaved at highly specific positions to create the 8 - 10 long peptide used as the immune system cognate. The particular peptide that categorizes a foreign protein appears to be part of an amphipathic helix structure within the viral protein (Delisi, et. al., 1985, [7] and Margalit, et. al. 1987, [12]), or contain a certain residue motif (Rothbard, et. al., 1988, [15]). If the flavivirus altered the proteasome cleaving so as to create peptides that do not match this amphipathic or motif requirement, the resulting full viral peptide may not be exquisitely tuned to the viral protein. Hence, presentation of the p-MHC-I complex on the cell surface exhibits a peptide in which only a few residues are specific to this viral protein. Hence, this p-MHC-I is broadly and not finely tuned. Hence, flavivirus tampering at this stage could create a population of broadly tuned complexes due to the MHC-I and other cell surface molecule upregulation. This could explain the collateral damage due to the broadly tuned CTLs attraction to the infected G_0 infections.
- The population of MHC-I molecules is allele h dependent and hence, independent of the peptides produced by the proteasome machinery, the population of p-MHC-I surface molecules depends on the allele h .

The CTL functional model $\Phi(h)$ is clearly a critical component of this modeling process. It creates the appropriate scalars for the parameter vectors \mathcal{P} needed for each choice of allele h . For a given human population, some fraction of the 800 possible MHC-I alleles are presented. A human population in a given geographical zone will consist of r alleles h_1 to h_r with allele h_ℓ occurring in ζ_ℓ percentage of the population with $\sum_{\ell=1}^r \zeta_\ell = 1$. Hence, the human host and flavivirus pathogen interaction model for a given human population with a specific allele mixture consists of a collection of human MHC-I allele models:

$$\begin{aligned}\mathcal{X}'_\ell &= F_\ell(\mathcal{X}_\ell, \mathcal{P}_\ell, h_\ell), 1 \leq \ell \leq r \\ \mathcal{X}_\ell(0) &= \mathcal{X}_{\ell 0}\end{aligned}\tag{5}$$

An efficient mechanism for allowing the components of System 5 to interact is as a group of interacting agents. A useful comparison of agent and equation based modeling is presented in (Parunak, et. al., 1998, [14]). Despite the age of the article, the comments and discussion it contains are not dated. Hence, equation based modeling is used in each component $\mathcal{X}'_\ell = F_\ell(\mathcal{X}_\ell, \mathcal{P}_\ell, h_\ell)$ and interactions between components are agent based. Let \mathcal{H}_ℓ denote the ℓ^{th} agent in the agent model. This agent contains the equation-based code which models the ℓ^{th} component of System 5 plus additional computational structures and decision making code. These additional components are traditional pieces in an autonomous software agent. Each agent communicates to other agents by both sending and receiving information packets at each time step in the model. Hence, the agent based model has switched to a discrete time step dynamical system. This is in contrast with the continuous time model which each component in System 5 uses.

The agents $\{\mathcal{H}_1, \dots, \mathcal{H}_r\}$ are typically embedded within a graph structure with non symmetric links between nodes. The links between nodes \mathcal{H}_i and \mathcal{H}_j are labeled \mathcal{L}_{ij} and in this context are not simple scalars. Each link is a communication pathway between given allele models using for example KIF (Knowledge Information Format) encoded data. Simulation of the agent based model is well suited to asynchronous distributed programming tools such as the *twisted* Python package. The agent based model is thus a graph $\mathcal{G}(\mathbf{H}, \mathbf{L})$ where \mathbf{H} and \mathbf{L} denote the agent node vector and linkage matrix for the graph. The state values in a given node \mathcal{H}_i are time dependent and the model tracks the time evolution of the nodal values of the graph. This time evolution takes the form of a difference equation. Let the time evolution of the system be the set $\{k\delta : 0 \leq k, k \in \mathbb{N}\}$ for some discrete time increment δ . The state transition map which transfers the graph from state $k\delta$ to $(k+1)\delta$ is a nonlinear function Ψ giving the time evolution.

$$\mathcal{G}((k+1)\delta) = \Psi(\mathcal{G}(k\delta)) \quad (6)$$

Usually, we do not write this so formally. We assume a time clock that increments in some unit (here δ) and write

$$\mathcal{G}_{t+1} = \Psi(\mathcal{G}_t) \quad (7)$$

The above discussion has been a careful exposition of the complete *decoy* model and tries to take into account many interacting systems. This model therefore depends on a huge number of parameters most of which are very difficult to choose intelligently based on the biology of flavivirus infections. The source literature is not written in a manner that is conducive to such choices. Although our interests are, of course, in the full model as disclosed here, we believe there is great value in working out a much simpler *decoy* model that is set within one host for some specific allele set. This removes a lot of the biological detail yet allows us to focus on various important modeling tasks. Hence, we will not model the $\Phi(h)$ function; we will see how far we can get in a substantive model just assuming that there is up-regulation.

4 The One Host Flavivirus Infection Model:

In this section, we develop a model inspired by the *decoy* model of (King and Kesson[10]), which simulates the human immune response in the presence of flavivirus in the body. Several simplifying assumptions allow to focus on main characteristics and may certainly be subject of further discussion. Nevertheless, while this model does not strive for completeness, we believe that a simple and easily understandable model provides significant benefits in terms of result traceability and model extensibility. Further, it sheds insight into the eventual structure of the full model System 5. Our model follows an agent based approach (ABM) (See [14] for a discussion of agent based and equation based modeling) and identifies six main objects of interests, namely the flavivirus Antigens, the Antigen Presenting Cells (APC), the Antigen Recognizer, Healthy Cells, T-Cells and the transcription of T-Cells by a T-Cell Factory. The Antigens, the APCs, the Healthy Cells and the T-Cells are modeled as populations, and the APCs are further distinguished into APC_0 and APC_1 , depending on if the infected cells are up-regulated and thus MHC-1 up-regulated or not, respectively. It is assumed that the size of the APC_0 population dominates the number of non-up-regulated APC1 cells. The above agents interact based on a collection of specified actions illustrated in Figure 1(a) and subsequently explained in further detail.

x_1	AntigenPop	Number in the Antigen population
x_2	HealthyCellPop	Number of Uninfected cells in the body
x_3	APC0CellPop	Number in Infected Up-regulated Cell Population
x_4	APC1CellPop	Number in Infected Non Up-regulated Cell Population
x_5	TCellPop	Number in the T-cell population

Table 1: State Variables

The goal of the model is to simulate the infection of a flavivirus in a population of hosts. The flavivirus kills a certain percentage of healthy cells in a given host, and the model generates this percentage over a population of 2500 hosts. Death of a host is determined by a threshold of 15% of healthy cells killed. Also note the model only simulates the adaptive immune system response after a flavivirus is introduced into a given host. We describe the model in detail and justify the assumptions that are necessary for its implementation. Further, all of the actions illustrated in Figure 1(a) are governed based on probabilities which are specified at the beginning of every simulation run, together with initial population sizes for Antigens, APCs, Healthy Cells and T-Cells.

4.1 The Mathematical Model:

To simulate the *decoy* model, we have identified state and control variables that describe enough about the flavivirus infection to enable us to model the $\Phi(h)$ for a given host and allele. We have chosen state variables to refer to cell populations and control variables to describe the interaction between agents, antigen recognition and T-Cell factory as illustrated in Figure 1(a). We list the state variables in Table 1, and the control variables in Table 2 and governing probabilities in Table 3.

Figure 1(a) can then be redrawn using the state variables as Figure 1(b).

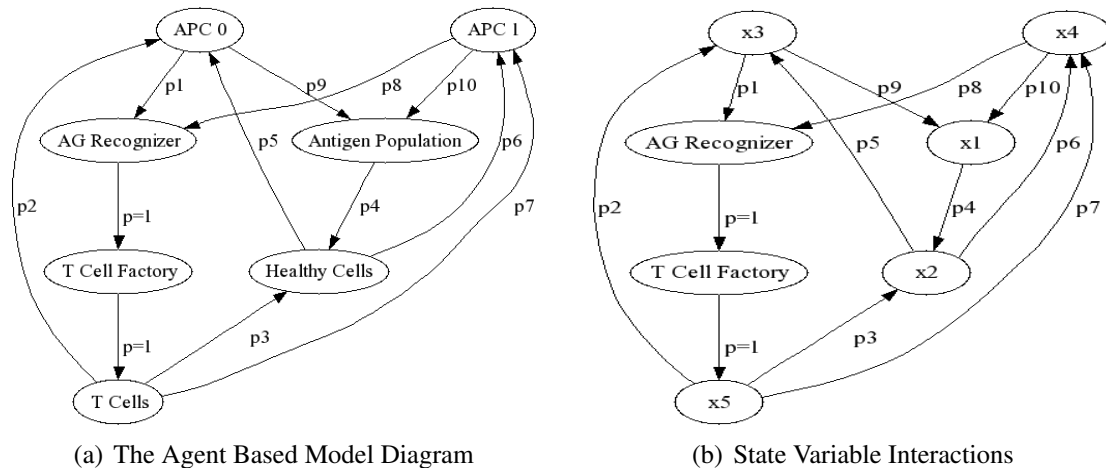


Figure 1: Agent Based Models

The AgFactor α is modeled with a probability distribution. There are also a large number of probabilities associated with our model as listed in Table 3.

In our model, we assume, $p_9 \ll p_8$ to take into account that up-regulated APC_0 are recognized to a greater extent than the non up-regulated APC_1 .

y_1	NewInfections	Number of new infections in the healthy cell population
y_2	NewAPC0Cells	Number of New cells added to APC_0 and APC_1 populations due to infection of healthy cells by virus antigen
y_3	NewAPC1Cells	
y_4	NewVirusesAPC0	New viruses added to the Antigen population due to rupturing of infected APC_0 and APC_1 cells
y_5	NewVirusesAPC1	
y_6	NewRecognizedAPCs	Number of infected cells recognized to be infected
y_7	NewTCells	Number of new T-cells made in the lymph nodes and released into the body to become part of the T-cell population
y_8	TCellAPC0Attacks	Number of attacks by T-cells against APC_0 , APC_1 , and healthy cell populations
y_9	TCellAPC1Attacks	
y_{10}	TCellHealthyCellAttacks	
α	AgFactor	Number of new viruses added to Antigen population by each rupture of an infected cell (APC_0 and APC_1)
β	TCellFactor	Number of new T-Cells produced upon recognition of a single antigen

Table 2: Control Variables

p_1	probability of Antigen infecting Healthy Cells
p_2	probability of a Healthy Cell becoming an infected, up-regulated cell (i.e. an APC_0)
p_3	probability of a Healthy Cell becoming an infected non-up-regulated cell (i.e. an APC_1), $p_3 = 1 - p_2$
p_4	probability of a APC_0 ruptures and produces more antigens
p_5	probability of a APC_1 ruptures and produces more antigens
p_6	probability of APC_0 cells being recognized by immune system
p_7	probability of APC_1 cells being recognized by immune system
p_8	probability of T-Cell attack against APC_0 cells
p_9	probability of T-Cell attack against APC_1 cells due to up-regulation of MHC complexes
p_{10}	probability of T-Cell attack against Healthy cells

Table 3: Probabilities

4.2 Linear Time Discrete model Formulation

The complete model consists of the *Antigen Agent*, the *Healthy Cell Agent*, the *APCO Agent*, the *APCI Agent*, the *Ag Recognizer Agent*, the *TCell Factory Agent* and the *TCells Agent*. The defining equations are inferred from the diagram Figure 1(b). These equations are defined using *arrow* notation. For example, from Figure 1(b), we see that since the value of x_1 is incremented over time due to viral infection, it depends on x_1 itself (feedback). In addition, it depends on the values of y_4 and y_5 whose sum adds new viruses to the antigen population. This gives the assignment $x_1 \leftarrow x_1 + y_4 + y_5$. We postulate that the number of new infections is a percentage of the antigen population x_1 which leads to the assignment $y_1 \leftarrow p_1 x_1$. Finally, since some antigen is now within cells, the value of x_1 decreases according to the assignment $x_1 \leftarrow x_1 - y_1$. These equations are written in a serial manner (reflecting our choice of implementation in MatLab) but it is clear the relationships are actually asynchronous. The asynchronous models will be addressed in future work. Note we can compress these relations into the more succinct form for the antigen agent

$$\begin{aligned}x_1 &\leftarrow x_1 + y_4 + y_5 - y_1 \\y_1 &\leftarrow p_1 x_1\end{aligned}$$

which uses “old” value of x_1 , y_1 , y_4 and y_5 to find the updated values of x_1 and y_1 . However, this misses some of the subtlety of the assignment. Note the relation

$$\begin{aligned}x_1 &\leftarrow x_1 + y_4 + y_5 \\y_1 &\leftarrow p_1 x_1 \\x_1 &\leftarrow x_1 - y_1\end{aligned}$$

can be seen more clearly if we label values as “old” and “new” with superscripts.

$$\begin{aligned}x_1^{new} &\leftarrow x_1^{old} + y_4^{old} + y_5^{old} \\y_1^{new} &\leftarrow p_1 x_1^{new} \\x_1^{new} &\leftarrow x_1^{new} - y_1^{new}\end{aligned}$$

From this formulation, we can see that forcing an asynchronous model into a serial mode introduces translational problems. However, for our purposes, the serial MatLab implementation performs well. The full assignment model for all the state and control variables is listed in System 8.

$$\begin{array}{ll}
\text{Antigen Agent} & \begin{array}{l} x_1 \leftarrow x_1 + y_4 + y_5 \\ y_1 \leftarrow p_1 x_1 \\ x_1 \leftarrow x_1 - y_1 \end{array} \\
\text{Healthy Cell Agent} & \begin{array}{l} x_2 \leftarrow x_2 - y_1 - y_{10} \\ y_2 \leftarrow p_2 y_1 \\ y_3 \leftarrow p_3 y_1 \end{array} \\
\text{APC0 Agent} & \begin{array}{l} x_3 \leftarrow x_3 + y_2 - y_8 - p_4 x_2 \\ y_4 \leftarrow \alpha p_4 x_3 \end{array} \\
\text{APC1 Agent} & \begin{array}{l} x_4 \leftarrow x_4 + y_3 - y_9 - p_5 x_3 \\ y_5 \leftarrow \alpha p_5 x_4 \end{array} \\
\text{Ag Recognizer Agent} & y_6 \leftarrow p_6 x_2 + p_7 x_3 \\
\text{T-Cell Factory Agent} & y_7 \leftarrow \beta y_6 \\
\text{T-Cells Agent} & \begin{array}{l} x_5 \leftarrow x_5 + y_7 \\ y_8 \leftarrow p_8 \min\{x_2, x_5\} \\ y_9 \leftarrow p_9 \min\{x_3, x_5\} \\ y_{10} \leftarrow p_{10} x_5 \end{array}
\end{array} \tag{8}$$

The assignment equations can then be translated into a discrete time system by thinking of the left hand side of each assignment at the value of the system at system clock time $t + 1$ and the values on the right as values due to time t . This gives System 9.

$$\begin{array}{ll}
\text{Antigen Agent} & \begin{array}{l} x_1(t+1) = x_1(t) + y_4(t) + y_5(t) - y_1(t) \\ y_1(t+1) = p_1 x_1(t) \end{array} \\
\text{Healthy Cell Agent} & \begin{array}{l} x_2(t+1) = x_2(t) - y_1(t) - y_{10}(t) \\ y_2(t+1) = p_2 y_1(t) \\ y_3(t+1) = p_3 y_1(t) \end{array} \\
\text{APC0 Agent} & \begin{array}{l} x_3(t+1) = -p_4 x_2(t) + x_3(t) + y_2(t) - y_8(t) \\ y_4(t+1) = \alpha p_4 (-p_4 x_2(t) + x_3(t) + y_2(t) - y_8(t)) \end{array} \\
\text{APC1 Agent} & \begin{array}{l} x_4(t+1) = -p_5 x_3(t) + x_4(t) + y_3(t) - y_9(t) \\ y_5(t+1) = \alpha p_5 (-p_5 x_3(t) + x_4(t) + y_3(t) - y_9(t)) \end{array} \\
\text{Ag Recognizer Agent} & y_6(t+1) = p_6 x_2(t) + p_7 x_3(t) \\
\text{T-Cell Factory Agent} & y_7(t+1) = \beta y_6(t) \\
\text{T-Cells Agent} & \begin{array}{l} x_5(t+1) = x_5(t) + y_7(t) \\ y_8(t+1) = p_8 \min\{x_2(t), x_5(t)\} \\ y_9(t+1) = p_9 \min\{x_3(t), x_5(t)\} \\ y_{10}(t+1) = p_{10}(x_5(t) + y_7(t)) \end{array}
\end{array} \tag{9}$$

This system can be reorganized into the more standard form shown in System 10.

State Equations

$$\begin{aligned}
x_1(t+1) &= x_1(t) + y_4(t) + y_5(t) - y_1(t) \\
x_2(t+1) &= x_2(t) - y_1(t) - y_{10}(t) \\
x_3(t+1) &= -p_4x_2(t) + x_3(t) + y_2(t) - y_8(t) \\
x_4(t+1) &= -p_5x_3(t) + x_4(t) + y_3(t) - y_9(t) \\
x_5(t+1) &= x_5(t) + y_7(t)
\end{aligned}$$

Control Equations

$$\begin{aligned}
y_1(t+1) &= p_1(x_1(t) + y_4(t) + y_5(t)) \\
y_2(t+1) &= p_2y_1(t) \\
y_3(t+1) &= p_3y_1(t) \\
y_4(t+1) &= \alpha p_4(-p_4x_2(t) + x_3(t) + y_2(t) - y_8(t)) \\
y_5(t+1) &= \alpha p_5(-p_5x_3(t) + x_4(t) + y_3(t) - y_9(t)) \\
y_6(t+1) &= p_6x_2(t) + p_7x_3(t) \\
y_7(t+1) &= \beta y_6(t) \\
y_8(t+1) &= p_8 \min\{x_2(t), x_5(t)\} \\
y_9(t+1) &= p_9 \min\{x_3(t), x_5(t)\} \\
y_{10}(t+1) &= p_{10}(x_5(t) + y_7(t))
\end{aligned} \tag{10}$$

4.3 Simulation Implementation:

We model the population of healthy cells remaining in a host over a period of 10 days. This population is updated corresponding to time intervals of fifteen minutes. We initialize values which represent the relative size of each population. The number of antigens is assumed to be 1000. We choose to model the adaptive response only and thus the number of infected up-regulated APC_0 cells is set to zero. In a similar manner, the number of infected non up-regulated cells APC_1 is also set to zero. This population is significantly smaller than the number of infected up-regulated cells because most cells in the body are in cell cycle state G_0 . The number of uninfected cells in the body is set to 10^7 . Also, the number of T-cells is initially zero since we are concerned with modeling adaptive immune response. Of course, a non-zero value could be used allowing us to model innate immune response as well. However, that is not the focus of this particular research.

New viruses are added to the Antigen population due to rupturing of APC_0 and APC_1 cells. New cells are also added to APC_0 and APC_1 populations due to infection of healthy cells by virus antigen. We likewise calculate the number of attacks by T-cells against APC_0 , APC_1 , and healthy cell populations. In addition, a choice is made for the number of new viruses added to the Antigen population by each rupture of an infected cell (both APC_0 and APC_1). We model this using a normal probability distribution. The number of new T-Cells produced upon recognition of a single antigen increases by a factor of one. We know that an infected cell may not be recognized by the adaptive immune system. Hence, we keep track of this type of infected cell. We set the number of infected cells that are actually recognized as infected, the number of new infections in the healthy cell population and the number of new T-cells made in the lymph nodes and released into the body to become part of the

AntigenPop_0	1000	TCellPop_0	0
APC0CellPop_0	0	NewVirusesAPC0_0	0
APC1CellPop_0	0	NewAPC0Cells_0	0
HealthyCellPop_0	10^7	NewVirusesAPC1_0	0
Threshold	$0.85 * \text{HealthyCellPop}_0$	NewAPC1Cells_0	0
AgFactor	500	TCellAPC0Attacks_0	0
TCellFactor	1	TCellAPC1Attacks_0	0
p1	$0.01 + \text{randn} * \text{stddev}$	TCellHealthyCellAttacks_0	0
p2	0.99	NewRecognizedAPCs_0	0
p3	$1 - p2$	RecognizedAPC0Cells_0	0
p4	$0.0006 + \text{randn} * 0.0002$	RecognizedAPC1Cells_0	0
p5	$0.0006 + \text{randn} * 0.0002$	NewInfections_0	0
p6	0	NewTCells_0	0
p7	0	p6_1	0
p8	0	p7_1	0
p9	0	p8_1	0
p10	0	p9_1	0
stddev	0.002	p10_1	0

Table 4: Initial Simulation Parameters

T-cell population initially to zero. The model is dependent upon probabilities denoted p_1, p_2, \dots, p_{10} which represent the likelihood of interaction between agents which are defined in Table 3. The values of p_1, p_4 , and p_5 are determined by normal distributions with standard deviation 0.002 with the mean for p_1 set to be 0.01 and the mean for p_4 and p_5 set at 0.0006. We want a fairly small deviation to insure probability values that are greater than zero. Our simulation code uses the initializations listed in Table 4. In the model, p_1, p_4 , and p_5 change at each iteration. In particular, specifying a positive initial T-Cell population may account for the presence of an innate immune response, whereas an initial value of zero confines the consideration to adaptive responses.

4.3.1 Antigen Agent

The Antigen Agent subroutine calculates the new size of the Antigen population and the number of New infections based on the influx of new viruses from rupturing infected APC_0 and APC_1 agents. The number of new infections is computed by multiplying probability p_1 by the Antigen population. From the existing Antigen population, we subtract the New infections to get the updated Antigen population. The number of new infections calculated is used in the HealthyCell Agent subroutine Listing 1.

Listing 1: Antigen Agent

```
function [NewInfections, AntigenPop] = ...
    Antigen_Agent(NewVirusesAPC0, NewVirusesAPC1, AntigenPop, p1)
AntigenPop = AntigenPop + NewVirusesAPC0 + NewVirusesAPC1;
NewInfections = p1 * AntigenPop;
AntigenPop = AntigenPop - NewInfections;
```

The Antigen Agent spreads the flavivirus by introducing additional infected cells. A certain proportion of healthy cells from the Healthy Cell Agent is then transformed into APCs. The APC Agent is

comprised of APC_0 (up-regulated) and APC_1 (non up-regulated) agents. The APC agent also determines the number of additional Antigens caused by the rupturing of these infected cells. The emerging flavivirus infection causes an increase of the number of antigen cells present in the human body system. The proposed model simulates this increase by probabilities that reflect the likelihood of interactions between different cells, based upon their current population sizes. We are well aware that the performance of the simulation and the significance of the results derived depend crucially on the choice of these probabilities, in addition to other parameters such as the initial population sizes, and motivates the choices for probability p_1 causing a healthy cell infection by an antigen cell and probability p_4 responsible for rupturing of antigen presenting cells, thereby producing new antigen cells quantified by the `AgFactor`. In [2] a flow cytometry study of a West Nile virus (WNV) infection revealed an infection rate of 70% after a time period of 48 hours. Based on the assumption of comparability we have chosen values for p_1 , p_4 and `AgFactor` to achieve an according number of antigen cells based on the initial population of healthy cells. Due to the chosen time scale of 4 population recalculations per hour, the simulation is run for 192 time increments for varying values of p_1 , p_4 and `AgFactor`, and thereafter the obtained number of antigen cells is compared to the number of initial healthy cells. Our final choices are given in Table 5. We hope that by fitting parts of our model to the experimental information reported in [2], our model gains several benefits in terms of relevance and significance of the results obtained. Nevertheless, the difficulties in identifying more concrete data remains as one of the main difficulties and drawbacks of any simulation model, and we promote and encourage the further release and discussion of experimental data to enable more accurate models of the infection cycle in future work.

Initial Healthy Cell population	p_1	p_4	<code>AgFactor</code>
10,000,000	0.01	0.0006	500

Table 5: Infection cycle calibration (p_1 , p_4 and `AgFactor`)

4.3.2 HealthyCell Agent

The `HealthyCell` Agent subroutine calculates the current size of the Healthy Cell population and determines the number of newly infected APC_0 and APC_1 cells.

The current size of the Healthy Cell population is determined by subtracting from the existing Healthy Cell population the New infections calculated above and the number of T-Cell attacks against healthy cells from the T-Cell Agent subroutine. This models collateral damage from the immune response.

It is remarked that there might be reason to continuously add healthy cells in order to account for mitosis effects, and furthermore to continually subtract healthy cells due to natural cell death. Based on the brave assumption that these two processes automatically cancel each other they are not included in the current model.

The computed number of new APC_0 and APC_1 are computed by multiplying probabilities p_2 and p_3 with the corresponding New infections. These populations are used by the APC_0 and APC_1 Agent subroutines in order to adjust the population of APC_0 and APC_1 by the newly infected cells as shown in Listing 2.

Listing 2: Computing New ACP1 and ACP2 Cells

```
function [NewAPC0Cells, NewAPC1Cells, HealthyCellPop] = ...
```

```

HealthyCell_Agent(HealthyCellPop , NewInfections , ...
TCellHealthyCellAttacks , p2 , p3)

HealthyCellPop = HealthyCellPop - NewInfections - ...
TCellHealthyCellAttacks ;
NewAPC0Cells = p2 * NewInfections ;
NewAPC1Cells = p3 * NewInfections ;

```

4.3.3 APC Agents

The APC_0 Agent subroutine calculates the current population size of the up-regulated APC_0 cells, the number of new viruses from ruptured APC_0 cells and the number of newly infected APC_0 cells as computed by HealthyCell Agent.

The updated APC_0 population is derived by adding to the existing APC_0 population the new up-regulated cells and subtracting the number APC_0 cells successfully attacked by T-cells as well as the number of ruptured cells releasing new viruses.

In addition, the number of new viruses produced from APC_0 is computed for use by the Antigen Agent subroutine. The new viruses coming from APC_0 cells is modeled by a normal distribution with a mean of AgFactor and a standard deviation of AgStdDev (see Listing 3)

Listing 3: Computing New Viruses From APC0 Cells

```

function [NewVirusesAPC0 , APC0CellPop] = ...
APC0_Agent (APC0CellPop , NewAPC0Cells , TCellAPC0Attacks , AgFactor , ...
AgStdDev , p4)

APC0CellPop = APC0CellPop + NewAPC0Cells - TCellAPC0Attacks - ...
p4 * APC0CellPop ;

NewVirusesAPC0 = (AgFactor + randn * AgStdDev) * p4 * APC0CellPop ;

```

The APC1 Agent subroutine calculates the current population size of the non up-regulated APC1 cells, the number of new viruses from ruptured APC1 cells and the number of newly infected APC1 cells as computed by HealthyCell Agent.

The updated APC1 population is derived by adding to the existing APC1 population the newly infected G_1 cells and subtracting the number APC1 cells successfully attacked by T-cells as well as the number of ruptured cells releasing new viruses.

In addition, the number of new viruses produced from APC1 is computed for use by the Antigen Agent subroutine. The new viruses coming from APC1 cells is also modeled by a normal distribution with a mean of AgFactor and a standard deviation of AgStdDev as shown in Listing 4.

Listing 4: Computing New Viruses From APC1 Cells

```

function [NewVirusesAPC1 , APC1CellPop] = ...
APC1_Agent (APC1CellPop , NewAPC1Cells , TCellAPC1Attacks , AgFactor , ...
AgStdDev , p5)

APC1CellPop = APC1CellPop + NewAPC1Cells - TCellAPC1Attacks - ...
p5 * APC1CellPop ;

NewVirusesAPC1 = (AgFactor + randn * AgStdDev) * p5 * APC1CellPop ;

```

4.3.4 Immune Response

Simultaneously to the further infection of healthy cells and resulting increase of the APC population, the Antigen Recognizer detects this increase in APCs and sends a signal to the T-Cell Factory to produce additional T-Cells. Thereafter the T-Cell Agent specifies the number of T-Cells attacking and destroying either an APC or a Healthy Cell by sending a signal to the corresponding APC or Healthy Cell Agent to update the respective cell population. The Antigen Recognizer Agent subroutine

knownprotcount	0	ViralProtsRecog	0	NormalProtsRecog	0
numprots	12	numvirals	0.75 * numprots		

Table 6: Initial Protein Factors

computes the number of recognized antigen presenting cells based on the current APC_0 and APC_1 population and their corresponding recognition probabilities p_6 and p_7 . The computed number of recognized infected cells is used by the T-Cell Factory Agent in order to increase the transcription of T-Cells. Here the probabilities model the proportions of the APC_0 and APC_1 populations that are recognized by the immune response. The code is given in Listing 5.

Listing 5: Proportionate Immune Response

```
function [NewRecognizedAPCs , p6_1 , p7_1 , p8_1 , p9_1 , p10_1 , Recog , knownprots_1 , ...
    ViralProtsRecog_1 , NormalProtsRecog_1] = ...
    Ag_Recognizer_Agent ( APC0CellPop , APC1CellPop , p6_0 , p7_0 , NewAPC0Cells_1 , ...
    NewAPC1Cells_1 , Recog , knownprots_0 , ViralProtsRecog_0 , ...
    NormalProtsRecog_0 , virals , UpRegFactor , power2 )
```

In Listing 6, we compute the number of newly recognized APC based on the current APC_0 and APC_1 population and the recognition probabilities p_6 and p_7 .

Listing 6: Number Of Newly Recognized APC Cells

```
NewRecognizedAPCs = p6_0 * APC0CellPop + p7_0 * APC1CellPop;
```

Here the probabilities are taken to be proportions of the APC_0 and APC_1 populations that are recognized by the immune response. The computed number of recognized infected cells is then signaled to and used by the T-Cell Factory Agent in order to increase the transcription of T-Cells. Hereafter, the new probabilities p_6 and p_7 are updated for the next iteration. This computation is based on the counter for the total number of protein snippets that the immune system newly recognizes in the current iteration, `knownprotcount`, initially set to zero, the counter for the number of viral protein snippets that the immune system newly recognizes in this iteration, `ViralProtsRecog`, initially set to zero, the counter for the number of self protein snippets that the immune system newly recognizes in this iteration, `NormalProtsRecog` initially set to zero, the number of protein snippets displayed by infected cells, `numprots`, initialized by 12, and the proportion of displayed protein snippets that are viral or non-self, `numvirals`, initialized with 75% of `numprots`. These values are listed in Table 6.

At last, the number of newly infected cells that the body tries to recognize is given by `numcells` and computed as the sum of new APC_0 and APC_1 (Listing 7).

Listing 7: Number Of Newly Infected Cells Recognized

```
numcells = round( NewAPC0Cells_1 + NewAPC1Cells_1 );
```

The Ag Recognition Agent subroutine uses the variables listed in Table 7

The displayed proteins are recorded in the two-dimensional array `ViralCells` (Listing 8).

Listing 8: Displayed Proteins

```
for i = 1: numcells
    ViralCells( i , numprots ) = 0;
end
```

For each APC_0 up-regulated cell and each APC_1 non-up-regulated cell, the self and non-self proteins which are displayed on its membrane are randomly selected according to the specified proportions as illustrated in Listing 9.

Variables	Description	Value
knownprots	number of proteins currently recognized by the body	init = 0
ViralProtsRecog	viral proteins progressively recognized by the body	init = 0
NormalProtsRecog	self proteins recognized as foreign proteins	init = 0
virals	number of peptide snippets that are non-self	const
Recog	vector of initially identifiable viral peptide snippets	
knownprotcount	number of newly recognized protein snippets	int = 0
ViralProtsRecog	number of newly recognized viral protein snippets	int = 0
NormalProtsRecog	number of newly recognizes self protein snippets	int = 0
numprots	number of protein snippets displayed by infected cells	const = 12
numvirals	number of viral or non-self protein snippets displayed	const = 9
numcells	number of newly infected cells tried to be recognized	
ViralCells	array of displayed proteins	

Table 7: Variables and constants in the Ag Recognition Agent subroutine

Listing 9: Selecting Proteins For Display

```

for count = 1:round(NewAPC0Cells_1)
  for i = 1:numprots
    if (i <= numvirals)
      APC.I(i) = floor(rand * virals) + 1;
    else
      APC.I(i) = floor(virals + rand * virals * .2) + 1;
    end
  end
  ViralCells(count,:) = APC.I;
end

```

This main loop is performed for all new APC_0 for which the proteins 1 to `numvirals` are selected as self and the remaining of the `numprots` proteins are selected as non-self. The vector of proteins is recorded in the `ViralCells` array. In case the model allows for APC which are not up-regulated, that is a nonzero number of `APC1`, then the same steps are repeated for simulation of the `APC1`. The full loop is shown in Listing 10.

Listing 10: New APC0 Loop

```

if (NewAPC1Cells_1 ~= 0)
  for count = 1:round(NewAPC1Cells_1)
    for i = 1:numprots
      if (i <= numprots/UpRegFactor/2)
        APC.II(i) = floor(rand * virals) + 1;
      elseif (i > numprots/UpRegFactor/2) & (i <= numprots/UpRegFactor)
        APC.II(i) = floor(virals + rand * virals) + 1;
      else
        APC.II(i) = 0;
      end
    end
    ViralCells(round(NewAPC0Cells_1 + count),:) = APC.II;
  end
end

```

The additional `UpRegFactor` takes into account that `MHC1` complexes are increased over non-up-regulated cells. After initializing an index vector `temp`, the displayed proteins `ViralCells` of each of the `numcells` APC subject to recognition are compared to the vector `Recog` of known proteins. The loop which compares the displayed proteins to the known proteins is deeply nested (hence, computationally expensive) and is shown in Listing 11.

Listing 11: Comparison To Known Proteins

```

for i = 1:numprots+1
  temp(i) = 0;
end
for i = 1:numcells

```

p6_1	$\frac{ViralProtsRecog_0}{virals}$	p7_1	$\frac{p6_1}{UpRegFactor}$	p8_1	p6_1
p9_1	p7_1	p10_1	$\frac{NormalProtsRecog_0}{virals}$		

Table 8: Computing New Detection Probabilities

```

knownprotcount = 0;
for j = 1:numprots
    if (ViralCells(i,j) ~= 0)
        for k = 1:knownprots_0
            if (Recog(k) == ViralCells(i,j))
                knownprotcount = knownprotcount + 1;
                temp(knownprotcount) = j;
                break
            end
        end
    end
end
end

if knownprotcount ~= 0
    index = 1;
    for k = 1:numprots
        if k ~= temp(index)
            knownprots_0 = knownprots_0 + 1;
            Recog(knownprots_0) = ViralCells(i,k);
            if ViralCells(i,k) <= virals
                ViralProtsRecog_0 = ViralProtsRecog_0 + 1;
            else
                NormalProtsRecog_0 = NormalProtsRecog_0 + 1;
            end
        end
        else
            index = index + 1;
        end
    end
end
end
end

```

Given a protein is recognized for a certain APC, then its index is stored in the temporary index vector and dependent on its categorization as self or non-self the corresponding variable is incremented. Eventually, the new probabilities are computed as given in Table 8.

4.3.5 TCells:

The T-Cell Factory Agent subroutine computes the number of new T-Cells to be transcribed, based on the number of APC cells that have been newly identified as computed in the Ag Recognizer Agent subroutine. The computed number of new T-Cells to be produced is used for the T-Cells Agent subroutine given in Listing 12.

Listing 12: TCell Factory Agent

```

function [NewTCells] = TCell_Factory_Agent(NewRecognizedAPCs, TCellFactor)
NewTCells = TCellFactor * NewRecognizedAPCs;

```

The T-Cell factor allows to model different ratios between the numbers of new T-Cells, and is assumed to be much less than the number of recognized APCs. The T-Cells Agent subroutine computes the sizes of the APC_0 and APC_1 population that is destroyed by T-cells and the number of healthy cells population that is destroyed by T-cells as a consequence of collateral damage. The computations are based on the current T-Cell population which is updated based on the number of new T-Cells arriving from lymphoid tissue computed as part of the T-Cell Factory Agent subroutine. The computed sizes are reported to the APC_0 , APC_1 and HealthyCell Agent subroutines as indicated in Listing 13..

Listing 13: T Cells Agent

```

function [TCellPop, TCellAPC0Attacks, TCellAPC1Attacks, ...
TCellHealthyCellAttacks] = TCells_Agent(TCellPop, NewTCells, ...
APC0CellPop, APC1CellPop, p8, p9, p10)

TCellPop = TCellPop + NewTCells;

```

```
TCellAPC0Attacks = p8 * min(TCellPop, APC0CellPop);
TCellAPC1Attacks = p9 * min(TCellPop, APC1CellPop);
TCellHealthyCellAttacks = p10 * TCellPop;
```

4.3.6 Recognition model

The model of interest determines how many (and how quickly) the number of viral proteins are recognized by the immune system, where we assume that half of the possible peptide snippets presented by infected APC are viral and the other half are healthy. The number of proteins that are up-regulated is determined by the constant `numprots` and in the current model specified as 12. 75 % of the total number of up-regulated proteins are assumed to be viral, and the remaining 25 % are considered healthy. The number of recognized proteins is specified by the variable `knownprots` and can be initialized with a nonzero value to account for the body's ability to remember foreign proteins from previous infections. Throughout the simulation this number is subject to change since the body develops the capability to potentially recognize many more foreign proteins. The recognition model will scan the up-regulated proteins and compare these proteins to the currently known proteins. Thereby a first loop will go through a certain number of infected cells. Then the code will scan the up-regulated proteins and compare the up-regulated proteins with the known viral protein population. If a protein is recognized, i.e. there exists a match between an up-regulated protein and a known protein already in the list, then the associated up-regulated proteins are added to the list (without double counting). Note that the original known proteins are initially considered to be viral. After the system first recognizes a protein, the population of known proteins will increase and then contain healthy and viral proteins since there are both healthy and viral proteins that are potentially up-regulated on a given cell. The overall goal of the simulation is to output the number of recognized proteins, and this number will be used to calculate the probability of recognition from the APC_0 class.

The probabilities associated with the recognition of APCs are modeled time dependent and are updated together with the cell populations during the simulation. In particular, these are probabilities p_6 and p_7 that the immune system recognizes a cell of type APC_0 or APC_1 , respectively, with an initial value of 0. The three variables `knownprots`, `ViralProtsRecog` and `NormalProtsRecog` represent the foreign proteins initially recognized by the body, the viral proteins progressively recognized by the body and the self proteins recognized as foreign proteins by the body, the later used to simulate auto-immune reaction to Flavivirus infection. It is assumed that the body initially recognizes 10 foreign and none self proteins.

The power term `power2` determines the number of possible peptide snippets presented by infected APC and is modeled as constant (14). Based upon this term, the number of snippets that are non-self is denoted by `virals` and computed as $virals = 0.5 * 2^{power2}$. The viral snippets that are initially identifiable by the body's innate system are randomly selected and stored in the vector `Recog` (Listing 14).

Listing 14: Initial Recognized Viral Snippets

```
for i = 1:knownprots_0
    Recog(i) = floor(rand * virals) + 1;
end
```

The factor `UpRegFactor` represents how many times MHC1 complexes are increased over non-up-regulated cells and is also modeled as a constant (6). Together with the recognition probabilities p_6 and p_7 , the probabilities that a T-Cell attacks against APC_0 , APC_1 or a healthy cells is modeled by time dependent probabilities p_8 , p_9 and p_{10} with initial values of zero. More precise details

and justification of the antigen recognition procedure were provided in the discussion of the Antigen Recognizer Agent subroutine in Section 4.3.4.

5 Simulation Results:

Using this model, we simulated a population of 1000 hosts with each host randomly initialized so that their APC_0 and APC_1 viral signatures were distinct. In Figure 2(a) we plot the level of Antigen as a function of time increment. Antigen starts at 1000 and each time increment is 15 minutes. There's no innate response, and so the adapt immune system has to “catch up” with the infection. Thus, there is a maximum in the Antigen growth. Notice the little “pinch” occurring on the declining slope side, probably due to the APC_1 population still growing. We compare the growth and decline of the up-regulated (APC_0) and non up-regulated (APC_1) infected cell populations in Figure 2(b). The upregulated population is noticed relatively quickly by the immune response. Thus, it peaks earlier.

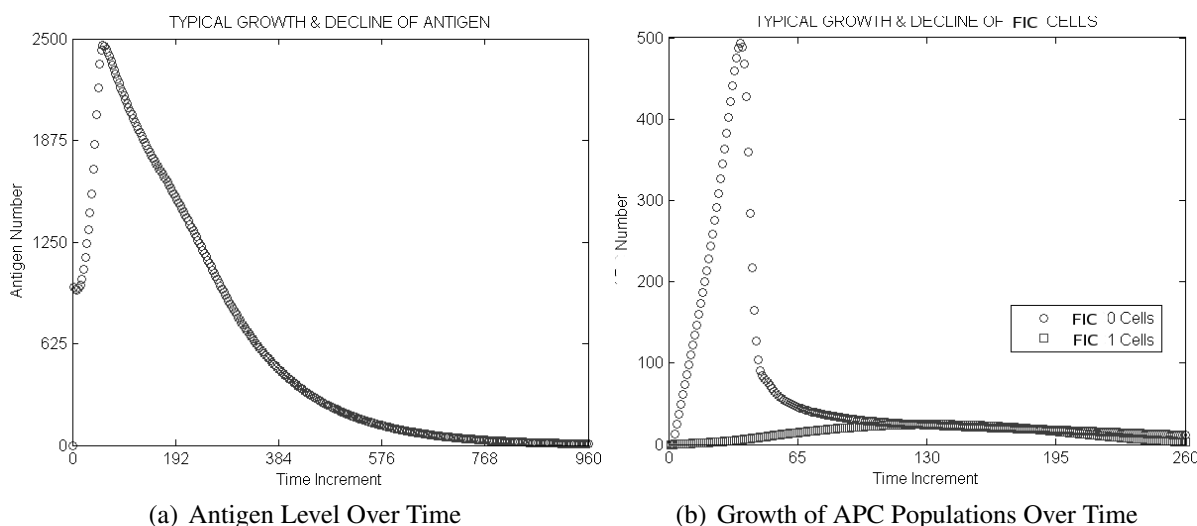


Figure 2: Antigen Asymptotic Results

In Figure 3(a), we see a bar chart which plots the number of hosts in our population which have had a given percentage of healthy cells destroyed by the infection versus that percentage. The chart gives a comparison on the severity of infection over 5000 infections. Each bar has a width of 0.125. Values closer to 1 mean an individual overcame the infection easily. Values farther from 1 mean an individual had difficulty fighting the infection. We note that if we define *host death* in our model to mean that 85% to 90% of the healthy host cells have been killed, we obtain collateral damage that is approximately 3% to 8% of the host population. One of our assumptions in our modeling process has been that the flavivirus preferentially targets certain cells in the brain and hence once a threshold of collateral damage has been reached, the host dies. We want our model to have recognition probabilities that are reasonable from a biological standpoint. Figure 3(b) shows the growth of recognition probabilities. Up-regulated APC_0 cells are recognized relatively quickly, while non up-regulated APC_1 cells remain “under the radar” for awhile. For the flavivirus infections, we know the probability an infected cell is recognized approaches one for both APC populations. We see this behavior in the plot.

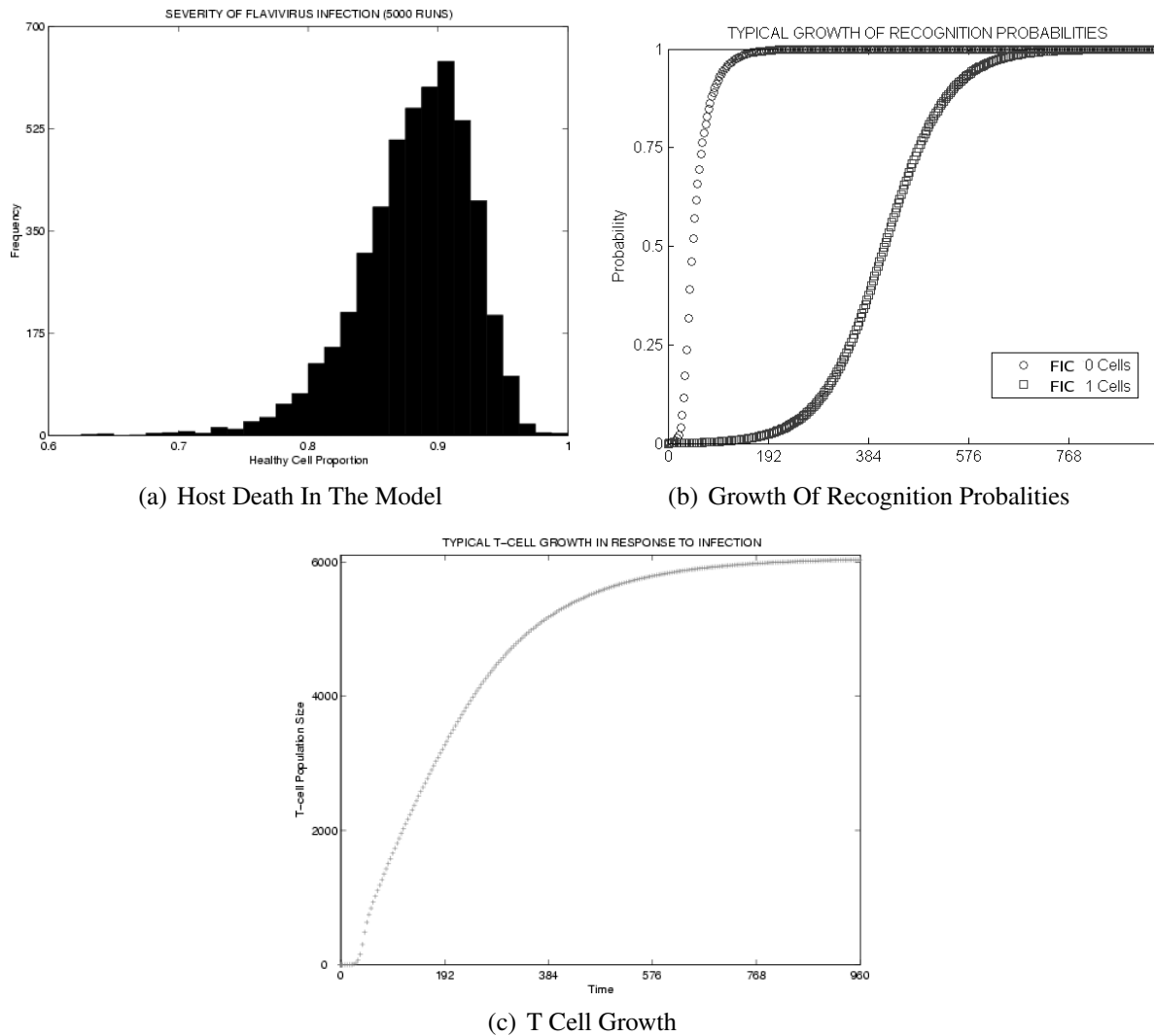


Figure 3: Asymptotic Results

We also want to see how fast T-cells are produced by the immune response. This is shown in Figure 3(c). At first, no T-cells are manufactured allowing the infection to increase without resistance. Then T-cells are produced quite quickly, leveling off asymptotically to a maximal value. There are many parameters in this model that need to be chosen, but we have attempted to choose their values based on as much accurate scientific information as we could find. These simulations do support the *decoy* model of (King, [10]).

6 Conclusions:

In this paper, we have developed a simulation in MatLab which generates results that support the *decoy* model of (King, [10]). At this stage of our modeling, we strove to find as simple a model as possible that was reasonably correct biologically and delivered correct results. It is clear that much more work can be done and we are hopeful of establishing a dialog with researchers in the epidemiological

community. The full source for the simulation as well as a user guide to both the writing and running of the code can be found in (Crawford and Voller, [6]). We used an agent based approach because it was easy to translate the basics of the flavivirus biology and epidemiology into the asynchronous model presented in Figure 1(a). The full host model requires biological and epidemiological knowledge to set a vast number of parameters and as such it is quite easy to lose sight of the need for the model to be scientifically relevant. Since much of the known scientific literature in this topic is couched in qualitative – not quantitative – terms, it is easier to capture broad flows of information rather than details of parameter size. While the implementation of the agent based model into the serial programming environment of MatLab loses the asynchronous nature of the model, we have still found the simulation captures real biology. Future work includes implementation of additional portions of the full host model and a transition to an asynchronous model using distributed programming tools.

This paper was written as part of a mathematical modeling course, MTHSC 974, taught at the Mathematical Sciences Department of Clemson University during Spring 2005.

7 Acknowledgements:

References

- [1] S. Arnold, S. Osvath, R. Hall, N. King, and L. Sedger. Regulation of antigen processing and presentation molecules in west nile virus-infected human skin fibroblasts. *Virology*, 324:286 – 296, 2004.
- [2] S. Bao, N. King, and Dos R. Flavivirus induces mhc antigen on human myoblasts: a model of autoimmune myositis? *Muscle Nerve*, 15:1271 – 1277, 1992.
- [3] M. Brigl and M. Brenner. Cd1: Antigen presentation and t cell function. *Annual Review of Immunology*, 22:817 – 890, 2004.
- [4] M. Brinton. The molecular biology of the west nile virus: A new invader of the western hemisphere. *Annual Review Microbiology*, 56:371 – 402, 2002.
- [5] Y. Cheng, N. King, and A. Kesson. Major histocompatibility complex class i (mhc-i) induction by west nile virus: Involvement of 2 signaling pathways in mhc-i up-regulation. *Journal of Infectious Diseases*, 189:658 – 668, February 2004.
- [6] S. Crawford and Z. Voller. *A Flavivirus Decoy Model Simulation Code*. Clemson University, www.ces.clemson.edu/~petersj/MyBooks/Flavivirus.html, 2005. MatLab Model.
- [7] C. Delisi and J. Bernowski. T-cell antigenic sites tend to be amphipathic structures. *Proc. Natl. Acad. Sci. USA*, 82:7048 – 7052, 1985.
- [8] Y. Iwasa, F. Michor, and M. Nowak. Some basic properties of immune selection. *Journal of Theoretical Biology*, 229:179 – 188, 2004.
- [9] A. Kesson and N. King. Transcriptional regulation of major histocompatibility complex class i by flavivirus west nile is dependent on nf- κ b activation. *Journal of Infectious Diseases*, 184:947 – 954, 2001.

- [10] N. King and A. Kesson. Interaction of flaviviruses with cells of the vertebrate host and decoy of the immune response. *Immunology and Cell Biology*, 81:207 – 216, 2003.
- [11] A. Kumánovics, T. Takada, and K. Lindahl. Genomic organization of the mammalian mhc. *Annual Review of Immunology*, 21:629 – 657, 2003.
- [12] M. Margalit, J. Spouge, J. Cornette, K. Cease, C. Delisi, and J. Berzofsky. Prediction of immunodominant helper t cell antigenic sites from the primary sequence. *Journal of Immunology*, 138(7):2213 – 2229, 1987.
- [13] K. Natarajan, N. Dimasi, J. Wang, R. Mariuzza, and D. Margulies. Structure and function of natural killer cell receptors: Multiple molecular solutions to self, non self discrimination. *Annual Review Immunology*, 20:853 – 885, 2002.
- [14] H. Parunak, R. Savit, and R. Riolo. Agent-based modeling vs. equation based modeling: a case study and users' guide. In *Proceedings of Multi-Agent Systems and Agent Based Simulation (MABS'98)*, pages 10 – 25. Lecture Notes in Artificial Intelligence, 1998. Volume 1534.
- [15] J. Rothbard and W. Taylor. A sequence pattern common to t cell epitopes. *EMBO Journal*, 7:93 – 100, 1988.
- [16] M. Santoro, A. Rossi, and C. Amici. Nf- κ b and virus infection: who controls whom. *The EMBO Journal*, 22(11):2552 – 2560, 2003.
- [17] N. Shastri, S. Schwab, and T. Serwold. Producing nature's gene chips: the generation of peptides for display by mhc class i molecules. *Annual Review Immunology*, 20:463 – 493, 2002.