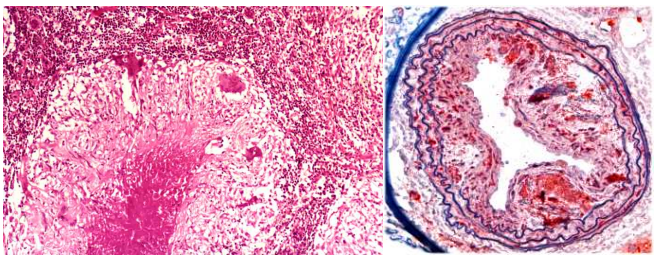


Modelling TB and Atherosclerosis using Multi-Agents

Jaspreet Deo
Department of Computing
Imperial College London
jd705@doc.ic.ac.uk

18th June 2009

Supervisor: Dr. Krysia Broda, kb@doc.ic.ac.uk
Second Marker: Professor Marek Sergot

Abstract

Mycobacterium tuberculosis and atherosclerosis are major world health problems. In this report I will detail how to implement a java agent based model to simulate mycobacterium tuberculosis (Mtb) infection. This has been previously attempted successfully by professionals within this field using C/C++. I will also discuss how to use the same approach for a different problem domain; atherosclerosis. To the best of my knowledge there has been no attempt to simulate the process of atherosclerosis using an agent based model.

Acknowledgements

I would like to take this opportunity to thank my supervisor Dr Krysia Broda for her continued support and patience throughout the project. I would also like to thank Dr Rob Krams and Dr Claudia Monaco for their proposal of the atherosclerosis model and answering my persistent questions. A thank you to Christian Ray who helped provide me with the correct parameter values from the original TB implementation.

Contents

1 Introduction

Mycobacterium tuberculosis(Mtb) and atherosclerosis are inflammatory diseases which are leading causes of mortality in the world today. Despite extensive research within both these fields treatments for prevention and immunisation are yet to be elucidated. By modelling the adaptive immune response to Mtb and atherosclerosis we can identify new areas of research and predict possible patterns of disease progression.

To gain a better understanding of Mtb infection researchers have predominantly used mathematical models to capture the process of infection, however, they have been unable to capture the spatio-temporal heterogeneity of individual cells. A limiting factor for mathematical approaches is the number of mathematical terms that are required to represent all of the relative interactions. As the number of mathematical equations increase the model becomes more computationally infeasible.

Segovia *et al* undertook an agent based model defined in [1], which allowed them to define the individual interactions among entities at the cellular level and observe the dynamics between these entities at the tissue level. In this project I re-implemented their C/C++ implementation using Java and was able to obtain similar results.

I then used this approach to model for the first time, atherosclerosis in a mouse. The current body of literature has little quantitative analysis of atherosclerosis in humans. Murine models are essential for testing mechanistic hypotheses in a controlled manner and are a good surrogate for studies of human diseases.

The Mtb simulation models granuloma formation. Granulomas are thought to be the means by which the adaptive immune response achieves and maintains latency in tuberculosis. The atherosclerosis simulation attempts to model plaque rupture. Plaque rupture is a complication of an already complex atherosclerotic process, and the precise mechanisms of which remain hypothetical.

1.1 Contribution

The core contribution of this project is the implementation of two java based discrete event simulations, which model the immune response to Mtb and atherosclerosis. Both

simulations can be run with or without a visual console allowing the user to modify model parameters before initialising a simulation.

Java's portability, extensive application programming interface(API) and large number of plug-ins makes it an attractive language to implement a multi-agent approach to simulate biological models.

The discrete event simulation for both models have also utilised a java based multi-agent simulation toolkit MASON. MASON is an open source toolkit, which provides a GUI interface for multi-agent simulations in 2D and 3D.

The key aims whilst modelling both diseases was to ensure that each model was computationally efficient and was a realistic representation of the biological process. I attempted to achieve comparable speeds to those obtained in [1] for the tuberculosis(TB) simulation. Although paper [1] provided details of the simulation and rules of the agents, the underlying implementation was unknown and was not elucidated.

The TB simulation runs a 200 day simulation within 16 minutes similar to the speeds obtained by [1]. The code and approach extends itself very well to the atherosclerosis model. The TB model proved to be very flexible and 60% of the code was found to be reusable for the atherosclerosis model.

The atherosclerosis model was a much more complex biological process than the TB model. The contradictory research material made the design of the model extremely difficult. However, through intensive testing and parameter modifications I was able to simulate plaque formation and rupture. Experts in the field believe it models the process closely although there are a number of extensions that could be added to the model for further realism. The atherosclerosis model proved to be robust as it underwent a number of modifications with the introduction of new literature and understanding of the process.

There are a number of key differences between the two models, which made the design and implementation of the atherosclerosis model much more complex. These include:

- A previously implemented TB model already existed.
- The TB immune response already contained a plethora of extensive literature within the medical field.
- Bacteria replicated within agents in the TB model, whereas lipids continually accumulate in the environment as well as within agents.
- The agents within the TB model were of the same size or negligible size, whereas the atherosclerosis model had agents of varying sizes.

- The TB model contains 5 agents whereas the atherosclerosis model contains 6 agents.

I hope to write a paper on the model, which will serve as a foundation for possible extensions and researching new aspects of atherosclerosis. Sensitivity and uncertainty analysis was conducted in the original TB model in [1]. I hope that the atherosclerosis model can be similarly progressed to provide valuable information on possible patterns of disease progression and prevention.

1.2 Overview

This report has been broken into two significant parts. Initially I will look at the Mtb model (chapter 2) and provide a thorough background into existing work(section 2.1) as well as providing specification(section 2.4) and implementation(section 2.5) details of the model. I will then explain how the atherosclerosis model was designed by providing background knowledge(section 3.1) of the disease and detailed information on the specification(section 3.2) and implementation(section 3.3) of the model describing the overall rules used for each agent. Next the two models will be evaluated(chapter 4) both quantitatively and qualitatively. Finally I have declared possible extensions to the project in the conclusion(chapter 5).

I have also attached an appendix where I investigated a rule based model. I found that the model was difficult to expand to provide an implementation for the TB model.

2 Mycobacterium Tuberculosis

2.1 Background

Infection with Mycobacterium tuberculosis is a major world health problem. An estimated 2 billion people are presently infected and the disease causes approximately 2 million deaths per year. Mycobacterium tuberculosis is a contagious disease and like the common cold, its transmission is airborne. It is estimated that one-third of the world population is infected with Mycobacterium tuberculosis [8].

There is a large discrepancy between the number of infected individuals (in the order of billions) and the number of annual cases resulting in death (in the order of millions). This is because M.tuberculosis may not progress to active disease. It is believed that only 5–10% of infected individuals develop active disease within 2 years of initial infection [9]. The vast majority contain the infection without clearing it (*latency*) by mounting a successful adaptive immune response.

When an individual's immune system is weakened, the chances of TB progressing to an active state become greater. TB is a leading cause of mortality among people living with human immunodeficiency virus(HIV), who have weakened immune systems. About 200 000 people with HIV die from TB every year, most of them in Africa [8].

To prevent active disease the immune system "walls off" the TB bacilli which, protected by a thick waxy coat, can lie dormant for years. Granulomas are thought to be the means by which the adaptive immune response achieves and maintains latency in tuberculosis. Since the majority of infections remain isolated at focal sites, granuloma formation initially occurs when an individual has been diagnosed with pulmonary tuberculosis[10]. There are several interactions involved in the formation of granulomas.

Exposed individuals typically inhale bacteria; the bacteria will make its way to the lung alveolai where they encounter resident macrophages. Macrophages phagocytose bacteria that they may kill, resulting in clearance of initial infection via the innate immune response [11].

However, resident macrophages are not as efficient at killing bacteria as macrophages which are activated for the task. Moreover, M.tuberculosis has evolved mechanisms for

evading killing by its host macrophages [12]. Although there is an opportunity for resident macrophages to clear the infection it is much more likely that host macrophages will become 'infected' as bacteria replicates over time.

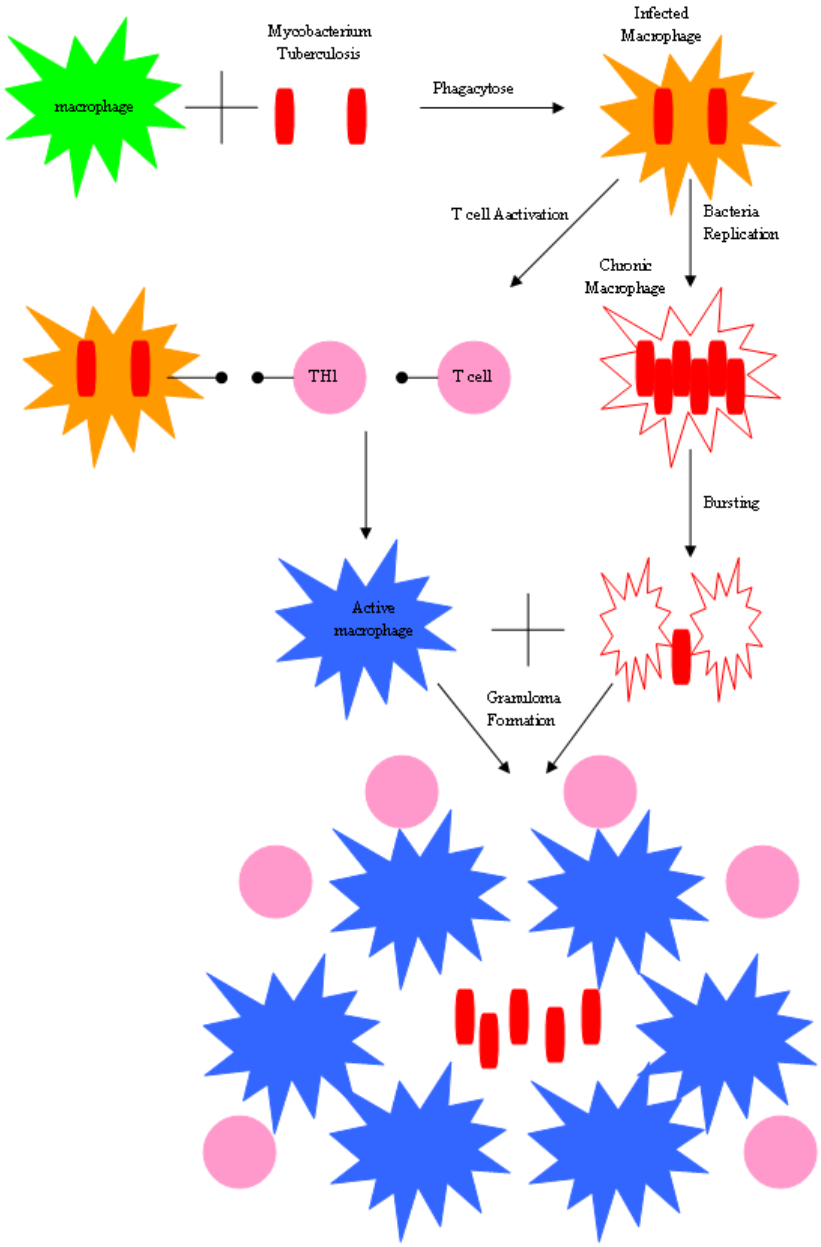
Mycobacteria replicate well within infected macrophages. M. tuberculosis prefer the intracellular environment, and replicate at a faster rate than they do in the extracellular environment [13]. Infected macrophages will eventually burst due to the excessive number of intracellular bacteria. Intracellular bacteria will spread into the surrounding extracellular environment, thus inducing further infected macrophages and contributing to the spread of the infection.

As the disease spreads it is highly unlikely the infection can be cleared, however, it can be contained by the host's immune system. Infected macrophages release cytokines and chemokines that attract dendritic cells and additional macrophages [14, 15].

Dendritic cells engulf bacteria and then migrate to the nearest draining lymph node where they present antigen to naive T cells. This induces differentiation and activation of T cells that subsequently migrate back to the lung and to specific sites of infection. T-cells are guided back to the site of infection by adhesion molecules and chemokine signals produced by infected macrophages [16, 10].

Both CD4+ and CD8+ T cells help control infection by activating macrophages. Activated macrophages are very efficient at phagocytosing and eliminating extracellular bacteria. Only infected macrophages can be activated. When a macrophage becomes chronically infected (nearing bursting point) the macrophage cannot be activated. However, T cells can help eliminate chronically infected macrophages and thus reduce the amount of intracellular bacteria that is released into the environment when the chronically infected macrophage is removed. The combination of activated macrophages, resident macrophages and T cells can contain the infection by forming a granuloma. On the next page is a simplified diagram detailing the process of granuloma formation.

Figure 2.1: Process of Granuloma Formation



2.2 Modelling Mtb Infection

To study the process of Mtb infection several mathematical models have been proposed and implemented. A range of approaches from continuous deterministic models to discrete stochastic models have been discussed. I will explain some of the different approaches analysed and the results obtained, much of this section sources [4].

2.2.1 Baseline Ordinary Differential Equation(ODE) Model

Through the use of nonlinear Ordinary Differential Equations (ODE) a mathematical system was devised based on the interaction of key cells and cytokines to model TB infection in the lung. [2] Both extracellular and intracellular bacteria levels were tracked as well as the following cell populations:

- Lymphocytes (T0, T1 and T2 cells).
- Macrophages (Resting, Activated and Infected).
- Cytokines (IFN-γ, IL-12, IL-10 and IL-4).

Mathematical expressions were developed representing the interactions between cell populations and cytokines. The model was simulated by solving the differential equations using an appropriate numerical method. Below is an example of ODE used for describing resting macrophage dynamics during Mtb infection.

Resting macrophage dynamics: There is a source of new cells coming into the site (s_m) and death of cells that stay the remainder of their life span ($-\mu_r M_R$). When bacteria are present additional resting macrophages are recruited to the site of infection. in the lung in response to chemokine released by activated(M_A) and infected(M_I) macrophages at rates of α_4. The chemokine released by infected macrophages depend on the contribution that intracellular bacteria plays in affecting its chemokine secretion ($0 \leq w \leq 1$). On reaching the site the resting macrophage may become activated in response to (I_γ) at a maximal rate of $k_3$and deactivated at a maximal rate $k_4$and μ_{da}. Resting macrophages can become chronically infected at a maximal rate of k_2. α_{21}represents the max M_R recruitment by resting macrophages. s_3, s_8, c_8, and c_{28} represent half life of Interferon, IL-10, bacteria on M_R activation and bacteria on M_R recruitment by M_R respectively.

$$\frac{dM_R}{dt} = s_m + \alpha_4(M_A + wM_I) + \alpha_{21}M_R(\frac{B_T}{B_T+c_{28}}) + k_4M_A(\frac{I_{10}}{I_{10}+s_8})$$
$$-kM_R(\frac{B_E}{B_E+c_9}) - k_3M_R(\frac{I_\gamma}{I_\gamma+s_3})(\frac{B_T}{B_T+c_8})$$
$$+\mu_{da}M_A(\frac{s_3}{I_\gamma+s_3})(\frac{c_8}{B_T+c_8}) - \mu_rM_R$$

The model indicated that it was possible to be exposed to initial bacteria and then clear the infection with no memory of that response. This is clearly possible since only 30% of individuals that are exposed to Mtb become infected.

It was found that only 6 model parameters out of a possible 99 lead to different infection outcomes when modified. Infection either became latent or progressed to active disease rather than clearance.[2] The 6 model parameters that were identified include:

1. Chronic Macrophage infection rate
2. Macrophage activation rate
3. Maximal T cell killing of infected macrophage
4. Max. killing of extracellular bacteria by active macrophages
5. Rate of INF-γ produced by CD8+ T cells
6. IL-4 production by Th0 cells

It was also found that the role macrophages and their activation as well as T cells were key in determining infection outcome.

2.2.2 Multiple Compartment ODE model

A two-compartmental mathematical model was devised, which extends the previous model by including a physiological compartment (the lymph node) and five new variables each with their own mathematical equation. This extension allowed the model to represent the migration patterns between the draining lymph node and the lung as well as important processes performed by the dendritic cells of antigen presentation and cellular activation.[4]

The results of this ODE model were in line with the results obtained by the previous model. The model highlighted the role macrophages and dendritic cells play in initiating and directing adaptive T cell immunity. The model also indicated that if there is a delay in either the dendritic cell migration to the draining lymph node or T cell trafficking to the site of infection then the outcome of Mtb infection will be altered.

It is believed that fast dendritic cell turnover at the site of infection as well as strong activation by dendritic cells can lead to a maximal antigen presentation and production of key cytokines, which could represent a strategy for the development of a new generation of treatments against Mtb. [21, 20]

2.2.3 Metapopulation Model

The metapopulation framework was applied at the cellular level and studies the spatial aspects of the adaptive immune response to Mtb infection. The spatial domain is a two-dimensional cross section of alveolar lung tissue consisting of $N \times N$ compartments. Each compartment contains sub-populations of various cell types.

Nonlinear ODEs are used to model the local interactions of the sub-populations within each compartment as well as the migration of sub-populations between adjacent compartments. Unlike in previous models for simplicity cytokine variables were removed and their effects incorporated by the cells that produce them namely, macrophages and T cells. [3]

All three T cell populations are represented as a single T cell population. There are still three macrophage classes and both extracellular and intracellular bacteria are monitored. Unlike in previous models resting and activated macrophages as well as T cells have the ability to move. Cell movement is dictated by chemokine, which also has the ability to move. Chemokines were not included directly in previous models that have been discussed.

The ODEs for T cells, resting and activated macrophages include movement terms. Each movement term contains a set of 'movement coefficients.' The coefficients represent the proportion of cells that move up, down left, right and stationary. These coefficients are calculated as functions of the chemokine gradient between a given compartment and its neighbours.

Movement ODE: Assume movement of subpopulations from a given compartment (i,j) to adjacent compartments, that is, up(U) to (i, j-1), down(D) to (i, j+1), left(L) to (i-1, j) and right(R) to (i+1, j). X_w *represents the rate of movement for each cell type w.* α *are a set of movement coefficient which change over time. The coefficients* δ_l *are included do that only the appropriate movement terms appear in the differential equations boundary compartments.*

$$
\begin{aligned}
Mov^{w}_{(i,j)}(t) = & -X_w(1-\alpha^{w}_{i,j,S})w_{(i,j)} \text{Movement out} \\
& +\delta_R\, X_w\, \alpha^{w}_{i-1,j,R}\, w_{(i-1,j)} \text{Movement in from left} \\
& +\delta_L\, X_w\, \alpha^{w}_{i+1,j,L}\, w_{(i+1,j)} \text{Movement in from right} \\
& +\delta_D\, X_w\, \alpha^{w}_{i,j-1,D}\, w_{(i,j-1)} \text{Movement in from above} \\
& +\delta_U\, X_w\, \alpha^{w}_{i,j+1,U}\, w_{(i,j+1)} \text{Movement in from below}
\end{aligned}
$$

Through the metapopulation model simulations, all three disease outcomes were obtained clearance, latency and active disease depending on the bacterial load in the centre of the lattice. However, clearance was only achieved through innate immunity where resting macrophages kill off infection before T cells arrive at the site of infection.

Clearance was not achieved through adaptive immunity where T cells have also arrived to help fight the infection. The model did demonstrate pseudo-clearance where bacterial load remains at such low levels for 150-400 days. This allows small but nonzero

bacterial numbers to grow again, which re-triggers the adaptive immunity, reducing the bacterial load to negligible levels again. This cycle repeats every 1 -2 years. [3]

Disease progression depends on how well the immune system responds to initial infection. Parameter values were found for both latency and active disease. It was found that increasing either the growth rate of T cells or macrophage activation rate lead to clearance. The increase in the number of T cells contributed to the increase in the number of infected macrophages dying as well as increasing the number of active macrophages. Both factors were found to limit both the intracellular and extracellular bacterial load.[4]

Another factor which was found to affect latency or pseudo-clearance was chronic macrophage infection rate. As chronic macrophage infection rate decreased the model outcome changed from latency to pseudo-clearance. If initially there are a large number of resting macrophages on the lattice then there will be higher initial levels of infected macrophages, leading to a stronger chemokine signal. The higher levels of T cells induced by the chemokine signal balances the higher rate of infection. Moreover, the higher levels of active macrophages limit extracellular bacteria growth.[4]

It was also found that the rate of infected macrophage death due to bursting significantly affected the outcome of the model. If the rate of infected macrophage death due to bursting is low then there is a decrease in the number of T cell deaths of infected macrophages, leading to active disease due to the large amounts of bacteria becoming extracellular. If the rate of infected macrophage death due to bursting is large then infected macrophages burst more rapidly. Intracellular bacteria have less time to replicate and then, depending on the rate at which active macrophages kill extracellular bacteria, can clear the bacteria.

2.2.4 Partial Differential Equations (PDE) Model

The Partial Differential Equation (PDE) model was developed to further investigate the innate immune response to Mtb infection. The PDE model also looks at granuloma size, growth as well as recruitment of resting macrophages to the site of infection, cell movement inside the granuloma and phagocytosis. The PDE model is one-dimensional.

The metapopulation model described above is a discretized version of a PDE model. In the metapopulation model it was assumed that bacteria were confined to a single compartment, which does not help estimate the size of the granuloma or the importance of bacteria movement parameters. [4]

The PDE model contains five dependent variable types:

1. Extracellular bacteria.

2. Macrophages(uninfected and infected).

3. A bacteria produced Chemokine.

4. Internal velocity of cells within the granuloma.

5. Granuloma radius.

In previous models there were different classes of macrophages such as resting, infected etc however, in the PDE model a ladder of infection is used. A macrophage is given by $M_w(t,x), w = 0, \ldots, N$ where w is an integer which denotes the number of bacteria inside a particular macrophage .

It is assumed that extracellular bacteria release a chemokine that attracts resting macrophages to the growing granuloma. Macrophages are then able to phagocytose and kill bacteria. However, bacteria could evade killing and replicate within the host macrophage. If the macrophage contains too much bacteria it loses functionality and it is assumed that it loses the ability to phagocytose extracellular bacteria and kill intracellular bacteria once intracellular bacteria reaches some critical threshold value. [4, 22] It is also assumed that chemokine diffuses through tissue and cell types move via diffusion. Cells in the granuloma are affected by an internal velocity due to bacterial growth, phagocytosis, death and other movement in and out of the granuloma.

The PDE model simulations found that phagocytosis, intracellular bacterial killing and both extracellular and intracellular bacteria growth have an important effect on the development of a granuloma.

Decreasing rates of phagocytosis and bacterial killing lead to granuloma growth. This is because bacteria survive and proliferate. Granuloma growth is also induced when bacteria growth rates are increased as there are a greater number of bacteria that cannot be dealt with by the innate immune response. [22]

It was also found that if the chemokine value at the boundary of the granuloma decreased, macrophage recruitment also decreased. This allowed continued growth of the bacteria, thus encouraging granuloma growth. Decreasing the rate of macrophage chemotaxis (movement of resting macrophage towards chemokine gradient) gave an advantage to bacteria as macrophages recruited into the granuloma are not moving directly toward bacteria.[4]

2.2.5 Conclusion of Previous Models

It is important to note the scales at which the above models were developed. The ODE models give a measure for the overall lung environment and thus are a more global

indicator of infection status. Since neither ODE model represented spatial domain, chemotactic signals such as chemokine were not directly included as cell movement was not explicitly stated.

The metapopulation and PDE frameworks both model Mtb infection locally. They represent the immune response at a single focal point of infection since the spatial domains of the models represent a small portion of lung tissue and an initial infection within the inspected lung tissue. The ABM will model Mtb infection locally allowing us to include chemotactic signals required for cell movement. There are a number of reasons why the models have moved away from mathematical expressions and towards mathematical and computational models.

Mathematical models have been unable to capture the spatio-temporal heterogeneity of individual cells such as macrophages and T cells. [4] Our approaches will also allow us to define more clearly individual interactions among entities at the cellular level and observe resulting dynamics at the tissue level. A limiting factor for mathematical approaches is the number of mathematical terms that are required to represent all of the relative interactions. As the number of mathematical equations increase the model becomes more computationally infeasible. Table 2.1 summarises the different models discussed and how the approach I will be implementing differs from the above models.

Table 2.1: *Table Summary Abbrev: C = Continuous, D = discrete, S = Stochastic, DE = deterministic, CC = Computational Complexity, MC = Math Complexity, BC = Biological Complexity, ↑ = Expensive, ↑↑ = v.Expensive, ↓ = inexpensive.*

Model Type	**Dynamics**	**Environment**	**Spatial Dom**	**Complexity**
Baseline ODE Model	C, DE	Lung (Global)	N	CC↓, MC↑
2-Compartment ODE Model	C, DE	Lung (Global) and Lymph node	Y	CC↓, MC↑
Metapopulation Model	C, S, DE	Lung Tissue (Local)	Y	CC↑, MC↓
PDE Model	C, DE	Lung Tissue (Local)	Y	CC↑, MC↑, BC↓
Agent Based Model	D, S	Lung Tissue (Local)	Y	CC↑↑, MC↓, BC↑

2.3 Agent Based Model

Cellular automata models have been applied widely within theoretical biology. Agent-based modelling is an extension and outgrowth of cellular automata models.

Agent-based models consist of elements of the system, which are represented primarily as discrete agents with several unique attributes. Individual agents reside in an explicitly represented spatial environment. They interact with one another and with the environment according to sets of rules.

The above mathematical models have all been based on continuous and deterministic interactions at the global or local level. In an agent-based model, the local, possibly stochastic, individual-level interactions give rise to global, system-wide dynamics and patterns. Thus, ABMs are particularly useful for studying complex systems in which individual heterogeneity and spatial interactions are important allowing for greater biological realism[1].

Extracellular bacteria and chemokines are continuous variables within this model whereas T cells and macrophages are discrete agents. Chemokines will create a field, which attracts macrophages and T cells to move towards higher chemokine concentrations. Therefore larger lattices do not affect the dynamics of T cells and macrophages or the outcome of the infection.

Unlike previous models there are four macrophage classes. The ABM makes a clear distinction between chronically infected macrophages which are defined as those macrophages that cannot be activated and contain many more intracellular bacteria than infected macrophages, which can still be activated by T cells.

The agents themselves are reactive agents. The agents will observe the environment before reacting in accordance to their predefined rules. Agents are only awoken when they are expected to perform some task. Agents are awoken to move in a particular direction or to execute some rule within their internal state depending on the environment. This may include doing nothing.

This approach has already been implemented for Mycobacterium Tuberculosis and the results discussed in the paper [1]. I have implemented the same model in Java rather than C/C++. By implementing this system I have gained a familiarity with the problem domain allowing me to modify the model to simulate atherosclerosis.

2.4 Specification

The previous section provided a detailed description of existing work and information on the problem domain. I will now discuss the overall requirements and aims of the implementation. Much of the specification has been sourced from the rules outlined in [1]. I have chosen to implement the system in Java. There are a number of reasons for this choice.

Java's object oriented approach allows implementation of agent rules to be fairly easy. It also allows me to incorporate MASON (Multi-Agent Simulator Of Neighbourhoods). MASON is a fast discrete-event multi-agent simulation library core in Java, designed to be the foundation for large custom-purpose Java simulations. MASON contains both a model library and an optional suite of visualisation tools in 2D and 3D. [23]

There are a number of beneficial features provided by MASON that lends itself to being an appropriate tool kit:

- Models are completely independent from visualisation, which can be added, removed, or changed at any time.
- Models may be check pointed and recovered, and dynamically migrated across platforms.
- Can generate PNG snapshots, Quicktime movies, charts and graphs, and output data streams.

Also with the use of inheritance and polymorphism I can segregate rules for each type of macrophage. I have also separated the simulation logic from the graphical depiction of the simulation. It was also important to use java primitive types whenever possible to improve efficiency.

2.4.1 Model Description

The model has the following key components:

1. The environment: A $2mm \times 2mm$ cross section of lung tissue, where entities reside.
2. Entities: Both agents representing macrophages and T cells and continuous chemokine and bacteria variables
3. Rules: These determine the dynamics of the system and represent the interactions between the different entities
4. Time Scale: Indicating which rules are executed.

2.4.1.1 Environment

The environment I am modelling represents a portion of the alveolar lung tissue. The environment will be represented as a $N \times N$ lattice of micro-compartments. The lattice is a torus allowing us to avoid boundary effects.

Each micro-compartment contains at most a single macrophage and T cell as well as any extracellular bacteria and chemokine. The dimensions of each micro-compartment are $2\mu m \times 20\mu m$. This is because the diameter of a human alveolar macrophage is approximately 20µm [17], which is the largest entity that we will represent in our model. All other entities have negligible size.

The dimensions of the entire lattice are $2mm \times 2mm$, which corresponds to dimensions 100×100. These dimensions have been chosen because dissemination and progression to disease has been observed with large necrotic granulomas with a diameter of 2mm or more. This has been noted in both humans [11] and more recently in non-human primates [18]. Containment has been associated with smaller solid granulomas of diameter 2mm or less as observed by non-human primates infected by Mtb[18].

The environment also consists of 4 source compartments. These compartments represent locations where blood vessels enter the lung tissue. Each source compartment can introduce new macrophages or T cell to the lattice depending on the chemokine gradient at the source compartment.

There are four entities that must be modelled. I will detail the rules for each one below.

2.4.1.2 Entities

Chemokine

In both models chemokines will act as attractors for macrophages and T cells. The model uses a single generic chemokine to represent chemoattractants as specified in the paper [1]. Chemokine is secreted by active, infected and chronically infected macrophages. Initially there is no chemokine on the lattice.

The chemokine concentration at any particular micro-compartment is represented as a continuous variable. The chemokine concentration value C_{ij} will diffuse and decay over time. The concentration value for each micro-compartment will diffuse to and from the four micro-compartments in its immediate Von-Neumann neighbourhood. The following formulae should be used to determine the diffusion of the chemokine concentration value C_{ij} for readability I abbreviate the subscripts of the four neighbours by N, E, S, W.

$$C_{i,j}(t+1) = C_{i,j} * (1-\lambda) + \lambda * (C_N + C_E + C_S + C_W)/4$$

The λ is a diffusion constant and is the proportion of C_{ij} that diffuses out of micro-compartment (i,j) during each time-step. Below is the decay formula to model chemokine degradation this is where at each time-step a proportion δ decays.

$$C_{i,j}(t+1) = C_{i,j} - \delta * C_{i,j}$$

Extracellular Bacteria

Initially we insert 16 extracellular bacteria into four adjacent micro-compartments closest to the centre of the lattice. i.e. $BE_{49,49} = BE_{50,49} = BE_{49,50} = BE_{50,50} = 4$. The bacteria are added to the centre of the lattice allowing for easy visualisation.

A macrophage becomes infected when it contains 10 bacteria. We assume that a micro-compartment can only hold an upper bound of 200 extracellular bacteria and bacteria do not diffuse to other micro-compartments. Extracellular bacteria follows the following growth formula with respect to maximum number of extracellular bacteria a micro-compartment can hold.[1]

$$B_E(t+1) = B_E(t) + \alpha_{BE} * (1 - (B_E(t)/K_{BE} * 1.1))$$

T Cells

T cells can share a micro-compartment with a macrophage. T cells only have age and position attributes. T cells are not allowed to enter the lattice until day 10 of infection. This captures recruitment of immune cells to the site of infection in response to chemotactic signals[1]. Each T cell is recruited via a source micro-compartment and has a lifespan between 0 and 3 days [19]. The T cell is removed from the lattice when its age reaches its lifespan.

T cells move depending on the local chemokine gradient. T cells only recognise chemokine values of $C_{ij} > 1$. T cells movement is a biased random walk with probabilities calculated as a function of the chemokine concentration of the eight neighbouring (Moore neighbourhood) micro-compartments [1]. If the new location is empty then the T cell moves to the new micro-compartment. If the new location contains a T cell then the T cell remains stationary. If the new location contains a macrophage then the T cell moves to the new location with a given small probability this is to capture the crowding effects imposed by macrophages on T cell movement.

Using this approach we combine the effects of CD4+ and CD8+ T cells into a composite T cell population. CD4+ and CD8+ T cells perform different cytotoxic function leading to the death of infected macrophages. This is captured in the rules of the following section. [1]

Macrophages

There can only be at most one macrophage in any particular micro-compartment. Macrophages have the following attributes:

- Position
- Age
- Number of intracellular bacteria(B_I) such that $0 \leq B_I < K_{BI}$ where K_{BI} is a parameter representing the maximum bacterial carrying capacity of macrophages.

Like T cells macrophages are recruited via source micro-compartments depending on the chemokine value at the source cell. The threshold value to determine whether a macrophage or T cell is recruited is currently set to 1.

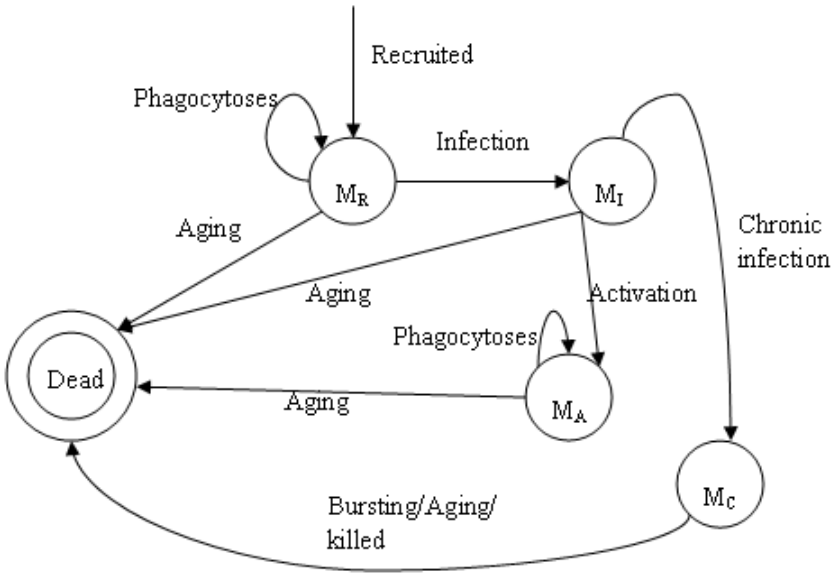
Each Macrophage is randomly assigned a lifespan between 0 and 100 days and is removed from the lattice when it reaches its lifespan. Macrophages move chemotactically similar to T cells as described in the previous section. However, macrophages move more slowly than T cells. If there is no chemokine on the lattice Macrophages move randomly otherwise they walk biased towards neighbouring micro-compartments with higher concentration of chemokine.

There are four types of macrophage:

- **Resting Macrophage**: These are initially placed on the lattice and recruited via source micro-compartments. Resting macrophages take up bacteria and either clear them or become infected.
- **Infected Macrophage**: Infected macrophages occur when resting macrophages become infected with a small number of intracellular bacteria.
- **Chronic Macrophage**: When the number of intracellular bacteria within an infected macrophage grows above a certain threshold, the macrophage is classified as chronic. The macrophage may burst due to excessive number of intracellular bacteria or killed by T cells.
- **Active Macrophage**: Infected macrophage can become active via T cells. Active macrophages are extremely good at clearing extracellular bacteria.

The rules and algorithmic details for the above macrophages are detailed in the rules section. Figure 2.2 depicts the interactions between the different macrophage types.

Figure 2.2: Macrophage State Diagram



2.4.1.3 Time Scale

Each time step corresponds to 6 seconds of 'real time.' The state of the system at time-step $t+1$ is computed by applying a set of rules to the system at time step t. Most rules occur at a slower timescale than chemokine diffusion and decay, which occurs every time step. Extracellular bacteria growth is calculated every minute thus 10 time-steps. T cells, resting macrophages, infected macrophages and active macrophages move approximately every 10, 100, 142857 and 4000 time-steps respectively.

Simulations should be performed for 200-500 days. The typical time frame for development of a granuloma is anywhere between 14 to 100 days in non-human primates [18].

2.4.1.4 Rules

Below are the algorithmic rules obtained from [1] for the entities within the models and an explanation of the algorithm.

Resting Macrophage:

Numbers in brackets(?) represent line numbers with associated algorithm.

If there are a small number of extracellular bacteria present in the same micro-compartment as a resting macrophage N_{RK} then the macrophage phagocytoses and kills those bacteria and remains in the resting state(1-3). If there are a large number of extracellular bacteria then the resting macrophage may still successfully remove the extracellular bacteria by some probability p_k(5-8). However, it is more likely that it cannot remove the bacteria and the macrophage becomes infected(12-14).

Algorithm 2.1 Resting Macrophage Rules

1: if $B_E i, j(t) \leq N_{RK}$ then
2: $B_E i, j(t+1) = (B_E i, j(t) - N_{RK}) \geq 0$
3: return $\mathrm{M_R}$; Resting macrophage
4: endif
5: if $B_E i, j(t) > N_{RK}$ then
6: $r \leftarrow RandomUniform[0, 100]$
7: $B_E i, j(t+1) = (B_E i, j(t) - N_{RK})$
8: if $r \leq p_k$then
9: $\mathrm{M_R}$ succeeds in killing the bacteria and remains in resting state
10: return $\mathrm{M_R}$; Resting macrophage
11: else
12: $B_I i, j(t+1) = N_{RK}$
13: $\mathrm{M_R}$does not succeed in killing bacteria The macrophage is infected.
14: return $\mathrm{M_I}$; Infected macrophage
15: endif
16: endif

Infected Macrophage:

Numbers in brackets(?) represent line numbers with associated algorithm.

At every time-step each infected macrophage releases a given amount $ci = 5000$ units of chemokine into its micro-compartment(2). Intracellular bacteria is replicated within each infected macrophage(4), if intracellular bacteria increases above a threshold N_c the macrophage becomes chronically infected(6,7). However, the macrophage can be activated if there are sufficient T cells in or around the micro-compartment an infected macrophage resides. Activation depends on the number of T cells in the immediate (Moore) neighbourhood of an infected macrophage (with a maximum number N_{tact}) and the probability that T cells will activate a macrophage (T_{actm})(11-22).

Algorithm 2.2 Infected Macrophage Rules

1: Chemokines released by macrophage:
2: $C_{i,j}(t+1) = C_{i,j}(t) + c_I$
3: Intracellular bacterial replication:
4: $B_I(t+1) = B_I(t) + \alpha_{BI} * B_I(t)$
5: Chronic infection:
6: if $B_I(t) > N_c$ then
7: return M_C; Chronically infected macrophage
8: endif
9: Macrophage activation:
10: if $B_I(t) \leq Nc$ then
11: N_T= Number of T cells in the neighbourhood
12: if $N_T > N_{tact}$ then
13: $N_T = N_{tact}$
14: endif
15: If the macrophage is activated, then it eliminates its intracellular bacterial load:
16: $r \leftarrow RandomUniform[0, 100]$
17: if $r < (N_T * T_{actm}) \leq 100$ then
18: $B_I i, j = 0$
19: return M_A; Activated Macrophage
20: else
21: return M_I; Infected Macrophage
22: endif
23: endif

Chronic Macrophage:

Numbers in brackets(?) represent line numbers with associated algorithm.

Like infected macrophages chronic macrophages secrete chemokine into their micro-compartment(2). Chronic macrophages replicate intracellular bacteria with respect to the intracellular bacteria carrying capacity of macrophages K_{BI}(4). If the intracellular bacterial load of a chronically infected macrophage exceeds the carrying capacity the macrophage bursts and the intracellular bacteria is released into the current and neighbouring micro-compartments(6-11). However, if this is not the case and there exists a T cell in the same micro-compartment as the chronically infected macrophage(14). The T cell can kill the macrophage with probability p_{Tk}(16). We assume 50% of the bacteria load is killed and the remainder spreads to the current and neighbouring compartments(17-22).

Algorithm 2.3 Chronic Macrophage Rules

1: Chemokines released by macrophage:
2: $C_{i,j}(t+1) = C_{i,j}(t) + c_I$
3: Intracellular bacterial replication:
4: $B_I(t+1) = B_I(t) + \alpha_{BI} * B_I(t)/(1 + (B_I(t) = (K_{BI} + 30)))$
5: Bursting:
6: if $B_I(t) > K_{BI}$ then
7: $B_{Ei-1,j}(t+1) = (B_{Ei-1,j}(t) + B_I(t) = 9$
8: $B_{Ei+1,j}(t+1) = (B_{Ei+1,j}(t) + B_I(t) = 9$
9: $B_{Ei-1,j+1}(t+1) = (B_{Ei-1,j+1}(t) + B_I(t) = 9$ etc:
10: Macrophage is eliminated
11: return Death
12: endif
13: T cell killing:
14: if $(B_I(t) \leq K_{BI})$ *and* $\exists T - cell$ *at* (i,j)then
15: $r \leftarrow RandomUniform[0, 100]$
16: if $r < (p_{Tk})$ then
17: $B_{Ii,j}(t) = (P_{kill}/100) * B_{Ii,j}$
18: $B_{Ei-1,j}(t+1) = B_{Ei-1,j}(t) + B_I(t)/9$
19: $B_{Ei+1,j}(t+1) = B_{Ei+1,j}(t) + B_I(t)/9$
20: $B_{Ei-1,j+1}(t+1) = B_{Ei-1,j+1}(t) + B_I(t)/9$ etc:
21: Macrophage is eliminated
22: return Death
23: else
24: return M_C; Chronic Macrophage
25: endif
26: endif

Active Macrophage:

Numbers in brackets(?) represent line numbers with associated algorithm.

As with infected and chronically infected macrophages, active macrophages also secrete chemokine into its micro-compartment(2). Active macrophages are highly efficient at phagocytosing and eliminating extracellular bacteria. We assume a certain number N_{phag} of extracellular bacteria are eliminated from a micro-compartment where an active macrophage resides(6).

Algorithm 2.4 Active Macrophage Rules

1: Chemokines released by macrophage:
2: $C_{i,j}(t+1) = C_{i,j}(t) + c_I$
3: if $B_{Ei,j}(t) < N_{phag}$then
4: $B_{Ei,j}(t+1) = 0$
5: else
6: $B_{Ei,j}(t+1) = B_{Ei,j}(t) - N_{phag}$
7: endif

Efficiency is extremely important as a 200 day simulation needs to run within 16 minutes. By using Java I was able to use profiling tools to help identify bottle necks and areas of code that if altered can provide the greatest performance increase.

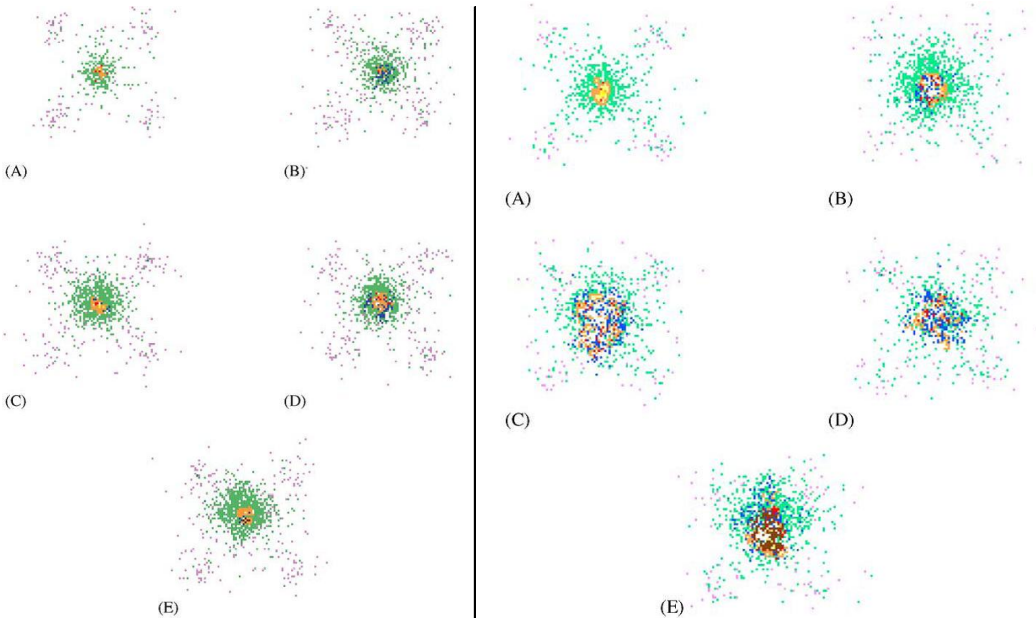
Depending on the values set for the parameters previously listed the simulation should show all three outcomes, clearance, latency and dissemination. To ensure that the system has been correctly implemented the following table of parameters(taken from paper [1]) should be used to test if the system does indeed simulate containment using column A parameters and dissemination using column B parameters.

Table 2.3: *Model Parameters*

Symbol	Parameter Description	A	B	Units
λ	Chemokine diffusion coefficient	0.64	0.65	0.1min
δ	Chemokine degradation coefficient	0.001	0.0004	0.1min
p_k	Prob. of bacteria being killed within M_R	8.51	2.36	%
pT_k	Prob. T cell kills a macrophage	6.31	3.61	%
P_{kill}	% of BI being destroyed by killing	50	50	%
K_{BE}	Carrying capacity (B_E) of micro-compartment	200	200	Scalar
α_{BI}	Intracellular bacteria growth rate	0.00021	0.00049	min
α_{BE}	Extracellular bacteria growth rate	0.00015	0.00015	min
N_c	No. of intracellular bacteria defining transition to chronically infected state	10	10	Scalar
K_{BI}	No. of bacteria that make a macrophage burst	20	20	Scalar
N_{tact}	No. of T cells needed to activate a macrophage	4	5	Scalar
T_{recr}	Prob. of T cell recruitment	32	12.31	%
T_{move}	Prob. of T cell movement	4.97	5.26	%
T_{actm}	Prob. a T cell will activate a macrophage	6	12.2	%
T_{ls}	T cell lifespan	3	3	days
T_{delay}	T cell delay	10	10	days
M_{recr}	Prob. of macrophage recruitment	2.11	6.82	%
M_{rls}	Resting macrophage lifespan	100	100	days
M_{als}	Activated macrophage lifespan	10	10	days
T_{sp}	T cell speed	10	10	μm/min
M_{rsp}	Resting macrophage speed	1	1	μm/min
M_{asp}	Activated macrophage speed	0.025	0.0632	μm/min
M_{isp}	Infected macrophage speed	0.0007	0.0007	μm/min
M_{init}	Initial No. of resting macrophages	105	97	Scalar
N_{RK}	No. of bacteria killed byM_R	2	2	Scalar
N_{phag}	No. of bacteria killed by M_A	10	10	Scalar

I should receive similar images to the following for containment and dissemination. Since each simulation is uniquely different due to random probabilities I will not receive images that are exactly the same as those depicted by [1] but these images provide an approximation as to what should be occurring at the specified time intervals.

Figure 2.3: *Taking parameters from column A(left) and column B(right) an ABM produced the following captions of granuloma formation at days 12(A), 25(B), 50(C), 100(D) and 200(E). Containment is depicted on the left and dissemination is depicted on the right. The colours represent resting macrophages(green), activated macrophages(blue), infected macrophages(orange), chronically infected macrophages (red), T cells (pink), necrotic tissue (brown), and extracellular bacteria (yellow).*



Overall Aims

- Java based implementation.
- Simulation to run between 200-500 days.
- Simulation should show all three outcomes of clearance, latency and dissemination.
- User should be able to vary parameters via a text file
- User should be able to run a simulation with or without a visible console.

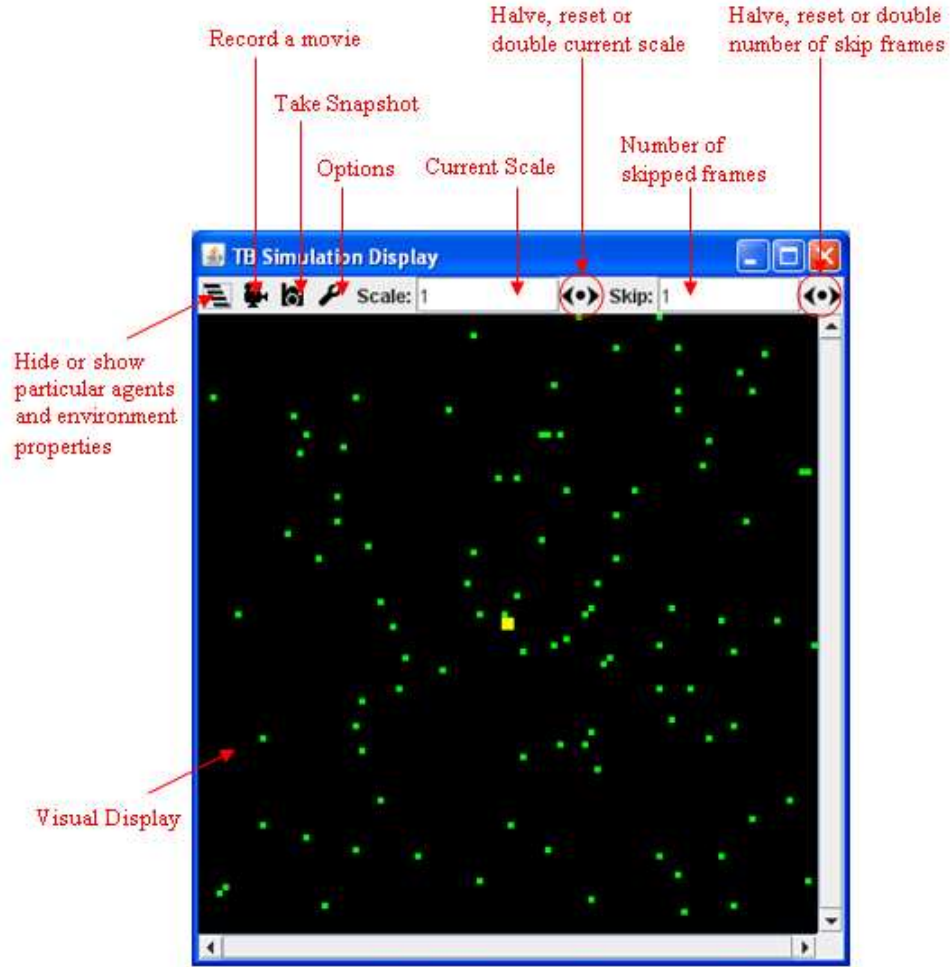
- Graphical window should display Resting Macrophages, Infected Macrophages, Chronic Macrophages, T cells and Extracellular Bacteria.
- User should be able to pause and stop simulation at any time.
- As a bench mark a 200 day simulation of latency should run without the console within 16 minutes.

2.5 Implementation

The simulation was implemented in accordance to the specification set out above. The simulation can be run in one of two ways, with or without a visual console. The user can input parameters in one of two ways depending on whether the visual console is present. The simulation runs within 16 minutes without the visual console.

If the simulation is run without the visual console, the simulation will checkpoint files every 10 days. These checkpointed files can then be loaded into the visual display and run from that point in the simulation. The GUI was provided by the MASON toolkit which I integrated to run with my simulation. To run the simulation with the GUI visible, the user must run the VisualSimulation.java file. To run the simulation without the GUI the user must run Simulation.java. The visual console can be hidden when running the simulation with the GUI. I have detailed in Figure 2.4 the different aspects of the display console.

Figure 2.4: Visual Display Console



Alongside the visual display appears a console window, where the user can change different properties of the simulation. There are five tabs in the console window, the user can play, pause or stop the simulation from any of the different panels within the console.

The Play button allows the user to start a simulation. If the simulation is paused, pressing the play button advances the simulation a single time step. The Pause button allows you to pause the simulation. Once paused, the simulation may be either advanced by a single step (by pressing the play button), or continued by pressing the pause button again. The Stop button completely stops a simulation. Once stopped, a

simulation cannot be restarted. The other two labels on the console show the current time step of the simulation, and the frame/second rate.

2.5.1 Console Panel

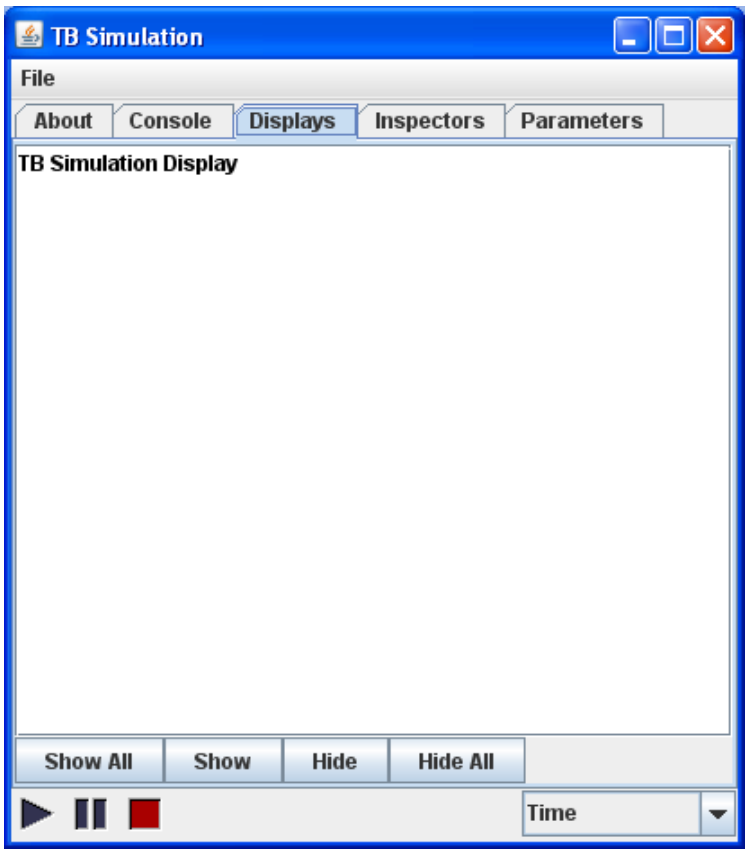
The Console panel contains useful settings for the simulation. The first three slides allow you to insert some delay time between each time step (in case the simulation is too fast), to increase the priority of the simulation thread, and to execute any number of steps upon pressing the play button (when the simulation is paused). There are three text areas where you can specify the time step where the simulation should stop or pause (once stopped, a simulation cannot be resumed), or set the random seed for the simulation to a certain value. By default, the random seed is incremented whenever the simulation is stopped (so that the next time the simulation is played, it runs with a different seed). Setting the last check button, Repeat Play on Stop, sets MASON to automatically start a new simulation whenever the old one is stopped.

Figure 2.5: Console Panel

2.5.2 Display Panel

The Displays panel allows you to select which displays will be shown, and which will not. The simulation runs much faster without the visualisation component. As such, you may choose to hide the display, at least until the emergent effect occurs (for example, the agents cluster to some specific locations), and show the display later on. To hide a display, the user needs to select it in the list of displays, and then press the Hide button. Alternatively, clicking on the displays close button will hide that display. To show a display, the user needs to select it in the list of displays, and then press the Show button. The user may show/hide all displays by pressing the Show all / Hide all buttons. We can have multiple displays open for different TB simulations running at the same time.

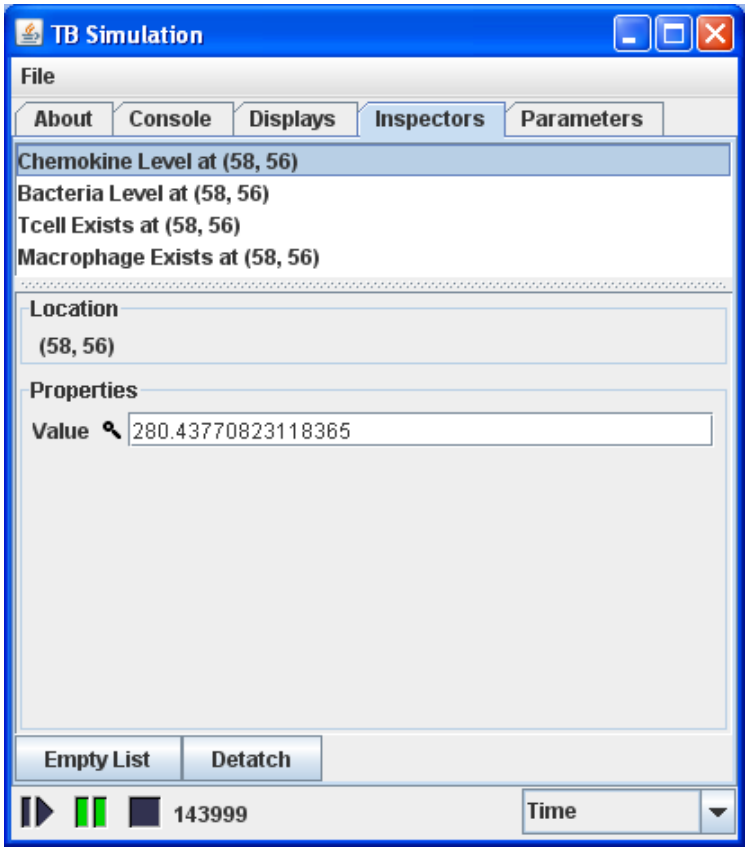
Figure 2.6: Display Panel



2.5.3 Inspector Panel

The Inspectors panel allows the user to inspect objects in the simulation. In this simulation by double clicking on any area of the visual display an inspector panel will open for that cell. It will indicate the chemokine and extracellular bacteria level at that location, as well as indicate which type of macrophage exists at the location and whether a t-cell is present.

Figure 2.7: Inspector Panel

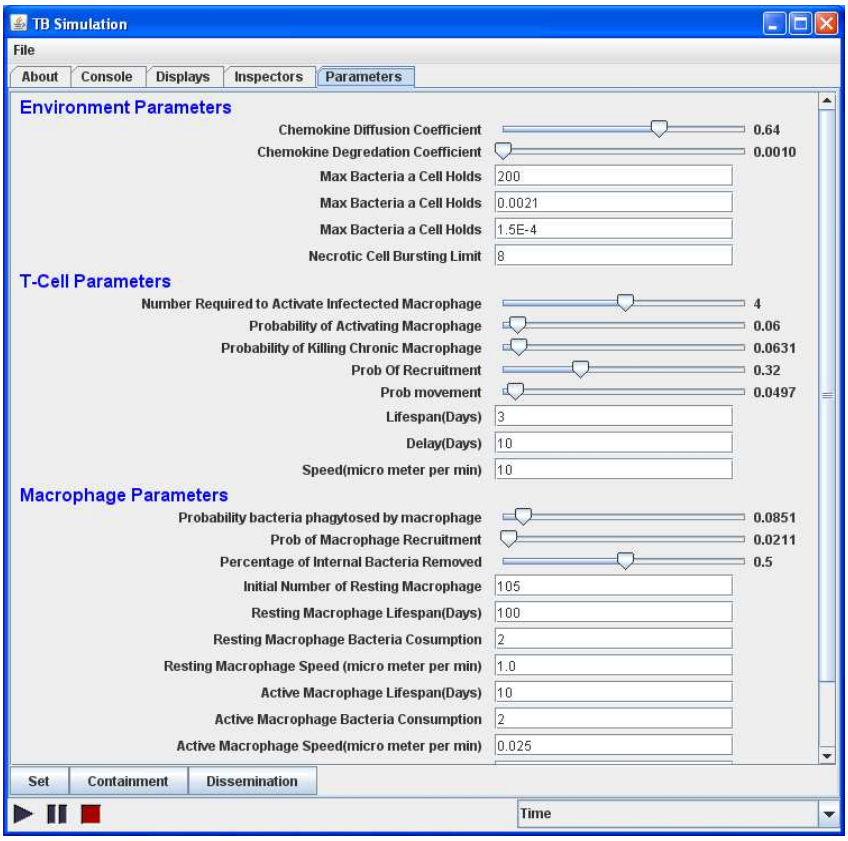


2.5.4 Parameters Panel

The user can modify the different parameters in the model. By using sliders I have ensured that the value entered by a user is valid. For text fields I have added validation checks to ensure that only numbers are entered. Prior to the start of the simulation

the user can modify the parameters and select the 'Set' button to confirm changes to model parameters. The user can select the 'Containment' tab to set the parameters to column A values as outlined in Table 2.2. The user can select the 'Dissemination' tab to set the parameters to column B values.

Figure 2.8: Parameters Panel



For efficiency reasons each cell in the lattice is not represented as a single object. There is a single Grid class which represents the entire lattice environment. We update all cells in a given step together rather than going through each cell object individually.

We update the grid every time step and thus it is the most computationally expensive event in the system. Therefore it was essential to minimise the number of computational rules within the time step. I used 1D arrays of a suitable primitive type to represent the chemokine value and bacteria level for each cell in the lattice. I also did this to represent the presence of a particular type of Macrophage and/or the presence of a T-cell in a given compartment.

By using 1D arrays to represent the entire grid I could access each compartment by simply indexing. E.g a compartment with coordinates (x,y) translates to the index $x * LatticeSize + y$. By using simple primitive 1D arrays we also reduce memory consumption, which becomes problematic as the number of agents that enter the environment increase.

Numbers in brackets(?) represent line numbers in algorithm 2.5.

Below is the outline for the algorithm used in the Grid step.

Algorithm 2.5 Grid step

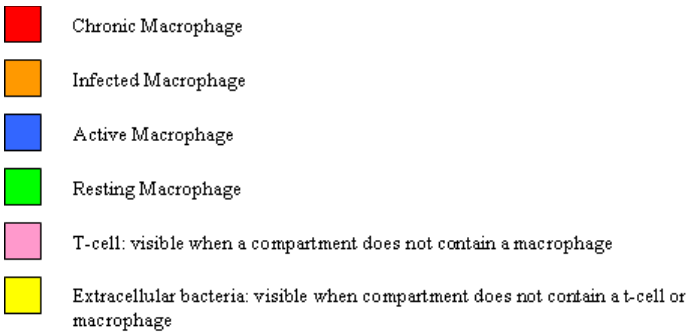
1: if(*timestep*){
2: foreach cell
3: lookup VonNeumann Neighbourhood for cell
4: $chem(t+1)_{(i,j)} = chem(t)_{(i,j)} + \lambda * (C_N + C_E + C_S + C_W)$
5: $chem(t+1)_{(i,j)} = \delta * chem(t)_{(i,j)}$
6: $B_E(t+1) = B_E(t) + \alpha_{BE} * (1 - (B_E(t)/K_{BE} * 1.1))$
7: endfor
8: else {
9: foreach cell
10: $chem(t+1)_{(i,j)} = chem(t)_{(i,j)} + \lambda * (C_N + C_E + C_S + C_W)$
11: $chem(t+1)_{(i,j)} = \delta * chem(t)_{(i,j)}$
12: endfor
13: endif

Through the use of Jprofiler I was able to identify performance bottlenecks. One such bottleneck was computing the Von-Neumann neighbourhood when computing the amount of chemokine that diffuses into each cell from its neighbouring cells(3). This proved to be an expensive process as the lattice is a toroidal environment. I found that by precomputing the moore-neighbourhood for each cell which is also required for the T-cell and InfectedMacrophage classes, it considerably reduced computation time.

The 1D arrays are also used as an interface to the graphical user interface(GUI). Depending on the combination of values within the arrays the pixel for each compartment will be a differing colour. Unfortunately MASON the java GUI plug-in interface does not expect java primitives as input into the graphical classes. It expects the programmer to use in-built classes, however, I found that these reduced performance times considerably. Therefore I modified MASON source code to accept java primitives as input into the graphical display. Figure 2.9 expresses the colour coding for each entity.

Figure 2.9: Colour Chart

Each cell can only contain a single Macrophages thus macrophage colours do not overlap



There are four source cell compartments in the lattice. They are situated at coordinates (25,25), (25,75), (75,25) and (75,75). Resting macrophages and T-cells are recruited from these source compartments. They are only recruited if the chemokine gradient at the source compartment is above 1. There is also an additional 10 day delay for T-cells before they can enter the lattice.

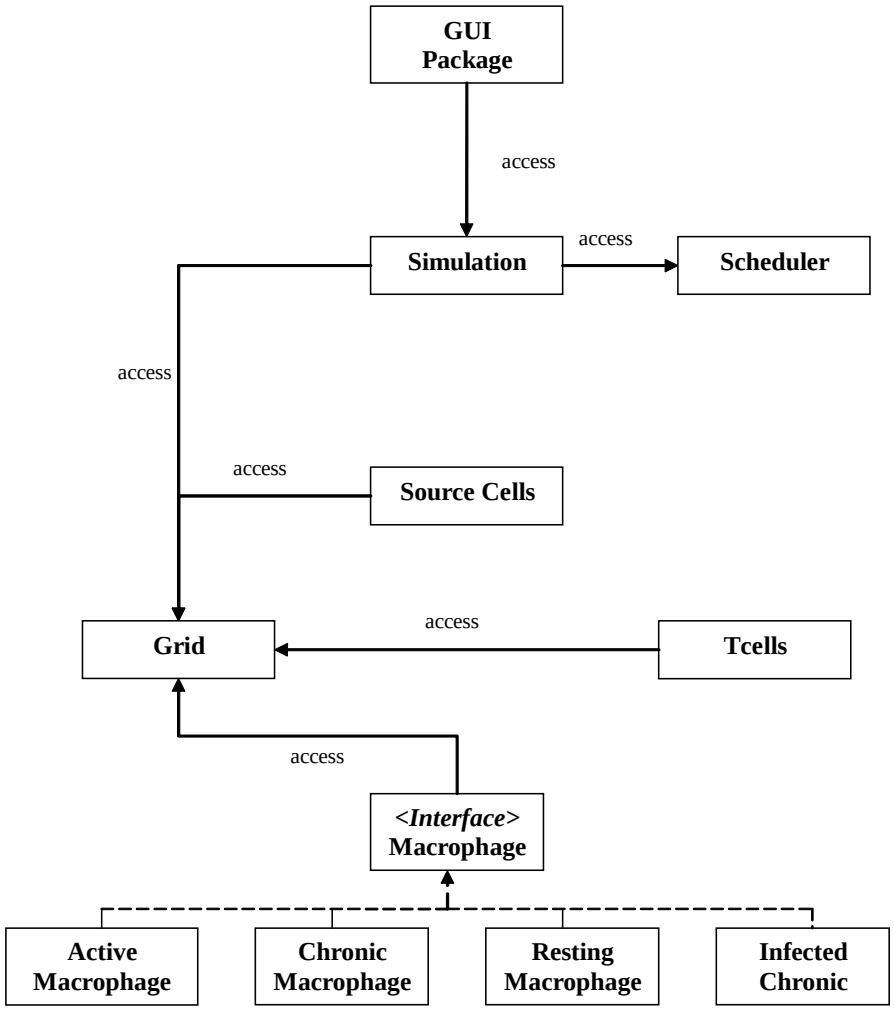
The Simulation class represents the simulation proper, and extends a MASON SimState class which contains a random number generator and the simulator's schedule. The simulation is serializable therefore we can checkpoint the simulation to a file and read from checkpoints. Each object in the simulation is serializable. This allows us to quickly obtain snapshots of the simulation at varying time points without having to watch the entire simulation. We can also continue a simulation from a given time point.

The underlying queueing mechanism of the scheduler is implemented by MASON as a binary heap. Although it would be more efficient to implement my own queueing mechanism I found that the binary heap gave a reasonable result.

Each macrophage type is represented as an individual class. They all inherit properties from the Macrophage interface. All the agents in the simulation implement the Steppable and Stoppable interfaces provided by MASON. A steppable object can be scheduled on the scheduler to have its step() method called i.e. to be "stepped". A stoppable object is created when a steppable object is scheduled repeatedly, a Stoppable is returned, with a function called stop() to stop the repeating from continuing on.

On the next page is a UML diagram depicting the general structure of the application. The diagram does not display the cyclic dependency that exists between the agent classes and the Grid class. These dependencies were required for efficiency reasons.

Figure 2.10: *Class Diagram of ABM*



2.5.5 Testing

Throughout the implementation of the above model I carried out unit tests to ensure that the rules for each agent and the environment were correctly coded. There are a number of advantages to gain by unit testing code, these include:

- Reduce Bugs in New Features
- Are Good Documentation: Concise code examples can be better than many paragraphs of documentation.
- Improve Design: Tightly coupled code is difficult to test, thus this ensures that code will be more loosely coupled.
- Allow Refactoring: You are more able to change code throughout the lifetime of an application.
- Reduce Fear: Fearing to make a change to a piece of code and not knowing what is going to break.

Below are the unit tests that I created to ensure that the agents and environment were correctly coded.

2.5.5.1 Environment Tests

- Chemokine Diffusion Test: Ensured that chemokine diffused and degraded over the environment according to the diffusion rate.
- Extracellular Growth Test: Ensured that extracellular bacteria grew in each compartment at the correct rate.
- Total Neighbouring T-cells: Given a location in the environment, ensures that we calculate the number of surrounding T-cells correctly.
- Macrophage Recruitment Test: Ensures that we can recruit Resting Macrophages from the source compartment if chemokine gradient is sufficient at the source compartment.
- T-cell Recruitment Test: Ensures that we can recruit T-cells from the source compartment if chemokine gradient is sufficient at the source compartment.

2.5.5.2 T-cell Tests

- T-cell Movement: Ensure T-cell moves in all 8 directions depending on chemokine gradient. Also checks that agent can move into a compartment that contains another macrophage.
- T-cell Death: Ensure that agent is removed from the environment when its life span has expired.

2.5.5.3 Resting Macrophage Tests

- Resting Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains another macrophage.
- Resting Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it is transitioning into an infected macrophage.
- Transition into Infected Macrophage: Ensures that if the intracellular bacterial load exceeds some threshold the macrophage may transition into an infected macrophage.
- Phagocytose Bacteria: Ensures that the macrophage can remove small amounts of extracellular bacteria if present.

2.5.5.4 Infected Macrophage Tests

- Infected Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains another macrophage.
- Infected Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it is transitioning into a chronic macrophage.
- Transition into Chronic Macrophage: Ensures that if the intracellular bacterial load exceeds some threshold the macrophage may transition into a chronic macrophage.
- Intracellular Bacteria Growth: Ensures that intracellular bacteria growth in a macrophage is correct.
- Bacteria Spread: Checks that if the macrophage dies and contains intracellular bacteria, then the bacteria is spread into the surrounding environment.
- Transition into Active Macrophage: Checks if there are sufficient T-cells in Infected macrophage's neighbouring cells. If there are sufficient T-cells and they activate macrophage then the macrophage transitions into an Active Macrophage.

2.5.5.5 Chronic Macrophage Tests

- Chronic Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it has accumulated too much intracellular bacteria. A chronic macrophage may also be removed if there are sufficient T-cells that can remove the macrophage before it bursts.

- Intracellular Bacteria Growth: Ensures that intracellular bacteria growth in a macrophage is correct.

- Bacteria Spread: Checks that if the macrophage dies and contains intracellular bacteria, then the bacteria is spread into the surrounding environment.

2.5.5.6 Active Macrophage Tests

- Active Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains another macrophage.

- Phagocytose Bacteria: Ensures that the macrophage can remove small amounts of extracellular bacteria if present.

- Active Macrophage Death: Ensure that agent is removed from the environment when its life span has expired.

3 Atherosclerosis

3.1 Background

Atherosclerosis is established as the leading cause of death in the developed world, and this is predicted to project to the developing world within the first quarter of the next century.[25]

Atherosclerosis is an inflammatory disease and is a type of arteriosclerosis. It is a condition in which fatty material accumulates along the walls of arteries. This fatty material thickens, hardens, and may eventually block the arteries. Atherosclerosis affects an extensive number of organ systems, including the cardiovascular, respiratory, gastrointestinal and urinary systems, as well as other vital organs such as the brain and extremities.

Atherosclerosis is a slow and progressive disease that may start in childhood. It is characterised by lipid retention and inflammation. Inflammation is a response to a disturbance in tissue or organ homeostasis. For a given cholesterol level, the rate at which atherosclerosis progresses can vary considerably. For example, it has been shown that although siblings may have identical cholesterol levels, one may suffer a fatal myocardial infarction as young as 10 years, whilst the other may survive beyond 50. In concordance with the evidence given, it can be stated that atherosclerosis is a multifactorial disease of great complexity. [28]

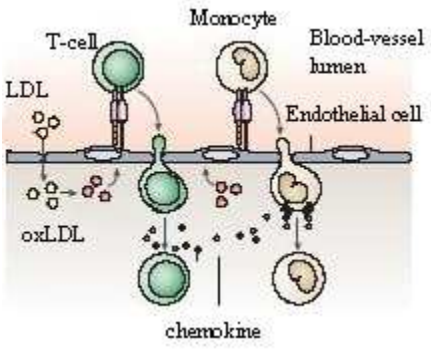
There are several risk factors related to the progression of atherosclerosis, most notably the following:

- High cholesterol
- Hypertension
- Smoking
- Diabetes
- Obesity

- Inflammatory Markers

Cholesterol can not dissolve in the blood, and instead, it has to be transported to and from cells via carriers called lipoproteins. Atherosclerosis is initiated by an accumulation and subsequent oxidation of low-density lipoprotein (oxLDL) in the intima layer of arteries. The precise mechanism responsible for oxLDL are yet to be elucidated [29]. High density lipoproteins (HDL) are known as 'good cholesterol' due to their ability to carry cholesterol away from arteries and back to the liver, where they can be re-utilised or excreted from the body. [24]

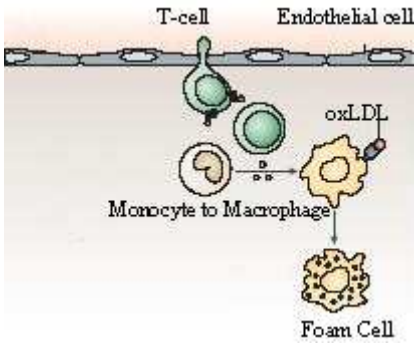
Figure 3.1: Monocyte and T-cell Recruitment



OxLDL itself is a chemoattractant and induces the expression of MCP-1 chemokine by endothelial cells.[29] In response to the oxidised LDL, endothelial cells express adhesion molecules such as vascular cell adhesion molecule-1 (VCAM-1). Endothelial cells and vascular smooth muscle cells (SMC's) release chemokines and chemoattractants, which lead to the recruitment of monocytes and T-cells into the arterial wall at specific sites where oxLDL deposits are located as depicted in Figure 3.1. [35]

Monocytes migrate through the endothelial layer due to the stimulation of chemokines, such as MCP-1, into the intima layer where they differentiate further into macrophages. [29] MCP-1 is expressed in significant amounts in all stages of the atherosclerotic process. Studies in atherosclerotic mice have shown that those mice deficient of MCP-1 had a significant reduction in lesion formation. This demonstrates that chemokine dependent migration of mononuclear cells into the intima is an important process in atherosclerosis. [31]

Figure 3.2: Foam Cell Formation

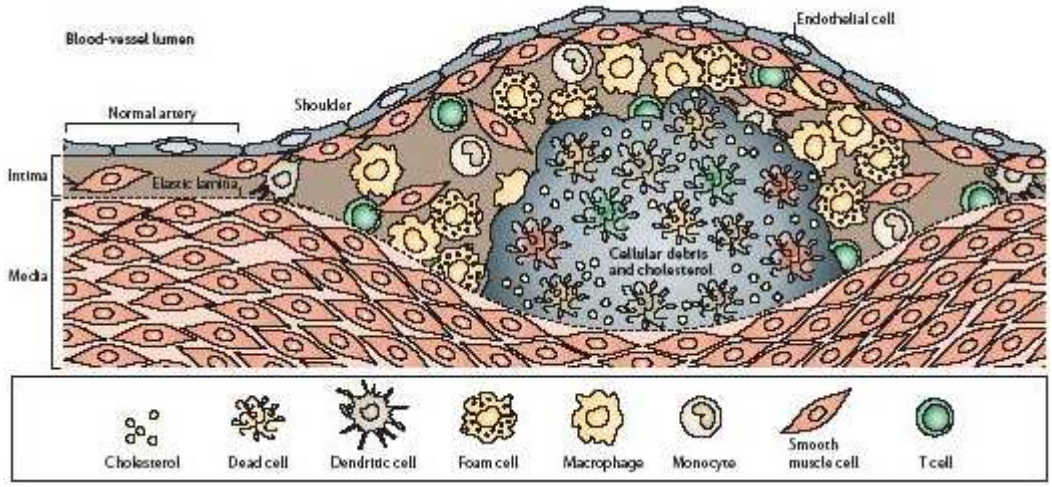


Oxidised LDL is taken up avidly by macrophages, leading to intracellular cholesterol accumulation and the formation of foam cells as shown in Figure 3.2 [28, 35]. T-cells predominantly CD3+ and CD4+ also migrate through the endothelial layer and release interferon-γ (IFN-γ) and activate macrophages, which in turn secrete more chemokines and lipids are accumulated at a faster rate. This early period of atherosclerosis is known as fatty streak lesions. These lesions can be found in the aorta shortly after birth and appear in increasing numbers in those aged 8-18 years [25].

Due to genetic and environmental risk factors, the fatty streak lesions progress into more complex lesions as T lymphocytes, platelets and smooth muscle cells also join the foam cells, expanding the plaque size. Advanced lesions begin to develop when individuals are aged approximately 25 years [25]. Lesions are not evenly distributed over the arterial system; they tend to reside in curved vessels and side branches of arteries where local sheer stress is low. Sheer stress is the force generated by the blood flow and induced by the flowing blood.[30]

As lesions progress in size, the initial expansion is outward, consequently minimising the effect on lumen cross-sectional area. Eventually, the growth of lesions extends inward causing narrowing of the vessel's lumen. This is responsible for the presentation of clinical symptoms such as angina. [29] Below you can see a diagram depicting plaque formation and the effect it has on the intima.

Figure 3.3: Plaque Formation
The images in this chapter have been taken and modified from [37].



SMC's synthesise collagen around the lesion, which forms the fibrous cap of the plaque. Macrophages are activated by T-cells and release matrix metalloproteinases(MMPs), which are responsible for plaque destabilisation. As the plaque destabilises, the fibrous cap thins and ruptures, allowing blood to access the tissue factor-rich interior of the lesion[35]. As the plaque expands, it increases the difficulty for blood to flow. If the coronary arteries become narrow, flow of blood to the heart can slow down or stop. This can cause the onset of signs and symptoms such as chest pain (stable angina), shortness of breath and myocardial infarction. Blood clots can also form around the plaque deposits, preventing the flow of blood. If the clot dislodges to the heart, lungs or brain, it can cause a myocardial infarction, pulmonary embolism or stroke respectively.

The frequency of coronary heart disease in the Far East is significantly lower than that documented in the West. CHD is preventable, although mortality rates in England remain at more than 110,000 people per year. More than 1.4 million people suffer from angina and 275,000 people have a myocardial infarction annually. Such statistics clearly outline why CHD is the major cause of mortality in the UK [26].

Few medications have been found to resolve and remove plaque formation. Most medications are only used to treat the complications of atherosclerosis. Anticoagulants may be given to prevent clots from forming. Medications are often prescribed to treat the risk factors for coronary artery disease, including those to:

- Lower lipids (statins)
- Keep blood sugar under control
- Lower high blood pressure

Similar to viruses on a computer, it is easier to prevent atherosclerosis than to treat it. There are several lifestyle changes that can help prevent atherosclerosis.

- Eating a low-cholesterol, and low-salt diet.
- Eating fish. Adding fish to the diet at least twice a week has been shown to be helpful.
- Exercise: It is recommended that the average individual should exercise for 30 minutes every day. If they are overweight, they should get 60 - 90 minutes of exercise a day.
- Stop smoking.

By understanding the complex inflammatory responses we can identify key processes that could be targeted to inhibit progression of the disease and its complications.[31]

3.2 Specification

In this section I will discuss the model that was devised to simulate the atherosclerotic process, as well as the overall requirements of the implementation. The model was devised through intensive research in the problem domain and regular meetings with experts within the field.

3.2.1 Model Description

As far as I am aware there are currently no models that simulate the process of atherosclerosis in the body. Having been able to reproduce the mycobacterium tuberculosis model successfully this was seen as a viable opportunity to replicate the model, as both are inflammatory responses and have similar agents involved in the process.

The model has the following key components:

1. The environment: The intima of the carotid artery of a mouse.

2. Entities: Agents representing macrophages, smooth muscle cells and T cells as well as continuous variables representing chemokines and lipids.

3. Rules: These determine the dynamics of the system and represent the interactions between the different entities

4. Time Scale: Indicating which rules and at what time they are executed.

Throughout the design process of the atherosclerosis model I consulted Dr Rob Krams a Professor of Molecular Bioengineering and Dr Claudia Monaco a Senior Clinical Lecturer at Imperial College Bioengineering. It quickly became apparent that the TB model would be too simple for the more complex atherosclerosis model.

The TB model was based on the Segovia paper, which listed the required agents and activities, however, as there has been no model initiated for atherosclerosis I needed to determine through reading research papers and discussions with Rob Krams and Claudia Monaco which agents needed to be added to the model. As well as discussing what the final result of the simulation should be.

Unlike the TB model whereby the simulation was a simulacrum of the TB process within the human body the atherosclerosis model would simulate the atherosclerosis process in a mouse body. This was decided due to the limited data collected on human subjects, which would be required for setting parameters within the model.

3.2.1.1 Environment

The environment I am modelling represents a portion of the intima at the curved Branch of the carotid artery of a mouse. The environment will be represented as a $N \times N$ lattice of micro-compartments. The lattice is a torus allowing us to avoid boundary effects.

Each micro-compartment will contain at most a single macrophage or smooth muscle cell and a T cell as well as any extracellular bacteria and chemokine. The largest agent in the model is the foam cell which consists of 9 micro-compartments.

The environment will also consist of six source compartments. One source compartment will represent locations where monocytes and T-cells enter the intima through a gap in the endothelial cells. Another source compartment will represent the location where LDL enter the intima and become oxLDL. One source compartment will represent the chemokine sources MCP-1 and IP-10 secreted by the endothelial cells which will attract agents towards the lipids. The remaining three source compartments represent locations where smooth muscle cells are recruited.

There are seven entities that must be modelled. I will detail the rules for each one below.

3.2.1.2 Entities

Chemokine

I will be modelling two chemokines MCP-1 and IP-10. The MCP-1 chemokine acts as an attractor to macrophages and SMC's, whereas the IP-10 acts as an attractor to T-cells. There is a third prominent chemokine 'Fracktalkine' which has the same effect as MCP-1. Thus to simplify the model and save computation we will not be modelling this chemokine. Both MCP-1 and IP-10 are secreted by active, deactive, upregulated macrophages and foam cells Chemokine is initially secreted by a source compartment, which represents the initial chemokine secreted by the endothelial cells used to attract monocytes to the focal site.

As with the TB model the chemokine concentration at any particular micro-compartment is represented as a continuous variable. Both chemokines are represented and calculated exactly as the generic chemokine was computed in the TB model in section 2.4.1.2.

Lipids

In the TB model bacteria was ingested by macrophages and they replicated within the macrophage. In the atherosclerosis model bacteria has been replaced by lipids. Lipids can not self replicate, the quantity of lipids increase as an individual digests foods. For visibility we assume that lipids accumulate in the centre of the lattice. To represent the introduction of new lipids, the source lipid site will be updated at some appropriate rate. Currently every hour in the simulation.

Each compartment can only hold a max threshold of $maxLip$ lipids. When the recruitment of lipids exceeds this threshold from the source lipid source, we assume that there is an overflow of lipids which causes lipids to pool out into neighbouring micro-compartments.

T-cells

As with the TB model T cells can share a micro-compartment with a macrophage and have only age and position attributes. T cells are not allowed to enter the lattice until day 10 of the inflammation. T-cells are recruited by the IP-10 chemotactic signals. Each T cell is recruited via a source micro-compartment and has a lifespan between 0 and 3 days. The T cell is removed from the lattice when its age reaches its lifespan.

T-cell movement is based on the chemokine gradient for IP-10. The movement rules for T-cells remained consistent to those used in the TB model. We combined the effects of CD3+ and CD4+ T cells into a composite T cell population. T-cells can upregulate active macrophages this is modelled by looking at the number of T-cells surrounding an active macrophage.

Macrophage

We have retained the attributes of position and age for macrophages from the TB model but instead of storing the number of intracellular bacteria we now store the number of internal lipids phagacytosed by the macrophage.

Macrophages are recruited via a source micro-compartment depending on the MCP-1 chemokine value at the source cell. The threshold MCP-1 value at the source compartment must exceed 1 and the source compartment must be empty to recruit a new macrophage.

Each Macrophage is randomly assigned a lifespan between 0 and 100 days and is removed from the lattice when it reaches its lifespan. Any internal lipids are spread

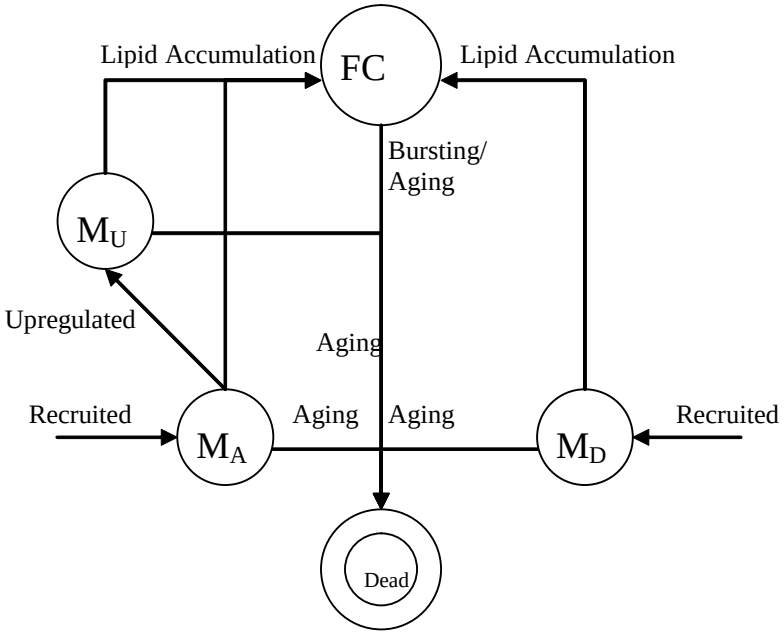
into neighbouring compartments. Macrophages move biased towards neighbouring micro-compartments with higher concentration of the MCP-1 chemokine.

There are four types of macrophage:

- **Active Macrophage**: These are recruited via a source micro-compartment and consume lipids. Can transition into an upregulated macrophage by T-cells.
- **Deactive Macrophage**: Are also recruited via a source micro-compartment and consume lipids. However, they move at a slower rate in comparison to Active macrophages.
- **Upregulated Macrophage**: They phagocytose lipids at a greater rate than both the active or deactive macrophage. They also secrete chemokines at a greater rate.
- **Foam Cells**: When the above three macrophage types consume sufficient number of lipids nF they transition into foam cells. Foam cells secrete a greater quantity of MCP-1 and IP-10 than the above macrophages and phagocytose lipids in greater quantities.

The rules and algorithmic details for the four agents above are detailed in the rules section. Below is a state diagram depicting the interactions between the different macrophage types.

Figure 3.4: Macrophage finite state diagram



SMCs

Smooth muscle cells are recruited via three source compartments after a delay of 12 days. They move chemotactically similar to macrophages. They have a biased random walk towards strong concentrations of MCP-1. However, unlike the macrophages in the environment SMC cells can not move through the right side of the environment near the media to the far left side of the environment near the lumen. SMCs can release collagen into the environment depending on the number of macrophages surrounding the SMC.

SMCs can also undergo mitosis and split into 2 or more cells. This enables us to model the proliferation of the SMCs in the simulation. The proliferation of SMCs is limited by the number of T-cells in the environment.

SMCs are assigned a random lifespan of 0 to 200 days and move at the same rate as deactive macrophages. They are removed from the environment if they have reached of age or if there are sufficient active macrophages surrounding the SMC causing the SMC to be killed. SMCs and macrophages will move in a random direction if there is no chemokine in the environment.

Collagen & Necrotic cells

Collagen is released into the environment by SMCs. By placing collagen into a micro-compartment it causes the chemokine value of both MCP-1 and IP-10 at that location to become 0. It also prevents other agents from moving through that compartment. Collagen represents the fibrous cap of the plaque.

Necrotic compartments represent dead tissue. This occurs as foam cells repeatedly burst over the same compartment. As with collagen agents can not move over necrotic compartments. However, unlike collagen chemokine still diffuse and degrade over the compartment.

3.2.1.3 Time Scale

As with the TB model each time step corresponds to 6 seconds of 'real time.' MCP-1 and IP-10 diffusion and degradation occurs at every time-step. Most rules occur at a slower rate. Lipid recruitment is calculated every hour i.e. 600 time-steps. T cells, active macrophages, deactive macrophages, upregulated macrophages and SMC move every 10, 50, 100, 100 and 100 time-steps respectively. Foam cell movement is negligible and thus has not been added to the model.

3.2.1.4 Rules

Below are the algorithmic rules devised for the entities within this model.

Active Macrophage:

Numbers in brackets(?) represent line numbers with associated algorithm.

An active macrophage will initially secrete both MCP-1 and IP-10 chemokine at its present location(1,2). If there are lipids in the same micro-compartment as the macrophage, it will phagocytose a small number N_{mak} of lipids(3-9). If the number of intracellular lipids for the Active Macrophage exceed a threshold N_f then the macrophage will transition into a foam cell(11). If the number of intracellular lipids does not exceed N_f then the macrophage could be upregulated by surrounding T-cells(13). Upregulation of the macrophage depends on the the number of T-cells N_{tup} in its moore-neighbours and the probability T_{upam} that those T-cells successfully upregulate the macrophage(15,18).

Algorithm 3.1 Active Macrophage Rules

```
1: mcp_1 += ci
2: ip_10 += ci;
3: if Lip_E i, j(t) ≥ 0 then
4:     if Lip_E i, j(t) < N_mak then
5:         Lip_I i, j(t + 1) = Lip_E i, j(t)
6:         Lip_E i, j(t + 1) = 0
7:     else
8:         Lip_I i, j(t + 1) = N_mak
9:         Lip_E i, j(t + 1) = Lip_E i, j(t) − N_mak
10: endif
11: if Lip_I i, j(t) > N_f then
12:     return M_FC; Foam Cell
13: else
14:     N_T= Number of T cells in the neighbourhood
15:     if N_T > N_tup then
16:         N_T = N_tup
17:     endif
18:     r ← RandomUniform[0, 100]
19:     if r < (N_T ∗ T_upam) ≤ 100 then
20:         return M_U; Upregulated Macrophage
21:     else
22:         return M_A; Active Macrophage
23:     endif
24: endif
```

Deactive Macrophage:

The rules for a deactive macrophage are similar to those of an Active macrophage although a deactive macrophage will have different parameter values for speed etc. The only difference is that a deactive macrophage can not be upregulated into an Upregulated macrophage. However, as with the active and upregulated macrophage it can transition into a foam cell if it has consumed a sufficient number of lipids.

Upregulated Macrophage:

Since the macrophage has already been upregulated it can not be further upregulated so the rules for the upregulated macrophage are exactly the same as the rules for a deactive macrophage. Again the parameter values of the agents will be different as an Upregulated macrophage secretes a greater quantity of chemokine and consumes lipids in greater quantities. Unlike active and deactive macrophages an upregulated macrophage can move through and remove collagen from the environment.

Foam Cell:

Numbers in brackets(?) represent line numbers with associated algorithm.

Once a macrophage has transitioned into a foam cell it can not move. It consumes lipids and releases chemokines at a far greater rate than the other macrophages. The foam cell can consume lipids from all nine micro-compartments that it occupies(3-12). Once the foam cell has consumed lipids exceeding a threshold Kli the foam cell will burst spreading its lipid contents around the environment(13-16). This bursting damages the tissue of the intima and this is reflected by keeping a count of the number of burstings for that micro-compartment(17,18). The foam cell and other macrophage will also spread their lipid content when they have reach the end of their respective life spans.

Algorithm 3.2 Foam Cell Rules

1: mcp_1 += ci
2: ip_10 += ci;
3: for each micro-compartment of foam cell
4: if $Lip_E i, j(t) \geq 0$ then
5: if $Lip_E i, j(t) < N_{fk}$ then
6: $Lip_I i, j(t+1) = Lip_E i, j(t)$
7: $Lip_E i, j(t+1) = 0$
8: else
9: $Lip_I i, j(t+1) = N_{fk}$
10: $Lip_E i, j(t+1) = Lip_E i, j(t) - N_{fk}$
11: endif
12: endfor
13: if $Lip_I i, j(t) > Kli$ then
14: $Lip_{Ei-2,j}(t+1) = Lip_{Ei-2,j}(t) + Lip_I(t)/25$
15: $Lip_{Ei+2,j}(t+1) = Lip_{Ei+2,j}(t) + Lip_I(t)/25$
16: $Lip_{Ei-2,j+2}(t+1) = Lip_{Ei-2,j+2}(t) + Lip_I(t)/25$ etc:
17: $Necrotic_{i-2,j}(t+1)++$
18: $Necrotic_{i-1,j}(t+1)++$etc:
19: Foam cell is eliminated
20: return Death
21: else
22: return M_{FC}; Foam Cell
23: endif

SMC:

Numbers in brackets(?) represent line numbers with associated algorithm.

SMCs move towards locations of higher chemokine concentrations(5,6,7,12). However, they also produce collagen an important factor in stabilising plaques. In our model SMCs will synthesise and drop collagen in a micro-compartment which it is departing. Collagen synthesis is dependent upon the number of macrophages in its moore-neighbour cells *nmcoll*(8-10).

Algorithm 3.3 Smooth Muscle Cell Rules

1: ranProb;
2: cumChemTotal = 0;
3: totalMooreChem;
4: for each moore-neighbour micro-compartment of smc
5: $chem_{(i',j')} = Chem_{(i',j')}/totalMooreChem$
6: if $ranProb \geq cumChemTotal$ && $ranProb < cumChemTotal + chem_{(i',j')}$ then
7: SMC(t+1)$_{(i,j)}$ = SMC(t)$_{(i',j')}$
8: if($MacrophagesInMoore > nmcoll$) then
9: Coll(t+1)$_{(i,j)}$ = true
10: endif
11: else
12: cumChemTotal = $cumChemTotal + chem_{(i',j')}$
13: endif
14: endfor

Many of the parameters used in this model will be taken directly from the TB model. This is because there has been very little quantitative analysis undertaken in the atherosclerosis field. Depending on the values set for the parameters the simulation demonstrates plaque formation and rupture. On the next page are a list of parameters that will be used in this model. As you can see the number of parameters have increased alongside the number of agents within the model.

Table 3.1: *Model Parameters*

Symbol	Parameter Description	A	Units
λ_{mcp}	MCP Chemokine diffusion coefficient	0.64	0.1min
λ_{ip}	IP Chemokine diffusion coefficient	0.64	0.1min
δ_{mcp}	MCP Chemokine degradation coefficient	0.001	0.1min
δ_{ip}	IP Chemokine degradation coefficient	0.001	0.1min
mcp_{init}	Initial MCP chemokine amount	100000	Integer
ip_{init}	Initial IP chemokine amount	100000	Integer
LDL_{init}	Initial LDL amount	20	Integer
LDL_{recr}	Recruitment rate of LDL	10	0.1min
N_f	Number of lipids required to Transition from macrophage into a Foam Cell.	10	Scalar
Kli	Number of lipids that make a foam cell burst	50	Scalar
N_{tup}	No. of T-cells required to upregulate a macrophage	7	Scalar
T_{recr}	Prob. of T-cell recruitment	32	%
N_{tact}	No. of T-cells needed to activate a macrophage	4	Scalar
T_{recr}	Prob. of T-cell recruitment	1	%
T_{move}	Prob. of T-cell movement	4.97	%
T_{upam}	Prob. a T cell will upregulate a macrophage	6	%
T_{ls}	T cell lifespan	3	days
T_{delay}	T cell delay	10	days
SMC_{delay}	SMC delay	15	days
M_{recr}	Prob. of macrophage recruitment	2.11	%
SMC_{recr}	Prob. of smooth muscle cell recruitment	32	%
am_{recr}	Active macrophage to deactive macrophage ratio	50:50	%
M_{als}	Active macrophage max lifespan	100	days
M_{dls}	Deactive macrophage max lifespan	100	days
SMC_{ls}	Smooth Muscle Cell mac lifespan	200	days
T_{sp}	T cell speed	10	µm/min
M_{asp}	Active macrophage speed	1	µm/min
M_{dsp}	Deactivate macrophage speed	0.5	µm/min
SMC_{sp}	Smooth muscle cell speed	0.5	µm/min
N_{necr}	Max number of burstings before cell is necrotic	8	Scalar
N_{mak}	No. of lipids phagacytosed byM_A	2	Scalar
N_{mdk}	No. of lipids phagacytosed byM_D	2	Scalar
N_{upmk}	No. of lipids phagacytosed byM_U	3	Scalar
N_{fck}	No. of lipids phagacytosed byM_{FC}	4	Scalar
Ci_{am}	Chemokine secretion by M_A	1	Scalar
Ci_{dm}	Chemokine secretion by M_D	1	Scalar
Ci_{um}	Chemokine secretion by M_U	2	Scalar
Am_{ksmc}	No. of M_A needed to remove SMC	8	Scalar
P_{amksmc}	Prob M_A kills SMC	6	%
$maxLip$	Max LDL in a single compartment	200	Scalar
N_{mcoll}	No. macrophages required to synthesise collagen	3	Scalar

I have already outlined my reasons for implementing the system in Java in section 2.4. Below are list of implementation aims.

Overall Aims

- Java based implementation.
- Simulation to run between 6-9 weeks.
- Simulation should show plaque rupture.
- User should be able to vary parameters via a text file
- User should be able to run a simulation with or without a visible console.
- Graphical window should display Collagen, SMCs, Foam Cells, Active Macrophages, Deactive Macrophages and Upregulated Macrophages, T cells, necrotic compartments and OxLDL.
- User should be able to pause and stop simulation at any time.
- As a bench mark the simulation should run as quickly as the TB simulation for 200 days.

3.3 Atherosclerosis Implementation

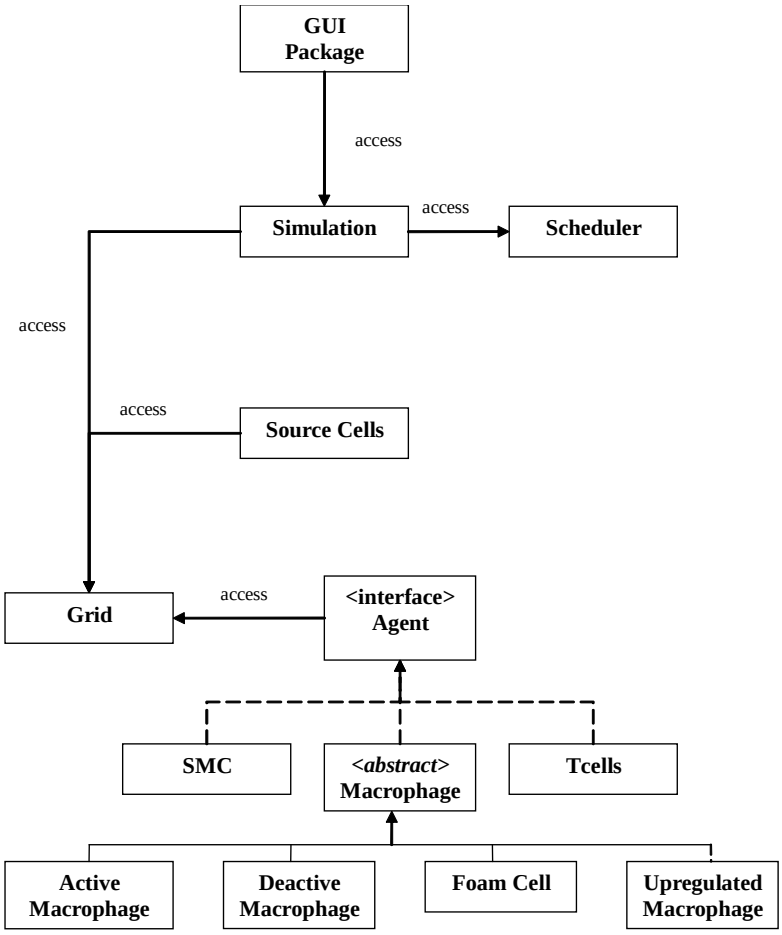
The GUI display is exactly the same as that shown in section 2.5. The only difference is that the parameters tab contains more parameters for the atherosclerosis model than the TB model.

The user can modify parameters prior to running the simulation in one of two ways. If the user is running the simulation without the GUI he/she can modify the parameters by accessing the Parameters.txt file. The user can modify the parameters when they run the simulation via the GUI by accessing the parameters tab, the user can modify the parameters and must select the 'Set' button to confirm the changes. The user can reset the parameters to those set originally by selecting the 'Reset' button.

On the next page is a UML diagram depicting the class structure. However, this is just a simplified UML diagram and does not depict the GUI class structure. I modified various GUI classes provided by MASON to fully integrate the model. Also there are cyclic dependencies that are not visible in the diagram between the environment and each agent. Conventional object orientated design suggests adding a further layer of

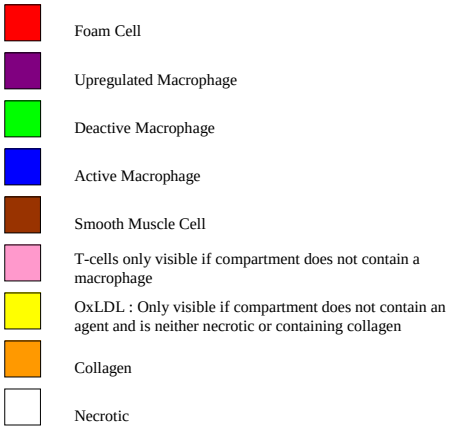
indirection to avoid cyclic dependencies, however, this would impact on the efficiency and computational speed of the system.

Figure 3.5: Atherosclerosis UML Diagram



The implementation uses the colour chart in Figure 2.9 to distinguish between the different elements in the system.

Figure 3.6: Colour Chart

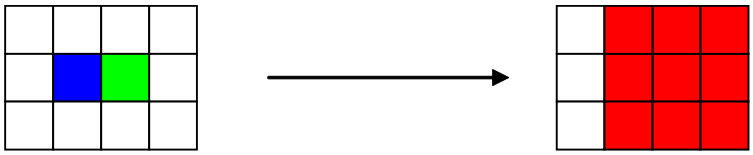


3.3.1 Foam Cell Formation and Agent Re-location Algorithm (FCFARA)

Foam cells are nine times larger than the size of other agents within the model. This caused unexpected problems within the simulation. There were two significant problems that arose due to this seemingly trivial situation, which are outlined below.

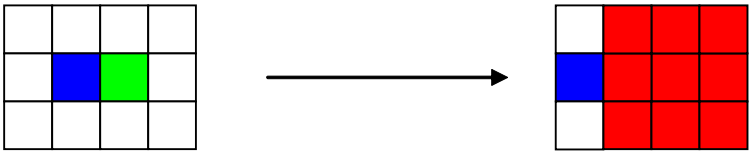
The first problem occurs when an Active, Deactive or Upregulated macrophage becomes a foam cell whilst other agents surround the transitioning macrophage. As the transitioning macrophage uptakes lipids it moves closer to the state of becoming a foam cell. This may also be true for neighbouring agents. When the macrophage becomes a foam cell instead of requiring just a single compartment it now needs nine compartments. The question arises what should we do with the neighbouring agents? One could argue that the surrounding agents are consumed by the foam cell as shown in Figure 3.1. However, this is not a realistic assumption as macrophages do not consume each other.

Figure 3.7: Deactive Macrophage(green) transitions into a foam cell and consumes Active Macrophage(blue)



Imagine placing a small set of equally sized marbles randomly near to each other. We pick up and replace one marble with one that is significantly larger. If the space left by the departing marble is sufficient enough to accommodate the larger marble then the other marbles are undisturbed. However, if the space is small then the larger marble will push those marbles closest to it further away, and these marbles in turn may push other marbles further away. We use a similar approach when a macrophage becomes a foam cell. We assume that the foam cell would not consume neighbouring agents but would 'push' them away into other compartments as shown in Figure 3.2.

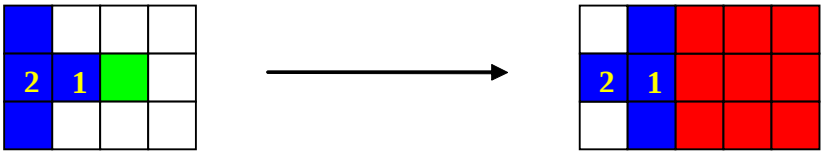
Figure 3.8: Deactive Macrophage(green) transitions into a foam cell, 'pushing' Active Macrophage(blue) to the left



Now we must decide which location the pushed agents should move into. Those marbles closest to the replaced marble are most likely still the closest marbles to the new larger marble. As in the example of marbles we can assume that those agents that were surrounding the transitioning macrophage will still be the closest agents when the macrophage has become a foam cell. Therefore the agent needs to move to an empty compartment within its moore neighbours away from the foam cell, as each macrophage can only move to a neighbouring micro-compartment at a time.

However, the agent may be surrounded by other agents and there is no empty compartment a single step away i.e. one compartment away. In such a scenario we need to obtain a path of agents that the initial agent would need to push outwards to create an empty space nearby that the agent can move into.

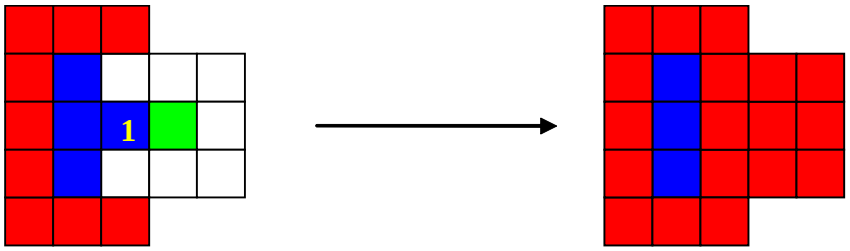
Figure 3.9: Deactive Macrophage(green) transitions into a foam cell and 'pushes' Active macrophage(2) left and the Active Macrophage(1) to the left.



The final case we must consider is what should happen if the agent that needs to move to a neighbouring empty compartment is in fact surrounded by foam cells? Again returning to the marble example, imagine we have several large marbles surrounding a few smaller marbles and we replace one of these small marbles with a larger one. The

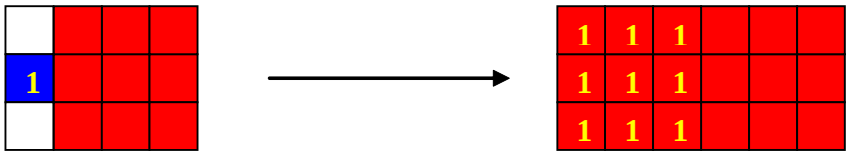
surrounding smaller marbles can not 'push' outwards as the larger marbles will refuse to budge from the weight of a smaller marble and therefore these small marbles are trapped. The large marble will just rest on top of the surrounding smaller marbles. The only way we can create sufficient space for the larger marble is if we remove the surrounding smaller marbles to create space for the larger marble.

Figure 3.10: Deactive Macrophage(green) transitions into a foam cell and Active macrophage (1) can not find an empty cell and thus must be removed.



The second problem occurs when a macrophage is ready to transition into a foam cell, but is currently next to a foam cell. We must decide whether the transitioning macrophage should move to a new location, should push surrounding foam cells away or should simply perish. After discussing with Rob Krams about the situation we decided that the macrophage must move away and become a foam cell, but it does not push away neighbouring foam cells. Therefore it finds the closest position to neighbouring foam cells.

Figure 3.11: Active Macrophage must move away so that it can transition into a foam cell



If the agent is unable to move to a location where it can transition into a foam cell, i.e. it is surrounded by foam cells, necrotic tissue and collagen then the macrophage is assumed to die.

3.3.1.1 FCFARA Implementation details

Numbers in brackets(?) in the following text represent line numbers in algorithm 3.4.

When a macrophage has accumulated sufficient lipids greater than some threshold then it transitions into a foam cell. To transition into a foam cell it needs to ensure that its moore neighbourhood compartments are cells that are not necrotic, contain collagen and does not contain some part of a neighbouring foam cell. The method canMorph(2) checks for this property and if true, allows the agent to become a foam cell immediately; otherwise the macrophage must find the closest location where it can transition into a foam cell(3). If it is unable to find a path where it can transition into a foam i.e. it is surrounded by foam cells, necrotic tissue or collagen then it is removed from the environment and spreads its lipid content into the environment(4,5).

Once a macrophage has found a location where it can morph, it needs to clear the surrounding compartments of other agents(8,9 and 14,15). We first need to find if there are indeed any agents in neighbouring cells, if there are no agents nearby then the macrophage can transition into a foam cell(11,17). However, if there were agents in neighbouring cells then we need to identify empty compartments where each agent can be moved into(9 and 15).

Algorithm 3.4 Macrophage To Foam Cell Transition

```
1: if Lip_I i, j(t) > N_f then
2:     if (!canMorph) then
3:         path = findPathToTransitionIntoFoamCell
4:         if (path == null) then
5:             return death;
6:         else
7:             executePath;
8:             if(neighbouringAgents) then ;
9:                 PushNeighbouringAgentsAway
10:            endif
11:            returnFoamCell;
12:        endif
13:    else
14:            if(neighbouringAgents) then ;
15:                PushNeighbouringAgentsAway
16:            endif
17:            returnFoamCell;
18:    endif
19: endif
```

Numbers in brackets(?) in the following text represent line numbers in algorithm 3.5.

To locate empty spaces we search in circular paths around the transitioning macrophage to identify free compartments(6). The number of free cells we need to obtain is equivalent to the number of surrounding agents.

I have limited the depth of the search to 15 circular paths around the transitioning macrophage(1). We then assign each empty compartment to the closest agent to that compartment of the surrounding agents. The search depth limit of 15 was an arbitrary number chosen to keep computation times low. Also realistically it is unlikely that the macrophage will need or be able to 'push' more than 15 other macrophages.

If whilst searching for empty spaces we find that we are surrounded by collagen, necrotic tissue and foam cells(10) then we assign as many empty spaces as possible and remove those agents furthest away from empty spaces(12). If we have not found any empty spaces and we are surrounded then we must remove all neighbouring agents(14).

Algorithm 3.5 Find Empty Spaces

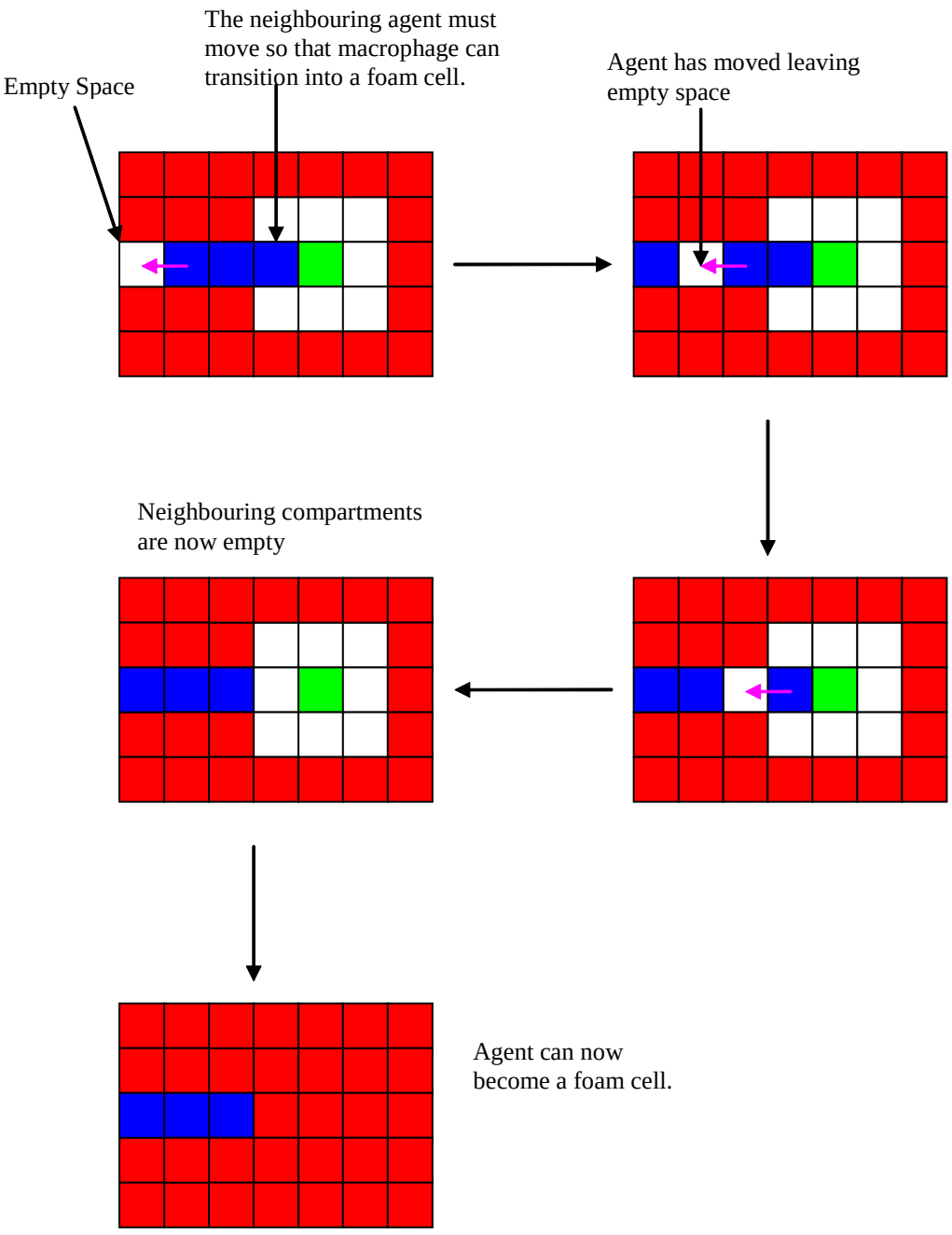
```
1: maxDepth = 15
2: emptySpacesReq = NumberOfSurroundingMac
3: emptySpacesFound = 0
4: depth = 0
5: while(emptySpacesFound < emptySpacesReq && depth <= maxDepth) then
6:     emptySpacesFound = SearchForEmptySpaces
7:     if(emptySpacesFound ⩾ emptySpacesReq) then
8:         AssignEmptySpacesFindAndExecutePlans
9:     else
10:        if(Surrounded) then
11:            if(emptySpacesFound > 0) then
12:                AssignEmptySpacesFindAndExecutePlans
13:            else;
14:                KillAllNeighbouringAgents
15:            endif;
16:        endif
17:    depth++
18:    endif
19: endwhile
20: if(depth > maxDepth) then
21:     if(emptySpacesFound > 0) then
22:         AssignEmptySpacesFindAndExecutePlans
23:     else;
24:         KillAllNeighbouringAgents
25:     endif
26: endif
```

Having assigned each agent an empty compartment we then build a plan from the empty compartment back to the agent assigned that empty space for each agent. This plan indicates what sequence of agents collides allowing the initial surrounding agent to move out away from the transitioning macrophage. To keep plan building efficient, whilst searching for an empty space I store locations where the plan can extend to. Therefore when building a plan we will avoid plans that will definitely be unsuccessful as they may require moving collagen, a foam cell or moving through necrotic tissue.

I have used an iterative deepening search algorithm to build plans. This includes building plans for agents to move away from a macrophage that is transitioning into a foam cell and when a macrophage looks to move away to find space to transition into a foam cell. An iterative deepening search will guarantee that we observe all plans within a reasonable time unlike depth-first. Also iterative deepening has lower memory requirements than breadth-first search.

These plans are executed from the initial empty space towards the assigned agent. The first element in the plan will be the empty space, which we originally assigned to the agent. Until the plan is empty we remove each new location indicated by the plan. If the next location removed from the plan contains an agent that is not the original agent assigned to the empty space we move it to the previous location removed from the plan. If the new location is a cell within the Moore neighbourhood of the transitioning macrophage, then we can immediately move the agent assigned to the empty space to the last location removed from the plan. Below is a diagram to help explain the execution of the plan.

Figure 3.12: Executing Plan



3.3.2 Testing

Throughout the development of the model I used unit tests to ensure that my code was working correctly. With the model changing on each new discussion of the model, unit testing was essential in ensuring new features did not break existing working code. As with the TB model I created JUnit tests for each agent and the environment itself.

To create unit tests I used JUnit. JUnit is a simple framework that allows you to write repeatable tests in java. JUnit can be easily integrated into many popular Integrated development environments (IDE's). This includes Eclipse the IDE which I used to develop both models. IDE's are editing environments with tools to help programmers write code quickly and efficiently.

However, due to the stochastic nature of the model, not all the code could be tested using JUnit tests. To test the FCFARA algorithm I created different simple scenarios to test the code. By using simple examples I was able to test all conditions and through a graphical output was able to determine whether the algorithm had worked correctly.

Below are the unit tests that I created to ensure that the agents and environment were correctly coded.

3.3.2.1 Environment Tests

- Lipid Overflow Test: Ensured that when the source lipid compartment contained more lipids that it could hold, lipids spill over into the neighbouring Von-Neumann neighbourhood.
- Chemokine Diffusion Test: Ensured that chemokine diffused and degraded over the environment according to the diffusion rate.
- Total Neighbouring T-cells: Given a location in the environment, ensures that we calculate the number of surrounding T-cells correctly.
- Total Neighbouring Macrophages and Collagen: Given a location in the environment, ensures that we calculate the total number of surrounding macrophages and collagen.
- Total Neighbouring Active Macrophage: Given a location in the environment, ensures that we calculate the total number of surrounding Active Macrophages correctly.
- Macrophage Recruitment Test: Ensures that we can recruit both Active and Deactive Macrophages from the source compartment if chemokine gradient is sufficient at the source compartment.

- SMC Recruitment Test: Ensures that we can recruit smooth muscle cells from all three SMC source compartments if chemokine gradient is sufficient at the source compartments.

- T-cell Recruitment Test: Ensures that we can recruit T-cells from the source compartment if chemokine gradient is sufficient at the source compartment.

3.3.2.2 SMC Tests

- SMC Movement: Ensure SMC moves in all 8 directions depending on chemokine gradient. Also ensure that SMC does not not move into a compartment which contains collagen, another macrophage or is necrotic regardless of chemokine gradient.

- SMC Death: Ensure that agent is removed from the environment when its life span has expired. Or if there are sufficient number of surrounding active macrophages.

- SMC Collagen Deposit: Checks that agent drops collagen if there a sufficient number of macrophages in neighbouring compartments.

3.3.2.3 T-cell Tests

- T-cell Movement: Ensure T-cell moves in all 8 directions depending on chemokine gradient. Also ensure that T-cell does not not move into a compartment which contains collagen or is necrotic regardless of chemokine gradient. Checks that agent can move into a compartment that contains another macrophage.

- T-cell Death: Ensure that agent is removed from the environment when its life span has expired.

3.3.2.4 Deactive Macrophage Tests

- Deactive Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains collagen, another macrophage or SMC and the compartment is not necrotic regardless of chemokine gradient.

- Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it is transitioning into a foam cell and can not move into a location where it can become a foam cell.

- Transition into Foam Cell: Ensures that agent can transition into a foam cell if there is sufficient space.

- Lipid Accumulation: Ensures that the macrophage can accumulate lipids if present.

- Lipid Spread: Checks that if the macrophage dies and contains intracellular lipids, these lipids are spread into the surrounding environment.

3.3.2.5 Active Macrophage Tests

- Active Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains collagen, another macrophage or SMC and the compartment is not necrotic regardless of chemokine gradient.

- Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it is transitioning into a foam cell and can not move into a location where it can become a foam cell.

- Transition into Foam Cell: Ensures that agent can transition into a foam cell if there is sufficient space.

- Lipid Accumulation: Ensures that the macrophage can accumulate lipids if present.

- Lipid Spread: Checks that if the macrophage dies and contains intracellular lipids, these lipids are spread into the surrounding environment.

- Transition into Upregulated Macrophage: Checks if there are sufficient T-cells in Active macrophage's neighbouring cells. If there are sufficient T-cells and they upregulate macrophage then the macrophage transitions into an Upregulated Macrophage.

3.3.2.6 Upregulated Macrophage Tests

- Upregulated Macrophage Movement: Ensure macrophage moves in all 8 directions depending on chemokine gradient. Also ensure that macrophage does not not move into a compartment if it contains collagen, another macrophage or SMC and the compartment is not necrotic regardless of chemokine gradient.

- Macrophage Death: Ensure that agent is removed from the environment when its life span has expired. Or if it is transitioning into a foam cell and can not move into a location where it can become a foam cell.

- Transition into Foam Cell: Ensures that agent can transition into a foam cell if there is sufficient space.

- Lipid Accumulation: Ensures that the macrophage can accumulate lipids if present.

- Lipid Spread: Checks that if the macrophage dies and contains intracellular lipids, these lipids are spread into the surrounding environment.

3.3.2.7 Foam Cell Tests

- Foam Cell Death: Ensure that agent is removed from the environment when its life span has expired.

- Transition into Foam Cell: Ensures that agent can transition into a foam cell if there is sufficient space.

- Lipid Accumulation: Ensures that the macrophage can accumulate lipids if present.

- Lipid Spread: Checks that if the macrophage dies and contains intracellular lipids, these lipids are spread into the surrounding environment.

4 Evaluation

4.1 Mycobacterium Tuberculosis

The TB implementation successfully demonstrated latency, clearance and dissemination as described in numerous papers. I was also able to achieve comparable results to those obtained by [1]. The simulation was run on a Dual Core 3.2Ghz machine with a Linux operating system. I was able to compute 200 day simulations within 16 minutes, however, this time could be improved as the application did not make use of the dual core capacity through concurrent threads.

To ensure that the system had been correctly implemented I used the parameters from the table 2.3 and was able to show containment using the parameters from column A and and dissemination using column B parameters. The following two pages show output received from the simulation. We can see that the output is very similar to the expected output explained earlier in Figure 2.3.

In Figure 4.1 we can see that initially for the first 100 days the bacteria continually spreads and are surrounded by chronic and infected macrophages and a few active macrophages. However, by day 200 the active macrophages have surrounded the bacteria forming a granuloma and containing the spread of infection.

In Figure 4.2 we can see that the bacteria continually spread throughout the simulation. We can see at day 100 a small portion of bacteria breaks away from the main contingent and by day 200, unlike the containment simulation the active macrophages could not contain the bacteria in a granuloma and the bacteria have partitioned into two distinct sections and display the infection spreading.

Figure 4.1: TB containment

Containment: The ABM produced the following captions of granuloma formation at days 12(A), 25(B), 50(C), 100(D) and 200(E). The colours represent resting macrophages(green), activated macrophages(blue), infected macrophages(orange), chronically infected macrophages (red), T cells (pink) and extracellular bacteria (yellow). The graphical legend is shown in Figure 2.9

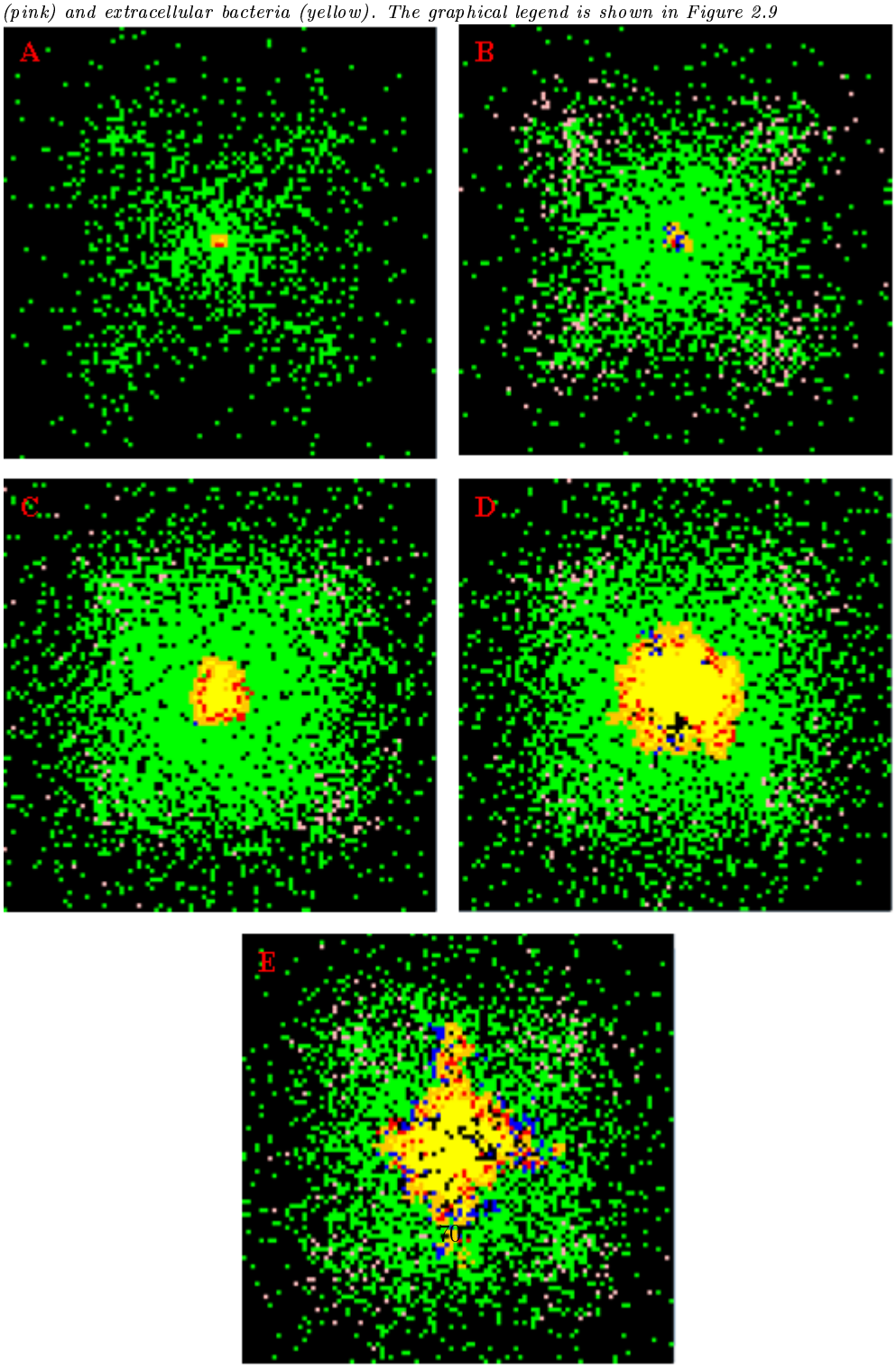
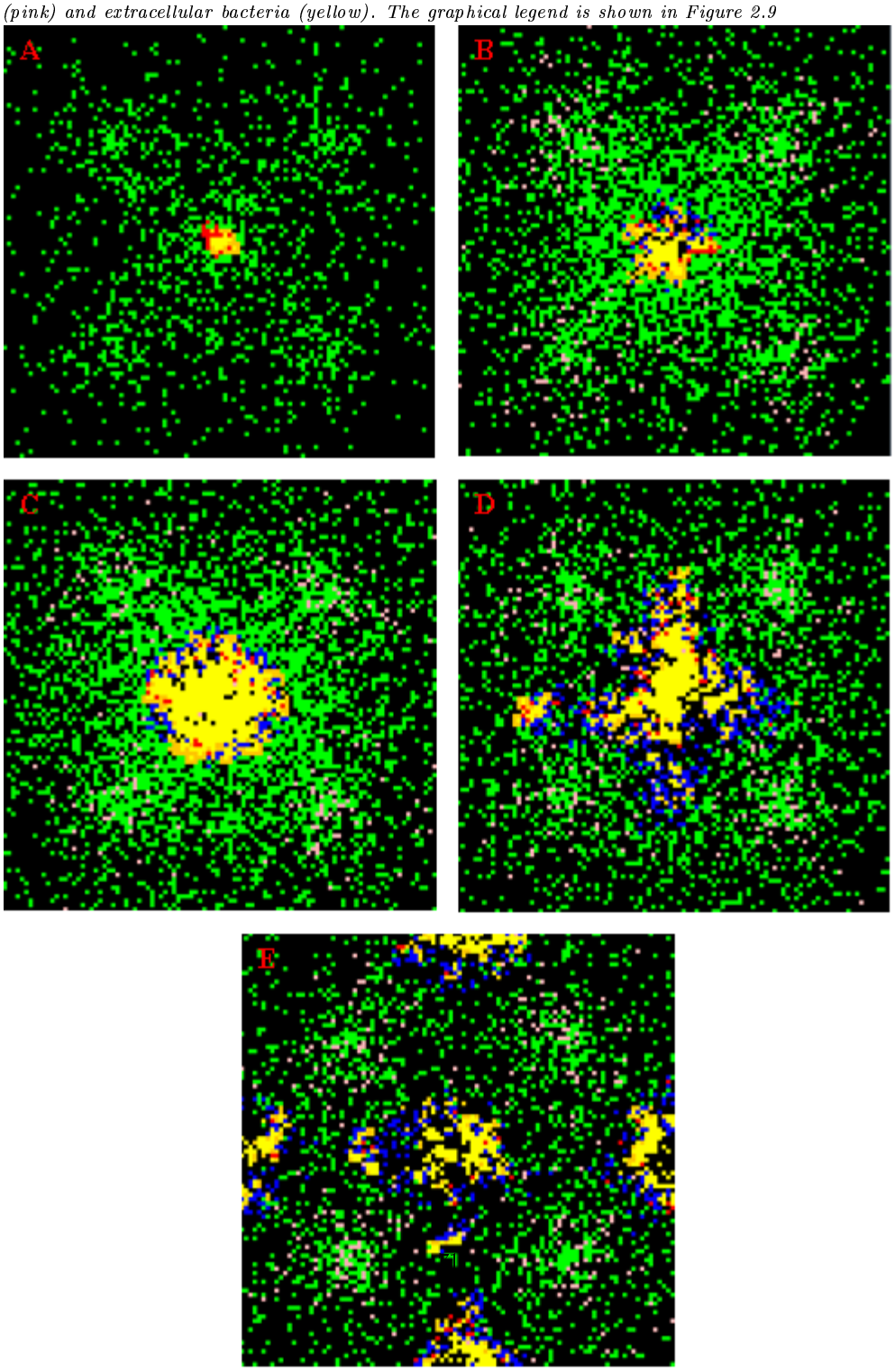


Figure 4.2: TB Dissemination

Dissemination: The ABM produced the following captions of granuloma formation at days 12(A), 25(B), 50(C), 100(D) and 200(E). The colours represent resting macrophages(green), activated macrophages(blue), infected macrophages(orange), chronically infected macrophages (red), T cells (pink) and extracellular bacteria (yellow). The graphical legend is shown in Figure 2.9



4.2 Atherosclerosis

The extensibility of the TB code and the multi-agent approach was proven by the successful implementation of the atherosclerosis model. Approximately 60% of the TB code was reusable. The atherosclerosis model needed to be flexible as new literary work can introduce unexpected changes. The model proved to be robust as it underwent a number of modifications and additions.

The simulation displayed plaque formation and rupture as described in literary work that was cross-referenced. The application runs a 9-week simulation within 12 minutes on Dual Core 3Ghz machine. Despite the impressive turnaround I believe further computational efficiency can be gained by refining the queueing mechanism used as detailed in section 5.1.

On the following page you can see an example run of the simulation. In the first week we start to realise fatty streaks within the intima, which are the first grossly visible lesions in the development of atherosclerosis. By week 3 we can see that the smooth muscle cells have formed a boundary of collagen around the lesion and surround the plaque, as shown in Figure 3.3. As the simulation progresses onto week 5 we can see the fibrous cap thinning and eventually rupturing as explained earlier in section 3.1. These were the expected results for the simulation.

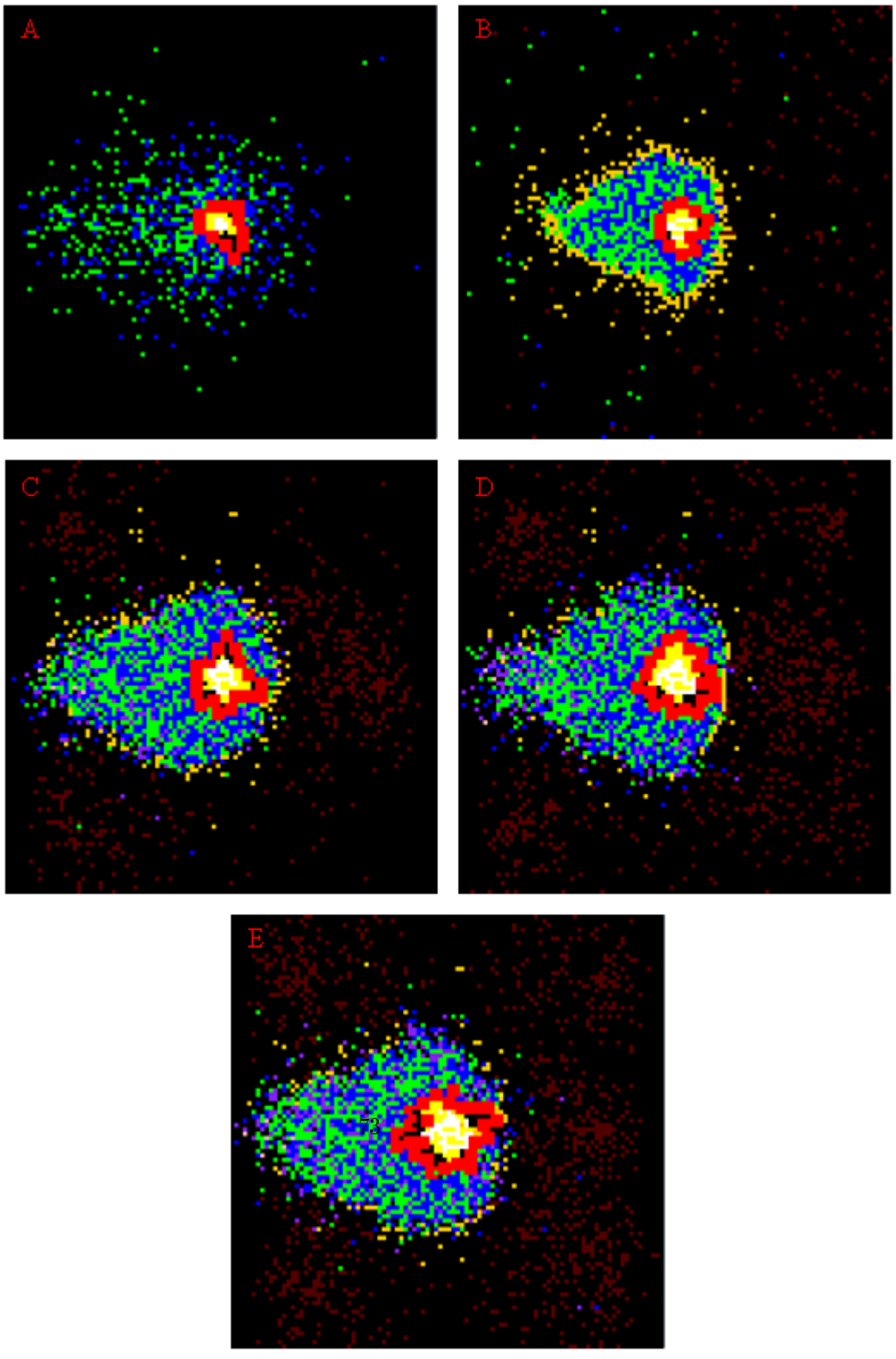
Although the simulation is an accurate representation of atherosclerosis, the model can be improved. There are a number of agents that can be added to the simulation including Toll-like receptors (TLRs), which help macrophages recognise lipids in the environment, cytokines, lymphocytes and many more covered in various literature associated with atherosclerosis. We have modelled the key agents involved in the atherosclerotic process. The behaviour of these agents is well defined and agreed upon by the majority within current literature.

As there is limited quantitative information on the process of atherosclerosis, unlike the TB model, most parameters were estimated. I retained many of the parameters from the TB model but additional parameters were estimated as a binary relation. E.g. active macrophage is faster than a deactive macrophage or active macrophage is smaller than a foam cell. However, I am confident that the exact parameters would provide very similar results.

Having successfully implemented the process I hope to write a paper on the current model, which will serve as a starting point for possible extensions to the model. The model can now undergo a sensitivity and uncertainty analysis as was conducted in the original TB model in [1]. I hope that the atherosclerosis model can be similarly progressed to provide valuable information on possible patterns of disease progression and prevention.

Figure 4.3: Atherosclerosis Simulation

Plaque rupture: The ABM produced the following captions of plaque formation and rupture at week 1(A), 3(B), 5(C), 7(D), 9(E). The colours represent deactive macrophages(green), active macrophages(blue), upregulated macrophages(purple), foam cells(red), smooth muscle cells(brown), T cells (pink), collagen(orange), lipids(yellow), necrotic cells(white), empty space(black). The graphical legend is shown in Figure 3.6



5 Conclusion

In conclusion I have found that we can use multi-agents to model both TB and atherosclerosis successfully. Although there were similarities in the models, a number oft changes were required as agents performed different tasks. Due to the complexity and varying information on the process of atherosclerosis, it was particularly difficult to devise and implement the atherosclerosis model. Contradictory medical papers also made it difficult to design a model that represented all researched material.

The scope and depth of both processes in reality is much more complex. However, using multi-agents provided a succinct solution that could offer researchers valuable information and insights into each process. The atherosclerosis model can now undergo sensitivity and uncertainty analysis as was undertaken by [1] for the TB model, which may provide information on possible methods of prevention of atherosclerosis.

5.1 Future Work

There are a number of modifications that can be made to the model and implementation such that the process of atherosclerosis can be modelled more accurately and realistically. There are also modifications that can be made to the implementation to improve the speed of the simulation.

5.1.1 Scheduler Improvement

As the TB simulation progressed I found through using Jprofiler that the amount of time spent calculating the next event to be processed in the simulation and adding/removing events to the scheduler increased. At the end of a 200 day simulation over 50% of the computation time was spent on the scheduler. Although still considerable, this was less of an affect in the atherosclerosis model as it was run for only 63 days.

Currently the queueing mechanism used in the scheduler for both the TB and atherosclerosis model is a standard binary heap. Although the simulation times are currently

acceptable, as the models progress in complexity this will become more of a factor. Therefore I would recommend changing the queueing mechanism within the scheduler to a calendar queue.

The calendar queue was proposed in the paper [36], the algorithm consists of using uniform buckets. Its performance is proposed to be of O(1) in comparison to binary heaps which have a performance of O(log n). Therefore by implementing this queueing mechanism we should obtain an improvement in computation time of the simulation.

5.1.2 Random Movement

Currently the movement of agents is fixed to a specific time, (i.e. to occur every 100 time-steps) and does not change. To more realistically model the movement of cells, agents should be more stochastic by moving at random times.

5.1.3 Concurrency

Despite achieving respectable speeds for both simulations as the models increase in complexity using concurrent threads can help decrease computation times and better utilise multi-core machines. MASON does provide concurrent threading ability, however, parts of their threading facilities are untested and may provide unexpected problems.

5.1.4 Atherosclerosis Model Improvement

There is a large body of theoretical work detailing the process of atherosclerosis and the different immune cells involved. Although the key cells have been modelled there are a number of other agents that can be added to the simulation. Currently we do not model cytokines in either model. Cytokines act as messengers between immune cells. We also do not model the chemokine 'Fractalkine' which recruits macrophages and is involved in the process of activating T-cells.

The paper [31] also suggested that some plaques were found to contain SMCs that had transformed into foam cells. The paper [31] also suggested that oxLDL can upregulate active macrophages, currently the simulation does not support these processes.

T-cells upregulate(activate) macrophages by secreting IFN-γ as well as cell to cell interactions. However, currently the model approximates this complex process in a very simplistic manner. Perhaps once IFN-γ is included into the model the importance of macrophage upregulation will be better observed.

5.1.5 FCFARA algorithm improvements

The FCFARA algorithm can be improved to provide a more realistic solution. Currently agents that are pushed away from a location by a foam cell will always find spaces in a particular order. We could add a stochastic element to the simulation such that agents when re-located will not always move into exactly the same cell given the same condition. This would be more realistic as collisions between agents do not always result in agents moving to the same location. Also agents could move further than a single compartment away depending on the size of the force from being pushed by a foam cell.

5.1.6 Foam Cell Formation

Currently when active, deactive and upregulated macrophages transition into a foam cell they instantaneously increase nine times in size. This is not realistic as the foam cell itself grows in proportion to the number of lipids it consumes. It would be more realistic if the transition to a foam cell and growth as a foam cell was more gradual. However, this would be more difficult to implement and will effect computation time of the simulation.

5.1.7 Uncertainty and Sensitivity Analysis

The atherosclerosis model can now undergo a series of uncertainty and sensitivity analysis as was undertaken by [1] for the TB model. By modifying the parameters of the model we can obtain different outcomes, such as plaque rupture or plaque stabilisation. We can calculate the effect changing a particular parameter has on the necrotic core at the centre of the lattice or the number of foam cells present in the plaque. By performing various tests we can conclude what factors can affect plaque rupture.

Appendix

Rule Based Model(RBM)

As well as investigating a multi-agent approach to model Mtb and atherosclerosis I also investigated using a rule based model. Although Mtb infection has never been modelled using RBMs other complex biological systems have been developed using RBM such as cellular signalling pathways. By expressing biological interactions in terms of rules we can avoid the combinatorial explosion that can beset differential equations.

By using a RBM we can express rules in an intuitive graphical form that transparently represents biological knowledge[5]. Allowing us to naturally build, modify and discuss the developed model. By attempting a rule based model we had hoped to uncover which aspects of the sequence of events that lead towards a granuloma formation are key.

I was hoping to implement the RBM using prolog. Prolog would allow me to state the facts and rules which relate objects in the problem domain to each other. The rules and facts are computed by the inference engine at run-time. Prolog code is easy to read and write programs which build structures.

The RBM has been implemented in other biological models. I implemented the RBM to cellular signalling as detailed by the paper [5]. I implemented a simple example that consisted of only 12 rules. The example contains three agents Kinase K, a target T with two sites and and a Phosphatase P. A kinase agent K can bind and unbind to a target site, the kinase may perform the action of phosphorylating the site (it is not important to know what this action means). A phosphatase agent P can also bind and unbind to a target site and it can unphosphorylate the site. The rules below are only for site x the rules are exactly the same for site y.

Textual notation: ! = indicates two end points to a link.~u = unphosphorylated~p = phosphorylated

$K(a), T(x) \rightarrow K(a!1), T(x!1)$
$K(a!1), T(x!1) \rightarrow K(a), T(x)$
$K(a!1), T(x\tilde{}u!1) \rightarrow K(a!1), T(x\tilde{}p!1)$

$P(a), T(x) \rightarrow P(a!1), T(x!1)$
$P(a!1), T(x!1) \rightarrow P(a), T(x)$
$P(a!1), T(x^{\sim}p!1) \rightarrow P(a!1), T(x^{\sim}u!1)$

I translated these rules into a form of event calculus, which allowed me to reason about the actions of each agent and their effects. I was able to achieve similar results to those obtained in the paper for the above example, however, it was difficult to expand the approach to a larger biological system such as Mtb and thus was not pursued.

Glossary

Active Macrophage(TB): An infected macrophage which has been activated for the specific task of eradicating bacteria.

Active Macrophage(Atherosclerosis): A macrophage that is recruited to LDL focal sites by endothelial cells.

Adhesion_Molecules Proteins located on the cell surface, are involved in binding with other cells or with the extracellular matrix.

Antigen: Surface marker made of protein specific to that cell.

Chemokine: Chemoattractive cytokine- attract cells to source.

Chronic Macrophage: An infected macrophage, whose intracellular bacteria load has passed a specific threshold passing the macrophage from the state of infected macrophage to chronic. Can not be activated via T-cells.

Cytokines: Small soluble intercellular messenger.

Deactive Macrophage(Atherosclerosis): A macrophage that is recruited to LDL focal sites by endothelial cells.

Dendritic Cells: Effector cell of the immune system- important in antigen presentation to T-cells.

Extracellular Bacteria: Bacteria that has not been phagocytosed and has not been neutralised and continues to replicate.

Endothelial_Cells The thin layer of cells that line the interior surface of blood vessels.

Fatty Streak: A small collection of monocytes. Are the first grossly visible lesions in the development of atherosclerosis.

Granuloma: An area of chronically inflamed tissue by the mononuclear phagocyte system to a slowly soluble antigen or irritant.

Infected Macrophage: A resting macrophage that has been unable to remove bacteria through the process of phagocytoses and the intracellular bacteria replicate within the host macrophage. Can be activated via T-cells.

Intima The innermost layer of an artery

Intracellular Bacteria: Bacteria that has been phagocytosed and has not been neutralised and continues to replicate.

IP-10 IFN-γ inducible protein 10 , is a small cytokine secreted by several cell types in response to IFN-γ.

LDL Low-density lipoproteins. A type of lipoprotein that transports cholesterol and triglycerides from the liver to peripheral tissues.

Lymph Node: Specialised tissue containing lymphocytes.

Lymphocyte: Major group of effector cells and includes T and B cells.

Macrophage: Is a type of mononuclear phagocyte, an effector cell of the immune system.

MCP-1 Monocyte Chemotactic Protein-1 (MCP-1) , A chemokine which recruits monocytes.

Matrix Metalloproteinases Zinc-dependant enzymes (or endopeptidases) that degrade extracellular matrix proteins.

Phagocytose: Engulfing a foreign antigen.

Plaque An accumulation of substances such as lipoproteins, inflammatory cells and collagen in the inner lining of an artery.

Resting Macrophage: Resident macrophage within the lung, which has the ability to phagocytose bacteria.

Smooth Muscle Cells An involuntary non-striated muscle, found within the tunica media.

T-Cells: Effector cell of the immune system important in regulating other immune cells and cytotoxicity.

Upregulated Macrophage: An Active macrophage that has been upregulated by T-cells.

Bibliography

[1] J. Segovia-Juarez, S. Ganguli, and D. Kirschner, Identifying control mechanisms of granuloma growth during Mycobacterium tuberculosis infection using an agent based model, J. Theoret. Biol., 231 (2004), pp. 357–376.

[2] J. Wigginton and D. Kirschner, A model to predict cell-mediated immune regulatory mechanisms during human infection with Mycobacterium tuberculosis, J. Immunol., 166 (2001), pp. 1951–1967.

[3] S. Ganguli, D. Gammack, and D. Kirschner, A metapopulation model of granuloma framework in the lung during infection with Mycobacterium tuberculosis, Math. Biosci. Eng., (2005), pp 535-560.

[4] D. Gammack, S. Ganguli, S. Marino, J. Segovia-Juarez and D. Kirschner, Understanding the immune response in tuberculosis using different mathematical models and Biological scales. (2005) pp. 312-345.

[5] V. Danos, J Feret, W. Fontana, R. Harmer and J. Krivine Rule-based modelling of cellular signalling. (2007) pp. 17-41.

[6] Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem 81, 2340–2361 (1977).

[7] A. Phillips and L. Cardelli A correct abstract machine for stochastic pi-calculus. (2004)

[8] World Health Organization, 2001. WHO Report 2001: Global Tuberculosis Control. Technical Report, World Health Organization

[9] Comstock, G., 1982. Epidemiology of tuberculosis. Am. Rev. Respir. Dis. 125, 8–15.

[10] Tufariello, J., Chan, J., Flynn, J., 2003. Latent tuberculosis: mechanisms of host and bacillus that contribute to persistent infection. The Lancet 3, 578–590.

[11] Dannenberg, A., Rook, G., 1994. Pathogenesis of pulmonary tuberculosis: an interplay of tissue-damaging and macrophage activating immune responses—dual mechanisms that control bacillary multiplication. In: Bloom, B. (Ed.), Tuberculosis: Pathogenesis, Protection, and Control. ASM Press, Washington, DC.

[12] McDonough, K., Kress, Y., Bloom, B., 1993. Pathogenesis of tuberculosis: interaction of Mycobacterium tuberculosis with macrophages. Infect. Immun. 61, 2763–2773.

[13] Zhang, M., Gong, J., Lin, Y., Barnes, P., 1998. Growth of virulent and avirulent Mycobacterium tuberculosis in human macrophages. Infect. Immun. 66, 794–799.

[14] Orme, I., Cooper, A., 1999. Cytokine/chemokine cascades in immunity to tuberculosis. Immunol. Today 20 (7), 307–312.

[15] Sadek, M., Sada, E., Toossi, Z., Schwander, S., Rich, E., 1998. Chemokines induced by infection of mononuclear phagocytes with mycobacteria and present in lung alveoli during active pulmonary tuberculosis. Am. J. Respir. Cell Mol. Biol. 19, 513–521.

[16] Saunders, B., Cooper, A., 2000. Restraining mycobacteria: role of granulomas in mycobacterial infections. Immunol. Cell Biol. 78, 334–341.

[17] Krombach, F., Munzing, S., Allmeling, A., Gerlach, J., Behr, J., Dorger, M., 1997. Cell size of alveolar macrophages: an interspecies comparison. Environ. Health Perspect. 105 (5), 1261–1264.

[18] Capuano, S., Croix, D., Pawar, S., Zinovik, A., Myers, A., Lin, P., Bissel, S., Fuhrman, C., Klein, E., Flynn, J., 2003. Experimental Mycobacterium tuberculosis infection of cynomolgusmacaques closely resembles the various manifestations of human M. tuberculosis infection. Infect. Immun. 71, 5831–5844.

[19] Sprent, J., 1993. Lifespans of naive, memory and effector lymphocytes. Curr. Opin. Immunol. 5, 433.

[20] E. Giacomini, E. Iona, L. Ferroni, M. Miettinen, L. Fattorini, G. Orefici, I. Julkunen, and E. M. Coccia, Infection of human macrophages and dendritic cells with Mycobacterium tuberculosis induces a differential cytokine gene expression that modulates T cell response, J. Immunol., 166 (2001), pp. 7033–7041.

[21] M. D. Iseman, Tuberculosis therapy: Past, present and future, Eur. Respir. J. Suppl., 36 (2002), pp. 87s–94s.

[22] D. Gammack, C. Doering, and D. Kirschner, Macrophage response to Mycobacterium tuberculosis infection, J. Math. Biol., 48 (2004), pp. 218–242.

[23] http://www.cs.gmu.edu/~eclab/projects/mason/

[24] www.americanheart.org

[25] http://emedicine.medscape.com/article/150916-overview

[26] Dpeartment of Health UK government

[27] Kumar and Clark Clinical Medicine Sixth Edition

[28] Daniel Steinberg, Atherogenesis in perspective: Hypercholesterolemia and inflammation as partners in crime, (2002), pp. 1211-1217.

[29] Andrew C.Li & Christopher K.Glass, The macrophage foam cell as a target for therapeutic intervention, (2002), pp. 1235-1242.

[30] C. Cheng , R. de Crom, R.van Haperen, F.Helderman, B. Mousavi Gourabi, L.C A. van Damme, S.W. Kirschbaum, C.J. Slager, A.F.W van der Steen and R.Krams The role of sheer stress in Atherosclerosis

[31] Goran K. Hansson, Immune Mechanisms in Atherosclerosis, (2009)

[32] Fernando O. Martinez, Laura Helming and Siamon Gordon. Alternative Activation of Macrophages: An immunologic fuctional perspective, (2008), pp. 451-483.

[33] Fabio Martinon, Annick Mayor and Jurg Tschopp. The inflammasomes guardians of the body, (2008), pp. 229-265.

[34] David M. Mosser and Justin P. Edwards. Exploring the full spectrum of macrophage activation, (2008)

[35] Zhong-qun Yan, Goran K, Hansson. Innate immunity, macrophage activation and atherosclerosis, (2007), pp. 187-203.

[36] Randy Brown, Calendar Queues: A Fast O(1) Priority Queue Implementation for the Simulation Event Set Problem. 1988.

[37] Goran K, Hansson, Peter Libby, The immune response in atherosclerosis: a double-edged sword. 2006.