INVESTIGATING THE DYNAMICS OF HEPATIC INFLAMMATION THROUGH SIMULATION

DANIEL MOYO

PhD

University of York
Department of Computer Science

April 2014

ABSTRACT

Inflammation is a fundamental mechanism for the body to induce repair and healing in tissues, and exacerbated inflammatory responses are associated with a wide variety of diseases and disorders. Categorising the various cells, proteins, and precise mechanisms involved in initiating and driving inflammation poses significant challenges, due to the complex interplay that occurs between them.

In this thesis, I will introduce a deadly parasitic disease called Visceral Leishmaniasis (VL) as a case study in using computational modelling techniques to elucidate the mechanisms underpinning inflammation. During VL infection, inflammatory aggregations of immune system cells form, these are called granulomas. Granulomas function to contain and subsequently remove infection. Whilst immunological studies have provided insights into the structure and function of granulomas, there remains a breadth of questions which laboratory techniques are currently incapable of answering. As such, the challenges facing biologists from a scientific perspective will be addressed, I will then argue after a thorough review of the relevant literature, that agent-based computational modelling is a logical choice for research into granuloma formation, and that such models can help answer some outstanding questions in the field.

The thesis presents the process of designing and developing the first spatially resolved model of liver localised granuloma formation during VL. The development and use of modelling and simulation to study granulomas has involved close collaboration with immunologists at all stages through conceptualisation, modelling, implementation, and also results interpretation. I describe the use of established statistical techniques to instill confidence in both the model, and the results it can produce through simulation.

Through iterative hypothesis generation and testing, the research undertaken has allowed for several predictions to be made, some of which have biological significance and which were later validated experimentally. Specifically, transcriptomic data analysis revealed that both infected and uninfected Kupffer cells are equally capable of responding to infection in a similar manner, something which wasn't previously evident in the literature. Using this transcriptomic data, I investigated through simulation, several experimental scenarios and elucidated a novel mechanism of immune system regulation in the liver microenvironment. Using an experimental model of Leishmania *donovani* infection, I demonstrated that such an immune regulatory mechanism can be overcome with the expansion of early promoter cells called Natural Killer T cells.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

Sleep, it's over...

— Tubelord

ACKNOWLEDGMENTS

I was privileged to have such an amazing and supportive supervision team during my years at York. Thank you to Jon Timmis and Paul Kaye for introducing me to such an interesting project and for advising me throughout. Also, none of this work would have been possible without the guidance of Lynette Beattie and Paul Andrews, thank you both.

Thank you to the York Centre for Complex Systems Analysis (YCCSA) for being a great environment for collaboration, and to the York Computational Immunology Lab for our weekly lab meetings and helpful feedback.

The CD1d Tetramer used in part iii of the thesis was a kind gift to our lab from Professor. Vincenzzo Cerundolo (Oxford).

On a personal note, I couldn't have done this without my lovely supporting girlfriend Charlene and her family, thank you for putting up with years of late nights spent in front of the laptop.

Thank you to my parents Phillip and Leslie for supporting me, we got there in the end!

Lastly, the biggest thanks go to my mum Alison who I wish could be here to watch me see this through.

Finally, this thesis would not have materialised if the music of Tubelord was not in my life; "Our First American Friends" is a masterpiece.

DECLARATION

I declare that the research presented in this thesis is the result of my own work, except where stated explicitly. Some of the ideas and figures in this thesis have appeared in the following publications:

Moore JWJ, **Moyo D**, Beattie L, Andrews PS, Timmis J and Kaye PM (2013) Functional complexity of the Leishmania granuloma and the potential of in silico modeling. Front. Immun. 4:35, 2013.

York, April 2014

Daniel Moyo

Date

To my mum.

Part I

INTRODUCTION

1 RESEARCH CONTEXT AND MOTIVATION

In 2007 the World Health Organization (WHO) published a report on the global impact of Leishmaniasis, a disease that can affect humans [World Health Organization, 2007a]. The report highlighted that nearly 350 million people are at risk of contracting the disease. There are three main types of Leishmaniasis, often transmitted through the bite of infected female sandflies [World Health Organization, 2014]. *Cutaneous* leishmaniasis (figure 1(**a**)) is the most common of the diseases, causing ulcers and lesions. *Mucocutaneous* leishmaniasis (figure 1(**b**)) is a more extreme case of the disease that can causes "partial or total destruction of the mucous membranes of the nose, mouth and throat cavities and surrounding tissues" [World Health Organization, 2014, Page 1.]. Finally, *Visceral* leishmaniasis (VL) (figure 1(**c**)) is a deadly form of the disease that causes inflammation of the spleen and liver, and is ultimately fatal if left untreated. In recognition of the 2007 report, the WHO approved a resolution to provide member states with guidance on how they should tackle this global health problem [World Health Organization, 2007b]. Recommendations include increasing awareness through public-health initiatives, and to promote both the prevention and control of the disease. Development of vaccines is also a goal of the WHO resolution.

Focusing on VL, researchers at the Centre for Immunology and Infection (CII) at the University of York have been developing therapeutic vaccines to reduce the numbers of deaths caused by the disease. Recent findings estimate that VL has an annual global incidence of approximately 58k, though true estimates are expected to be in the region of 202-389k [Alvar et al., 2012]. Several factors contribute to the variability in disease estimates; for example due to socio-economic factors in many countries, there is limited or no access to health care. This leads to either misdiagnosis due to poor medical care, or the disease carries on unreported if those infected have no avenue by which to seek aid [Desjeux, 2004]. The disease is present in 98 recorded countries [Alvar et al., 2012] and many of these are still considered underdeveloped [Desjeux, 2001]. Understanding the immune system response to VL is fundamental to the study of the disease and its pathogenesis.

In this thesis I explore the use of Agent-based Modelling and Simulation (ABMS) to compliment the biological research at the CII, as part of wider computational immunology research conducted in the York Computational Immunology Lab. Whilst significant content is devoted to biological research, and explaining the biological context underpinning that research, the core Computer Science contributions I provide are in the areas of simulation engineering, simulation use, and Agent-based Modelling (ABM).

This first chapter introduces the motivation for research, outlining the immunological domain that provides inspiration for the design and development of a computational model of liver inflammation. The challenges facing the field will

(a) (b) (c)

Figure 1: The Leishmaniases. The three forms of Leishmaniasis. **(a)** Cutaneous, **(b)** Mucocutaneous, and **(c)** Visceral.

be addressed, and several research questions defined. Finally, I summarise the research contributions of this thesis.

1.1 VL–INDUCED GRANULOMAS

Granulomas are multi-cellular inflammatory aggregations of immune system cells that form in a number of infectious diseases, including: *VL*, *Mycobacterium tuberculosis* (Mtb) and *Sarcoidosis*. Granulomas form around an initial infected cell focus, which is typically a macrophage (a type of phagocyte). Granuloma function has long been purported to contain and subsequently remove infection, though the efficacy of the granulomatous response at removing infection has recently been debated, with a review into the tuberculosis granuloma implicating them in the perpetuation and exacerbation of infection [Ramakrishnan, 2012]. Granulomas can be grouped into two categories: ineffective and effective [Murray, 2001]. Effective, or hypertrophied, granulomas are enlarged and exhibit properties that facilitate parasite or bacterial clearance. Ineffective granulomas can exhibit all of the characteristics and common structural traits associated with an effective granuloma, but for unknown reasons, provide inadequate or no response in terms of parasite/bacterial clearance.

In VL, hepatic (liver) granulomas are formed subsequent to infection with *Leishmania donovani* or *Leishmania infantum* parasites, and Kupffer cells (KCs) are the characteristic infected phagocyte that initiates the inflammatory response (section 3.2). The accepted canon is that granulomas are associated with asymptomatic VL in dogs and humans [Pearson and Sousa, 1996; Sanchez et al., 2004], and also function to convey protection in a mouse model of VL called experimental VL (EVL). In EVL, granulomas have been shown to be heterogeneous in their structure, cellular composition and function [McElrath et al., 1988; Murray, 2001], though there is still uncertainty regarding the precise mechanisms underlying these heterogeneous responses, and early granuloma assembly in general.

1.2 MOTIVATION FOR RESEARCH

Despite the advent of new technologies, acquiring the experimental data required to further understand the complexities of cells in the hepatic microenvironment is challenging. Liver function is impaired rapidly *ex vivo*, due to the lack of blood perfusion which provides vital nutrients to liver cells. Liver sinusoids (small blood vessels) collapse without blood perfusion, leading to the obfuscation of the true structure of the hepatic microenvironment and its cellular contents (see section 3.1). *Ex vivo* studies also provide only static experimental observations that can merely allude to properties of cells in the sinusoids, such as gene expression, parasite burden, or cell morphology.

Aside from the plethora of more established wet laboratory tools and techniques used by biologists, novel experimental approaches, devised in the last couple of decades, have helped to build a more integrated picture of the complex cell responses and interplay relating to granulomas. Technology such as two-photon intravital microscopy [Piston, 1999; Oheim et al., 2006; Benninger and Piston, 2013] can be utilised to provide dynamic, real-time observations of immune system responses in tissues *in vivo*. Two-photon intravital microscopy is particularly useful for the study of the liver microenvironment, overcoming some of the limitations previously described relating to blood perfusion and the processing of tissues *ex vivo*. As such, two-photon imaging has been used to study and characterise the dynamics between T cells and macrophages during development of mycobacterial (*Mycobacterium bovis*) granulomas [Egen et al., 2008, 2011], and in VL has helped elucidate that KCs are the only type of APC capable of presenting antigen to $CD8^+$ effector T cells [Beattie et al., 2010a]. However, two-photon microscopy also has several limitations due to the nature of the technology which is based on laser excitation. Prolonged exposure to the laser causes tissue to become progressively damaged, and means that imaging studies can only be performed for several hours at best. The duration can be increased moderately, however this has several trade-offs including imaging resolution and not least the prolonged exposure of the mouse, albeit under anesthesia, to a procedure that is surgically invasive.

Where animal testing and experimentation is concerned, there has been a long-standing drive for that experimentation to be performed within the strictest ethical bounds. The 3Rs, *Replacement*, *Reduction* and *Refinement* are the pillars of the framework for ethical scientific experiments involving animals. The 3Rs were first introduced by Russel and Burch [1959], and are now promoted by organisations such as the NC3Rs [Accessed 29th July 2013]. *Refinement* is a broad topic relating to both animal welfare and capitalising on data and deliverables from experiments that are unavoidable. *Reduction* aims to reduce the numbers of animals required to acquire a set of experimental results, or in another sense, getting more data from those experiments with the same number of animals. Finally, *Replacement* is the drive to ultimately avoid using animals altogether, or replace the use of animals classed as 'protected', which includes all living vertebrates. There are many experimental techniques which can fulfil the role of replacement, such as *in vitro* methods that use human or animal material

that is cultured, or material that has been harvested through natural deaths or as a by-product of meat and food processing. Computer modelling is also an example of replacement, and has the potential to both reduce and replace certain aspects of animal testing in the future, which will ultimately refine any animal testing that must invariably be done.

Increasing computing power and methodological advances in modelling and simulation have facilitated growth in the field of computational approaches to systems biology [Germain et al., 2011]. Despite this, the computational study of VL and VL-induced granulomas is in its infancy. Computational modelling has proved useful for understanding inflammation in a variety of other contexts [Pogson et al., 2006; An and Christley, 2012; Pothen et al., 2013], and Agent-based Modelling (ABM) in particular lends itself to the study of tissue and cellular level inflammation (see section 2.1.3.2). ABM is a computational modelling approach that allows for intuitive mapping of entities, such as biological cells, to computational analogs called agents (see section 2.1.3), and is a useful tool for examining how low-level interactions and component behaviours can manifest themselves as higher-level observable phenomena.

At time of writing, the literature is devoid of a detailed ABM of VL-induced granuloma formation, though several exist for tuberculosis granulomas. Tuberculosis granuloma models are predominantly focused on the microenvironment of the lung (see section 2.1.4), and therefore are not particularly suited to providing any fundamental insight into VL-induced granulomas that develop in the liver. Given the difficulties and limitations of current experimental techniques, and the ethical and financial drive to promote the 3Rs in all aspects of research, the creation of a computational model is timely in this context. Creating an ABM of hepatic inflammation would allow us to investigate the dynamics of cellular interactions in the liver, and allow several hypothesis to be tested and experiments repeated for each, both contributing to the 3Rs drive by acting as a "plausibility filter for putative hypotheses" [An and Christley, 2012, p. 324], and providing alternative insight into the underlying processes which are currently challenging to study experimentally. Not only would the field benefit from having the first detailed ABM of VL-induced hepatic inflammation, but the process of developing a computational model in conjunction with experimental collaborators can be evaluated, giving further insight into how to better engineer and utilise computational models in a fast-moving, ever changing research landscape.

1.3 RESEARCH HYPOTHESIS AND STRANDS

Given the panoply of challenges facing researchers from both a technological and ethical perspective, and the breadth of outstanding questions in the field of VL-induced granulomas (see chapter 3), there is a strong case for the development of a computational ABM of VL-induced granuloma formation. This would also go some way towards evaluating whether such models can indeed contribute to the 3Rs drive, and ultimately benefit animal welfare if they are more widely adopted in the future. Therefore, this thesis will address the following hypothesis:

A spatio-temporal agent-based model can be used to simulate heterogeneous inflammatory responses in the liver microenvironment, and provide new insight into the underlying mechanisms responsible for inflammation.

To investigate my core research hypothesis, I must answer several underlying research questions:

1. Can a spatio-temporal agent-based simulation, incorporating the key cellular players and interaction dynamics into biologically plausible simulation environments, produce a heterogeneous population of granulomas?
2. What role does the structure of the hepatic microenvironment play in granuloma heterogeneity?
3. Does KC spatial location and mediators of cellular recruitment, account for the varying size and distribution of granulomas observed *in vivo*?

1.4 THESIS CONTRIBUTIONS

Based on the research questions listed in section 1.3, the thesis will make the following contributions to the field:

1. The first agent-based model of early hepatic granuloma initiation.

In designing and developing an ABM of granuloma initiation (chapters 4 and 5), CoSMoS [Andrews et al., 2010] principles will be adopted in order to carefully separate the biological domain from the proposed modelling abstraction of that domain, and the subsequent implementation. By adopting an agile approach to design and development, involving experimental collaborators in the entire process, coupled with the use of statistical techniques to establish confidence in the simulation, a novel ABM will be a product of this research.

2. A determination on the role of environmental structure on the formation of granulomas.

Statistically accurate simulation environments will be generated (chapter 5) and used in simulations to determine the extent to which structure influences the formation of granulomas (chapter 6).

3. Informed predictions relating to the role of cells and cellular influences on granuloma heterogeneity.

Based on iterative *in silico* experimental design, the simulation will be explored using different experimental hypotheses relating to various mechanisms that are not easily testable with current laboratory techniques. Specifically: functional cell heterogeneity, recruitment and retention dynamics in the liver microenvironment,

and the role of KC spatial location (chapters 6 and 7).

1.5 THESIS STRUCTURE

The thesis is comprised of 4 Parts and a total of 8 chapters excluding the appendix. Part i includes this introductory chapter, and will discuss the use of modelling and simulation to further understand biological processes (chapter 2), with particular emphasis on inflammation.

Part ii will outline the design of the model and development of the simulator in phases dictated by the CoSMoS framework (section 2.2.1), including: the domain, a literature review relating to the biological domain of interest (chapter 3); the domain model, a scoping study based on the domain that acts as a specification for the modelling and simulation endeavour (chapter 4); and finally, the platform model, an implementation-specific analog of the domain model (chapter 5).

Part iii is comprised of two chapters: the first presents the results from a statistical exploration of the simulation output, and addresses questions relating to structure and KC retentive mechanisms (chapter 6); the second chapter is a body of results relating to the function of attractive mediators, acquired through iterative hypothesis generation and prediction, which includes wet-lab experimentation and validation (chapter 7). Finally, Part iv critiques the body of research presented in this thesis, highlighting the contributions to the field and potential future work.

2 BIOLOGICAL INSIGHT THROUGH COMPUTATION

The purpose of this chapter is to introduce the current approaches in methodology for modelling of complex biological systems, with a focus on how models are engineered into software products. This chapter begins with a review of computational modelling, with a focus on modelling of inflammation, and I will justify why I focus predominantly on agent-based models and granuloma associated diseases. I will then discuss practices and approaches for modelling complex systems. Finally, gaps in the literature will be identified and these help justify the research proposed in this thesis.

2.1 COMPUTATIONAL MODELLING IN BIOLOGY

Systems biology emerged as a means to process and interpret the ever increasing deluge of biological data in a holistic context. Systems biology marries hypothesis-driven research and novel experimental approaches with the high-throughput data acquisition that advances in tools, technologies, and methodologies has provided [Ideker et al., 2001]. Whilst the profile of systems biology has increased, with attempts to advocate systems-level understanding as a means to further our knowledge of biological processes [Kitano, 2002], new tools and technologies are constantly evolving and revealing even more new questions than they are helping to answer. The field is still arguably in its infancy, and indeed has spawned sub-disciplines such as the application of systems biology to immunology, utilising genomic and proteomic data analysis and acquisition, bioinformatics and computer modelling and simulation to compliment and enhance the existing work-flow of biological research [Germain et al., 2011].

2.1.1 Graphical Modelling and Simulation

There are a variety of bespoke graphical notation tools and simulation environments for use in creating biological models. These include: BioUML, a visual modelling and simulation tool for systems biology [Kolpakov and Puzanov, 2006]; Molecular Interaction Maps (MIMs), graphical notation for modelling biochemical networks; and a proposed amendment to MIM notation by both Kitano and Kohn (Kohn 1999 as cited in Kitano [2003]; Kohn et al. [2006]. These software tools have varying degrees of usability, which is defined here as the extent to which a tool can be used without any prior experience or training. In order to improve the usability of modelling software, tools with visual interaction aim to make model creation accessible to both biologists and programmers [Germain

et al., 2011]. Examples of such tools include Cellware - a tool for modelling cellular processes [Dhar et al., 2004] and COmplex PAthway SImulator (COPASI), a simulator for biochemical reaction networks [Hoops et al., 2006]. Other modelling environments exist such as Reactive Animation (RA), which brings together reactive systems design and front-end animation tools [Efroni et al., 2005a]. Reactive systems are those that function by responding to various inputs, outputs and system behaviours in parallel, with information from the environment influencing the system heavily [Cohen and Harel, 2007]. Design and specification of these reactive systems is achieved primarily with statecharts [Harel and Gery, 1996], and a proprietary tool called Rhapsody can then be used to create executable models [Harel and Kugler, 2004]. Graphical user interfaces are employed to facilitate interaction and manipulation of these executed models [Efroni et al., 2005a]. RA merges the visual formalism of Statecharts with engaging, data generated, visual representations of the biological system [Efroni et al., 2005b].

There has also been some effort to create a universal format for storing models for many of the previously described tools. The Systems Biology Markup Language (SBML) [Hucka, 2003] is one such format. It aims to improve interoperability, allowing models to be utilised by multiple researchers across a wide variety of simulation platforms. Should software used to create a model become unsupported, the model can be opened and maintained using one of over 230 software tools known to be capable of interpreting SBML models [Bergmann et al., 2011]. Portability means that users can take advantage of application specific features of specific tools, for example a tool with a graphical editor might be used to create a model, then another tool more suited to analysis may be used to process and interpret the acquired data [Finney et al., 2006]. A constraint to SBML is that its specification encodes *entities* and *reactions*, *reactions* or processes are more commonly associated with mathematical models and expressions, and therefore it is difficult to encapsulate agent-based models, which are based on objects with states and interaction rules, into structure SBML definitions. Therefore SBML is context specific and more suited to particular types of models.

The tools described in this section have a variety of uses; however, none of them are clear choices for the creation of a spatio-temporal model to understand emergent phenomena.

2.1.2 Mathematical Modelling of Biological Systems

Ordinary differential equations (ODEs) are often used to model and describe a variety of biological systems. ODE models are scalable to allow the study of potentially large populations of cells and the subsequent population-level dynamics that arise. The core benefit of using ODE models is that often relatively few equations with small numbers of parameters are sufficient to describe a given system [Bauer et al., 2009], keeping models relatively simple and computationally tractable. These models assume that each model element or population is homogeneous in function and exists in a uniform space. In the context of biological cells, ODEs thereby ignore inherent differences in individual cell behaviour

and functionality over a given period of time, details that are of relevance from a biological perspective.

Partial differential equations (PDEs) can be used to incorporate spatial and temporal dynamics into mathematical equation based models. Some have previously argued that this invariably increases the complexity and therefore the computational requirements of those models, diminishing some of the original benefit of the approach [Bauer et al., 2009]. However, as with any model, the context, complexity, and implementation platform all play a role in the determining the resource requirements of a given model, and therefore it is possible to have extremely fast, computationally efficient implementations of ODEs and PDEs for a wide variety of purposes.

Differential equations have been used to model systems in a variety of inflammation related contexts. Equation based models include a model of infection induced acute inflammatory response [Reynolds et al., 2006], and the effect of pro and anti-inflammatory cytokines on the function of phagocytes during Mycobacterium tuberculosis (Mtb) infection [Marino et al., 2010]. Mathematical modelling in the biosciences is an extensive field, and in the context of acute inflammatory responses is already well reviewed [Vodovotz et al., 2004]. Referring back to several themes highlighted in the original research questions (section 1.3), environment and spatial location are two properties that form the focus of this research. Considering that approaches such as agent-based modelling exist (section 2.1.3), which is inherently based on capturing heterogeneous populations of entities interacting in a spatial environment, ODEs and PDEs are not as immediately suitable for examining the relationship between environment and the spatial location of individuals, and the influence both have on population-level behaviours.

2.1.3 Agent-Based Modelling

ABM is a computational approach for simulating a system by modelling its constituent components as autonomous entities called agents. In agent-based models of biological systems, agents can encapsulate entities, such as cells or molecules, at a variety of different biological scales. Simulated environments can be constructed and populated with agents. The powerful utility of ABM comes from the ability to investigate causality and emergence, and to determine how macro-level phenomena can manifest themselves as a result of low-level stochastic interactions between agents [Bonabeau, 2002]. Conditional rules determine how agents transition between pre-defined states, and how they move and interact with and upon their environment. Also, heterogeneous populations of agents can be instantiated to determine how functional differences between agents play a role in the system.

ABM allows for a great degree of flexibility with regards to both the quantities of agents and rules governing interactions, thus models can be modified to find the required degree of abstraction and to incorporate new theories and hypotheses about a given system, grounded in evidence-based assumptions. Where little or no evidence exists, reasonable assumptions can be made or inferred about agent

behaviour, provided the assumptions are plausible and that they are explicitly defined and documented. Through simulation and subsequent experimental validation of hypotheses, model assumptions can be refined or updated to reflect current understanding of the system, furthering the predictive capacity of the model.

Agent-based modelling and simulation, which has history in modelling cooperation and decision making in political and social science [Axelrod, 1985], did not gain more widespread adoption until the 1990s, predominantly due to its computational intensive nature at the time; though this limitation is not as significant in recent times. Aided by the advent of agent-based modelling software such as NetLogo and Swarm (reviewed in section 2.1.3.1), ABM has been adopted in fields as diverse as ecology [Grimm and Railsback, 2013], economics [Phan and Varenne, 2010], and biology [Forrest and Beauchemin, 2007].

Having introduced ABM, the next section will outline various agent-based modelling platforms that can be used to implement models into executable simulations, and the advantages and disadvantages inherent with some of those tools. I will then review agent-based modelling in the particular context of inflammation (see section 2.1.3.2) and elaborate on ABMs and computational models that are more relevant to granulomatous inflammation (see section 2.1.4), which is the focus of the domain of interest in this thesis (expanded on in chapter 3).

2.1.3.1 *Agent-Based Modelling Platforms*

The list of agent-based modelling platforms is extensive and as such this section will not discuss the list in its entirety. At the time of writing, Nikolai and Madey [2009] provide the most comprehensive categorisation of ABM software by: programming language, licence type, user support and documentation, and required operating system. An alternative review focuses on a more detailed comparison in terms of execution speed, input-output functionality, scheduling, and required programming expertise between the following ABM platforms: *Swarm*, *Java Swarm*, *Repast*, *MASON* and *NetLogo* [Railsback et al., 2005].

Dedicated agent-based modelling platforms are not a requirement for creating an ABM; scientific toolkits such as MATLAB® may also be used. Alternatively, modellers may write their own bespoke model in a computer language of their choosing. The advantage of many dedicated agent-based modelling platforms is that they generally provide a suite of pre-written software libraries, with specific functionality to implement the ABM into an executable form. These libraries may include functionality for simulation scheduling, visualisation, or code for utilising high performance grid computing facilities. The disadvantage of pre-written software tools is that the user is invariably trusting the authors of said tool in ensuring that the platform has been engineered rigorously and is suitably free from errors. Several ABM platforms have strong user communities, providing valuable feedback and bug reports to developers for future releases, such as NetLogo (proprietary-free-to-use) or MASON (open-source). These tools will now be described along with some other established ABM platforms, each having been referenced in various journal publications relating to computational models.

NetLogo is a free-to-use multi-agent modelling environment written in Java [Wilensky, 1999], which utilises an agent-centric modified version of the Logo programming language. The high-level programming language Netlogo uses is accessible to veteran programmers and novices alike, though it lacks some of the data structures and control logic of more established programming languages. The low barriers to entry of NetLogo make it a very popular tool for both education and research [Tisue and Wilensky, 2004], and it is the most mature platform in terms of abundance of documentation and user tutorials. There is debate as to the merit of NetLogo for scientific research [Railsback et al., 2005], as much of its history and functionality is grounded in pedagogy and in making ABM accessible for non-programmers. ABMs utilising NetLogo have established themselves in scientific journals, and some of the biological ABMs reviewed in section 2.1.3.2 make use of the platform.

Multi-Agent Simulator Of Neighborhoods (*MASON*) is an open-source discrete-event multi-agent toolkit developed in the Java programming language, with an emphasis on execution speed [Luke et al., 2004]. The MASON toolkit was designed to efficiently handle large numbers of agents, very useful for simulating large scale social science and insect systems [Balan et al., 2003], though support for substantial numbers of agents is a desirable feature of any agent-based simulation toolkit. MASON has libraries to handle scheduling, check-pointing and 2D/3D visualisation, as well as being platform independent and capable of producing identical simulation results across all major computing platforms using random number seeding. Demonstrated uses of MASON cover a diverse range of application areas including modelling of climate change [Hailegiorgis et al., 2010] and ant foraging [Panait and Luke, 2004]. In the context of immunology, MASON has been used to develop simulations of population dynamics of cells in Experimental Autoimmune Encephalomyelitis (EAE) [Read et al., 2011], a murine model of Multiple Sclerosis, and also in the development of a simulation of lymphoid organogenesis and the development of Peyer's patches in the small intestines of mice [Alden et al., 2012]. Both case studies benefited from the existing libraries provided by MASON by allowing the authors to dedicate more time towards establishing confidence in their simulations.

Repast Simphony [North et al., 2013] is the latest incarnation of the *Repast* family of ABM modelling libraries. Repast Simphony builds upon *Repast 3*, with a suite of open-source, freely available, extensible simulation libraries written in Java, C#.NET and a bespoke Python varient called Not Quite Python (NQPy). Repast Simphony code is extensible due to its code archictecture, giving users the ability to extend, modify or replace a wide variety of its functionality. The model development process normally involves first integrating Repast Simphony with the Eclipse Integrated Development Environment (IDE), and users can then write program code, develop flowcharts for program logic, and view simulation output and visualisation all from within the IDE. Similar to MASON, Repast Simphony is built with object-oriented design as the core paradigm of how user models are developed and also integrated with existing simulation libararies. Repast models can also be extended to work on High Performance Computing facilities [Collier and North, 2012].

Swarm [Minar et al., 1996; SDG, 2013] is a framework where models are developed as a collection of *swarms*, which themselves can be collections of various heirarchical objects. Swarm decouples the model from output and visualisation by enforcing the use of *observer swarms* to be constructed to interrogate the state of a model object. Swarm was first implemented in the Objective-C programming language, though *Java/Swarm* was later developed to allow users to develop Swarm models in the Java programming language.

Flexible Large-Scale Agent Modelling Environment (*FLAME*) is an ABM simulation platform built upon the principle of communicating X-machines [Coakley et al., 2006], similar to finite state machines but with the additional capacity to cater for state memory. The key benefit of modelling with X-machines is that they provide a formal means of describing a model, which can help with the validation and verification of simulation software. FLAME also boasts the capacity to work on both desktop machines and using massively parallel grid computing facilities. FLAME has been used predominantly in the context of both biological [Adra et al., 2010; Holcombe et al., 2012; Sun et al., 2007] and economic modelling [Deissenberg et al., 2008]. *FLAME GPU* [Richmond and Romano, 2008] extended the FLAME framework to exploit the parallel architecture of modern Graphics Processing Units (GPUs). Agent-oriented simulations are then developed using the FLAME framework and run as Compute Unified Device Architecture (CUDA) code, providing a significant performance boost over the original FLAME implementation when comparing two identical agent-based simulations [Richmond et al., 2009].

What is clear from the literature and my concise review, is that no ABM platform is considered the exemplar, it is user knowledge, implementation specific requirements, and also user preference that dictate the most suitable platform for a given modelling purpose.

2.1.3.2 *Inflammation Associated Agent-Based Modelling*

Inflammation is a fundamental immune system response, with the over-arching purpose of tissue repair, which results from a wide range of diseases and injury associated damage. Inflammation is such a complex response to understand and study because of the immune system's decentralisation, and the inherent ability for it to naturally balance pro and anti inflammatory responses. Problems occur when diseases or injury interferes with the immune systems capability to self-regulate inflammation, leading to exacerbated inflammatory responses that are associated with disorders such as cancer, obesity, and diabetes. The role of granulomatous inflammation in the context of VL is further elaborated in chapter 3.

The agent-based modelling paradigm represents a powerful utility for the study of inflammation in a variety of contexts, since inflammation often manifests itself as some form of tissue-scale observable phenomena, and agents and environments can be designed to represent cells and biological tissues where these phenomena occur. Figure 2 summarises some of the intuitive analogues that agents have compared with features of biological cells. There are several proponents of ABM for inflammation, and the large body of inflammation related modelling literature

is already well reviewed (see Mi et al. [2010]; An [2006, 2010]; An and Christley [2012]), though I will discuss some of the more recent examples in the literature.

Brown et al. [2011] created an ABM to study the responses of macrophages when introducing particulate matter into the lung. Their model was capable of reproducing a variety of the emergent phenomena, such as fibrosis and localised tissue damage, which they subsequently validated using histological analysis of mice lungs subsequent to smoke exposure. Whilst interesting, the authors specifically state that they have not made any attempts to calibrate their simulation. Much of their simulated data is based on averaging an arbitrary number of simulation runs, and many cell populations also have arbitrary baseline values. The authors are not alone in this respect, others have interpreted their simulation data without properly understanding how stochasticity affects simulations. Studies by both Pothen et al. [2013], testing the plausibility of a novel inflammatory 'twitch' hypothesis, and Li et al. [2008], predicting individual treatment outcomes using an ABM of vocal fold inflammation, also lacked the rigour that comes from using statistical analyses to mitigate the effects of stochasticity in simulations.

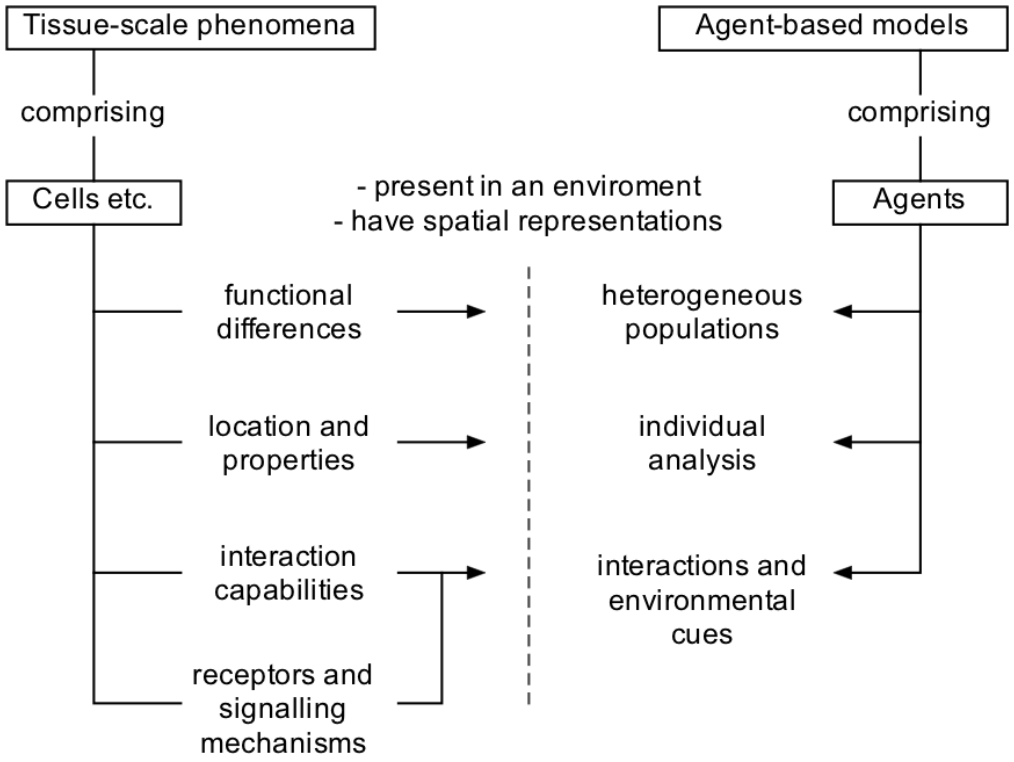


Figure 2: Suitability of agent-based models for studying tissue-scale phenomena; as published in Moore et al. [2013]

The Epitheliome project has contributed a large body of literature on the dynamics of wound healing in epithelial tissues. Walker et al. [2004b] created a proof-of-concept ABM to investigate the growth of epithelial tissue in simulated culture, demonstrating different tissue growth characteristics. Their model has

been subject to various extentions, firstly extending their POC to predict the effects of calcium concentration on wound closure [Walker et al., 2004a], and also integrating ordinary differential equation (ODE) reaction diffusion models of the epidermal growth factor receptor (EGFR) signalling pathway into their existing ABM, forming a hybrid multi-scale model [Walker et al., 2008]. Their approach demonstrates how stochastic simulations can be used to predict how low-level cellular heterogeneity can influence emergent system-level cell behaviour.

Stern et al. [2012] developed an ABM to study the restoration of epithelial barriers, simulating wound healing. Their model was later extended to study how host-pathogen interactions in the gut affect wound healing [Stern et al., 2013]. The authors predicted that impairment and delay of wound healing could be causes of a phenomenon known as an anastomotic leak, the leakage of fluids through epithelial barriers that leads to sepsis and morbidity after surgery. Their study is an example of how ABMs are powerful for hypothesis driven research, with several biologically plausible hypotheses examined and results interpreted in the context of the underlying biology. Where the study lacks is that there appears to be no calibration and validation of the simulation based on the model extensions. The authors state that the anastomotic leak study is based on a 'previously validated ABM'; however, as discussed by Greaves et al. [2013], extending simulations can lead to dramatic changes in simulation behaviour and data output. Therefore, careful consideration and a principled engineering approach must be adopted when extending simulations of complex systems.

Another example of ABM being used to study wound healing is in the context of non-healing diabetic foot ulcers (DFU) [Mi et al., 2007]. Whilst an interesting first step, and the first published ABM of DFU, the study does not appear to demonstrate that wound healing actually manifests itself at the tissue-level, and instead they present correlates of wound healing such as: increased levels of collagen, cytokines, and a tissue damage surrogate metric. From a predictive standpoint, the ABM provides several 'non-intuitive' findings that the authors believe could explain some of the dynamics of wounding healing in DFU.

The utility of ABMs is not restricted to studying tissue level phenomena. Agent-based modelling and simulation has been used to good effect in the study of single cells and various intracellular signalling pathways. The NF-κB signalling pathway, considered central for the regulation of immune response and inflammation, has been the focus of several agent-based modelling studies. Pogson et al. [2006, 2008] published their ABM investigations into the NF-κB pathway looking at Toll-interleukin receptor (TIR) mediated pathway activation. Their studies show how the NF-κB pathway affects inhibitor complexes responsible for regulating itself via IκB mediated negative feedback. They predicted key roles for several pathway components, namely IκB-action interactions in controlling the negative feedback loop, which they later validated experimentally. The authors state that ABM overcame several limitations of previous ODE models in this context, primarily the tendency of concentrations modelled with ODEs to reach an equilibrium, giving rise to non-biologically plausible results. Their results demonstrate the power of ABM for studying an intracellular level processes. Similarly, Dong et al. [2010] implemented an ABM of a single cell, and the propagation of extracellular

signals (LPS) as a signalling cascade inside the cellular membrane and towards a nucleus. Their model implements a simplified version of the NF-κB signalling pathway, which acts as hub for downstream inflammatory responses within a cell. Their efforts whilst interesting, seem to oversimplify the NF-κB pathway, which is extremely complex, and their calibration and validation of model results was limited to a parameter sweep, of which they do not explain their method, nor do the authors describe any robustness or sensitivity analysis.

I have described several computational models developed to elucidate features of inflammation in a wide variety of contexts. Later in chapter 3, I will elaborate on the complexities of granulomatous inflammation, which was briefly discussed in the introduction to this thesis (chapter 1). The following section will critique the literature of equation based, mathematical, and agent-based models that specifically address granulomatous inflammation, which presents during a variety of diseases.

2.1.4 Computational Insights into Granulomas

Most granuloma modelling literature is in the context of Mtb, though there are precedents in the fields of other granuloma inducing diseases including sarcoidosis and VL. Several predictions about the immune system factors contributing to both the formation of Mtb granulomas, and their efficacy in controlling infection, have been studied using agent-based, mathematical, and multi-scale modelling approaches.

2.1.4.1 *Mycobacterium Tuberculosis Granuloma Modelling*

Segovia-Juarez et al. [2004] published an ABM of granuloma formation using a two-dimensional discrete lattice environment representing lung tissue. Their ABM includes agents representing macrophages and lymphocytes, with abstractions for various cytokines and chemokines, and was designed to elucidate the influential processes and functions responsible for granuloma formation in Mtb. Their initial model was later extended to determine the efficacy of granulomas at controlling infection and the role TNF-α plays in this process [Ray et al., 2009]. The same laboratory published further research detailing a mathematical model that was coupled with their previously published ABM to create a multi-scale model that predicted the main factors controlling availability of TNF-α within granulomas [Fallahi-Sichani et al., 2010, 2011]. The study suggested that TNF levels and bacterial burden are intrinsically linked, making TNF dynamics critical in determining a range of possible infection outcomes pertinent to bacterial growth and level of inflammation. Their multi-scale model was later refined to incorporate the dynamics of NF-κB [Fallahi-Sichani et al., 2012], a transcription factor ubiquitous in cells and a key component in cellular response to infection.

Whilst the study of Mtb granulomas using computational modelling has been fairly prolific in comparison to any other granuloma modelling efforts, the validation of many of the predictions has proved elusive, primarily due to the current technical difficulties inhibiting detailed measurement of granulomas *in vivo*/*in*

situ; this is partly the reason for using computational modelling in the first place. The published works provide detailed examples of the integration of highly detailed mathematical and agent-based models across multiple scales, though this high level of detail makes the predictions heavily reliant on the assumptions and parameterisation of the models, despite their use of sensitivity analysis to calibrate simulation parameters. Another issue when drawing on such work for inspiration for modelling liver granulomas, is that the two-dimensional lattice approach adopted, whilst it may be acceptable for modelling the lung, is insufficient to represent the liver microenvironment for a variety of reasons that I outline in section 2.1.5.

2.1.4.2 *Sarcoidosis Granuloma Modelling*

Aguda et al. [2011] described an ordinary differential equation (ODE) based model of the dynamics of sarcoidosis using an immune network model of cellular players and cytokine mediators, with interactions and transitions having a corresponding ODE representation. Interpreting the formation of granulomas was achieved by monitoring steady states in their simulation, categorised by a sustained increases of proinflammatory cytokines, antigenic exposure and expansion of key cell populations in the model. Their model predicts that the immune response to sarcoidosis is biphasic in nature, depending on the level of antigenic challenge during infection, and that sufficient exposure of antigen can lead to the promotion of a Th1 (protective-type) response. Whilst interesting, the proxy measures used for measuring a granuloma are insufficient to truly understand the interplay between individual cells in their system. No environment was modelled explicitly, and as such, the dynamics of both the formation and maintenance of granulomas from a spatial perspective could not be dissected with such an equation based model.

2.1.4.3 *Visceral Leishmaniasis Granuloma Modelling*

Modelling of granulomas in VL is still in its infancy. Flügge et al. [2009] modelled the spatial aspect of hepatic granulomas, investigating whether heterogeneity in KC function might affect recruitment of innate liver cells. The limited scope of the model made it difficult for the authors to draw any useful biological conclusions, and it served primarily as a proof of concept for potential future work. An alternative, more complex, stochastic petri-net was used to predict the dynamics of various cells and cytokines across the full time course of *L. donovani* infection [Albergante, 2010; Albergante et al., 2010, 2013]. The authors predict a key role for KC produced autocrine IL-10 as a regulator of effector mechanisms within a granuloma. The complexity of this model allows it to be used as a tool for generating hypotheses and prediction as to the interplay between cells and cytokines, and how that ultimately relates to parasite clearance. However, the stochastic petri-net approach is incapable of modelling interactions within a spatially defined environment, thus, the role of the liver micro-environment and interactions at the cellular level were not investigated in these studies.

2.1.4.4 *Summary of Granuloma Modelling*

This section has highlighted the relevant literature relating to computational models of granulomas. Previous efforts have adopted a variety of techniques including equation-based models, agent-based models, stochastic petri-nets and hybrid approaches. Whilst many of these models have proven interesting and indeed have their own strengths, there are several issues with why they are insufficient to be emulated when creating a model of granuloma formation in the liver.

Firstly, the equation based and stochastic petri-net approaches ignore the inherent structure of the tissues within which the granulomas form, and the interplay between individual cells and forming granulomas. Secondly, the models that do introduce a spatial environment often use the standard two-dimensional grid environment approach. As I will discuss in sections 2.1.5 and 3.1, the structure and function of the liver are intricately linked; therefore, such an approach is not ideal for studying liver granulomas spatio-temporally. The next section will review the various attempts to model the liver and its microenvironment for use in simulation.

2.1.5 Computational Liver Representations

Tissue function is often inherently linked to its structure, particularly in the case of the liver (see section 3.1). In order to further understand the physiological relationship between cells and their environment, and any signalling mechanisms within, any computational representation of a particular tissue needs to be a plausible abstraction of that environment.

Accurately modelling the liver lobule micro-environment has been a challenge in a variety of areas. Wambaugh and Shah [2010] generated connected graphs representing the sinusoidal network (liver micro-environment) in order to investigate blood-flow, chemical metabolism and cell responses. The graph representations facilitated the computationally efficient solving of blood flow. Virtual liver lobules were constructed by amalgamating individual sinusoidal sections, representative of plates of liver lobule, but whilst qualitatively similar to observed liver lobule morphology, the authors acknowledge that their networks are not generated based on statistical data.

Statistical data describing the overall size of lobules, average non-branched sinusoid length, and branching angles between sinusoids was first published by Höhme et al. [2010], who used a combination of confocal microscopy and a novel image processing and volume analysis method chain, to perform statistical analysis of 26 individual mouse liver lobules. The authors then constructed a single statistically representative computational liver lobule, used as the basis for a model to investigate lobule regeneration after damage. Their initial models failed to accurately capture how regeneration manifests itself through proliferation and movement of hepatocytes alone. After testing multiple hypotheses, the authors determined that if their model included a rule forcing hepatocyte daughter cells to align to the nearest sinusoid, the proliferation of hepatocytes was then able to promote lobule regeneration, this process was dubbed “hepatocyte-sinusoid align-

ment" (HSA). The authors validated HSA experimentally, and their research is an exemplar in how computational models of complex systems can provide valuable insight into predicting otherwise unknown biological mechanisms. With respect to replicating the liver microenvironment, the methods employed by Höhme et al. [2010] are advanced and not trivial to replicate; therefore, a significant amount of time and resources would be required to repeat such a study and incorporate these environments into other models.

Bhattacharya et al. [2012] described a multi-scale model coupling a cellular level ABM of liver sinusoids to an ODE-based model of hepatocyte toxicity pathways. Whilst a novel step in its particular context, the model falls short in terms of accurately modelling the liver micro-environment. Only a small section of sinusoid was modeled, and the scales of individual cells in their model appear not to be calibrated to any biological data, particularly that of KCs. Similarly, in an early conceptual investigation into inflammatory cell recruitment, Flügge et al. [2009] only modelled a single Y-shaped branch of sinusoid. Though in fairness, sinusoidal structure was not the key focus for both of these studies.

In another study, Heiland et al. [2012] used the liver lobule as a case study for demonstrating the visualisation advantages of a modelling application called CompuCell3D (CC3D), which utilises the 'Visualization Toolkit' (VTK) [Schroeder et al., 2003]. The authors introduced 3D renderings of lobules with both sinusoids and hepatocytes modelled. Whilst interesting, the paper does not give any concrete details as to how the authors managed to mimic liver morphology, and whether or not their sinusoidal structures were algorithmically generated, and if so what algorithm was used.

Of the spatial representations of hepatic lobules discussed, that of Höhme et al. [2010] appears to be the most accomplished, as the structures acquired by the authors are those from real mice. However, the acquisition process requires expert knowledge and the need to follow a strict protocol to ensure that mice livers maintain their structure for imaging after harvesting. Also, their image processing tool chain is not openly accessible; therefore, their work would have to be replicated to turn lobule images into computational network representations. An alternative approach would be to generate artificial sinusoidal and lobule structures utilising the statistics published by Höhme et al. [2010], providing a means to create multiple structures whilst reducing the number of animals required for such an activity. This aspect of the research detailed in this thesis will be outlined in section 5.2.

2.1.6 Summary of Identified Literature Gaps

I have outlined some of the literature relating to computational modelling of inflammation, with a focus on agent-based models due to their capacity for spatially constrained models, including a review of various implementation platforms that can be used to develop those models. Subsequently, I highlighted previous efforts in the modelling of granulomas and granulomatous inflammation. This section summarises the gaps identified during my literature review, and makes clear the

motivation for the research detailed in this thesis. Based on the literature review, the following are the identified gaps in the literature, these gaps justify why the thesis contributions listed in section 1.4 are novel:

1. As outlined in the review, there is currently only one existing ABM that deals with the spatial formation of granulomas in the context of VL [Flügge et al., 2009], and that model was merely a proof of concept. The development of an ABM allowing the study of cellular migration and recruitment would provide the field with a valuable tool for more comprehensively studying potential mechanisms underpinning granuloma heterogeneity.

2. The majority of the ABMs reviewed in section 2.1.3.2 model environments as 2D lattice grids. Quite often, the tissue environments they represent are significantly more complex, and little to no reflection is made regarding how the simulated tissue environments are influencing their simulation results. To this end, an ABM that incorporates biologically plausible, statistically accurate simulation environments, would allow the role of structure to be investigated and understood in the context of the liver.

3. There is a clear need for better adoption of techniques to increase confidence in simulations and their results. Modellers must be confident that their models are engineered accurately and calibrated to available data, and both modellers and experimental collaborators must be confident that the models they use provide plausible abstractions of the underlying biology. Confidence can be garnered through clear documentation of design decisions and assumptions, understanding of stochasticity in simulation, and the robustness and sensitivity of a simulation to parameter perturbation. Therefore, the proposed granuloma model must be clearly documented to highlight design decisions and available data, as well as be calibrated and results interpreted using state of the art statistical techniques.

With the gaps in the literature defined, it is important to understand the methods that are available to adequately model and implement complex systems of this type. The following sections will investigate the process of engineering software for complex systems (section 2.2), and introduce the CoSMoS framework which can guide the development and use of complex systems simulations for research (section 2.2.1).

2.2 ENGINEERING SOFTWARE FOR COMPLEX SYSTEMS

There exists no widely accepted method of developing software for modelling complex systems. There have been various conceptualisations of the process, including Kitano [2002]'s cycle of research and experimentation, or Sargent [2007]'s

"Real world and simulation world relationships" (see figure 4). This section will introduce the CoSMoS framework for developing complex systems simulations, and discuss a variety of the commonly used modelling tools and approaches in the field of computational biology.

2.2.1 The CoSMoS Framework

The **Co**mplex **S**ystems **Mo**delling and **S**imulation (CoSMoS) framework [Andrews et al., 2010; Bown et al., 2012] is designed to guide the principled development and validation of complex systems simulations, with applications of the CoSMoS framework including a biological simulation of auxin transport in plants [Garnett et al., 2010], experimental autoimmune encephalomyelitis [Read, 2011] and Peyer's Patch organogenesis [Alden, 2012; Alden et al., 2012], illustrating a variety of biological problem domains suitable for the CoSMoS framework.

Adhering to CoSMoS provides flexibility in the development process. Models and software simulations can be developed iteratively, with both modeller and experimentalist (domain expert) working cooperatively to ensure a mutual understanding regarding the scope of the project and the research context. Collaborators, modeller and domain expert, are required to justify any abstractions made during the development process. Collaboration also extends to the analysis and interpretation of data gathered from simulations, as a modeller may not necessarily understand how their results are relevant and what contributions those results can make when interpreting them in the context of the domain interest.

2.2.1.1 *Phases of the CoSMoS Framework*

The CoSMoS framework is split into three phases, *Discovery*, *Development* and *Exploration*. The *Discovery* phase is concerned with identifying the domain of interest i.e. the biological question(s) to be addressed, then creating a *domain model* that represents, explicitly, both the domain experts' and the modellers' understanding of the system. Understanding of the domain of interest will not be absolute, or there would be no purpose for studying that domain, hence there will invariably be assumptions made about the domain, these must be made explicit. The *domain model* omits any simulation implementation details.

Development is the realisation of the domain model into a repeatable *simulation platform*. A *platform model* is created that serves as an engineering construct for the *simulation platform* (the implementation of the platform model in software executable on a chosen hardware platform). The platform model is derived from the domain model and as such it must incorporate its various assumptions and abstractions made. However, CoSMoS advocates that there is a strict separation between the domain and platform models, with the omission of any implementation specific concerns at the domain modelling stage. This separation ensures that system interactions or behaviours that may lead to a particular observable phenomena, are not explicitly coded into the simulation. Instead, a domain model will conceptualise how particular phenomena might manifest themselves from low-level interactions, and the platform model implements those

low-level interactions, and crucially, system-level phenomena 'emerge' from the simulation without being directly implemented.

Exploration utilises the *simulation platform* to answer the original questions posed of the domain [Andrews et al., 2010, 2011]. This phase records the results of running experiments with the *simulation platform*, creating the *results model* which interprets those results in the context of the original research questions or hypotheses. The iterative nature of the CoSMoS process dictates that the interpretation of the results model be compared to the original domain model, assessing the suitability of the simulation as a representation of the real domain. Ultimately, the goal is to utilise a suitable *simulation platform* and the subsequent results model to drive experimentation in the real domain.

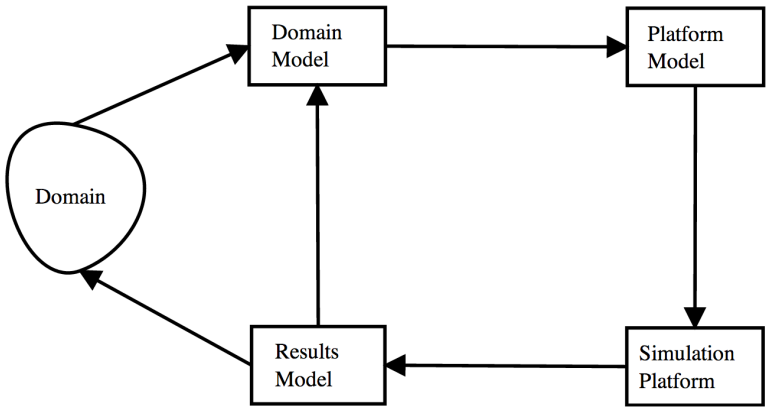


Figure 3: The CoSMoS framework. This diagram illustrates the phases of the CoSMoS framework. Information gathered about the domain is used to construct a domain model. This domain model forms the basis of the platform model. The platform model contains the required information to design and implement the domain model *in silico*. The simulation platform is the realisation of the platform model onto comprises of hardware and/or software. The results model is a record of observations acquired from experimentation with the simulation. A results model is used to interpret the simulation output in the context of the domain model and the research context. The research context encompasses the overriding goals of the research, such as research questions or hypotheses relating to the domain. Figure after Andrews et al. [2010]

Adhering to the CoSMoS framework ensures the separation of the domain of interest from the platform specification required to implement that domain in simulation. Whilst there are currently no tangible documents or deliverables advocated by CoSMoS, each stage of the framework is separate and followed in a specific order: domain -> domain model -> platform model -> simulation model and finally results model, though CoSMoS is iterative, and if newly gleaned insights can improve, or if they call into question, any features of the model then CoSMoS allows you to revisit earlier stages in the process and make the relevant changes before repeating the process.

The benefit of following the CoSMoS project structure also extends to the reuse, adaptability and extension of complex systems simulations. Li et al. [2013] demonstrated how an existing software simulation modelling the formation of vascular structures could be adapted to investigate tumour growth. The authors

outline how their original domain model, along with the aspects of its software implementation, could be reused for an alternate purpose. Whilst development time can be reduced by adopting such an activity of model re-purposing, it must be done with careful consideration of the original model assumptions, and how these might need adapting for the alternate purpose, and also the requirements for calibration of the new model Li et al. [2013]. Similar care is required when engaging in model extension. A CoSMoS project on the modelling of experimental auto-immune encephalomyelitis (EAE) [Read, 2011], a murine auto-immune disease, was extended by [Greaves et al., 2013] to investigate the role of a cellular pathway, CD200, involved in the regulation of T cell priming. As no widely accepted method for extending complex systems simulations exists, the original CoSMoS domain and platform models provided a basis by which to begin designing and implementing additions to the model.

2.2.1.2 *CoSMoS is Agile*

In situations where a software project requirements may be subject to a degree of change, traditional linear approaches to software development, such as the *waterfall* method, may not always be suitable. Academic projects are inherently unstable, due to a range of factors such as funding arrangements (PhD funding or grant limitations), fast turnover of human resource (short term contracts) and project time constraints. Those constraints have detrimental effects on the feasibility of long term software maintenance [Pitt-Francis et al., 2008]. As well as computer code, software refers to the relevant documentation and operational data that goes alongside the computer program.

CoSMoS advocates an agile approach to software development, though not any specific agile method. Agile is a type of software process model that focuses on iteration and flexibility, with an emphasis on customer and stakeholder engagement throughout the development of software. Customer, developer, and stakeholders work together to agree on software requirements, which are continually refined based on rapid prototype development and stakeholder feedback [Nerur et al., 2005]. This iterative and incremental development focuses on team engagement to bring multiple perspectives and problem solving capabilities to a project [Cockburn and Highsmith, 2001; Nerur et al., 2005; Sommerville, 2009].

There exist many agile methods (see Larman and Basili [2003]; Dyba and Dingsoyr [2008] for more comprehensive reviews), including: *Dynamic Software Development Method* (DSDM) [Stapleton, 2003], *Feature-Driven Development* [Palmer and Felsing, 2002], *Scrum* [Schwaber and Beedle, 2001], and *Extreme Programming* (XP) [Beck, 1999]. Each method has its own advantages and disadvantages, some better suited to different problem areas than others. Though a development team must inevitably adapt whichever variant suits them based on their own requirements.

The research outlined later in this thesis was developed by taking inspiration from the Scrum agile variant [Schwaber, 1997; Schwaber and Beedle, 2001], and I comment on the use of Scrum during this project later in the thesis (see section 8.5). The Scrum process is centered around three components; the *product*

backlog, the *sprint backlog*, and the *working software increment*. The product backlog contains a list of software features agreed upon by the stakeholders and the development team. The *Scrum Master* then has the task of prioritising items from the product backlog to be included in the sprint backlog. A *sprint* is a time-boxed unit of work devoted to the implementation or modification of a software feature. The sprint backlog is the ordered list of features assigned to the next sprint. After each sprint, a working software increment is created, and feedback from that increment feeds back to inform both the product and spring backlogs subsequent to discussions with the stakeholders.

2.2.2 Validating Models for Scientific Research

Grounding models and simulation in the real domain is the challenge of any modeller and domain expert. Demonstrating the validity of results and justifying these to the scientific community should be one of the primary concerns of anyone involved in computational modelling. Verification and validation (V&V) is a multifaceted problem for traditional software systems let alone software simulations of complex systems, which have their own unique developmental caveats. This section discusses V&V techniques in the context of traditional software systems, highlighting the challenges facing the application of these techniques during complex systems simulation development V&V.

In software engineering, *verification* and *validation* (V&V) is the practice of ensuring a given piece of software adheres to its specification (*verification*) and provides the expected functionality expected of it by the end user (*validation*) [Sommerville, 2009]. Methods of V&V can be either *static* or *dynamic*. Static methods can include activities such as source code analysis and formal verification of algorithms. Dynamic methods involve active system testing, for example, controlling input and monitoring for expected outputs. Static techniques can be performed at any stage of the software engineering process whereas dynamic techniques require an implementation of the program capable of being executed. It is impractical to validate and verify systems exhaustively, due to both the time and monetary constraints that V&V invariably places on a project. This makes it imperative that there be careful planning of any V&V activities before any development work has been undertaken, or as early in the process as possible. Sargent [2007] illustrated how the process of V&V can intergrate with the development of simulations for complex systems (figure 4). Sargent's depiction is very much similar in message to Kitano's 'Yin and Yang' description of hypothesis driven research, with computation as a complementary approach to the traditional cycle of research and experimentation [Kitano, 2002].

Nance and Sargent [2002] reviewed the history of *verification* and *validation* in the context of simulations, highlighting the chronology from V&V as two indistinguishable terms, to the current and clear distinction between both. Polack [2010] summarises that many of the techniques used by Sargent [2007] to verify and validate simulations derived from systems and software engineering methods. This in itself presents a problem in that validation techniques which make use

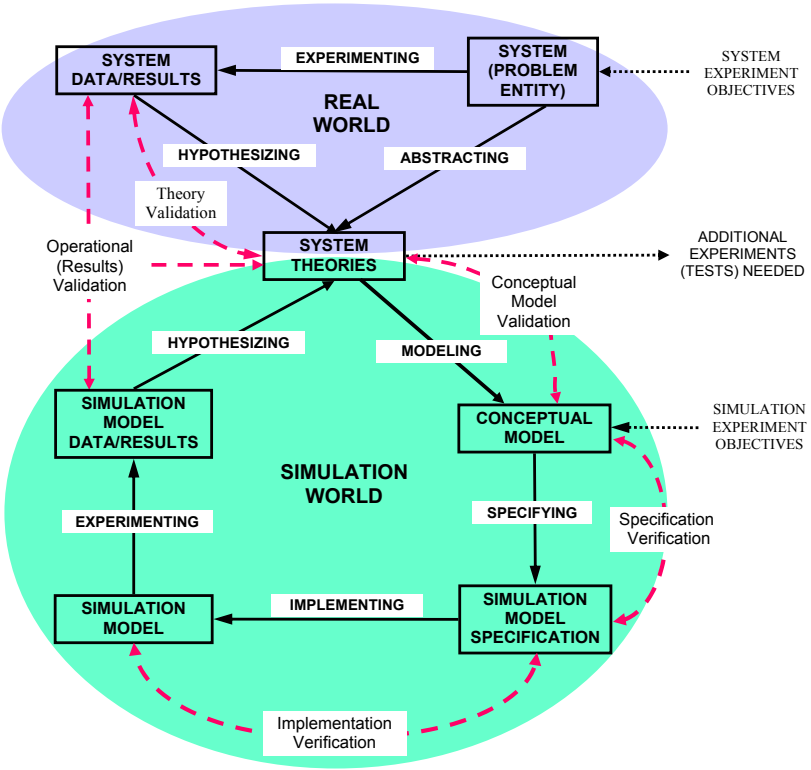


Figure 4: Real world and simulation world relationships with verification and validation - from Sargent [2007]. This figure illustrates how verification and validation can integrate with the process of modelling complex systems, by separation of the real world from the simulation world. One can formulate hypotheses and abstract from the real domain to establish system theories used to create a conceptual model, needed to define a simulation model specification. This is then implemented as a simulation that can then be used to run experiments and in some cases provide further insight into the system theories used earlier in the process.

of activities such as traceability analysis, cannot be truly suitable if the complex system that they are modelling is not fully understood - which in respect to biology, it never truly is [Polack et al., 2010; Andrews et al., 2008]. Instead, Polack [2010] suggests the use of *validity arguments*, inspired by argumentation which is used to demonstrate how evidence can satisfy requirements for safety [Weaver et al., 2005] (Argumentation is discussed in section 5.6.7). In the context of simulation engineering; particularly with agent based simulations, it is necessary to argue that the simulation is adequate in fulfilling its specification, as quite often simulations are made because certain aspects of the system are unknown.

2.2.2.1 *Calibration and Sensitivity Analysis*

The process of calibrating a simulation needs to involve the domain experts in order to ground the simulation in the biological domain. This has been highlighted recently by Read et al. [2011] who stress that sensitivity analysis is important to ensure that simulation results are robust to parameter alteration within the confines of biologically plausible parameter values, a practice also advocated by others that have used statistical techniques to validate their agent based models [Dancik et al., 2010; Marino et al., 2008; Alden et al., 2013]. It must be stressed that despite undertaking activities such as calibration and using this evidence in argumentation, it is never possible to provide absolute V&V for complex systems simulations. Calibration is discussed later in section 5.6.1, and statistical analysis of simulation output in section 6.1.

2.2.2.2 *Argument-Driven Validation for Simulation Science*

Argumentation is a term to describe how one validates assumptions based on a principled process of both reasoning and deduction, to arrive at a final conclusion. Argument-Driven Validation (ADV) applies this method to demonstrate the validity of approaches related to the design, engineering and interpretation of complex systems simulations [Ghetiu et al., 2009, 2010]. Validity is argued by explicitly detailing the design and development of a simulation, outlining the various assumptions made and any reasoned or deduced knowledge to substantiate any claims made. Subsequently, one argues that the engineered simulation adequately represents the underlying system and that the results are representative of observables from that system.

ADV makes use of an established graphical notation approach to document arguments, called Goal-Structured Notation. The use of diagrammatic notation and syntax used in argumentation allows arguments to be presented formally, thus more amenable to critical analysis [Polack, 2008]. The ability for arguments to be updated throughout the development process means that they can be improved if required.

GSN was originally developed for the purposes of argumentation for safety-critical systems, such as those used in the aerospace industries [Kelly, 1998]. Goal-structured arguments are tree like structures, beginning with a top-level *claim*, with *strategies* stemming from that initial claim, designed to comprehensively argue the validity of the top claim. Each *strategy* itself can have sub-claims, and

those in turn are backed up by *solutions/evidence*. GSN notation allows the *context* of any component to be noted, and also any *justifications* made. The graphical shapes used to represent the core components of GSN are depicted in figure 5.

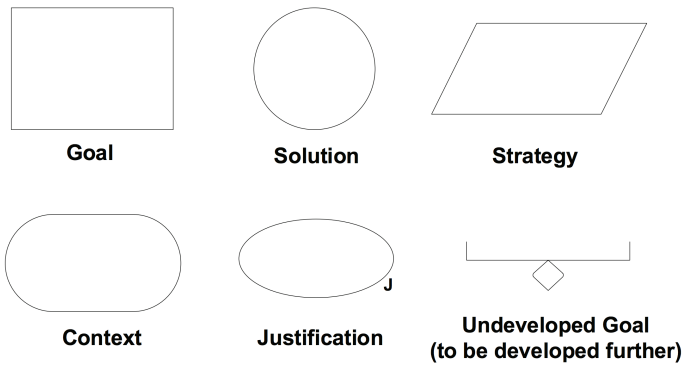


Figure 5: Goal Structure Notation, from [Ghetiu et al., 2010, Fig 1.].

There have been several applications of validity arguments applied in the complex systems modelling domain. Polack et al. [2011] used argumentation in parallel with developing a model of prostate cancer, capturing the behaviour of differentiating and dividing prostate cells. Their study covered argumentation of both the validity of the underlying cancer biology context, and the engineering of the software simulation of that biological domain. The authors posited that arguments are not static, and as well as having utility throughout the entire process from model to simulation, they can change through iterative updates, and quite often highlight gaps in understanding, or future avenues of research. Alden [2012] demonstrated further use of ADV within a biological context, developing claims to argue the validity of a simulation of Peyer's Patch development in the gut. That study built upon the previously described work by Ghetiu et al. [2009, 2010]; Polack et al. [2011], providing a further case study of how to incorporate various modelling assumptions and justifications into GSN constructed arguments.

Despite the limited use of argumentation in the complex systems modelling domain, yet the clear advantages that it brings from a validation perspective, argumentation appears a powerful technique to use throughout the process of engineering and utilising a simulation model. GSN structured goals, allowing one to both structure and document arguments that link evidence to assumption, help to instill confidence in that simulation model and facilitate transparent communication of the model and how it is built. Argumentation applied to the research outlined in this thesis is detailed in section 5.6.7.

2.2.3 Unified Modeling Language

Modelling and engineering complex systems simulations is a challenging task, and it is argued that the use of established software engineering practices can

help with the verification and validation of complex systems simulations [Polack et al., 2009]. This can be achieved in part by using visual modelling tools such as the Unified Modeling Language (UML); a multi-purpose, visual language for specifying components and interactions of a wide variety of object-oriented (OO) software systems [OMG®, 2010].

Diagrammatic system representations are constructed using a collection of up to fourteen different types of diagrams (as of UML version 2.3), grouped into three core categories: *structure*, *behaviour*, and *interaction* diagrams [OMG®, 2012], though interaction diagrams may be subsumed as a subset of behavioural diagrams. Collectively, UML diagrams can be used to visualise a system from both a high level (system-wide) and a low level (individual-entity) perspective.

There are several proponents of UML and the object-oriented paradigm for agent-based modelling and simulation [Read et al., 2009a,b; Alden et al., 2012, 2013], with the most prolific work by Bersini [2006, 2012], who has also demonstrated how UML can be used to depict equation based models in biology [Bersini et al., 2012]. Those who have adopted UML throughout the development of computational models argue that its core benefits are: the established nature of the technique in the software engineering domain, the accessible nature of the technique for non-experts, and the natural parallels that it has with systems capable of being modelled in an object-oriented fashion. However, adopters of UML do not have a consensus about how to proceed with UML diagrams once constructed. Some argue that automatic code generation is the logical way forward for the development of simulation software when modelling with UML [Bersini, 2006, 2012], and that the trend towards platform-independent model development is a beneficial one. Automatic code generation is made possible with tools that leverage the utility of techniques such as UML, or other abstract modelling techniques, and can interpret these high-level diagrammatic abstractions and generate programming code from them. James Rumbaugh, one of the original authors of UML, has expressed misgivings about the use of UML for automatic code generation, claiming that...“It never was meant to be a programming language. Use it to get the strategy right and write the final program in a suitable programming language” [Biancuzzi and Warden, 2009, pp. 343]. I echo Rumbaugh’s sentiments, and argue that the trend towards automatic code generation is not advantageous in a modelling context either, because as I discussed in section 2.2.1, directly linking the conceptualisation of a model with UML (the domain), to the implementation, or computer code of that model (the platform), can run the risk programming in the “answer”, or directly implementing system behaviours which should manifest themselves through emergence.

I have introduced the topic of UML, and will now discuss in more detail, some of the various diagrams that UML provides, and explain how they might be used in the context of biological systems modelling.

2.2.3.1 *UML for Models of Biology*

When modelling a biological complex system, *Class Diagrams* (structural) may be used to specify various objects, such as cells, and how they may have a rela-

tionship, as well as defining any plurality of those objects. However, as Read et al. [2009b] highlight, Class diagrams can be difficult to interpet when there is a high degree of connectivity between entities, and they are also only static representations of system entities. *Activity Diagrams* (behavioural) allow a modeller to depict conditional flow and concurrency in a system, and can also be used to form a more dynamic, system-wide, perspective to further detail how objects interact in a given event-driven scenario. *State Diagrams* (behavioural) on the other hand are suited to describing how individual entities function in pre-defined states, and how interactions or events may satisfy conditions, called gates, that govern transitions between those states. Each diagram in the suite of UML diagrams has its own unique and potential use, though not all are suitable for biological systems modelling. There is no right or wrong use for UML for modelling complex systems, diagram choices are often down to preference, and ultimately their use is justified if the diagram communicates the intended meaning.

Several authors both advocate and utilise UML for biological modelling, for example in the development of various models of the immune system [Bersini, 2006; Read et al., 2009a,b; Flügge et al., 2009; Alden et al., 2012; Patel et al., 2012]. UML was created for designing software systems and not for modelling biological systems, and therefore has no dedicated features to incorporate some of the core dynamics inherent in many complex systems, such as feedback mechanisms [Read et al., 2009a] or the ability to adequately represent an entity in multiple concurrent states. These limitations can be overcome with modifications to standard UML diagrams. Proponents of using UML to graphically model biological systems have argued its utility for model reuse [Bersini, 2012] and to facilitate communication between modeller and domain experts - those who ultimately validate the model [Read et al., 2009a,b].

UML and the OO paradigm are not only useful for specifying entities, relationships and functions of biological systems, but for allowing scalability in modelling endeavours [Webb and White, 2005]. Diagrammatic notations allow easier reuse and expansion of existing models, which can help ensure that researchers other than the original authors are capable of understanding how those software models work, facilitating scientific reproducibility. Particularly with the drive for ever more complex models, visual formalisms and anything that aids extensibility is ultimately beneficial.

There have been attempts to extend UML specifically for agent-oriented software engineering purposes, such as Agent UML (AUML) [Bauer et al., 2001]. AUML introduced some interesting additions such as *Protocol Diagrams*, designed to be a form of hybrid between UML *Sequence* and *State Diagrams*. The utility of *Protocol Diagrams* is that they can more clearly relay agent roles and constrain the types of communication that occurs between agents. It can be argued that there is no need for this type of hybrid diagram, as they could quite easily become very overwhelmingly complex with all but a simple model, and state and sequence diagrams separately can express states and interaction quite adequately. AUML style features were subsequently incorporated into UML version 2.0 by the Object Management Group (OMG®), the organisation responsible for standardising UML, causing proponents of AUML to re-evaluate its further

development and instead demonstrate how agent systems could be developed with the updated UML standard [Bauer and Odell, 2005].

Despite the plethora of UML diagrams, and different contexts in which they can be used, all adopters of UML for computational modelling use the technique as a some means of communicating their model, and to document their models and constituent components, and for that purpose, the literature seems to indicate that it can be very effective, despite the lack of widespread adoption in a variety of fields [Bersini, 2012]. The research detailed later in this thesis focuses primarily on using and in some cases extending the standard notation that UML provides, harnessing its power as a communicative medium, rather than the abstract syntax and semantics.

2.3 ON DOCUMENTING AGENT-BASED MODELS

There have been some published efforts to provide a standard means of documenting all of the relevant design details relating to agent-based simulations. Grimm et al. [2006] introduced the Overview, Design and Details protocol (ODD) as a method of capturing the relevant elements related to individual and agent-based models in the field of ecology. Elements include *purpose, state variables and scales, process overview and scheduling, design concepts, initialization, input* and *submodels* [Grimm et al., 2006, p. 117-119]. The authors argue that compartmentalising model information into these sub-categories makes it easier to write model descriptions, thus making them more understandable and also forcing the modeller to consider the relevant elements during their own modelling endeavour. Whilst the concept is interesting, and may be of use to those wishing to write up their papers for journals, the format may not always be the most appropriate choice for modellers. I would argue that whilst such a formulaic description of a model could prove useful, there may be cause for overlap between the relevant sub-categories, for example one may wish to describe design concepts alongside conceptualising entities and states, and ODD's categories force a separation between such elements. Within the economic modelling domain there are similar issues with describing models, and this inspired a set of ODD-like guidelines, but tailored specifically to economic ABMs [Wolf et al., 2013]. The fact that different disciplines have their own nuances makes a standardised protocol for documenting models difficult, and not always applicable.

A more recent extension argued against the common criticisms of the ODD protocol, advocating ODD for the documentation of any complex systems simulation [Grimm et al., 2010]. Alongside the limitations I described above, the authors describe some of the common issues of contention regarding ODD. For example, some feel that ODD is often overkill for simple models, or that there is often redundancy with having certain elements repeated in multiple ODD categories, though the authors argue that the ODD structure is flexible and can be made more concise, and that with careful consideration of their description, modellers can avoid repetition and redundancy in their model descriptions [Grimm et al., 2010, p. 2766].

Whilst there has been noticeable uptake of the ODD protocol, with 54 published model descriptions (predominantly within ecology, with minimal uptake in fields including epidemiology and the social sciences) [Grimm et al., 2010, p. 2761] I feel that such a protocol has benefits, but may not always be suitable for models of biological systems, with little evidence of ODD being used in the biosciences. To this end, the model described later in this thesis (see part ii) will be described according to no particular template, though the design and implementation will be separated into two over-arching chapters corresponding to the domain model and platform model components of the CoSMoS process.

2.4 SUMMARY

In this chapter, I began by reviewing a variety of methods for developing computational models, with a core focus on agent-based models due to their ability to be used for spatio-temporal modelling of individual entities. The literature relating to previous computational efforts in modelling inflammation were discussed, with a focus on granulomatous inflammation and efforts to create computational liver representations for the purposes of modelling.

Having identified a set of literature gaps in the computational modelling field relating to granulomas, I commented on a framework called CoSMoS, which can be adopted as an approach to developing complex systems simulations. I then discussed how complex systems models can be used and trusted as scientific tools for research. Lastly, I reviewed UML as a graphical notation tool that has several benefits for the documentation and communications of conceptual models, with the capability of being used as a software specification to some extent, should the modeller wish.

Next, chapter 3 introduce a more detailed description of the biological field of interest in this thesis (the domain). The domain highlights the gaps in understanding in the VL granuloma literature, and will form the basis for the conceptual domain model in chapter 4.

Part II

HEPATIC INFLAMMATION MODELLING AND SIMULATION – A COSMOS CASE STUDY

3 HEPATIC GRANULOMAS - THE BIOLOGICAL DOMAIN

This chapter will provide a comprehensive description of the biological domain, and the events believed to be responsible for granuloma initiation and subsequent formation and maturation. This domain chapter collates all of the relevant biological literature and information pertinent to the study, and provides an information template to facilitate formulating the more focused, scoped version of the domain which comprises the domain model (see chapter 4). A clear separation between the biological domain and the abstraction of that domain ensures transparency regarding what is and is not included in the model.

The chapter will begin by introducing the liver, as it is necessary to understand the structure of the liver microenvironment and how that influences liver function. Secondly, the role that liver resident macrophages and patrolling lymphocytes play in the formation of granulomas will be discussed, with an outline of the key signalling proteins and mechanisms pertinent to VL and hepatic granulomas. The literature outlined in this chapter pertains to experiments conducted on mice and not humans. I conducted the experiment to illustrate granuloma heterogeneity in figure 7 as detailed in section 8.4.1 using the protocol in Appendix 9.3.1.

Where not otherwise cited, biological information, in this section and throughout the thesis, was gathered from Janeway's Immunobiology 7th Edition [Murphy et al., 2008] and personal communication with Professor Paul Kaye and Dr Lynette Beattie, both resident at the University of York at time of writing. The reader is directed to the glossary for clarification of any terminology.

3.1 THE LIVER ORGAN AND MICRO-ENVIRONMENT

The liver is the largest vital organ and is comprised of five key systems/collective components: Hepatic Lobules, Stroma, Hepatic Vascular System, Hepatic Sinusoidal Cells and the Biliary System [Saxena et al., 1999; Ishibashi et al., 2009].

Lobules are the structural and functional sub-units of the liver, and are idealised as being columnar (hexagonal) in shape, though individual lobule structures can vary wildly from those typically illustrated throughout the literature (figure 6). Lobules consist of roughly hexagonal plates of hepatocytes that are surrounded by portal triads - vascular arrangements that consist primarily of a bile duct, a portal vein and a hepatic artery. Hepatocytes comprise the vast majority of liver mass, over 80%, and are responsible for much of the metabolic properties attributed to the liver (Arias et al. 2001, as cited in Kim and Rajagopalan [2010]). Lobules are divided into three zones: centrilobular/centrivenous (CV), located around central veins; the midzonal area (MZ), and the periportal (PP)

zone. These areas were categorised to represent several properties, including varying degrees of oxygenation with higher concentrations in the PP region carried by fresh blood, and an oxygen gradient towards the CV region [Jungermann, 1988; Jungermann and Katz, 1989; Arteel et al., 1995].

The Biliary System functions to transport hepatocyte produced waste to the gall bladder and small intestine, with vessels called bile canaliculi facilitating this waste transport, and stroma represents the collection of connective tissue present in the liver.

The Hepatic Vascular System is the network of veins and arteries of the liver. Blood flow originates from portal veins and hepatic arteries at the lobule periphery, and drains towards the central hepatic vein, providing vital nutrients and oxygen for liver cell survival. There are also hepatic arteries which provide highly oxygenated blood to the liver, and it is through channels, called liver sinusoids, that blood perfuses through, carrying platelets and red and white blood cells.

Hepatic Sinusoidal Cells (HSCs), the collective term for the various liver cells found within a liver lobule, are found lining the walls of the sinusoids. Types of HSCs include: Endothelial cells, Kupffer cells and Stellate cells. Kupffer cells in particular have a core role in response to a multitude of infections, and their function is intricately tied to the structure of the liver and formation of granulomas. Other non-HSCs are also known to be present in the liver, including neutrophils, which have a role in resolving bacterial infection and mediating inflammatory responses [Holub et al., 2009], before subsequently being ingested by Kupffer cells after providing their initial innate response to foreign pathogens.

Having provided a short review of the composition of the liver, I will now establish the various cells involved in driving inflammatory responses within the liver, with a focus on Kupffer cell responses as these are particularly pertinent to granuloma formation.

3.2 KUPFFER CELLS AND GRANULOMA FORMATION

Kupffer cells (KCs) play a pivotal role in the pathogenesis of VL and the formation of granulomas [Murray, 2001]. KCs are macrophages found distributed throughout the liver sinusoids [Naito et al., 2004], and they function as a surveillance network to rapidly filter any debris or pathogenic material from the blood. Within 2 hours of injecting mice with *L. donovani* parasites (amastigotes), the majority of parasites are removed from the blood stream by various phagocytes, and approximately 20% of KCs can be observed as being infected [Beattie et al., 2013]. However, after parasitisation of KCs, the precise mechanisms involved in the subsequent inflammatory response are not fully understood.

3.2.1 KC-Parasite Interactions

The dynamics of the KC-parasite interaction may have a role in determining the outcome of infection in EVL. Several characteristics of KCs have roles in manag-

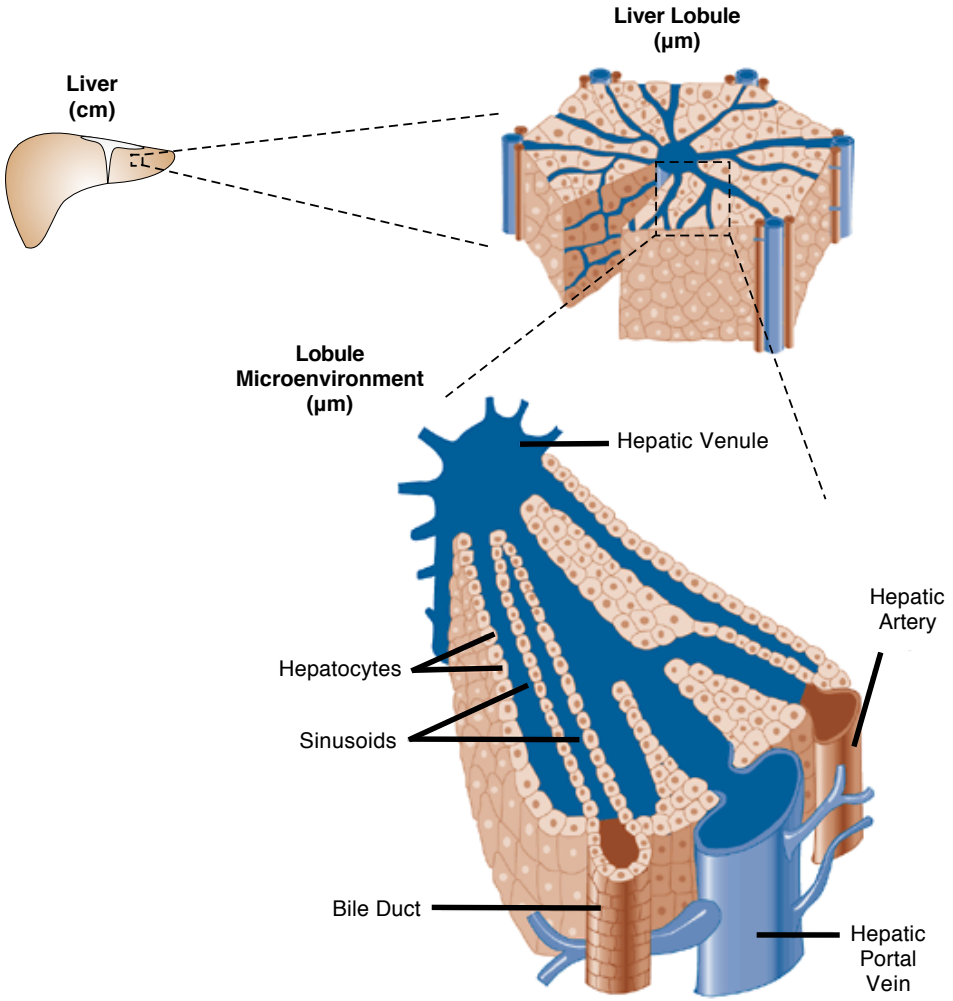


Figure 6: The Structure of the Liver Lobule. The liver is comprised of small functional sub-units called Lobules, and each lobule has a microenvironment with channels called sinusoids that contain hepatocytes and various hepatic sinusoidal cells (HSCs). The relevant scales are given in brackets. Adapted from [Cunningham and Van Horn, 2003, after Ross et al. 1995].

ing parasite burden, such as a natural resistance gene *Slc11a1* (also known as Nramp1 or Lsh), which was identified as a key factor in the control of *Leishmania* amastigote parasite growth [Crocker et al., 1984].

In terms of parasite removal, KCs have an impaired ability to produce reactive oxygen intermediates (ROI), such as hydrogen peroxide, which can act as a mechanism for the cell to become increasingly microbicidal [Lepay et al., 1985a,b]. The production of Nitric Oxide can also kill internalised parasites. Internal compartments called phagosomes play host to parasites, which can later fuse with other compartments called acidic lysosomes, forming phagolysosomes. Zilberstein [1991] (as cited in [Hommel, 1999]) showed that *Leishmania* has the ability to adapt to the change in the environmental pH, and that is just one of several mechanisms evolved by the parasites for surviving once phagocytosed. Intracellular *Leishmania* can also influence how their host cell behaves as part of the immune response. For example, *Leishmania* parasites have also been shown to have an influence on the presentation of antigen to $CD4^+$ T cells by APCs, inhibiting sustained TCR signalling and IFN-γ production [Meier et al., 2003].

3.2.2 KC Role in Granuloma Heterogeneity

Recruitment and migration of the various granulocytes, lymphocytes and monocytes, to the area of infected KCs helps initiate and physically form granulomas, though current understanding has yet to explain why only a fraction of the infected KC population develop fully formed, mature granulomas. This heterogeneity is exemplified in figure 7, a fluorescently labelled section of liver lobule harvested from a mouse at day 18 post-infection and imaged using confocal microscopy. The image shows: F4/80 expressing cells (green) - typical of mature KCs [Hume et al., 1984], amastigotes (tomato-red), cell nuclei stained with the fluorescent marker DAPI (blue), and finally tissue is stained with anti-glutamine synthetase to identify the central vein of the lobule (white - protocol in Appendix 9.3.1). Multiple infected KCs are visible, some immature granulomas with minimal cell infiltrate, others with large fused cores and greater mononuclear cell infiltrate, typical of medium to large sized mature/maturing granulomas [Murray, 2001].

Several studies have characterised the location, spatial distribution and function of KCs, and have provided mounting evidence that properties such as size, phenotype and age maturation, are heterogeneous in the KC population [Sleyster and Knook, 1982; Bouwens et al., 1986; Mochida et al., 1989; Naito et al., 2004], which may explain the resultant heterogeneous granuloma response. KCs, regarded for many years as a homogenous population, are now known to have two sub-populations, one motile and bone marrow derived and the other sessile and radiation resistant. Klein et al. [2007] determined that only motile KCs appeared to be responsible for immuno-inflammatory responses, which suggests that KC heterogeneity extends to cellular function.

Beattie et al. [2010a] sought to determine the origins of KCs found at the core of hepatic granulomas, demonstrating that blood borne monocytes are not found

inside mature granulomas. The authors postulated that only those KC already present inside the sinusoidal network are redistributed toward areas exhibiting inflammation. Further *in situ* analysis of KCs revealed visible change in KC form post-infection with *L. donovani*, with quantifiable membrane activity [Beattie et al., 2013]. These physiological changes may not necessarily transfer to downstream anti-leishmanial function; however, this can not be ruled out as a factor. Indeed Mackaness 1970 (as cited in [Mosser and Edwards, 2008]) gives credence to the fact that physiological changes may lead to increased anti-microbial properties of macrophages. There is undoubtedly complex interplay between KC host and pathogen in the context of *Leishmania* [Kaye and Scott, 2011].

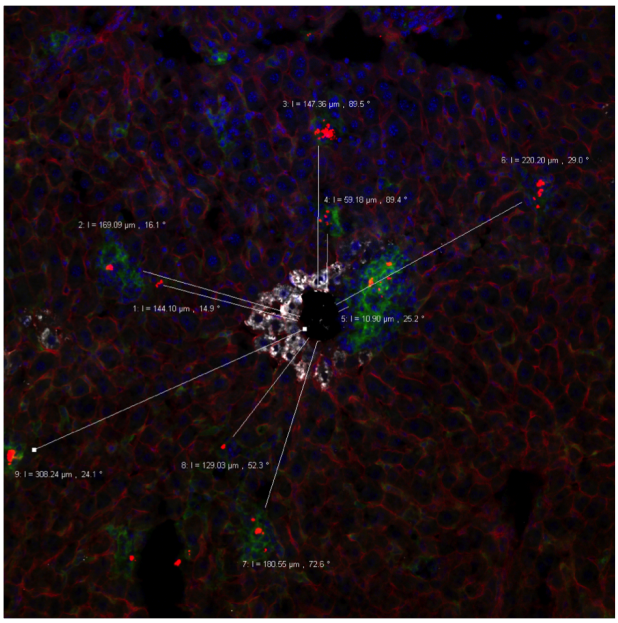


Figure 7: Confocal Microscopy of Hepatic Granulomas. Kupffer cells (F4/80), parasites (tdtomato), cell nuclei (DAPI), central vein (anti-glutamine synthetase - white). Image acquired from experiment detailed in Appendix 9.3.

3.2.3 Kupffer Cell Spatial Distribution

Kupffer cells are found distributed in different ratios throughout the PP, MZ, and CV zones of the liver lobule. Kupffer cell ratios have been reported as being in the order of 4:3:2 [Sleyster and Knook, 1982]. Similar quantitation reported mean distributions of 43%, 28%, 29%, and noted that only 64% of total KCs were labelled when latex beads were administered to mice in order to observe phagocytic uptake [Bouwens et al., 1986]. The results demonstrated that centrilobular KCs display impaired phagocytic capacity which can not be overcome with excess latex particle infiltrate. Centrilobular KCs were also reported as appearing smaller in size than their peri-portal counterparts. The varied KC localisation, size and phagocytic uptake might be a result of the dynamics of liver oxygenation and hypoxic gradients in the liver sinusoids, though the exact reasons are undefined.

3.3 NKT AND T CELL RESPONSES

T cells comprise the majority of cells within VL-induced granulomas, and they are capable of movement and interaction within both EVL and mycbacterial Calmette-Guérin (BCG) granulomas [Beattie et al., 2010a; Egen et al., 2011]. T cell subsets have identified roles in parasite clearance [Stern et al., 1988], and $CD4^+$ in particular are capable of producing interferon-γ, a KC stimulatory cytokine, during VL. Another T cell subset known as regulatory T cells (Tregs) are often associated with dampening immune responses; however, their role in VL-induced inflammation is poorly defined. Recent evidence suggests that $CD4^+$ T cells expressing both IFN-γ and IL-10 (a known property of regulatory type T cells) are present in the liver. Whether those regulatory $CD4^+$ T cells are actually present within granulomas is not yet known. The dampening function of Tregs is facilitated by the expression of cytotoxic T lymphocyte-associated antigen 4 (CTLA-4). When an antibody is introduced that is capable of blocking CTLA-4, the maturation of granulomas and subsequent anti-parasitic response is boosted, providing some evidence that T cells with a regulatory function are present within EVL granulomas [Zubairi et al., 2004].

Natural Killer T (NKT) cells are a subset of T cells that share properties with both T cells and Natural Killer (NK) cells [Kronenberg and Gapin, 2002], and are considered to be a bridge between innate and adaptive immunity, having been shown to rapidly activate NK cells upon activation [Carnaud et al., 1999]. Possessing the ability to produce immunostimulatory and effector cytokines, NKT cells are considered as key drivers of immunity [Juno et al., 2012; Mattarollo et al., 2012; Mussai et al., 2012; Pilones et al., 2012]. Heterogeneity which exists in the population also ultimately influences the overall response to both infection and autoimmunity [Godfrey et al., 2000]. Existing in various forms, one class of NKT cell has an invariant T cell receptor (TCR) and the other is more variable, similar to conventional T cells [Seino and Taniguchi, 2005]. The function of these *invariant NKT* (iNKT) cells makes them pertinent to infection with *L. donovani*. Recent evidence suggests that iNKT cells are an early driving force in the production of a favorable cytokine microenvironment that is conducive to optimal granuloma formation [Robert-Gangneux et al., 2012].

Whilst NKT cells are capable of driving adverse inflammatory responses, including chronic liver injury [Wehr et al., 2013], they represent a potent therapeutic, and their protective capacity has been demonstrated in a variety of disease contexts. After activation with α-galactosylceramide (α-GalCer), NKT cells are capable of reducing bacterial loads in a mouse model of Mtb [Chackerian et al., 2002]. Vα14 NKT cells have also been shown to confer protection [Gonzalez-Aseguinolaza et al., 2000].

The intravascular patrolling behaviour of NKT cells in the hepatic microenvironment is well documented [Geissmann et al., 2005; Lee et al., 2010]. NKT cells crawl through the liver sinusoids seemingly at random, and are capable of reversing direction spontaneously. NKT cells patrol at a speed of approximately 10-20μm/min, and their movement can be arrested subsequent to T cell receptor activation [Geissmann et al., 2005]. NKT cells can also demonstrate dramatically

altered behaviour during host infection, such as having their movement arrested subsequent to interaction with KCs after KC ingestion of the pathogen *Borrelia burgdorferi* [Lee et al., 2010].

The literature regarding NKT cells during EVL is conflicted, with some suggesting that they may play an important role in the later stages of EVL [Stanley et al., 2008]. Conversely, it has been shown that in *Leishmania major*, $CD4^{+}V\alpha$ iNKT cells play an important role in the early stages of infection [Ishikawa et al., 2000]. With such dichotomy in current understanding of NKT cell function during EVL and the leishmaniases, they are an intriguing focus for understanding EVL.

3.4 THE ROLE OF CYTOKINES AND SIGNALLING

Cytokines are molecules that facilitate or provide cell signalling functions. In order to ascertain the importance of particular cytokines, researchers often use knock-out studies, whereby mice or animals can be made deficient in a particular cytokine (denoted by $^{-/-}$).

The presence of both tumour necrosis factor (TNF) and lymphotoxin (LT)α are key for managing hepatic parasite burden, which was shown when $TNF^{-/-}$ and $LT\alpha^{-/-}$ C57BL/6 mice failed to adequately develop mature granulomas to manage infection [Engwerda et al., 2004]. Others have demonstrated that 129/Sv mice fail to present any inflammatory response at all to *L. donovani* in $TNF^{-/-}$ mice [Murray et al., 2000]. In either case, both TNF and LTα are essential for the control of *L. donovani*.

Knock-out studies involving the IL-13 cytokine revealed it as necessary for the maturation of hepatic granulomas and control of parasite burden, with $IL\text{-}13^{-/-}$ mice showing inhibited type 1 protective immune responses associated with cell-mediated immunity, and an increase in a type 2 response characterised by increased IL-4 and IL-10 levels [McFarlane et al., 2011]. Type 2 responses are generally not associated with favourable conditions for the removal of intracellular parasites, hence a type 1 response would be favoured in the context of VL. However, Stäger et al. [2003] also demonstrated a role for IL-4 and IL-4Rα signalling in optimal granuloma formation and management of *L. donovani* parasite burdens, both are generally associated with type 2 immune responses.

CXC chemokines regulate the infiltration of NKT cells into sites of infection and inflammation [Sato et al., 2005; Wehr et al., 2013]. Kupffer cells can produce these chemokines, including CCL1, CCL2 and CXCL10. During *L. donovani* infection, expression of CXCL10 is sustained when signal regulatory protein alpha (SIRPα) binds to CD47 on the surface of iNKT cells. This regulates increased IFN-γ cytokine production by hepatic iNKT cells [Beattie et al., 2010b], functioning as an amplification pathway to further regulate CXCL10 production [Svensson et al., 2005]. These responses are transient however, only occuring subsequent to KC-NKT cell interactions. The dynamics underpinning initial recruitment of NKT cells by infected KCs remain poorly understood.

These studies illustrate the complex interplay between cytokines and immune responses, and the implications of cell interactions and both cytokine and chemokine production for optimal host immunity to *L. donovani*.

3.5 IMPORTANCE OF A BALANCED IMMUNE RESPONSE

Without a balanced inflammatory response [Stanley and Engwerda, 2007], hepatic tissue may become damaged; as evidenced with delayed and exacerbated immune responses in the liver [Murray et al., 2000]. B cells, a component of the adaptive immune system, are involved in dampening the inflammatory response in VL, by controlling the influx of neutrophils and thus preventing adverse liver pathology [Moore et al., 2012]. B cell function is just one of many possible mechanisms of curbing overt liver pathology, alongside the previously described potential role of regulatory T cells. Natural killer (NK) cells also function in a immunoregulatory manner in EVL [Maroof et al., 2008], though whether the absence of NK cells modifies liver pathology is still an open question.

3.6 SUMMARY

In this chapter I have provided an overview of the liver microenvironment, and discussed current understanding relating to the function of liver resident macrophages called Kupffer cells in initiating an inflammatory response subsequent to pathogenic infection. The formation of liver granulomas, in response to infection, was discussed, along with mechanisms and properties that may be responsible for heterogeneity in the granulomatous response, such as: KC-parasite interactions, KC spatial distribution, and various cellular responses with known roles in the formation of VL-induced granulomas. The next chapter (see chapter 4), will elaborate on the core system entities and mechanisms that I hypothesise play the most important role in manifesting a heterogenous granulomatous response, and I will conceptualise a model of early granuloma initiation.

4 A DOMAIN MODEL OF GRANULOMA INITIATION IN VL

Having defined in chapter 3 the biological context that forms the domain of interest, it is then necessary to formulate a domain model to adequately scope that domain. As previously described in section 2.2.1, formulating a domain model is a key stage of the CoSMoS Process. When modelling, it would be intractable to include all of the components and mechanics outlined in the relevant literature. Highly granular computational models quite often heavily rely on assumptions, which are invariably made when constructing any model, only those assumptions are compounded with the added complexity of including all known and relevant biological components across the various scales. All cells and cytokines with identified roles in VL are not necessarily required to investigate the core dynamics of early cellular migration and recruitment into newly forming granulomas. The purpose of this chapter is to outline the scope of the modelling endeavour, detailing the cellular components and mechanics that I hypothesise are primarily responsible for the early stage initiation of granulomas, and to provide justification for the various levels of abstraction chosen.

N.B. Several assumptions are detailed in this chapter, though the reasoning behind these and the sources for evidence relating to those assumptions will be discussed in chapter 5. I believe it is more intuitive to show the mapping between domain assumption and platform implementation. Also, assumptions are included when constructing validity arguments, outlined in section 5.6.7, providing a means to both document and argue the validity of the simulation.

4.1 DEFINING THE SCOPE OF THE MODEL

The first method employed, which takes a systems level perspective to determine exactly what to model, was to condense the domain knowledge into an *expected behaviours diagram* as described by Read et al. [2009a]. The expected behaviours diagram (figure 8) allows the deconstruction of the problem into several parts: *observable phenomena*, *expected behaviours*, *biological components*, and *interactions/relationships*. Observable phenomena are the measurable system level behaviours that manifest themselves. Expected behaviours are the hypothesised events leading to the observable phenomena. Finally, a range of biological components may play a role in the system, each having their own relationships and interaction with each other.

The key *observable phenomena* in early granuloma initiation are as follows:

1. Infection with *L. donovani* parasites leads to parasitised KCs, generally within minutes of infection in EVL

2. Aggregations of immune cells are visible around a proportion of infected Kupffer cells as early as day 4 post infection

3. Both immature and mature granulomas are observable between 4 days and 8 weeks, with varying degrees of structural composition and parasite killing functional ability

Having defined the observable phenomena it is necessary to hypothesise and define the expected behaviours that may be responsible for those observable phenomena:

1. The parasitisation of KCs occurs because they are the predominant liver resident macrophages responsible for filtering the contents of the blood in the liver, and their effective phagocytic activity is well reported [Naito et al., 2004; Beattie et al., 2010a], having been shown to internalise pathogens within hours of infection [Lee et al., 2010].

2. Early aggregations of promoter cells including NKT cells, around infected KC foci, is observable within days of infection [Moore et al., 2013], and these NKT cell aggregations amplify the activation of KCs and subsequent mediators of mononuclear cell recruitment to sites of infection [Svensson et al., 2005; Beattie et al., 2010b].

3. Granuloma formation is an expected behaviour after the initial amplification stage, when other mononuclear cells provide an adaptive immune response due to subsequent interactions. Imaging has revealed heterogeneity in the granulomatous response, with a spectrum of small, medium, and large granulomas, with both parasite free and heavily infected foci observable at a variety of time-points (see section 3.2.3) [Murray, 2001; Moore et al., 2013; Beattie et al., 2010a]).

The varying sizes of cellular aggregations, and parasite burdens of KCs, is direct evidence that there is heterogeneity in granuloma structure and function, though the precise mechanisms that lead to fully functional granulomas are as yet unknown [Murray, 2001].

My hypothesis is that the amplification pathway previously described as a result of KC-NKT interactions, as well as both KC functional differences and spatial location, are all influencing factors in the determining the extent of granuloma formation and function - downstream of the amplification stage.

It is difficult to say with absolute certainty whether an inflammatory focus that appears parasite free was previously infected, or if uninfected sites are capable of forming cellular aggregates purely because of the inflammatory environment. The previously highlighted roles for KC-NKT cell interactions based on infection-induced upregulation of SIRPα by KCs, provide evidence that aggregations are associated solely with infection, though this is not a certainty.

The above expected behaviours make it clear that a model incorporating entities at the cellular scale is required to examine my hypotheses. I previously

highlighted the benefits that agent-based models hold for the study of inflammation (section 2.1.3.2), and those benefits are particularly relevant for the study of granulomatous inflammation. Whilst granulomas are surely the product of complex interplay between many molecular and cellular components across the various biological scales (figure 15a), this research focuses on the cellular interactions and subsequent aggregations which precede the second of the observable phenomena; the formation of aggregates a within days of infection.

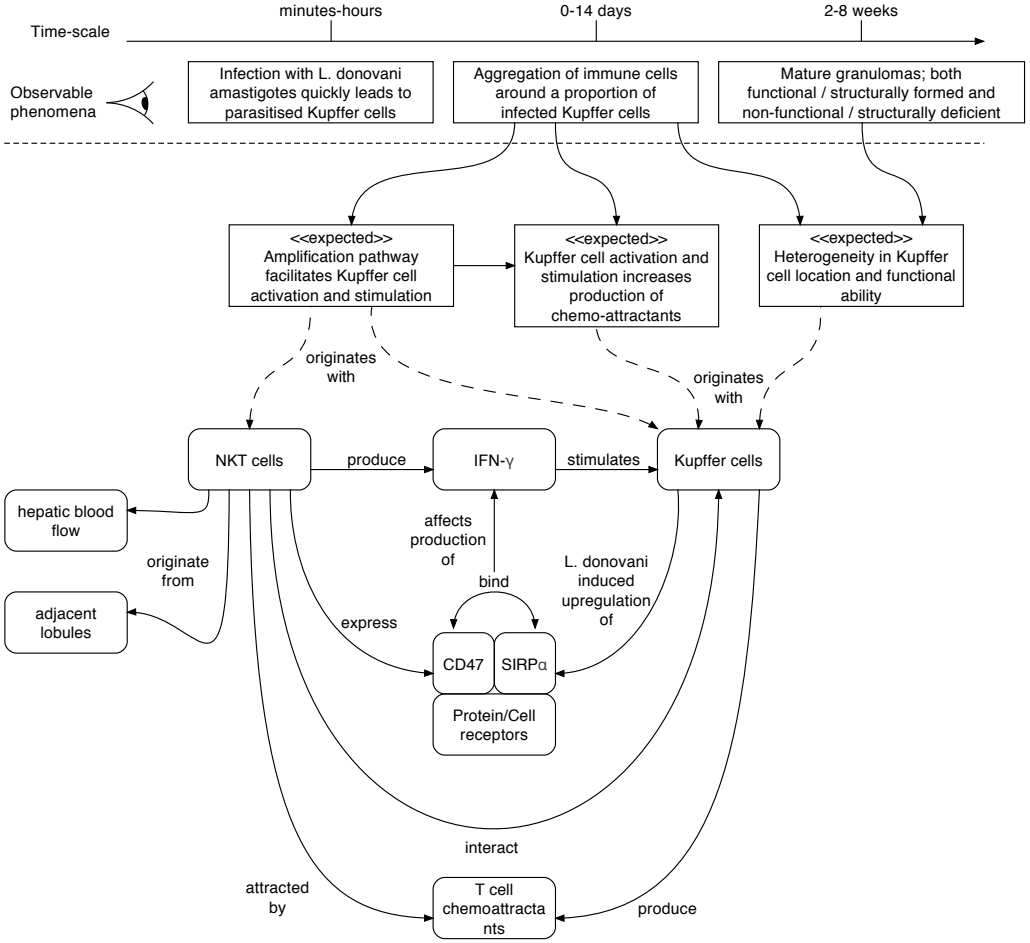


Figure 8: Expected Behaviours Diagram. This diagram outlines the observable phenomena from the domain of interest, and hypothesising as to how those observable phenomena manifest themselves based on interactions between known biological components. The diagram is influenced loosely by UML class diagram notation, though associations illustrated with arrows are not structural relationships, and in some cases are functions calls.

4.1.1 Infection Dynamics and Time-Scale

Whilst the dynamics of parasite survival are undoubtedly interesting and perhaps relevant for this study, there is a knowledge gap and lack of data pertaining to parasites and their influence on KCs. Also, as parasites are rapidly phagocytosed by KCs in experimental models [Beattie et al., 2013], I assume the dynamics of

initial infection do not heavily influence downstream events (domain expert assumption). Therefore, I ignore the parasitisation of KCs and instead abstract the influence of parasites into functional differences between KCs, namely uninfected and infected cells.

Fully formed granulomas are comprised of a wide variety of mononuclear cells, many of those are only recruited during the granuloma maturation stages of infection [Moore et al., 2013]. For this research, the assumption is that the observed heterogeneity and system dynamics can be explained by examining early initiation and amplification of KCs, prior to the recruitment of other mononuclear cells. For the domain model I chose to capture events within 48 hours of infection as this represents the amplification stage of KC infection.

4.2 DEFINING MODEL ENTITIES AND INTERACTIONS

As detailed previously in section 2.2.3, UML has been used effectively to document the interactions, relationships and states of biological components such as cells in a variety of contexts. It is important to reiterate that UML diagrams are not designed for specifying models of biological systems, but for specifying software systems. Therefore, the purpose of the diagrammatic notation in this case, particularly when formulating the domain model, is to facilitate communication between modeller and domain expert. The large suite of UML diagrams can be used not only in the documentation or specification of models to describe the domain, but later in using those models to engineer software for simulation.

4.2.1 Model Entities

I hypothesize that two core system dynamics are capable of manifesting cellular aggregations at the population level, namely dynamics of cellular attraction and cellular retention. Figure 9 encapsulates the mechanisms by which KCs might attract or encounter NKT cells, namely chemokine mediated attraction, or a random encounter as a result of a patrolling NKT cell. NKT cells are then retained through various cytokine, chemokine, and receptor mediated mechanisms. To document exactly which entities, mediators, and relationships would form the domain model, a UML Class Diagram was then constructed (figure 10), detailing three cell types: *infected KCs*, *uninfected KCs*, and *NKT cells*, whilst highlighting the relationships between those cells that are believed responsible for realising those cellular attraction and retention dynamics.

Uninfected and infected KCs have the same CD1d receptor which interacts with the T cell receptor (TCR) expressed by NKT cells. CD1d-TCR interactions will only occur when the CD1d is loaded with 'antigen'. This could be a parasite antigen in which case there might be a difference in the functionality of the interaction between infected and uninfected cells. CD1d antigens can also be self-antigens induced by stress, so this could occur on all KCs in the environment in which case there would be no discriminating functionality. For these reasons,

I assume that these interactions are implicit, with no discriminating function in this regard between infected and uninfected cells.

Both infected and uninfected KCs are capable of upregulating SIRP1α subsequent to infection; however, the mechanism by which uninfected cells induce upregulation has been shown to be *in trans* G-protein signalling [Beattie et al., 2010b]. IFN-γ is also then capable of further engaging infected KCs to increase expression of SIRP1α, and to produce more CXC-chemokines. Whilst both KC populations are capable of producing CXC-chemokines which facilitate cellular migration, due to the amplification mechanism, it is only the infected KCs that can utilise this mechanism to further retain those NKT cells.

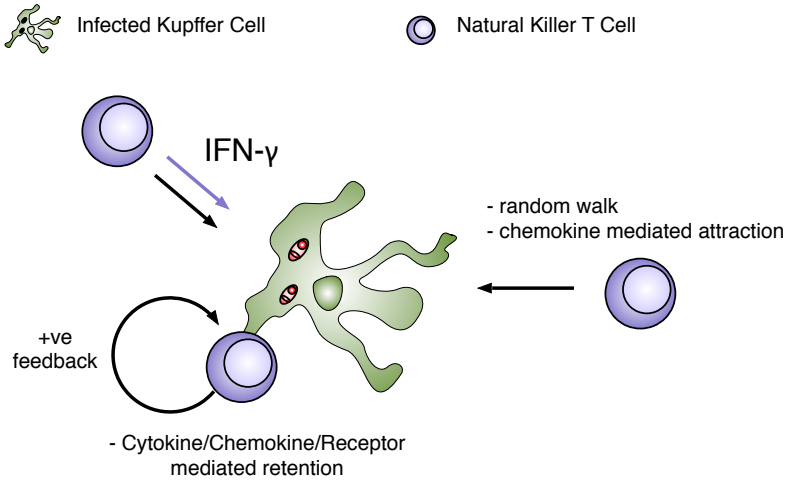


Figure 9: Domain Model Dynamics Summary. A cartoon to summarise the core interactions that the modelling endeavour needs to capture. KCs (green) attract, retain, and are stimulated by NKT cells (purple).

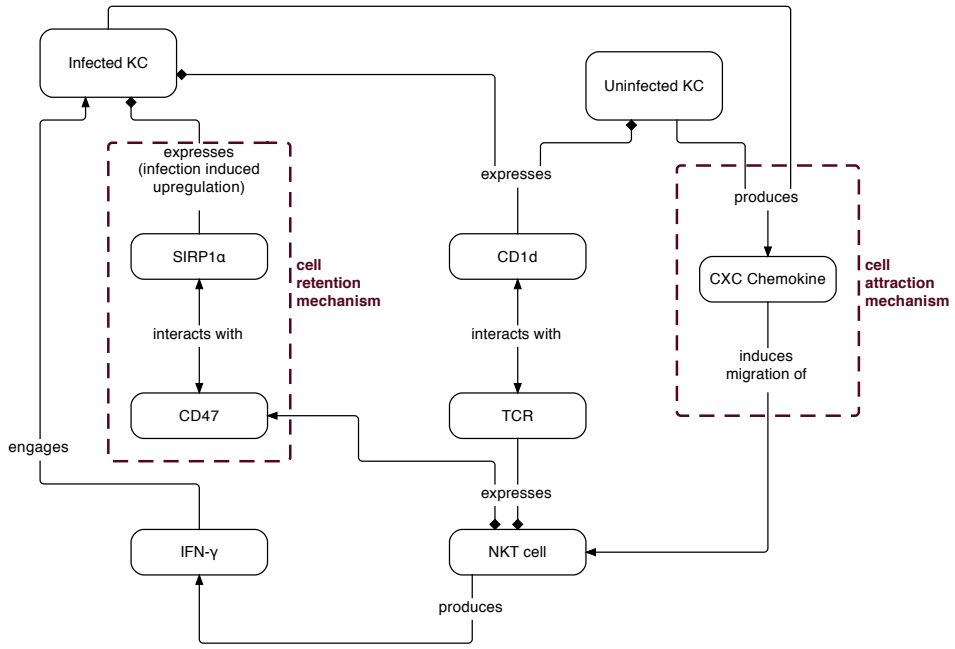


Figure 10: Domain Model Entities. These are the components that form the domain model. Not all are necessarily modelled explicitly when it comes to formulating and implementing a platform model. The diagram is influenced loosely by UML class diagram notation, though associations illustrated with arrows are not structural relationships, and in some cases are functions calls. Diamond arrows denote composition. Red dashed lines are used to simply to highlight how certain components and interactions implement a specific mechanism.

4.2.1.1 *Kupffer Cell States*

It is important to note that in the biological system, cells are not clearly defined as being in particular states. Whilst biologists can identify and differentiate cells into sub-populations, each capable of performing different actions, the tools and techniques used to identify them do not binarily dictate what their function will be. There is known heterogeneity in cell type and function, and it is us who impose cell types and states in an effort to better understand the heterogeneity. As a modeller, it is necessary to abstract cell functions into states in this context, since when it comes to forming a platform model and implementation, different cell functions must be logically independent.

State Diagrams were used to document the states of cells, and what conditions, environmental or internal, are required to be met in order for the cell to transition between states. Boxes denote states, and dashed lines denote when a cell can be in multiple states concurrently. Arrows determine transitions between states, and 'gates' provide conditions for state-transition. Figure 11 depicts the infected KC states and transitions, with the two core states described as *aware* and *engaged*. The domain model represents infected KCs subsequent to infection as this occurs rapidly (see section 3.2).

INFECTED KC STATES The state diagram for infected KCs is depicted in figure 11. I assume based on domain expert opinion that KCs sense and remain in an *aware* state until they encounter NKT cells, which produce IFN-γ, thus becoming *stimulated*. A threshold of stimulation is required for KCs to reach an *engaged* (activated) state. Subject to sustained interaction with NKT cells, KCs then begin to upregulate the production of chemo-attractants and retentive mediators. The evidence for the assumptions relating to infected KC states is listed in table 1.

UNINFECTED KC STATES The State Diagram for uninfected KCs is depicted in figure 12. I assume based on domain expert opinion that these cells are not receiving sustained CD1d-TCR signalling that is a consequence of the upregulation of SIRP1α upregulation; the lack of SIRPα enhanced response equates to no retention of NKT cells, though I assume they do produce chemo-attractant. The production of chemo-attractant by uninfected KCs was originally a design assumption (later validated in Beattie et al. [2013]) so that the model can determine the dynamics of competition between all KCs to recruit NKT cells, whether it be infected KCs competing to become activated, or uninfected KCs performing some other role, perhaps to limit inflammation. The evidence for the assumptions relating to infected KC states is listed in table 2.

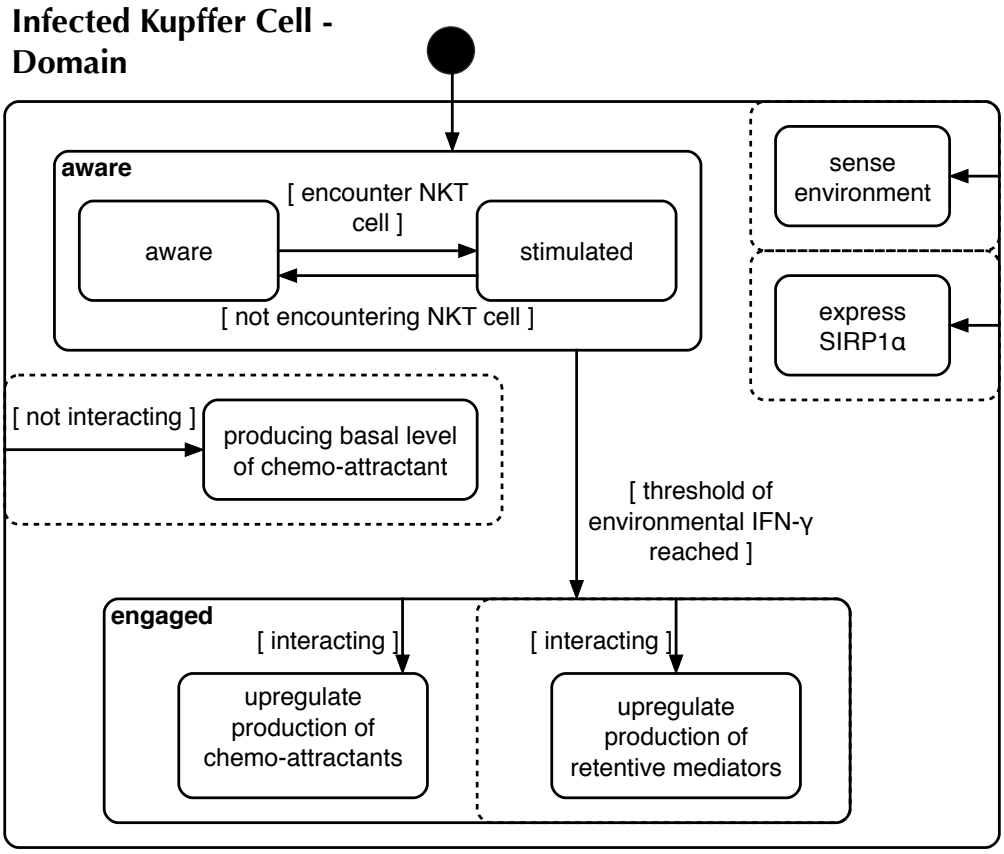


Figure 11: State Diagram (Domain) - Infected KC. UML state diagram notation is extended with the use of dashed lines to denote states the cell can be in concurrently. Conditions, gates, that must be satisfied before a state transition are denoted using squared brackets [].

Infected Kupffer Cell (Domain)		
State	***Associated transition***	***Evidence***
Stimulated	[encounter NKT cell]	Svensson et al. 2005
		Beattie et al. 2010b
Sense environment		Domain expert assumption
Express SIRPα		Beattie et al. 2010b
Producing basal level of chemo-attractant	[not interacting]	Domain expert assumption
Up-regulate production of chemo-attractants	[interacting]	Svensson et al. 2005
		Domain expert assumption
Up-regulate production of retentive mediators	[interacting]	Beattie et al. 2010b
		Domain expert assumption
Engaged	[threshold of environmental IFN-γ reached]	Domain expert assumption

Table 1: Evidence for states and transitions in figure 11.

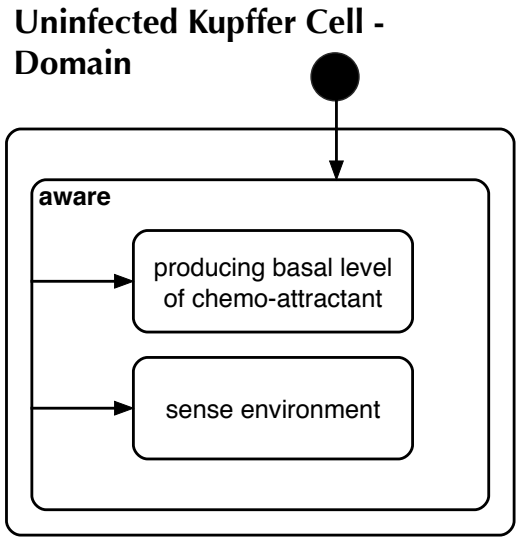


Figure 12: State Diagram (Domain) - Uninfected KC.

Uninfected Kupffer Cell (Domain)		
State	***Associated transition***	***Evidence***
Producing basal level of chemo-attractant		Domain expert assumption
Sense environment		Domain expert assumption

Table 2: Evidence for states and transitions in figure 12.

4.2.1.2 *NKT Cell States*

NKT cell active ability can include producing or not producing stimulatory cytokine, such as IFN-γ, and these can be classed as states. Since NKT cells perform a random walk in the sinusoids [Geissmann et al., 2005], yet are also capable of chemo-attractant induced taxis, these are also distinct states of movement. To capture behaviour such as NKT cells stimulating and being immobile, or not stimulating and performing a random walk, states are grouped into two concurrent types. The state-transition diagram depicted in figure 13 captures those states for NKT cells, which are grouped into *action* and *movement* states.

NKT cells begin by entering the lobule from the periphery, where they perform a random walk until they sense chemokine signals, which induces a chemotactic state. Once the cells encounter an infected KC, with its upregulating of SIRP1α and expression of $CD1^+$ antigen, it can then begin to aggregate around that KC until it probabilistically leaves, dictated by KC retentive mediators. Concurrently, the cell is initially in an inactive action state until it encounters an infected KC, at which time it begins to produce IFN-γ until it leaves the infected focus. NKT cells are then considered to be in a state of anergy after stimulation, a state reportedly induced by TCR-mediated signalling [Iyoda et al., 2010].

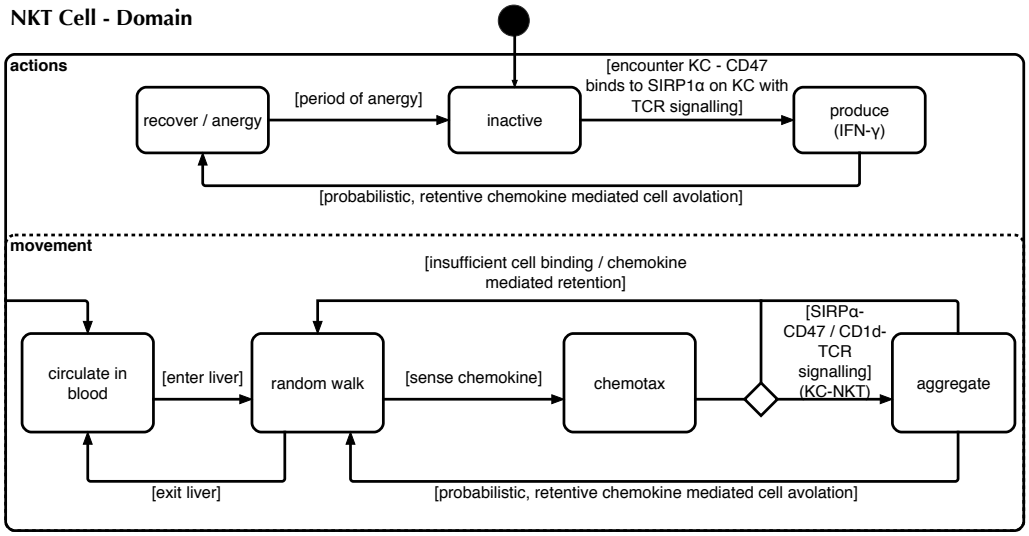


Figure 13: State Diagram (Domain) - NKT cell. The diamond denotes a choice dependent on a condition. UML state diagram notation is extended with the use of dashed lines to denote states the cell can be in concurrently. Conditions, gates, that must be satisfied before a state transition are denoted using squared brackets [].

NKT Cell (Domain)		
State	***Associated transition***	***Evidence***
Inactive		Domain expert assumption
Produce IFN-γ	[encounter KC - CD47 binds to SIRPα on KC]	Svensson et al. 2005
Recover / anergy	[probabilistic, retentive chemokine mediated cell avolation]	Iyoda et al. 2010
Circulate in blood	[exit liver]	Domain expert assumption
Random walk	[enter liver]	Geissmann et al. 2005
	[insufficient cell binding / chemokine mediated retention]	Domain expert assumption
Chemotax	[sense chemokine]	Domain expert assumption
Aggregate	[SIRPα-CD47 / CD1d-TCR signalling]	Beattie et al. 2010a,b

Table 3: Evidence for states and transitions in figure 13.

4.2.2 Model Interactions

The proposed model interactions and events were captured using UML Activity Diagrams. Activity Diagrams provide a means to document system events and the flow between them. Similar to the notation previously described for State Diagrams, arrows represent ordered flow, with black bars depicting occurrences where concurrent activities can converge or split, with conditional flow denoted by gates. Figure 14 details events that are capable of manifesting how NKT cells randomly patrol their environment until they either encounter a KC, or until they are influenced by some chemotactic gradient. The diagram incorporates an assumption that NKT cells will only aggregate around infected KCs, which are capable of interacting via the CD47-SIRPα pathway [Svensson et al., 2005]. Once an interaction occurs, infected KCs must then reach a threshold of activation before they can begin upregulating both the production of retentive mediators and chemo-attractants. Retentive mediators then influence the sustained retention of NKT cells at the infected focus. Uninfected KCs only function by producing a basal level of chemo-attractant in their environment, which can influence NKT cells to chemotax, but those NKT cells will subsequently not interact.

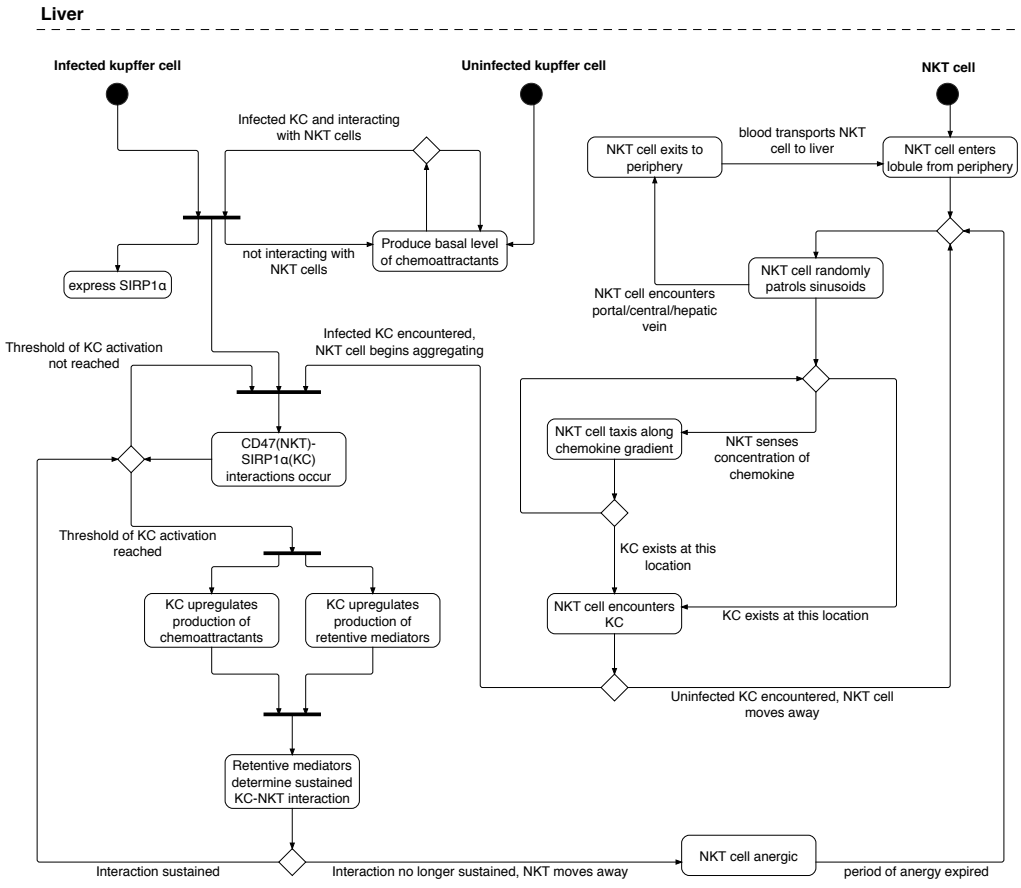


Figure 14: Activity Diagram (Domain): Infected KCs, Uninfected KCs and NKT cells interact.

4.3 SUMMARY

I began this chapter by discussing the need to adequately scope the knowledge of the biological domain into a domain model. The domain model outlined, represents a system-level perspective of various components and mechanics that I hypothesized could account for three observable phenomena relating to early granuloma initiation (section 4.1). Expected behaviours were constructed as assumptions regarding how and why observable phenomena manifest themselves, and an expected behaviours diagram constructed to illustrate the mapping between phenomena and behaviour. I posited that Kupffer cells, both uninfected and infected, along with NKT cells, were a sufficient set of cellular entities to model early granuloma initiation, based on a previously experimentally validated NKT cell mediated axis of KC activation. UML was used as a graphical modelling tool to convey how I perceived the cellular entities to function and interact, based on attributing them with states and state-transition capabilities.

In the following chapter, I will present a platform model of early granuloma initiation (chapter 5). The platform model is in essence a mapping of the domain model into an implementation specific format. Converting the various entities, mechanisms, and importantly assumptions of a domain model into computer code is not a trivial task, and the platform model clearly sets out how I tackled this problem.

5 A PLATFORM MODEL OF GRANULOMA INITIATION

Whilst the previous chapter outlined the scope of the modelling endeavour, detailing the components and interactions believed to be responsible for manifesting a range of observable phenomena in the biological domain of interest, the domain model is not a specification for a software system, nor does it describe the necessary logic required to encode the domain knowledge into an executable form. The platform model is the implementation specific translation of the domain model, forming a specification of how to encode the model and assumptions into a software system. Whilst the emphasis in the domain model was around emergent phenomena, for example the formation of cellular aggregations, it is important to maintain a clear separation with the platform model to ensure that the concept of a cellular aggregation is not hard-coded into the simulation, it should emerge as a result of the components modelled and their interactions.

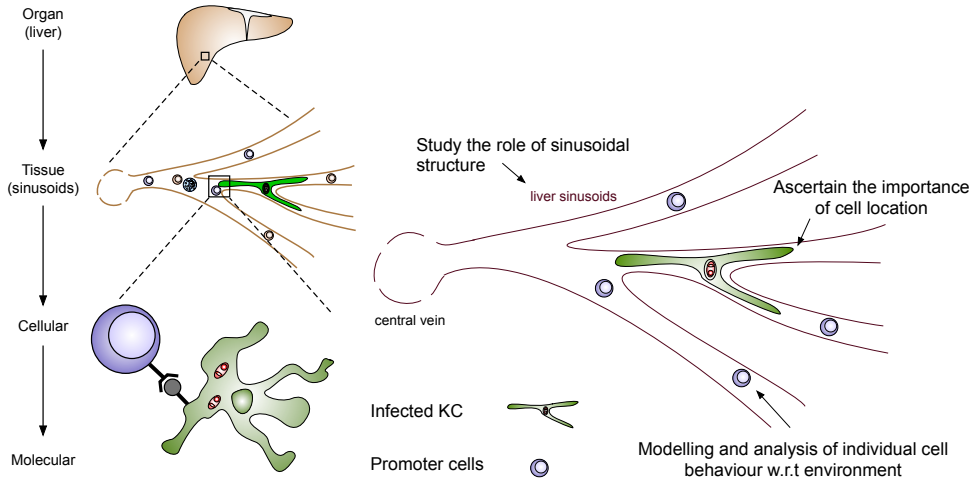
Sections 5.1 to 5.3, detail how the domain model theory and assumptions are conceptualised into a platform model, containing algorithms and logic required to turn the model into a computer simulation. Sections 5.4 and 5.5 conceptualise the simulator design. Section 5.6 details simulation calibration, and how various biological data are input into the simulation to parameterise the various cells and mechanisms outlined in the earlier sections of the platform model. Finally, in section 5.6.7 I detail my argumentation efforts, demonstrating that my simulation adequately represents the underlying biological system conceptualised in the domain model.

Declaration: the algorithm in section 5.2 was developed in collaboration with Dr. Paul Andrews at the University of York.

5.1 DECIDING ON A MODELLING PARADIGM

Figures 15(a),(b) illustrate how a tissue-level ABM incorporating interactions between individual cells, has an intuitive mapping for studying VL induced granulomas. Modelling at the tissue level makes it possible to determine the role of sinusoidal structure, assuming an accurate and statistically representative representation of the sinusoidal micro-environment is modelled. A tissue level ABM would also provide the opportunity to determine the importance of KC location, and the implications that this has for the role of the individual in forming granulomas. Lastly, given ABMs allow us to interrogate the status of individual agents, it is possible to analyse the behaviour of patrolling NKT cells with respect to their environment, allowing the dynamics of their recruitment to be dissected. Given that the ODE/PDE mathematical approach is more suited to population based models, and not suited to the study of individuals and the key require-

ments of a spatio-temporal model in this case, the ABM paradigm was chosen for the research detailed in this thesis. The next section will detail how spatial simulation environments were generated.



(a) Modelling Scales **(b)** ABM for Liver Study

Figure 15: Modelling Scale. Conceptualising the scale for a spatio-temporal model of granuloma initiation. Sub-figure **(a)** illustrates various biological scales. Conceptually, KC and NKT cell interactions occur at the tissue and cellular levels. Sub-figure **(b)** published in [Moore et al., 2013]

5.2 GENERATING IN SILICO LOBULE REPRESENTATIONS

Environment is an integral component of any agent-based model. For the purposes of this research, the objective was to generate realistic liver lobule sections that adopted a similar, computationally efficient connected graph approach as that used by Wambaugh and Shah [2010], whilst incorporating the statistical data reported by Höhme et al. [2010]. By generating multiple structures, it is then possible to analyse if there is variance in simulation results between different statistically representative structures, providing a means to determine the influence that sinusoidal structure has on the system.

In section 2.1.6 I argued that a 2D spatial grid approach to modelling the liver sinusoids was too simplistic for modelling certain tissues. This is certainly the case in respect to the intricacies of the liver microenvironment (figure 16**(a)**). Whilst a 2D grid approach is not appropriate, as it leads to implausible free movement of NKT cells (figure 16**(c)**), a 2D connected graph/network approach I argue is appropriate. Nodes and edges within a 2D network can be constructed to represent the channels and branches within the liver sinusoids, thus allowing restricted movement of patrolling cells as in the liver microenvironment (figure 16**(d)**). Constructed 2D networks could also be used as skeletons with which to construct 3D spatial representations of the same tissue. However, for the current model, investi-

gating the 3D spatial aspect of granulomas is not part of the underlying research hypotheses. Therefore, a connected graph approach was used to represent the sinusoids, allowing the creation of structures that are both computationally efficient and visibly similar abstractions of observed liver morphology.

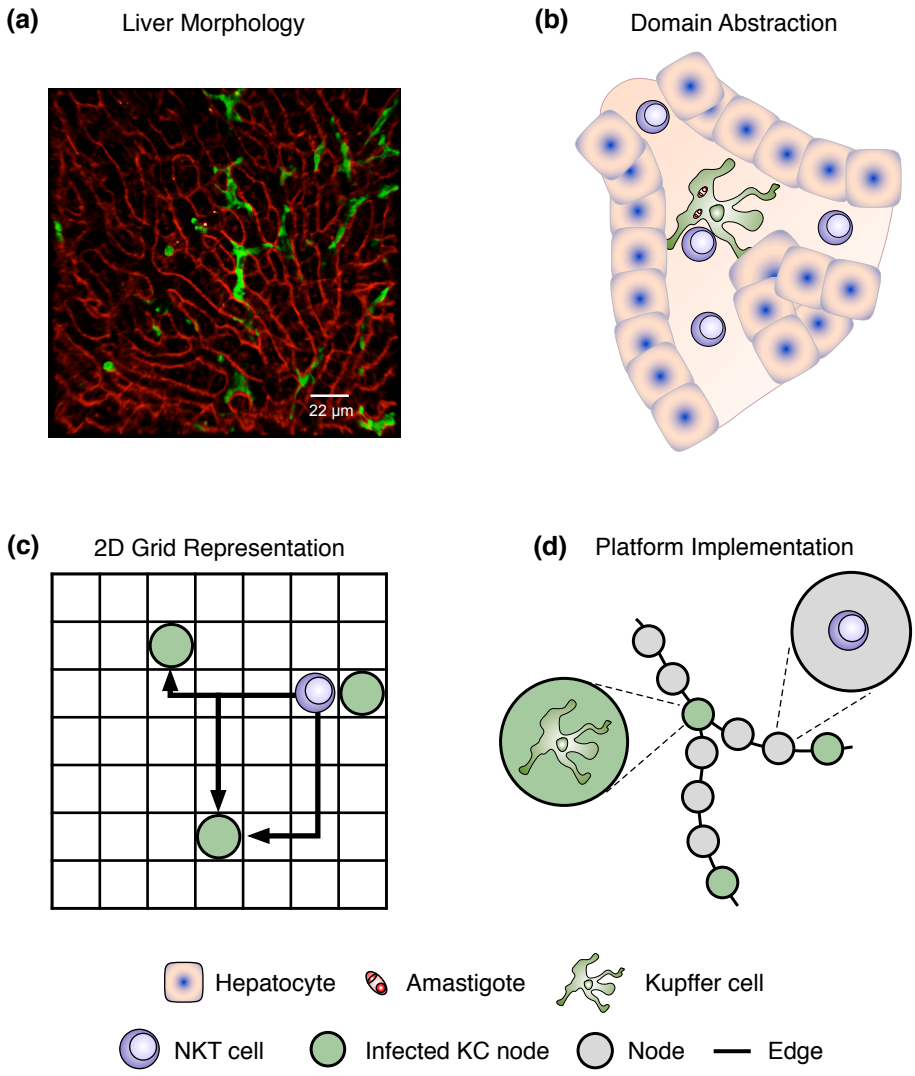


Figure 16: From liver morphology to platform implementation. **(a)** In situ analysis of liver sinusoids (red) and Kupffer cells (green) in (mT/mG x lysMcre)$_{F1}$ mice. **(b)** the domain abstraction of the sinusoids and two core cell types involved in early granuloma formation. **(c)** a 2D grid representation with biologically implausible free movement of NKT cells. **(d)** the engineered platform implementation using a 2D network that restricts cell movement as in the underlying liver microenvironment.

I approached Höhme et al. [2010] who kindly provided us with data representing 26 liver lobules amalgamated into a single statistically representative lobule structure comprised of nodes and edges. Whilst it would have been interesting to utilise this structure for my simulation, there were key properties of the structure, such as distorted sinusoids at the peripheries, that made it difficult to use their structure as presented. Rather than developing a strategy to remove those artifacts, I chose to instead proceed with the artificial generation of structures based on statistical data; this approach was also more appropriate, and needed for investigating variance between multiple sinusoidal structures in the system.

Figure 17 illustrates and describes the generation of the sinusoid structures. I assume that the sinusoid network exists in a 2D space (this is considered as a slice through a 3D lobule). I also assume that the lobule structure is roughly hexagonal with a single central vein in the centre and six portal triad areas placed at roughly regular intervals around the central vein. The flow of blood borne cells is assumed to be from portal triads to the central vein, so in the algorithmic description below the central vein is termed a drain node, and the portal triad regions are populated by entry nodes. Algorithm 3 (see Appendix 9.1.5) describes how the nodes and segments of the sinusoid network are generated, whilst algorithm 4 (see Appendix 9.1.5) describes how the overall sinusoid network (the lobule) is generated using algorithm 3. The statistical data published by Höhme et al. [2010]. that were augmented with algorithm 4 are as follows:

- Average length between central vein and portal triad = 284μm;
- Average length of a non-branched sinusoid = 43.1μm;
- Average angle between branching sinusoids = 32.5°.

Examples of the various structure networks generated can be seen later in figure 43. Having established a process to create simulation environments, it is necessary to define the implementation of cells that will populate those environments, this will be discussed in the next section.

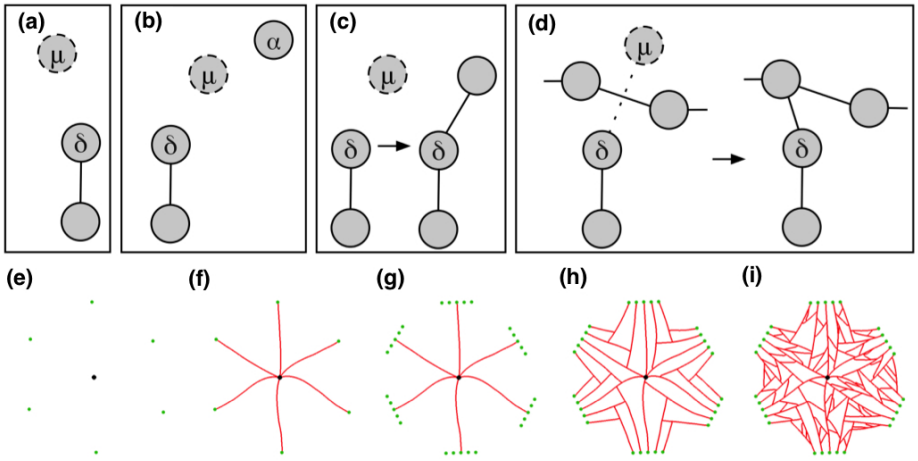


Figure 17: Structure generation algorithm. **(a)**, Candidate node, μ, generated based on position of current node, δ. **(b)**, Position of candidate node, μ, biased towards attracting node, α. **(c)**, Segment created between candidate node, μ, and current node, δ. **(d)**, New segment intersects with existing branch in sinusoid network so current node, δ, connected via a segment to the closest existing node. Sinusoid network construction algorithm stages: **(e)**, Drain node (black) placed in the centre of a 2D space with six surrounding entry nodes (green) forming an irregular hexagon layout. **(f)**, Sinusoids (red) grown from entry nodes to drain nodes. **(g)**, Additional entry nodes created around original entry nodes, conceptually forming a portal triad. **(h)**, Sinusoids grown from additional entry nodes. **(i)**, Additional sinusoid branches added between existing sinusoids.

5.3 IMPLEMENTING CELLS AND MECHANISMS

The domain model presented in chapter 4 outlined the scope of which cells and cell behaviours would be incorporated into the simulation. This was achieved by using UML to document both the overall activity of the system (Activity Diagram), and the states and transitions of individual cell types (State Diagrams).

The purpose of this section is to present the platform specific version of the earlier domain model, translating domain model assumptions and UML diagrams into modified versions incorporating logic and a consideration of the implementation platform chosen. The platform model is required to implement biologically plausible instantiations of the various cells and mechanisms I wish to model. Assumptions are documented in tables which list assumptions from both a domain and platform perspective, with justifications given for each. Domain assumptions may not translate directly across to the platform, this may be for example due to technological limitations, or lack of domain knowledge to justify a direct implementation.

5.3.1 Model Interactions - Platform

Previously in section 4.2.2 I utilised a UML activity diagram to outline the sequence of events that would lead to NKT cells encountering and interacting with KCs. The activity diagram detailing these events for the platform model is depicted in figure 18, incorporating conditional gates that link cell actions to the network environment. Aside from conditions that now link cells to the tree-node network environment, the primary difference between the original domain model (figure 14) is that cell interaction mechanisms are abstracted. Instead, interactions are assumed when NKT cells and infected KCs are co-located on a network node. Subsequently, the duration of interactions is modelled as a dynamic, probabilistic retention function (see section 5.3.4.3). The functions of both NKT cells and infected KCs whilst interacting are detailed in sections 5.3.2 and 5.3.3.

5.3.2 Kupffer Cells

Detailed technical information for KC implementation is detailed in Appendix 9.1.3.1. Infected KCs are initialised as infected cells. The state diagram depicted in figure 19 illustrates the platform translation of an infected KC, the figure also lists all the relevant assumptions made for both infected and uninfected KCs required for implementation. Infected KCs are infected from time zero of the simulation, and they are in an *aware* state whilst concurrently interrogating the level of stimulatory IFN-γ at their node location, and initially diffusing a basal level of chemo-attractant (figure 21(a)). Chemo-attractant is diffused a minimum network distance and at a minimum concentration (see section 5.3.4.2). Probabilistic interactions with NKT cells, denoted by δ(NKT), put the infected KC into a state of being *stimulated*, and if the level of IFN-γ at the node is above a specified threshold (ifnThreshold), the cell will become *engaged*. Subsequent

NKT cell interactions then allow the *engaged* infected KC to increment the functions that govern probabilistic retention and the production of chemo-attractant (figure 21(b)). *Assumption 1* details that infected KCs are considered infected at the initiation of the simulation, for reasons described in section 4.1.1. My domain experts wish to investigate the effects of uninfected KCs responding differently to infection, and these functional differences are realised by providing each population with a different capacity to attract and retain cells. Therefore, *Assumptions 2 and 3* specifiy that infected cells can both attract and retain cells, whereas uninfected KCs can only recruit - as a result of lacking sustained CD1d-TCR (see chapter 4). Since I model the environment as a network, I can abstract cell-cell interactions, including the SIRPα-CD47 mechanism, and instead just assume they occur when an NKT cell and an infected KC are co-located on the same node. The cognate SIRPα-CD47 interactions can then be modelled as a probabilistic retention function (see section 5.3.4.3).

Uninfected KCs serve only to introduce competition for infected KCs (figure 20), and their *aware* state functions only to produce a basal level of attraction as described for infected KCs.

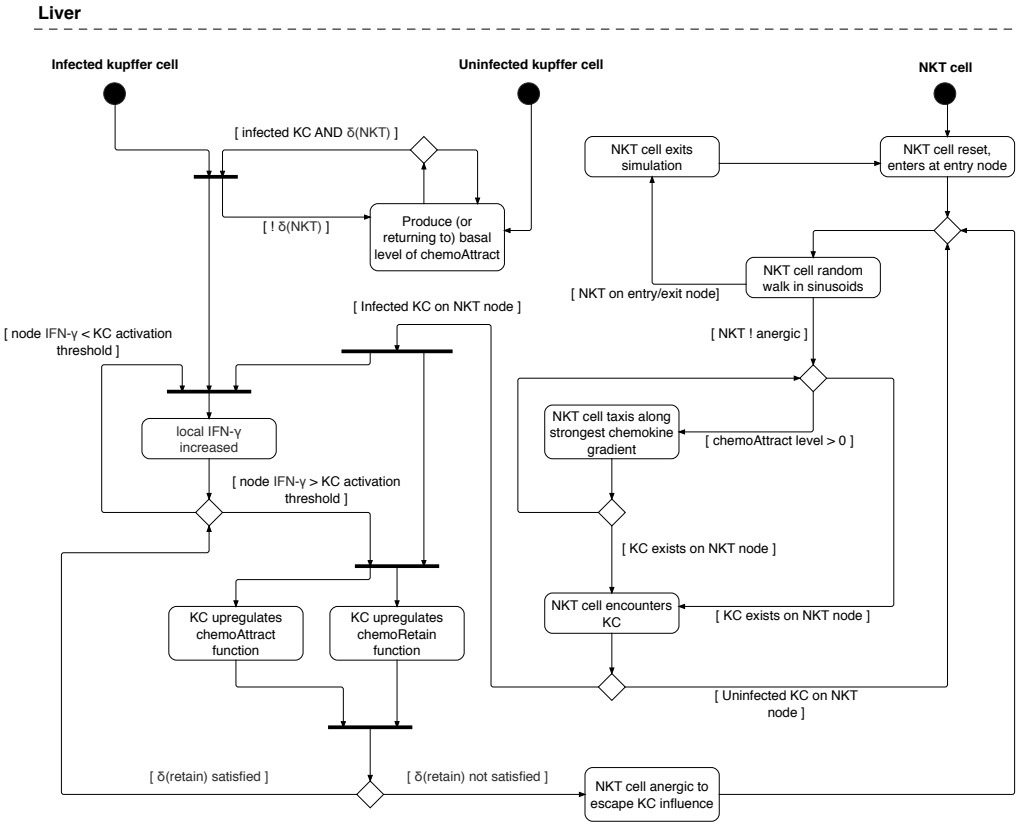
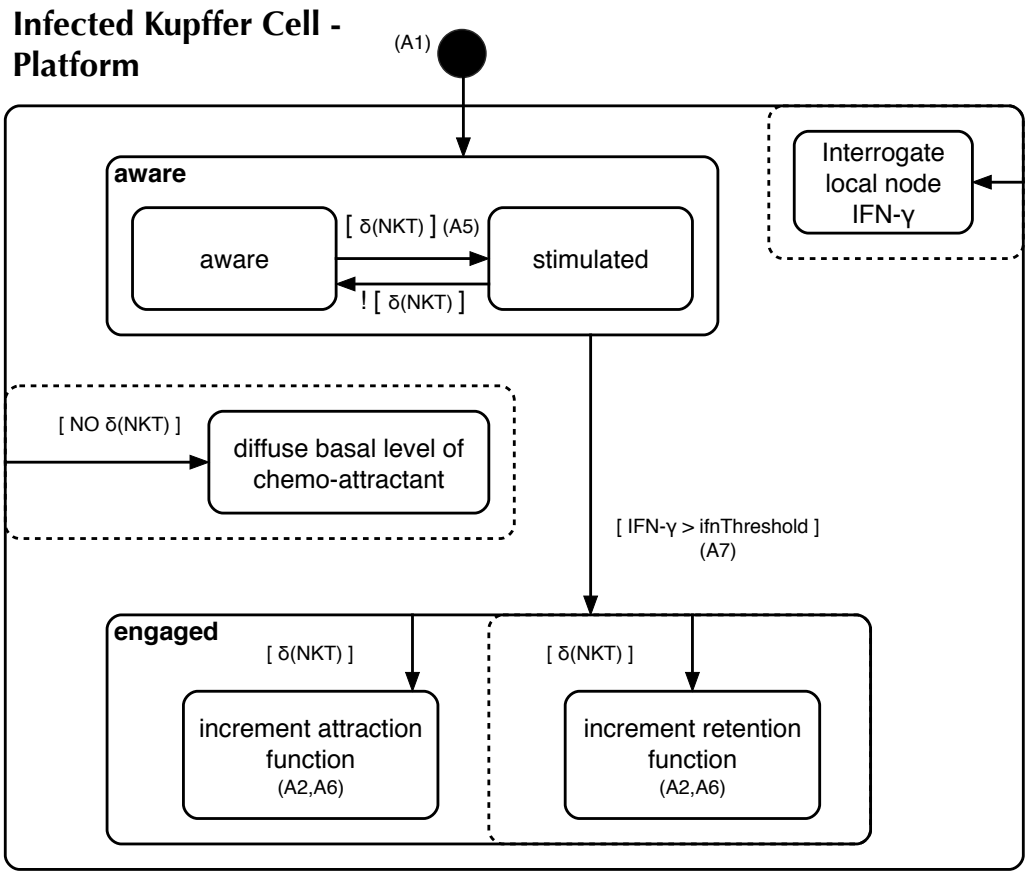


Figure 18: Activity Diagram (Platform). This diagram details the events that lead to interactions between infected KCs, uninfected KCs, NKT cells and the network environment.



	Domain	Platform	Justification
A1		Infected KCs are infected at initiation of the simulation, t(0).	KCs rapidly phagocytose parasites. We are not investigating conditions pre-infection. Evidence in section 3.2.3.
A2		Infected KCs can recruit and retain cells.	Domain expert assumption. Implementation discussed in section 5.3.2.
A3		Uninfected KCs can only recruit cells.	
A4	KCs are immobile	KCs have no movement rules	(Lee et al., 2010). Implementation discussed in section 5.3.2.
A5	KC retention of NKT cells is probabilistic, mediated by SIRP-α - CD47 interactions and local IFN-γ concentration.	Cognate interactions occur when two cells are co-located on the same node and are not modeled explicitly. Retention is modeled probabilistically using a function modified by local IFN-γ level	(Beattie et al., 2010; Svensson et al., 2005). Evidence highlighted in section 4.2.1.
A6	NKT produced IFN-γ affects KC attractive chemokine production.	Retentive chemokine level increases in-line with local IFN-γ level and increases the diffusion distance of attractive chemokine linearly. Retentive chemokine diffusion distance can decay.	Simplification, no experimental data available. Partial evidence in section 4.2.1.
A7	There is a threshold level of stimulation required to activate a KC.	KCs will not transition to an engaged state until a threshold level of IFN-γ is reached.	To facilitate implementation, no experimental data available. Implementation in section 5.3.2.

Figure 19: State Diagram (Platform) - Infected KC. UML state diagram notation is extended with the use of dashed lines to denote states the cell can be in concurrently. Conditions, gates, that must be satisfied before a state transition are denoted using squared brackets []. The δ symbol denotes a probabilistic event; in this instance an encounter with NKT cells.

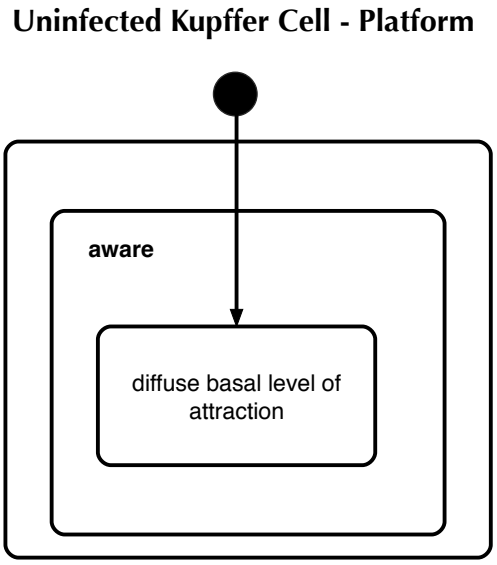


Figure 20: State Diagram (Platform) - Uninfected KC.

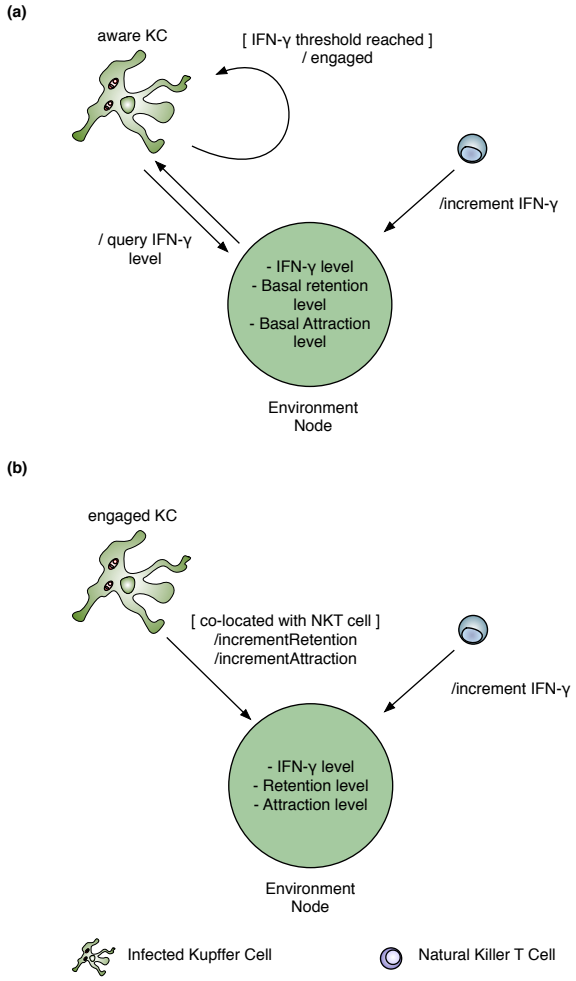


Figure 21: Cell Interaction With Environment Nodes. **(a)**, Infected KCs in the *aware* state will interrogate their nodes IFN-γ level until they reach a threshold, then they will become *engaged*. **(b)**, when in the *engaged* state and co-located with NKT cells, infected KCs will increase the levels of attraction and retention.

5.3.2.1 *Kupffer Cell Spatial Distribution*

To ascertain whether or not the reported abundance of KCs in the periportal zone could have a significant effect on early recruitment of NKT cells, two distribution algorithms were created for the simulation.

Prior to a simulation one of two algorithms is used to distribute Kupffer cells (each a minimum distance apart to avoid overlap) throughout the sinusoid network. The first, algorithm 1 (see Appendix 9.1.5), was designed to distribute KCs according to Bouwens et al. [1986]. Figure 22(b) illustrates uninfected KCs placed in percentages of 43%:28%:29%, and infected cells in percentages of 65%:25%:10% in the PP:MZ:CV zones respectively. The second algorithm distributes KCs randomly throughout the sinusoids, figure 22(a) illustrates this method, with both infected and uninfected KCs randomly placed throughout all zones of the lobule area, producing a more even distribution.

Both KC distribution algorithms ensure that KCs are placed a minimum distance apart, and the mechanics of this are outlined in figure 23 and algorithm 2 (see Appendix 9.1.5), this is to avoid the potential for large swathes of sinusoid not being allocated any KCs, which can occur due to random node sampling, and which would not be particularly biologically plausible.

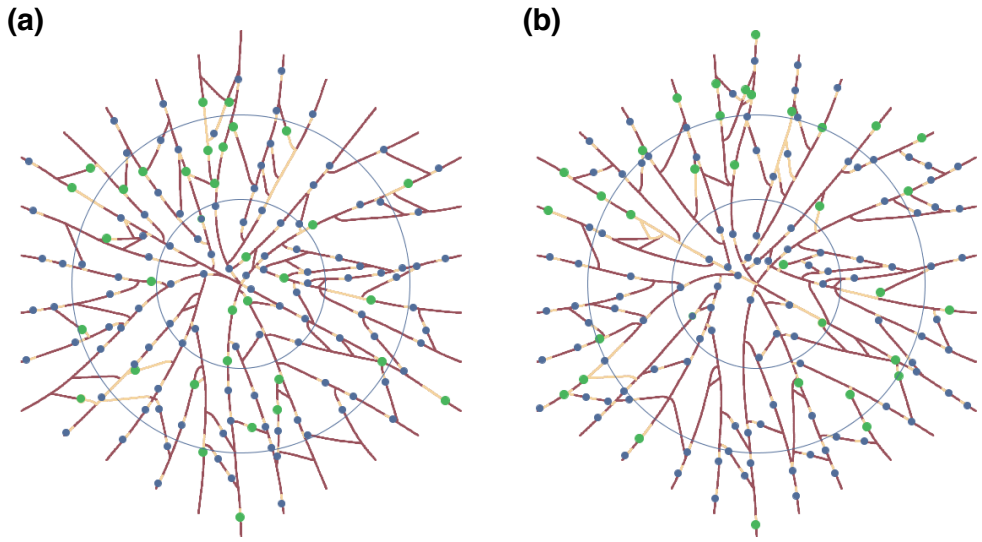


Figure 22: A comparison between KC distribution algorithms. Infected KCs (green), uninfected KCs (blue), sinusoids(Maroon), chemokine (orange), concentric circles separating the three lobular zones (sky blue). **(a)**, Even KC distribution across all nodes. **(b)**, A periportal skew of both infected and uninfected KCs.

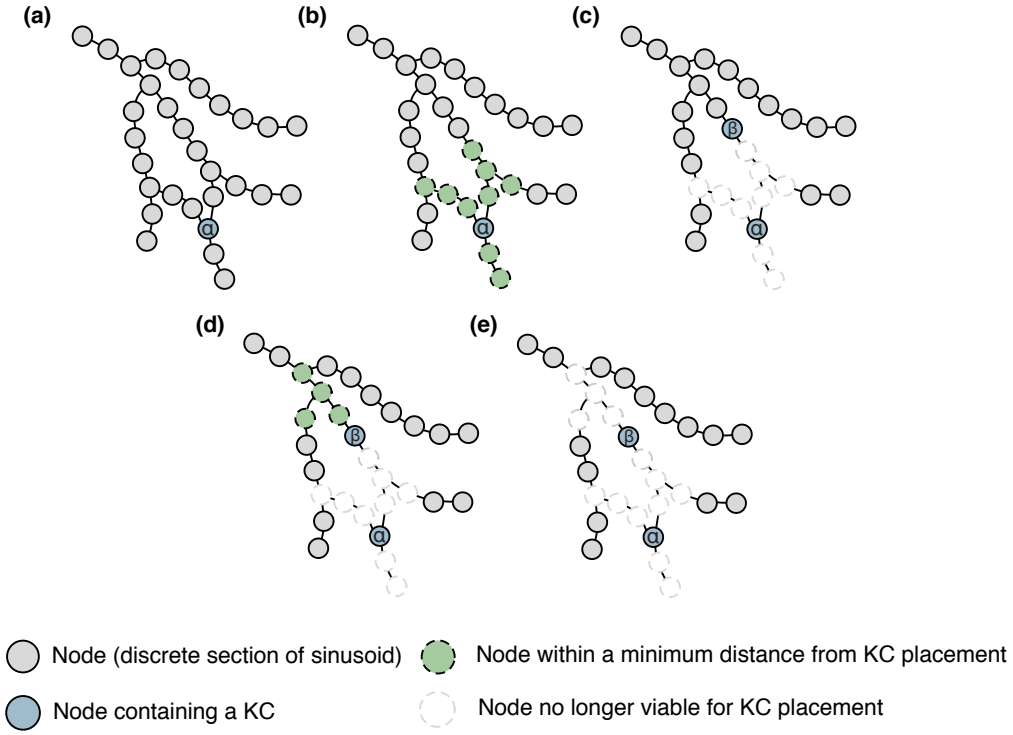


Figure 23: Distribution algorithm 2. Upon initialisation of a simulation, KCs are placed a minimum distance μ apart, ensuring even distribution in the required ratios of KCs throughout the periportal, midzonal and centrilobular zones of the computational representation of the sinusoids outlined in section 5.2 and figure 17. Grey nodes represent discrete sections of sinusoid, and white nodes represent nodes which are deemed no longer viable as a location on which to initialise a KC. **(a)**, A candidate node α is chosen from a set of viable nodes S. **(b)**, The tree is recursed up to distance μ, and all recursed nodes are added to set T. **(c)**, The set difference is calculated to remove proximal nodes from the set of viable nodes, S\T. **(d)**,**(e)**, the algorithm is repeated everytime a new candidate node β is chosen.

5.3.3 NKT Cells

Detailed technical information for implementation is detailed in Appendix 9.1.3.2. The simulation is required to capture several domain model behaviours highlighted for NKT cells. The state diagram for the platform model is depicted in figure 24 along with the relevant NKT cell assumptions. NKT cells are the only agents that were given movement rules, and are capable of traversing the generated sinusoidal structures from node to node, across edges of their current node (see figure 25).

Assumption 1 relates to directed movement (chemotaxis) of NKT cells when presented with multiple chemokine signals (see section 5.3.4.2 and figure 28). There is no data to suggest that NKT cells are capable of avoiding interactions with infected KCs. If this interaction was probabilistic, it would involve adding a further probability parameter to the simulation, one which is not traceable back to any measurable quantity. *Assumption 2* was created to ensure that there is

always an interaction between NKT cells and infected KCs, and a probabilistic calculation is made subsequent to that interaction to determine the duration of the interaction. *Assumption 3* was created as there are no available data pertaining to the dynamics of the NKT cells population in liver lobules over time. The NKT cell population per lobule section was calculated and remains constant throughout the simulation; though NKT cells that leave the simulation enter as new entities. *Assumption 4* was created to ensure that NKT cells do not continuously traffic from one KC to the next, interacting with each. Also, without some kind of recovery mechanism, the NKT cell would perpetually be attracted back, via chemokine signalling, to the previously stimulated KC. This assumption is justified since invariant NKT cells do display anergic behaviour after TCR-mediated activation coupled with co-stimulation [Iyoda et al., 2010]. *Assumption 5*, that NKT cells respond immediately to chemokine signals, is also to avoid introducing any further probabilistic calculations that have no measurable biological foundations. Lastly, *Assumption 6* relates to the NKT cell random walk behaviour reported in the literature [Geissmann et al., 2005; Lee et al., 2010].

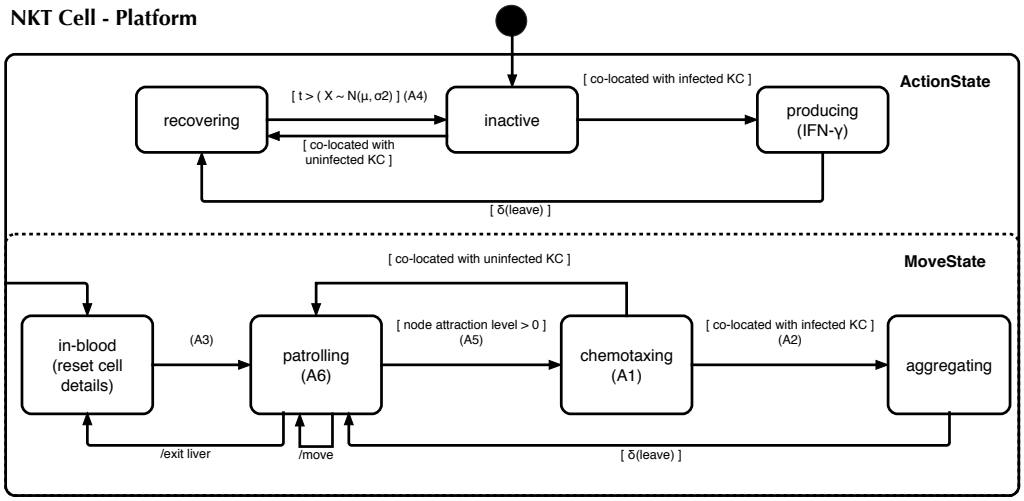
5.3.4 Modelling Chemokines

For the platform model of chemokines, the environment plays an integral role in how those chemokines influence the simulation. Nodes of the simulated sinusoidal network are capable of holding a value that can be attributed to either of the two types of chemokine, chemo-attractant and chemo-retention. Values are based on the calculation of a chemokine growth function (section 5.3.4.1). Chemo-attraction values can be stored on any node of the network, since chemo-attractant is capable of diffusing away from KCs (see section 5.3.4.2). Chemo-retention is an umbrella term used as an abstraction for all of the cytokine, chemokine and receptor mediated retention mechanisms discussed in the domain. Chemo-retention is only ever attributed to nodes that contain an infected KC (see section 5.3.4.3). Table 4 documents the full list of assumptions defined for chemokines prior to implementation. There are several parameter names used in this section that are shortened for brevity, table 5 highlights the mapping between any conceptual parameters and the actual simulator input parameters used to implement them (see tables 7 and 8 for simulation inputs parameters).

5.3.4.1 *Modelling Chemokine as a Growth Function*

A logistic S-shaped curve function was chosen as the means to calculate the value of a given chemokine at a given point in simulation. The S-shaped curves exhibit three separate phases; a low valued initial stage, an exponential-like accelerated growth phase, then finally saturation while approaching a climax. S-shaped curves have been used to describe growth in complex systems ranging from economics to biology and can be representative of cumulative growth [Kucharavy and De Guio, 2007].

The growth function was required to be not centered on zero in order to return a small value of chemokine initially, and the sigmoid represented by equation (1)



Domain		Platform	Justification
A1	NKT cells respond to chemo-attractant	NKT cells under chemotaxis, when presented with two or more attraction gradients, will choose a direction based on attractive chemokine strength.	To facilitate implementation, no experimental data available Implementation discussed in section 5.3.3.
A2	NKT cells interact with KCs	NKT cells will interact with the first infected KC they encounter whilst in chemotaxis.	
A3		The NKT cell population of the lobule section remains constant; cells exiting the environment will enter as new cells via an entry point.	See section 5.6.3.1.
A4	NKT cells are capable of becoming anergic	NKT cells are refractive to stimulation and take a period of time to recover after stimulating and leaving a KC.	(Iyoda et al., 2010). Assumption traced in section 5.6.5.2.
A5		NKT cells will respond immediately to a chemokine signal.	To facilitate implementation, no experimental data available. Chemokine implementation outlined in section 5.3.4.1.
A6	NKT cells can walk the sinusoids and switch direction probabilistically	NKT cells perform a random walk of the tree-node structure, and a probability governs their ability to turn in the sinusoids at random.	(Geissmann et al., 2005). Implementation part of NKT cell properties in section 5.6.4.1.

Figure 24: State Diagram (Platform) - NKT cell. UML state diagram notation is extended with the use of dashed lines to denote states the cell can be in concurrently. Conditions, gates, that must be satisfied before a state transition are denoted using squared brackets []. The δ symbol denotes a probabilistic event, in this instance the probability for the cell to leave a location.

Domain		**Platform**	**Justification**
A1	Attractive chemokines flow in the same direction as blood would.	Attraction diffuses downstream of infected KCs towards the central vein.	Chemotaxis in 3D environments is poorly understood (see section 5.3.4.2) (Haessler et al. 2011)
A2		Strength of attractive chemokine is a function of distance from source, calculated using a simplified Fick's Law of diffusion	See section 5.3.4.2.
A3		Chemokine growth is calculated using a sigmoid function.	See section 5.3.4.1.

Table 4: Chemokine Assumptions. Assumptions made for chemokines in both the domain and platform models.

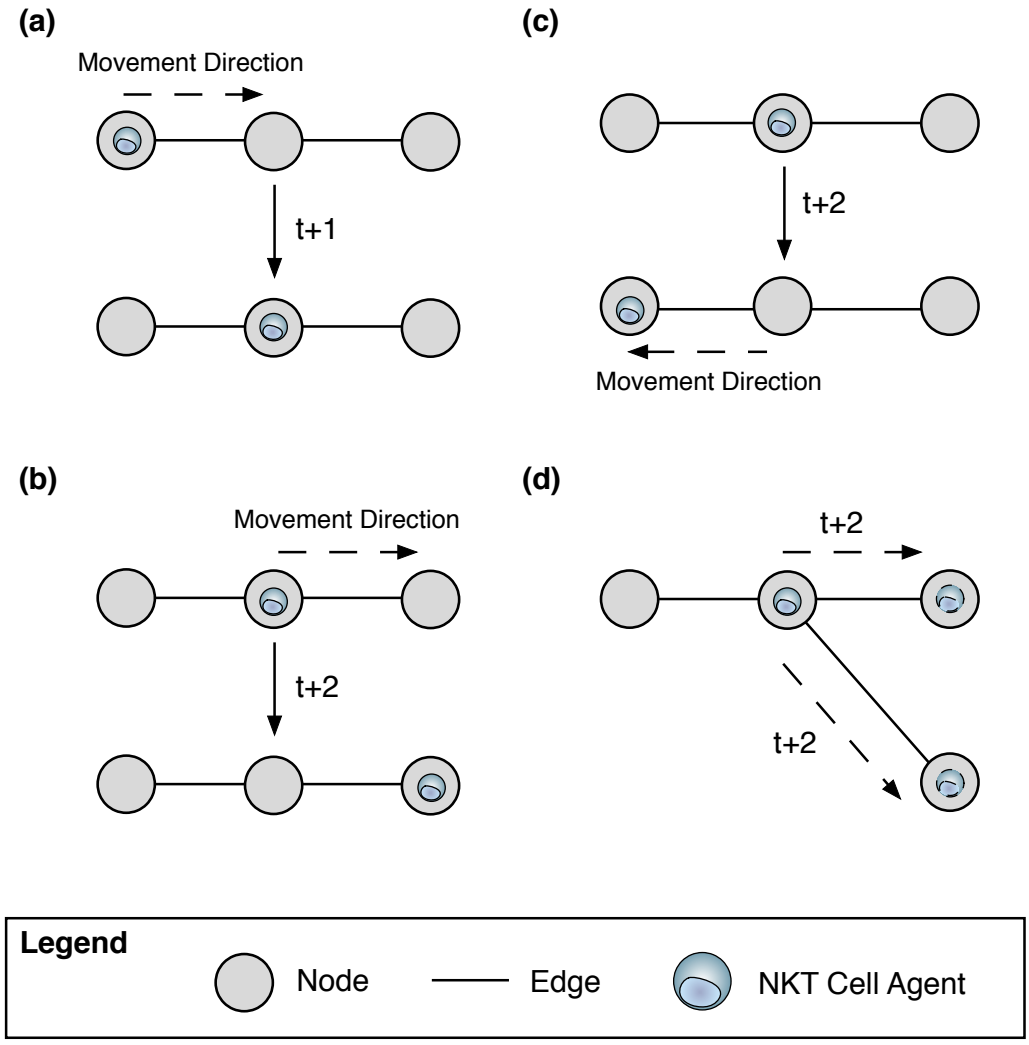


Figure 25: NKT cell movement within the edge-node sinusoid network. (a), NKT cells can move between nodes at each iteration *t*. (b), NKT cells can continue moving in their original direction, or (c) - probabilistically reverse direction. (d), when presented with a branch in the network, NKT cells can travel to any of the available nodes.

Conceptual Parameter	Simulation Parameter (input)
δmin	p_chemoDist
δmax	p_maxDist
δleave	p_leaveProb
δ	N/A runtime parameter
λ	N/A runtime parameter
sourceλ	N/A runtime parameter
d	N/A runtime parameter
ϕ	N/A runtime parameter

Table 5: Parameter Mapping. Mapping between concise conceptual parameters (see sections 5.3.4.1 to 5.3.4.3) and the actual simulator input parameters implemented (see tables 7 and 8).

can accomplish this. To avoid any inconsistances with floating point number calculations as this number approaches 1.0, a value can be parameterised as a maximum threshold. In equation (1), e is Euler's number and the function represents a specialised case of a standard logistic curve. The variable t represents a unit of time/simulation iteration. A standard logistic S-shaped curve is an odd function and has symmetry when rotated 180 degrees around the origin. Therefore, in order to capture the three aforementioned phases of the curve, it is necessary to modify the exponent. Low integer values are sufficient to translate the function and a value of 6 is used in equation (1). Figure 26 illustrates how different values for the constant c produce varying growth curve shapes between units of time ranging from 0 to 172800, 172800 iterations represents 48 hours of simulated time.

$$f(t) = \frac{1}{1 + e^{(-ct)+6}} \tag{1}$$

As an example, if I describe the growth of interferon, multiple NKT cells can produce chemokine, though the specific concentration they produce is abstracted away. Each NKT cell increments the value of t for each iteration that it is interacting with a KC. Therefore multiple NKT cells localised around a KC will increase the per iteration stimulation received and therefore cumulative chemokine produced. The presence of NKT cells also dictates when and by how many increments to the t value that KCs should make to produce attractive and retentive chemokines. To implement chemokine decay, the t time is merely decremented for every iteration that a KC is not receiving stimulation.

Several parameters: *p_chemoIFN, p_chemoAttract and p_chemoRetain* are used to define how many units of stimulation (increments) are required for the respective chemokines to reach the threshold maximum value. At runtime, the code in listing 15 is used to calculate the constant c required to reach a given threshold value in a stipulated number of iterations/increments.

5.3.4.2 *Modelling Chemokine Mediated Attraction*

As previously described, a network tree structure was chosen to represent the sinusoidal environment. The consequence of this, is that each node must con-

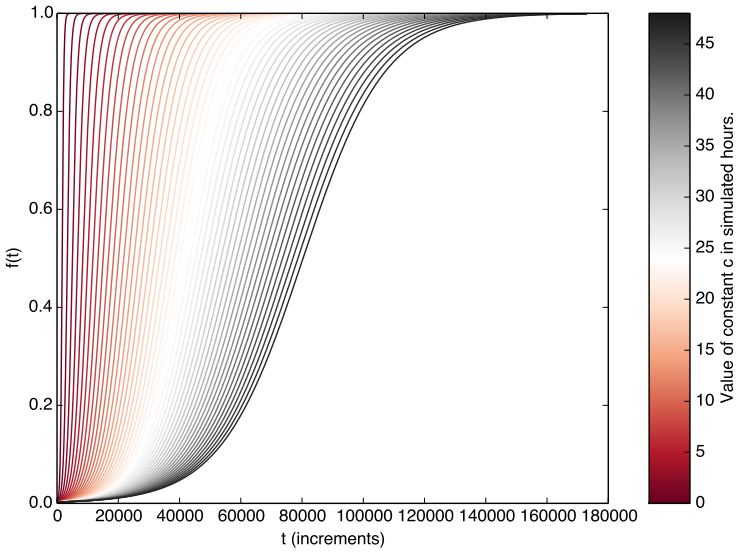


Figure 26: Chemokine growth function. For a given constant c, representing the number of hours required to reach maximum chemokine level (see colour bar), curves illustrate the growth of chemokine determined by equation (1) at each simulation iteration t (0 to 172800 iterations), assuming single increments of t.

tain not just cells, but also cell produced cytokines and chemokines. Since I previously assumed that KC-NKT interactions only occur when those cells are localised together on a node, it is relatively trivial, computationally, to represent NKT produced stimulatory cytokine and also the KC produced retentive cytokine values as floating point numbers due to the finite number of infected cells in the simulation. I make the reasonable assumption that chemo-attractant only diffuses downstream of infected KCs, towards the central vein of the lobule. In reality, the dispersion of chemo-attractants in the liver lobule has yet to be characterised in this experimental setting, though I do know that the expression of chemo-attractants is observable in areas of lymphocyte infiltration [Zeremski et al., 2008].

Attractive diffusion distance is determined by various factors which are illustrated in figure 27. Diffusion is a function of distance and is recalculated every simulation iteration with equation (2) (see also figure 27) - the function is floored to the nearest integer, and that is used as the updated diffusion distance.

$$\delta = |\mathrm{source}\lambda * (\delta max - \delta min)| \tag{2}$$

where:

$$\begin{aligned} \delta &= \text{calculated diffusion distance} \\ \mathrm{source}\lambda &= \text{the level of attractive chemokine} (0 \leqslant \mathrm{source}\lambda \leqslant 1.0.) \\ \delta min &= \text{minimum diffusion limit (nodes)} \\ \delta max &= \text{maximum diffusion limit (nodes)} \end{aligned}$$

The level of chemokine diffused to subsequent nodes is a function of the distance from source governed by equation (3) (see also figure 28).

$$\lambda = \frac{source\lambda}{d} \tag{3}$$

where:

$$\begin{aligned} \lambda &= \text{level of attractive chemokine at a given node} \\ source\lambda &= \text{the level of attractive chemokine at source KC} \\ d &= \text{distance from source (in nodes)} \end{aligned}$$

Two different assumptions for how attractive chemokine influences NKT cells was implemented. The first assumes that diffused attractive chemokine merely determines which direction an NKT cell in chemotaxis should travel, and where there are one or more competing chemokine signals, the cell calculates its next move probabilistically. The second implementation assumes that strength of chemokine plays a role in chemotaxis, and where there are multiple competing signals, an NKT cell will traffic via the strongest of those signals (figure 28).

5.3.4.3 *Modelling Chemokine Mediated Retention*

Retention is modelled as a function, which is dynamic based on the local concentration of retentive chemokine at an infected KC node. The relative quantity of retentive chemokine produced by KCs mediates the sustained retention of NKT cells at infected foci dynamically, in line with NKT cell-derived stimulatory IFN-γ. Thus, the probability of an NKT cell leaving an infected KC is governed by equation (4). To avoid the unwanted artifact of NKT cells becoming fixed as the value of δnktleave tends to 0, the simulation utilises a lower bound threshold for retention, parameterised using the input $p\_minLeaveProb$.

$$\delta nktleave = \delta leave - (\phi * \delta leave) \tag{4}$$

where:

$$\begin{aligned} nktleave &= \text{value of retention at the infected node} \\ \delta leave &= \text{maximum parameterised retention probability} \\ \phi &= \text{value of retention at infected node} \end{aligned}$$

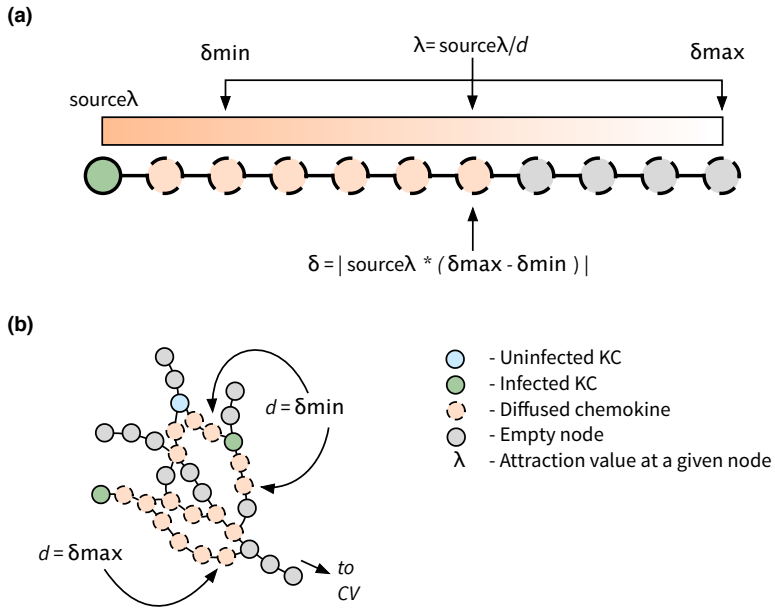


Figure 27: Conceptualising Chemokine Diffusion. **(a)**, The current chemokine diffusion distance is a function of the level of attractive chemokine at a source KC (sourceλ), with minimum and maximum limits of δmin and δmax respectively. **(b)**, Chemokine diffuses downstream through the network, towards the central vein.

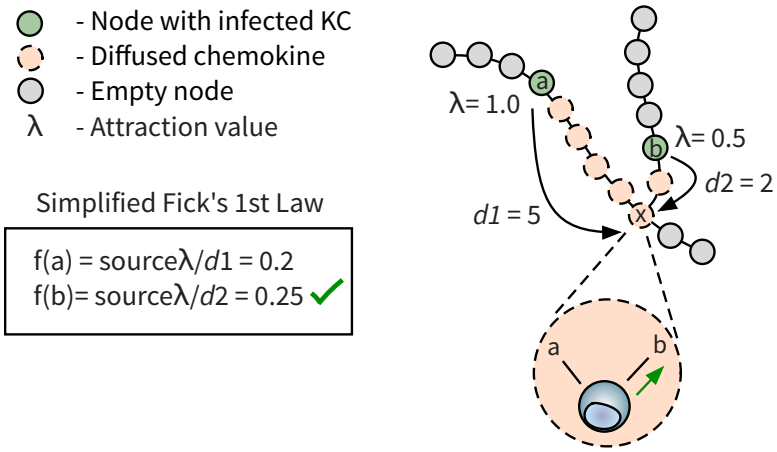


Figure 28: Chemokine Strength Decision. NKT cells when faced with 2 or more competing chemokine signals, will traffic via the strongest of those signals, calculated as a function of the distance from the each respective source. The example illustrated demonstrates that although node *a* has a stronger level of chemokine, the level at node *x* is lower than the chemokine diffused by the more proximal node *b*.

5.4 THE AGENT BASED MODELLING PLATFORM

The previous sections of this chapter outlined the platform model, detailing the various cells, algorithms, and functions required to implement the original domain model. The platform model acts as a system specification for the software development process to turn my model into computer code. In order to begin development, I must decide upon a platform to utilise to develop an agent-based model.

As discussed previously in section 2.1.3.1, where I outlined the various agent-based modelling platforms, there is a vast array to choose from and none particularly stand out as there are advantages and disadvantages to each. The only requirements for this research were that the platform had to be: freely available, open-source, capable of running across multiple platforms, and compatible with the object-oriented paradigm to facilitate future extensibility.

MASON [Luke et al., 2004] was chosen as it fulfills all of the previously outlined requirements. MASON utilises the object-oriented Java programming language, and there is a wealth of existing libraries and documentation that could facilitate model implementation. MASON also has many useful simulation libraries, though the only requisites needed in this research context were libraries to handle the scheduling of agents and simulation visualisation. Java classes were then written specifically for each cell agent. The core simulation logic does not require any additional library functionality provided by MASON.

The simulation update cycle and how MASON's scheduling libraries are used are detailed in Appendix 9.1.1, and visualisation library use in Appendix 9.1.1.1. The discrete-event, or clock based, aspects of MASON are used using the scheduling library, and events in the simulation are driven by rules and conditions declared within the agent classes.

5.5 SIMULATOR ARCHITECTURE

A top-down modular approach was taken to design the simulator. Individual package modules were conceived for the various simulator components. Packages are particularly useful for any future extensions of the simulator. Cell logic, visualisation and simulation output are all decoupled. The UML Class Diagrams in Appendix 9.1.4 outline the various packages, which include:

SIMULATION Main package containing the simulation objects, environment, and cytokine and visualisation managers (figure 63)

STRUCTURE The structure entity that defines a edge-node network (figure 64)

NETWORKS A file-only package containing the XML structures

CELLS Encapsulates the various cell types (figure 65)

CSV The logging package for outputting simulation data (figure 66)

ENUMS Various enumerations to handle states (figure 67)

CYTOKINES Defines cytokines/chemokines as a property that can be assigned to nodes (figure 68)

VIEWS Classes to create a 3D visual representation of the simulation environment and cellular components (figure 69)

The heirarchy of simulation objects is depicted in figure 29. The *Environment* class in the *Simulation* package encompasses all elements of the simulation. The environment object contains a single *structure* comprised of nodes and edges. The environment object receives updated state information from both *cell agents* and the *Cytokine Manager*, this data is then used to update the *structure* object. After individual cells have acted, the Cytokine Manager handles all the relevant calculations of cytokine and chemokine values. Implementation details for the Cytokine Manager are detailed in Appendix 9.1.3.3.

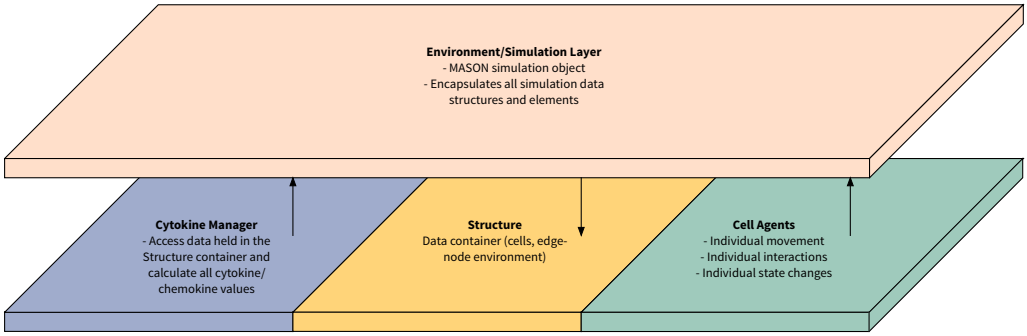


Figure 29: Simulation Architecture. The heirarchy of simulation components, showing the top level simulation/environment layer, which manages all cells, cytokines, and the underlying network environment.

5.5.1 Model - View - Controller

The simulator was designed to accommodate some form of visualisation. The Model-View-Controller (MVC) [Reenskaug, 1979, 2003] design pattern was chosen to separate simulation logic from visualisation and any potential in-simulation control mechanisms. MASON is also built around MVC and therefore this decision is in fitting with the underlying framework. Separation of simulation logic from visualisation allows the invocation of the simulation from the command line without visualisation or alternatively for a graphical user interface to be used.

3D *field portrayals* are used to store visual objects in a continuous environment, which is then displayed using a *simple portrayal*. The MASON *portrayal* package provides many useful functions for accessing and modifying individual or multiple visualisation objects. The simulator has been designed so that each cell type has a corresponding *Java3D* object which is added to its own 3D portrayal. Portrayals for each cell type allow individual cell populations to be toggled on or off when using MASON's visualisation inspectors. However, the views have no direct interaction with the underlying model. The use of *portrayals* allows the simulator to be extended in future to allow for user input from a graphical

user interface (GUI), facilitating the control of the underlying model from a GUI. Sample simulator visualisation can be seen in figure 30.

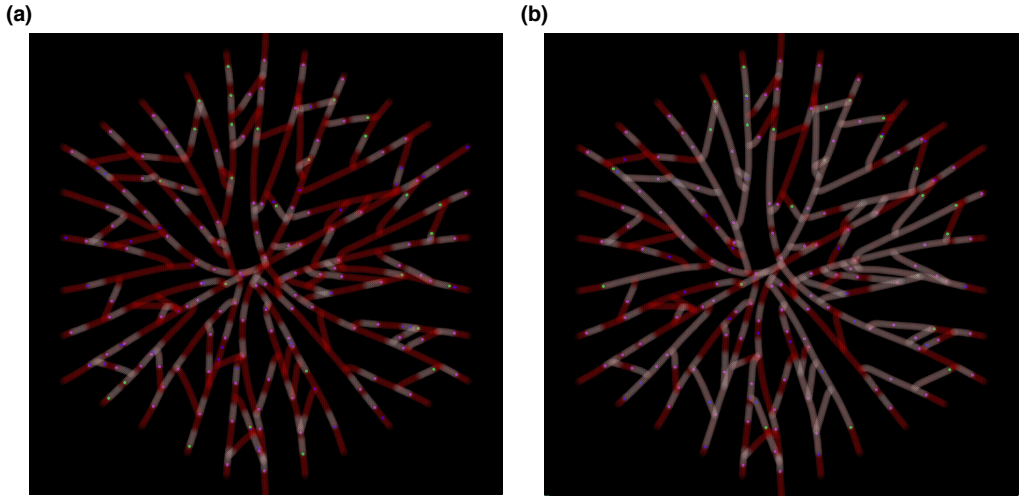


Figure 30: Simulator Visualisation. Sinusoids (Maroon), Infected KC (green), Uninfected KC (pink), NKT cell (blue), Chemokine (white). **(a)**, t(0) of a representative simulation. **(b)**, t(24hrs) of a representative simulation showing the increased diffusion of chemokine throughout the lobule section.

5.6 MODEL VALIDATION

Earlier in this chapter, sections 5.3.1 to 5.3.3 established platform representations of the cells being modelled, and detailing their mechanistic behaviours. The emphasis now turns to the validation of the model. Due to the inherent abstraction required to model particular biological functions, such as cell attraction and retention dynamics, parameters are created which have, or may not have, biological values or equivalents. It is important to firstly calibrate a model with known biological values, but to also understand how the model is influenced by those parameters that have unknown or no biological equivalents.

The calibration process is discussed first (see section 5.6.1), then I discuss a process of arguing the validity of my simulation in representing the underlying biology (see section 5.6.7). The use of statistical analyses to both determine and understand the influence that all simulation parameters exert on simulation output will be discussed as part of the CoSMoS results model in chapter 6 (see section 6.1.3 for latin-hypercube sensitivity analysis, and section 6.1.2 for aleatory analysis to mitigate uncertainty in the simulation output).

5.6.1 Calibration

The process of calibration was required to correctly parameterise the simulation, and has two stages. The first stage is to take biological values relevant to the behaviour of KCs and NKT cells in the hepatic microenvironment, and directly input them into the simulation; such as, numbers of particular cells, or percentages of infected cells. Where there is biological data that cannot be directly input into the simulation, those values may need surrogates for implementation; an example of this surrogacy is the implementation of variable NKT cell speed (section 5.6.3.2). The second stage of calibration is to ascertain values for simulation parameters that have either unknown biological values, or those that have no biological equivalent, but that are required for a functioning simulation - such as constants for mathematical functions or probabilistic calculations. The table in table 6 lists all the biological values incorporated into the simulation. This section will describe how those values have been implemented and how they specifically influence the execution of the simulation.

Biological Parameters	Value	Source
NKT cell velocity in the sinusoids	10-20μm/minute	(Geissmann et al., 2005). Calculation in 5.6.3.2, implementation in Appendix 9.1.3.2.
NKT cell numbers in a section of mouse liver lobule	~49 (~1:3 ratio with KCs)	Derived from (Lee et al., 2010) as discussed in section 5.6.3.1.
Kupffer cell numbers per gram	14-20 x 10^{6}/g	(Bouwens et al., 1986)
Infected KCs at 2 hours	~23%	(Beattie et al. 2013) data
KCs per lobule section	~146	See Appendix 9.2 data
Non-branched segment length	43.1 ± 2.25 μm	(Höhme et al., 2010) supplementary materials.
Mean branching angles of sinusoids	32.5° ± 11.2°	(Höhme et al., 2010) supplementary materials.
KC spatial distribution	Percentages: Periportal (PP) - 43 Midzonal (MZ) - 28 Centrilobular (CV) - 29 Ratios: 4(PP), 3(MZ), 2(CV)	(Bouwens et al., 1986; Sleyster and Knook, 1982) implemented in section 5.3.2.1.

Table 6: Table of biological parameters used for calibration.

5.6.2 Kupffer Cell Parameters

This subsection details the calibration of all KC related simulation parameters.

5.6.2.1 *Number of KCs and Infection Percentages*

To ascertain the representative number of KCs per liver lobule section, and the average infection rates in the experimental mouse model, two experiments were performed by my collaborators to quantify these values. Table 20 in Appendix 9.2 outlines data acquired from two groups of 3 mice, with 10 fields of view per mouse, to determine the number of uninfected motile and sessile KCs present, and additionally the number of infected motile and sessile KCs. Whilst two groups of 3 mice, 6 mice total, may seem insufficient to adequately capture the biological variation that exists, a balance must be struck between the acquisition of data to parameterise the simulation, and the sacrifice of mice. Part of the benefit of computational modelling is to replace certain aspects of animal testing, therefore whilst the values for KC numbers and infection percentages could be classed as preliminary, we can and do explore, through sensitivity analyses, the effects of different values for these parameters (see section 6.1.3).

From these data I calculated an average percentage infected rate across both experiments as approximately 23%. The average combined total of motile and sessile KCs per image section is 140.8 and each image section has the dimensions of 142.0*142.0*11.9μm^2; given my safe assumption that KCs can only be one deep given a depth of 11.9 μm, I ignore depth and calculate the total area across all image sections as 201649μm^2. The density of KCs across the imaged areas is then calculated as the average number of KCs divided by the total area, 140.8/201649 = 0.00069844. To normalise these data for biologically measured lobule sections, the area of a hexagonal lobule was calculated as 209550.435μm^2 from data by Höhme et al. [2010]; thus multiplying the lobule area by the calculated KC density, I arrive at approximately 146 KCs per lobule section. To parameterise the simulation with the calculated values, the number of uninfected KCs in the simulation is input using the parameter *numUninfected* and the number of infected KCs, 23% of the total, with the parameter *numGoodKCs*.

5.6.2.2 *KC Distribution*

Sections 3.2.3 and 5.3.2 previously outlined the KC distribution ratios, and these are input using parameters *p_ratioCV, p_ratioMZ and p_ratioPP*.

5.6.2.3 *KC Activation Threshold*

Section 5.3.4 described how growth functions are implemented for the simulation to calculate cytokine/chemokine values. As the sigmoid function used never reaches the theoretical maximum of 1.0, a threshold value is needed to act as a practical maximum for the simulator to use. A default value was chosen as 0.999 and this is input using the *p_ifnThreshold* parameter.

5.6.3 NKT Cell Parameters

This subsection details the calibration of all NKT cell related simulation parameters.

5.6.3.1 *Number of NKT cells*

An approximate ratio of KCs to NKT cells was calculated as 3:1, based on imaging data published by Lee et al. [2010]. Therefore, given the previously calculated KC numbers as 146 per lobule section, the number of NKT cells is calculated as 49 per section. The value is input into the simulation using the *p_numNKTs* parameter.

5.6.3.2 *NKT cell velocity*

NKT cell velocity is reported in the range of 10-20μm/min [Geissmann et al., 2005; Lee et al., 2010], and because of this range the model needs to be able to account for the varying speeds of the cells. A detailed analysis of NKT cell speed could not be undertaken in the scope of this research due to limitations with setting up the strain of mouse required to distinguish NKT cells visually from other mononuclear cells (see also section 8.4.2).

Parameterising NKT cell velocity requires several more parameters to be defined. In the engineered sinusoid lobule environments, 10-20μm translates directly to 10-20 nodes/min. The simulation was designed with each simulation iteration representing 1 second. If an NKT cell were capable of moving every iteration, the distance covered would be 60 nodes/min. Therefore, for a maximum speed of 20 nodes/min, the NKT would need to move every 3 simulation iterations. Similarly, for a minimum speed of 10 nodes/min, the NKT would need to move every 6 iterations. To implement this, two simulation parameters are defined, *p_moveMin* and *p_moveMax*, with values of 3 and 6 respectively. Each NKT cell has a *move* attribute with a default value of 0. When the move method is invoked (Appendix 9.1.3.2) the instance variable *move* value is 0, and a new time before the next movement (in iterations) is calculated randomly between the values *p_moveMin* and *p_moveMax*. The calculation ensures that NKT cell speed is maintained within the pre-defined boundaries and also ensures that NKT cell speed is dynamic, allowing for experimentally reported biological variation.

5.6.4 Unknown and Implementation Specific Parameters

The simulation contains 17 separate parameters, several of which were detailed in sections 5.6.2 and 5.6.3 which outlined how known experimental data was used to determine numerical values for elements of the simulation. However, there are several parameters for which it is currently not possible, or practical, to acquire data for the purposes of simulator parameterisation. Those parameters are as follows:

p_turnProb, *p_anergicItns*, *p_escapeItns*, *p_leaveProb*, *p_chemoIFN*, *p_chemoDist*, *p_maxDist*, *p_ifnThreshold*, *p_chemoAttract*, *p_chemoRetain*.

All of the listed parameters are needed for the simulation to function, however they are merely surrogates for some unknown biological quantity or mechanism, and I am unaware of those quantities or mechanisms in the specific context of Leishmania induced hepatic inflammation, or in all cases in any biological context. The purpose of each parameter is outlined in tables 7 and 8, and later I employ a Latin-Hypercube Sensitivity Analysis method [Marino et al., 2008; Read, 2011; Alden et al., 2013] to ascertain how robust the simulation output measures were to perturbations of these parameters (see section 6.1.3).

5.6.4.1 *NKT cell properties*

The seemingly random turning behaviour of NKT cells was previously described in section 3.3. The parameter *p_turnProb* was used to set a probability that an NKT cell will turn during each cycle of movement. If this value were to be set too high, the NKT cell could find itself repeatedly switching direction and becoming effectively immobile.

The *p_escapeItns* and *p_anergicItns* parameters perform a similar function. If I assume that in a simulation without these parameter, an NKT cell has provided stimulation to KC, but the condition for it to then leave that location has been met. Since the surrounding area will have attractive chemokine, if the NKT cell were to move directly to a patrolling state, it would leave the KC location and become immediately attracted back to the KC it just left. Therefore, there needs to be some mechanism to allow the NKT cell to ignore chemokine for a period of time, so that it may escape the immediate area. Similarly, as previously described regarding NKT cell anergy, it is not known whether NKT cells can continue to repeatedly stimulate other cells, or if there is a 'recharge' period required. I assume that NKT cells do become anergic, and therefore the *p_anergicItns* parameter defines the period of time post stimulation that an NKT cell will patrol the sinusoids though ignore all KCs and chemokines.

5.6.4.2 *Cytokine/Chemokine related properties*

Parameters *p_chemoIFN*, *p_chemoAttract* and *p_chemRetain* are all times, in simulation iterations, that it takes for a particular chemokine to reach a maximum concentration. There are so many compounding variables that would determine the growth of chemokines; number of stimulating promoter cells, duration of stimulation from each cell, blood flow, diffusion of chemokine, etc. I can therefore have to choose a time, and use statistical methods to determine how different times impact on the output measures.

Parameter		Value	Units	Description	Source
NKT cell	p_turnProb	0.005	probability	Probability that an NKT cell will reverse direction in the sinusoids.	No biological equivalent; explored and chosen through parameter sensitivity analysis (section 6.1.3).
	p_moveMin	3		Value given to link simulation iterations to NKT cell velocity.	Calibrated to published NKT cell speeds from (Geissmann et al., 2005). See sections 5.6.3.2 and Appendix 9.1.3.2.
	p_moveMax	6			
	p_anergicItns	3600	iterations	Time in iterations for an NKT cell to remain unable to stimulate a KC.	No biological equivalent; explored and chosen through parameter sensitivity analysis (section 6.1.3).
	p_escapeItns	600	iterations	Time in iterations for an NKT cell to,escape the influence,of KC produced,chemo-attractant.	
	p_leaveProb	0.000265306	probability	The probability of an interacting NKT cell leaving the location of an infected KC.	
	p_minLeaveProb	0.00005	probability	To guard against the probabilistic tipping point whereby retention causes cells to never leave.	
	p_chemoIFN	172800	iterations	Interaction time required to reach maximum attractive chemokine concentration.	
Kupffer cell	p_chemoDist	20	distance(nodes)	Starting diffusion distance for attractive chemokine.	No biological equivalent; explored and chosen through parameter sensitivity analysis. See section 5.3.4.2.
	p_ratioCV	0.1	percentage	Ratio of infected cells in the CV region of the lobule section	Bouwens et al., 1986. Sleyster et al., 1982. Data and implementation in section 5.3.2.1.
	p_ratioMZ	0.25	percentage	Ratio of infected cells in the MZ region of the lobule section	
	p_ratioPP	0.65	percentage	Ratio of infected cells in the PP region of the lobule section	
	p_maxDist	200	distance(nodes)	Maximum diffusion distance for attractive chemokine.	Calibrated to twice reported max (Weber et al., 2013), see also section 5.3.4.2.
	p_ifnThreshold	0.999	threshold	Threshold value of IFN-γ required to activate a KC. Chemokine function f(x)→1, therefore a threshold is required.	No biological equivalent, explained in section 5.3.2.

Table 7: Cell Specific Parameters.

Parameter		Value	Units	Description	Source
Simulation	p_numInfectedKCs	33	cells	Number of infected KCs in a π*(284μm)^2 sectional area of sinusoid.	Data in Appendix 9.2.
	p_numUninfected	113	cells	Number of uninfected KCs in a π*(284μm)^2 sectional area of sinusoid.	
	p_numNKTs	49	cells	Number of NKT cells in a π*(284μm)^2 sectional area of sinusoid.	See section 5.6.3.1 (Lee et al., 2010). Also domain expert assumption.
Chemokine	p_chemoAttract	43200	iterations	Stimulation time required to reach maximum attractive chemokine concentration.	No biological equivalent; explored and chosen through parameter sensitivity analysis. Retention discussed in assumption tracing 5.6.5.1.
	p_chemoRetain	172800	iterations	Interaction time required to reach maximum retentive chemokine concentration.	
	p_chemoIFN	172800	iterations	Interaction time required to reach activate infected KCs.	

Table 8: Simulation Specific Parameters.

5.6.5 Exemplar Assumption Tracing

The implementation of a modelling platform that I have detailed in this chapter and Appendix 9.1 is just one possible implementation of the underlying domain model. Some of the underlying assumptions made during the development of both the domain and platform models ultimately influence some of the design decisions taken when engineering the modelling platform. In this section I will detail two exemplar design decisions that were heavily influenced by the limitations in underlying knowledge from the original domain, and hence assumptions made in the domain model.

5.6.5.1 *Probabilistic Retention*

The mechanisms underpinning cell to cell interactions between NKT cells and KCs are poorly understood. Whilst much of the motivation for this research outlined the CD47 - SIRP-α pathway for stimulation of KCs by NKT cells, the specific pathways involved in cell to cell binding are unknown. Without specific knowledge of those mechanism, assumptions must be made to determine how cells will bind and unbind.

I made the assumption that some form of cell to cell interaction will always occur when an NKT cell is in direct contact with a KC (see assumption 2 figure 24). The duration of this encounter is then deemed to be a function of the retentive capabilities of KCs. This is a direct hypothesis based on the known CD47-SIRP-α interaction pathway [Svensson et al., 2005]. As NKT cells first encounter KCs, the interactions with KCs will be fleeting. However, as the KCs become ever more stimulated, and eventually engaged, the functional capacity to sustain interactions with NKT cells will be improved.

From an implementation perspective, the functionality I have described is deemed probabilistic retention. To implement probabilistic retention I required a default initial probability for a given NKT cell to leave a KC location. This is the highest possible probability for any given simulation. Equation (4) in section 5.3.4.3 then modifies this probability based on the current level of retentive chemokine at the current KC node. This ensures that when the value of retentive chemokine increases, the probability for an NKT cell to leave will decrease (listing 1). Thus, dynamic interaction behaviour between NKT cells and KCs is implemented, a direct influence of the limited knowledge of the cell to cell interactions between NKT cells and KCs. A range of possible initial retention probabilities are explored through sensitivity analysis (see section 6.1.3).

5.6.5.2 *NKT Cell Anergy*

The anergic properties of NKT cells have been highlighted in the literature [Iyoda et al., 2010], as discussed in section 4.2.1.2, though the underlying mechanisms responsible for cell anergy are unknown. As there is no known mechanism for this anergy, an assumption was required in order for me to implement this functionality. Without some form of anergic functionality, NKT cells engaged in the stimulation of a KC would never leave that KC. Should the cells probabilistically

leave the site of a KC, they would instantly be attracted back to that KC if the only two ActionStates were *inactive* and *producing* (see figure 24). Therefore, there needed to be a method for NKT cells to 'ignore' that attractive signal if they have recently ceased stimulating a KC.

I made the assumption that these cells will remain anergic for a given period of time after stimulation. As this time has no specific biological grounds, I explore through statistical analysis (see section 6.1.3) the influence that different parameterised anergic times have on simulation output measures. Fundamentally, any real mechanism in all intents and purposes would result in an anergic period of time. This anergic time assumption was detailed in assumption 4 (outlined in section 5.3.3).

From an implementation perspective, subsequent to the probabilistic calculation for NKT cells to leave the site of a KC, the NKT cell ActionState is updated to *recovering*. As detailed in Appendix 9.1.3.2, each state causes cells objects to call state specific methods. When in the recovering state NKT cells will call the *recover()* method, which simply decrements an instance variable which is initialised with the value parameterised for NKT cell anergy. Once the instance variable has decremented to zero, the cell state is then return to *inactive* and the cell can begin functioning as normal.

Listing 1: Anergic Implementation

```
/**
 * Produce Cytokine to stimulate Node/KC
 */
 private void produce()
{
    /* Decide the probability based on current retention level */
    double probability = leaveProb - (leaveProb*env.getRetain(node));
    if (Double.compare(probability,minLeaveProb) <= 0) {
        probability = minLeaveProb; /* mindful of a minimum retention prob */
    }
    /* Now compare with random number to decide leave/noleave*/
    int comp = Double.compare(env.random.nextDouble(), probability);
    if (comp < 0) { /* leave cluster */
        if (env.structure.getNode(node).getKC() instanceof KCUn)
            anergicTime = escapeItns;
        else anergicTime = anergicItns;
        action_state = ActionState.recovering;
        move_state = MoveState.patrolling; /* force out of aggregation */
        move();
    }
    else {
        env.incrementIFN(node);
    }
}

/**
 * Recover - decerement anergy - after stimulating KC.
 */
private void recover()
{
    anergicTime -= 1; // recovering
    if (anergicTime <= 0)
    {
        anergicTime = 0;
        action_state = ActionState.inactive; // recovered
    }
}
```

5.6.6 Validity Testing With Requirements-Based Testing and Trace Analysis

There is inherent complexity that arises when attempting to validate and test a software simulation that manifests behaviour that is emergent. The stochastic processes underlying the simulation model described in this thesis mean that no simulation will produce identical simulation results - unless the same seed is provided to the simulation's random number generator. This makes conventional software testing methods either difficult or infeasible to perform.

As a result, the predominant testing method employed to validate the simulator has been face-validity, requirements-based testing, code walkthroughs, animation and and parameter sensitivity analysis (see section 6.1.3).

An example of requirements testing of the simulation is in the analysis of NKT cell behaviour. Given the difficulty in automating such tests, animation and trace analysis of live simulation output was used to validate these kinds of tests. A sample structured requirements test is as follows:

- An NKT cell will patrol the sinusoids upon initialisation of the simulation.

- Subsequent to encountering chemokine in the sinusoids the NKT cell will transition from MoveState **patrolling** to **chemotaxing** (ActionState **inactive**), the NKT cell will then chemotax for a distance of 20 nodes (the basal distance that KC chemokine is diffused).
- After this chemotax, the NKT cell will encounter the KC and transition from MoveState **chemotaxing** to **aggregating**.
- As ActionStates are updated subsequent to movement, the simulation will take one iteration to update the ActionState from **inactive** to **producing**.
- After a period of time the NKT cell will probabilistically leave the KC and move from MoveState **aggregating** to **patrolling** and ActionState producing to **recovering**

Listing listing 2 shows sample from the trace output with annotations. Sit downs with domain experts helped to ensure that the output observed had face validity with how they expected the system to operate given the underlying assumptions of the model. The trace output was analysed in combination with the simulator visualisation to allow myself and the domain experts to study processes such as cell movement and chemokine diffusion. All of these activities helped facilitate the validation of the simulator.

Listing 2: NKT Requirements Test

```
Iteration: 281
Node ID: 4564
MoveState: patrolling
ActionState:inactive

Iteration: 282
Node ID: 4564
MoveState: chemotaxing
ActionState:inactive

... <- Chemotaxing NKT cells move every iteration;
       20 nodes travelled in 20 iterations (minimum chemokine distance)

Iteration: 302
Node ID: 8231
MoveState: chemotaxing
ActionState:inactive

Iteration: 303
Node ID: 8231
MoveState: aggregating
ActionState:inactive

Iteration: 304
Node ID: 8231
MoveState: aggregating
ActionState:producing

... <- cell fulfils probabilistic leave condition

Iteration: 1753    <- approx 29 minutes of simulated time
Node ID: 8231
MoveState: patrolling
ActionState:recovering
```

35 ...

5.6.7 Argumentation

A discussed previously in section 2.2.2.2, argument-driven validation (ADV) can be used to establish an additional level of confidence in simulations used for scientific purposes [Ghetiu et al., 2010]. The use of arguments, described using the visual notation depicted in figure 5 (originally from [Ghetiu et al., 2010, Fig 1.]), can help to validate my model and provide a visual link between evidence base and assumption. The core goal is to argue that my engineered simulation is an adequate representation of the underlying biology.

I begin by presenting my top-level claim, that my engineered model is an adequate representation of the biology (Claim 1, figure 31). The model is constructed utilising a variety of abstractions, and yet is expected to be able to replicate measurable biological responses, therefore, one must argue that despite those abstractions, my model is capable of producing results that are representative of the biological system, and thus allowing us to formulate reasoned predictions. Arguing the adequacy of my abstractions, and the experimental data used to justify or infer them, is the top-level strategy to support claim 1, and that strategy is tackled in for main parts. Firstly, sub-claim 1.1 relates to justifying the accuracy/validity of the underlying biological data (see figures 32 to 34). Secondly, sub-claim 1.2 will justify that the platform model implementation is representative of the biological system (see figure 35).

Sub-claim 1.1 argues that my simulation utilises data which is both accurate and relevant to my modelling study. This argument requires multiple strategies, and is split across four figures (see figures 32 to 34). Strategy 1.1a formulates an argument with data pertaining to NKT cells (figure 32). Strategy 1.1b argues the validity of data relating KCs (figure 33). Finally, Strategy 1.1c defends the accuracy of statistical data I have regarding sinusoid environments (figure 34).

Sub-claim 1.2 (figure 35) argues that the platform model adequately represents the underlying biological system that the domain model represents. Two strategies compose claim 1.2, one which relates to the attractive and retentive mechanisms implemented in the model, and the other that argues the simulated environment is representative of the liver sinusoids.

Argumentation with respect to simulation results and their validation is established in section 7.4.

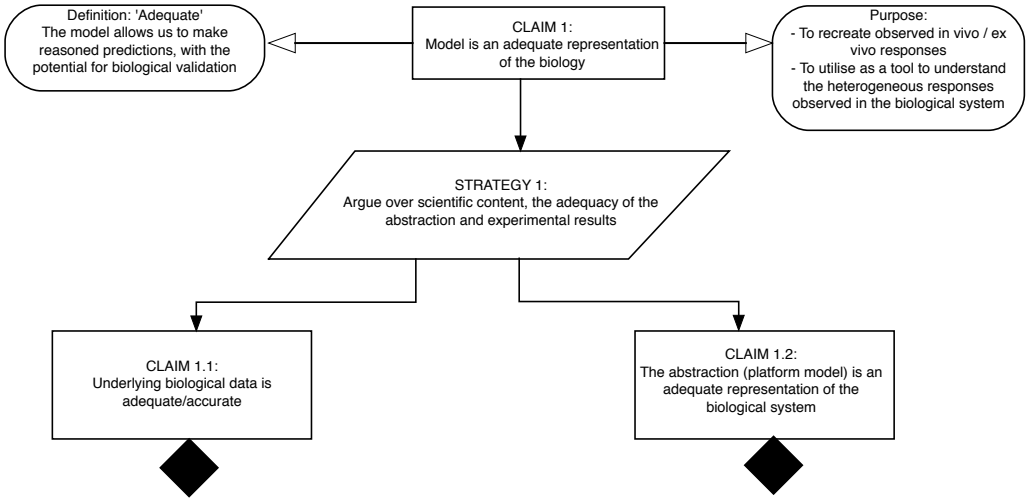


Figure 31: Argument Based Validation. This high level claim posits that the model is an adequate representation of the underlying biology. A strategy is defined to argue the top level claim, with the strategy itself being composed of several sub-claims. Black diamonds highlight where a claim is too large to be contained in the current figure, and is given it's own figure. For claim 1.1.1 see figures 32 to 34.

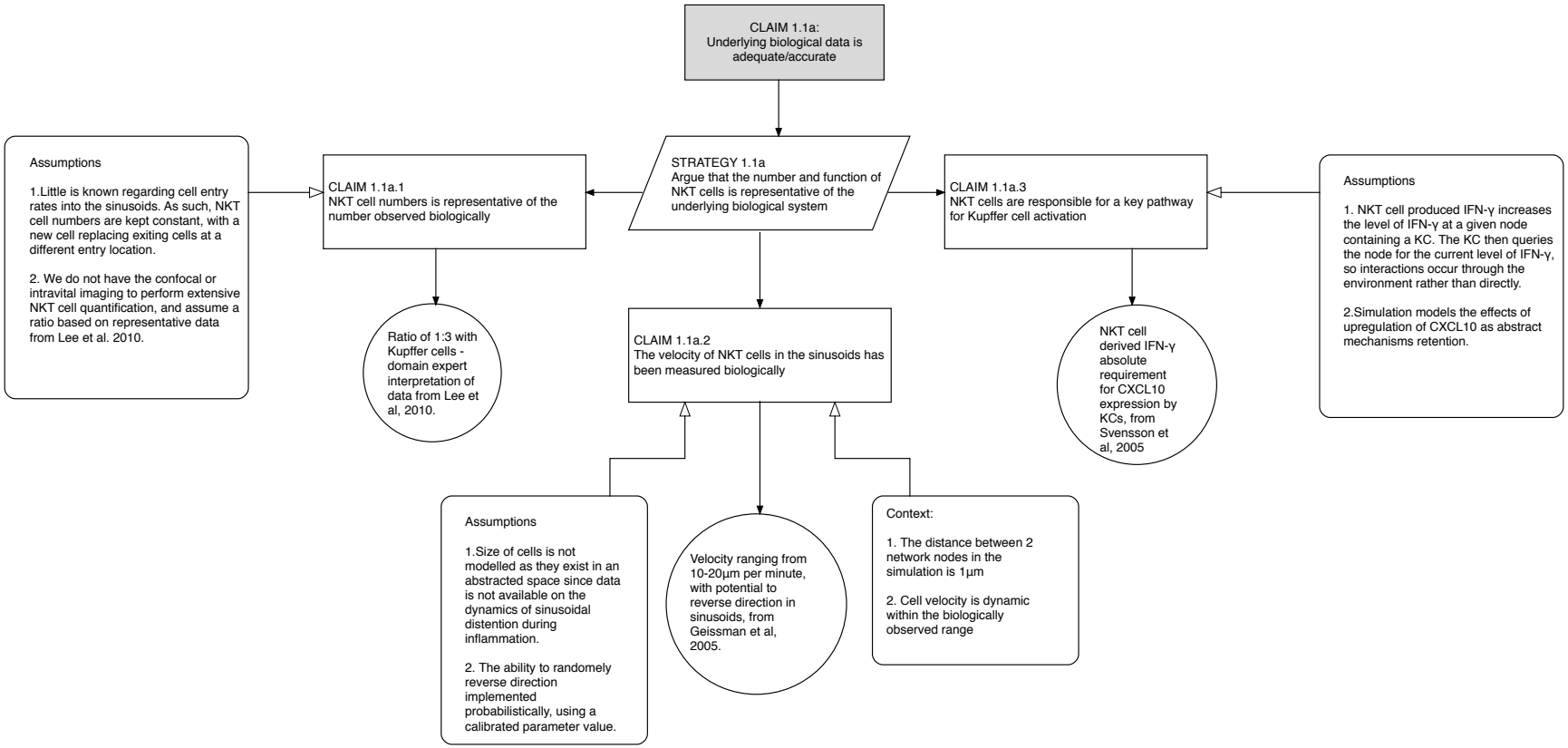


Figure 32: Argument Based Validation: Claim 1.1a. Part a to sub-claim 1.1 argues that the biological data pertaining to the number and function of NKT cells is suitable and adequate for the modelling endeavour.

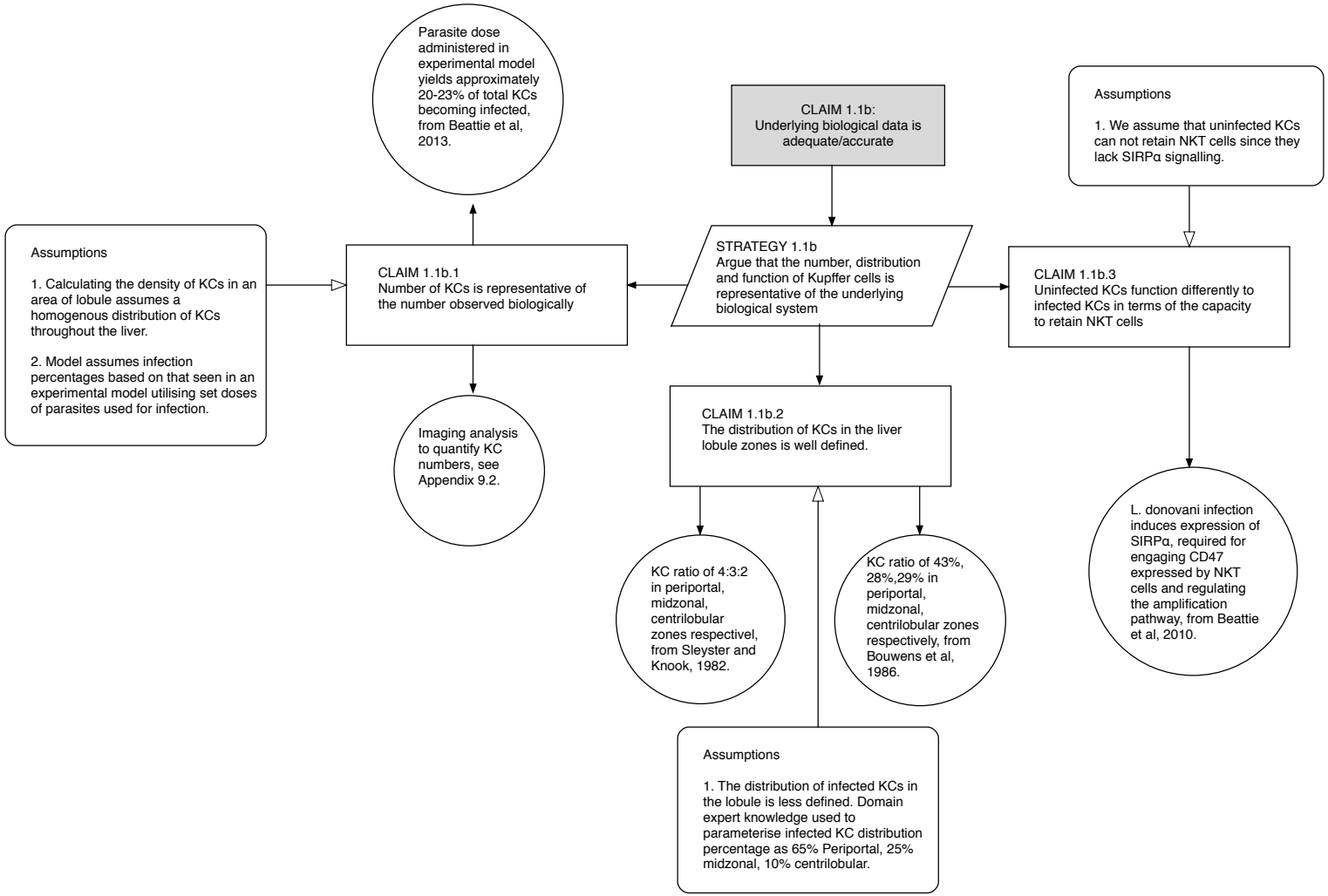


Figure 33: Argument Based Validation: Claim 1.1b. Part b to sub-claim 1.1 argues that the biological data pertaining to the number, distribution and function of Kupffer cells is suitable and adequate for the modelling endeavour.

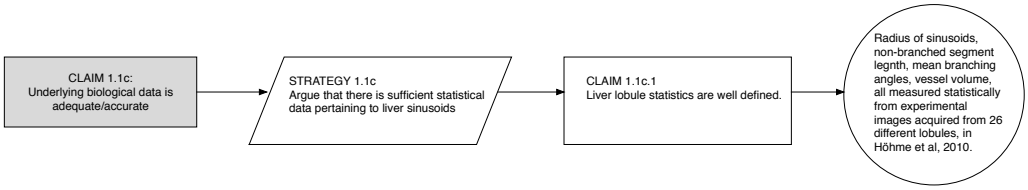


Figure 34: Argument Based Validation: Claim 1.1c. Part d to sub-claim 1.1 argues that the biological data pertaining to the statistics of the liver sinusoids is suitable and adequate for the modelling endeavour.

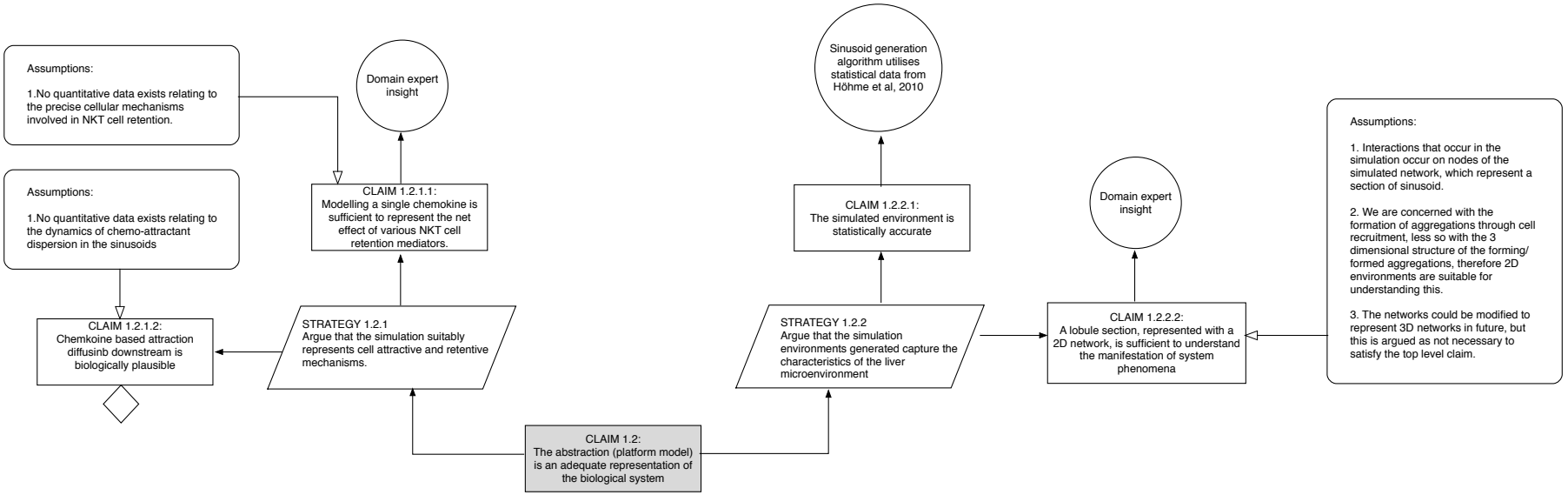


Figure 35: Argument Based Validation: Claim 1.2. Sub-claim 1.2 argues that the platform model adequately captures the biological domain outlined in the domain model. Incomplete or unavailable data to substantiate a claim is denoted using a blank diamond.

5.7 MODEL VERIFICATION

A detailed description of the model implementation is described in Appendix 9.1. Some of the functions and processes as currently implemented are difficult to test formally. Therefore, a mixture of both static and dynamic testing was used to help verify the simulation model.

5.7.1 Static Testing

The predominant method of static testing used to verify the implementation of the model was structured walk-throughs. The behaviour of individual cells, and the logic used to implement those behaviours was described to the domain experts. This activity allowed me as a developed to evaluate the program code, and the domain experts to interrogate the model and ensure that it adheres to their interpretation of both the domain and platform models.

5.7.2 Dynamic Testing

Dynamic testing is important for understanding how the simulation will operate based on a set of given conditions. This allows one to check that the implementation of the computer code performs the desired function [Sargent, 1998]. I now describe two examples of dynamic testing performed on the simulator.

5.7.2.1 *Example: NKT Cell Speed*

To test specific components of the simulator, analysis of output data was performed, to test that certain behaviours were implemented correctly, and that the outputs corresponded to certain input parameters. An example of this is with the validation of NKT cell speed. A method and parameter files were created to test NKT cell speed in the sinusoids. As described in section 5.6.3.2 this should be within the range of 10-20 nodes per minute (60 iterations). The parameters that govern cell movement are used in the move method for NKT cells (listing 13). The simulation does not allow extreme values outside of a $moveMin$ value of 3 and a $moveMax$ value of 6, as this range is the only possible range to manifest the calibrated range for NKT cell speed.

UPPER To parameterise a maximum speed of 20 nodes/m the simulation is parameterised with $moveMin$ and $moveMax$ both equal to 3 (Test 1, table 11). When $moveMin = moveMax$ the number of iterations until the cell moves will always be $moveMin$ iterations i.e. 3. If the cell moves every 3 iterations the average speed will always be 20 nodes/m (table 10).

LOWER Oppositely, for a maximum speed of 10 nodes/m the simulation is parameterised with $moveMin$ and $moveMax$ both equal to 6 (Test 2, table 11). When $moveMin = moveMax$ the number of iterations until the cell moves will still always be $moveMin$ iterations i.e. 6. If the cell moves every 3 iterations the average speed will always be 10 nodes/m (table 10).

Test	numUninfected	numGoodKCs	numNKTs	turnProb	...	moveMin	moveMax	...
1	0	0	1	0.005	...	3	3	...
2	0	0	1	0.005	...	6	6	...
3	0	0	1	0.005	...	3	6	...

Table 9: NKT Speed Test CSV File.

Test Parameter Values		Speed	
moveMin	moveMax	expected range	calculated average
3	3	20 nodes per min	20
6	6	10 nodes per min	10
3	6	10-20 nodes per min	14⩽nodes⩽16 per min

Table 10: Sample NKT Speed Test Results.

5.7.2.2 *Example: Chemokine Diffusion*

Another example of model testing is the testing of chemokine diffusion dynamics. Chemokine diffusion plays an integral role in influencing NKT cell movement throughout the environment, and it is important to ensure that the area of influence of chemokine fits the constraints defined by the minimum and maximum diffusion distance parameters.

Chemokine diffusion was tested on both the algorithmically generated structures, and also on simple test structures such as that in figure 36. Simulation parameter files were created to explore different extremes for chemokine diffusion minimum and maximum distances. A sample of the values explored a during testing are shown in table 12. Extreme values were also tested, such as negative values which raise errors and inform the user that such occurrences are invalid. Maximum diffusion distance is unlimited, as the chemokine will diffuse to the drain node in any case where the value is positively extreme. In situations where the maximum diffusion distance exceeds the minimum diffusion distance this also raises an informative error for the user.

As figure 36(**a**) illustrates for sample test 1 (table 11, minimum diffusion starts at 2 nodes, and by simulations end after having been activated and allowed to reach maximum attractive value at the source KC, the chemokine has diffused the parameterised maximum 6 nodes (figure 36(**b**)). Similarly, for one of the algorithmically generated structures, sample test 2 shows similar behaviour in a more realistic simulation setting, where several cause chemokine to engulf large swathes of the sinusoids and drain to the central node (figure 36(**c**,**d**)), as to be expected with a parameterised maximum distance of 500 - a value greater than the distance between any edge and the centre.

Test	numUninfected	numGoodKCs	numNKTs	turnProb	...	chemoDist	maxDist	...
1	0	1	8	0.005	...	2	6	...
2	113	33	46	0.005	...	20	500	...
n	...	...	...	...	...	...	...	...

Table 11: Chemokine Diffusion Test CSV File.

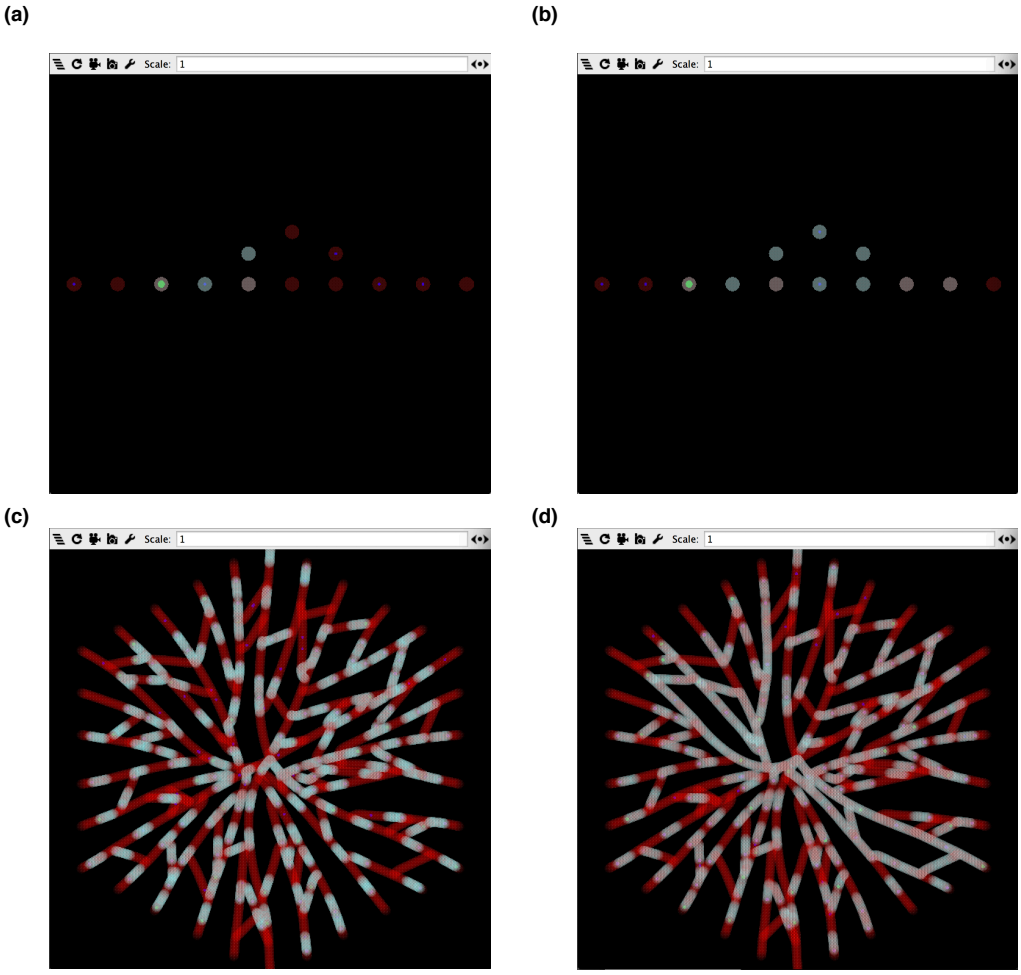


Figure 36: Chemokine Diffusion Test. Chemokine diffusion, illustrated by nodes highlighted in teal, at **(a)** t(0) and **(b)** t(24) hours of simulated time on the simple test structure. **(c)** t(0) and **(d)** t(24) hours of simulation time on a sample algorithmically generated structure.

Test Parameter Values			
minimum	maximum	structure	diffusion
0	1	simple test	*within constraints*
0	2	simple test	*within constraints*
...	...	...	...
2	6	simple test	*within constraints*
...	...	...	...
0	500	algorithmically generated	*within constraints*
20	500	algorithmically generated	*within constraints*

Table 12: Sample Chemokine Diffusion Test Results.

5.8 SUMMARY

In this chapter, I justified the use of the agent-based modelling paradigm to fulfill the domain model presented in chapter 4. I presented an algorithm for generating artificial liver microenvironments, and described how I implement the domain model conceptualisation of cells and mechanisms of interactions into logic for coding. The MASON agent-based modelling platform was chosen as a development platform, and I describe the software development process adopted, and describe the simulator architecture. Finally, the process of model validation and verification, through a variety of activities such as parameter calibration and argumentation, code walk-throughs, trace analysis, and a mixture of static and dynamic testing was outlined.

The next chapter comprises the first part of my CoSMoS results model, presenting results from statistical analyses of simulation output, and an initial round of experimentation to answer research questions relating to the role of liver structure in the formation of cellular aggregations.

Part III

PREDICTIVE MODELLING AND SIMULATION

6 RESULTS MODEL

The output from the simulation platform constitutes what CoSMoS defines as the results model. The results model comprises a wide variety of products including but not limited to: all of the simulation output data, both processed and unprocessed; any detail inferred through observation and statistical analysis; also any interpretations of the simulation output with respect to the underlying domain. The results model should ideally provide valuable insight into the domain to identify novel avenues for future experimentation and research, or perhaps even provide evidence to answer outstanding questions about the underlying domain.

This chapter comprises only part of the CoSMoS results model, setting up the foundation for interpreting simulation results and exploring issues of simulation environment and probabilistic retention mechanisms. The main body of simulation results is contained in chapter 7.

I will begin by introducing the core simulation output metrics, and present results from various statistical analyses aimed at understanding how stochasticity (section 6.1.2) and parameter perturbation (section 6.1.3) influence simulation output, these sections follow on from the model validation and parameterisation section discussed in the previous chapter (section 5.6). I will then introduce the first round of experimentation, with results to quantify how structure influences simulation output (section 6.2), followed by a factorial analysis of KC retention of NKT cells (section 6.3).

6.1 STATISTICAL ANALYSIS OF SIMULATION OUTPUT

Statistical analysis is required to understand how the simulation adequately captures the underlying biology, and how simulation parameters might be influencing the data. Many established statistical tests make assumptions about the underlying structure of the data (such as assuming normality). Non-parametric statistical methods can be used to provide a statistical metric that is robust, as the methods often make few or no assumptions about the underlying data [Whitley et al., 2002]. This section outlines two statistical techniques used in the field to understand both a simulations robustness to parameter perturbation (section 6.1.3), and to elucidate how stochasticity, inherent in agent-based simulations, influences simulation output (section 6.1.2).

6.1.1 Defining Simulation Output Metrics

In order to utilise a simulation and gather any data, it is important to create simulation output metrics that allow a user to both interrogate and interpret the

various states that a simulation can manifest. Chapter 4 highlighted several high-level observable phenomena in from the underlying biological system of interest (see section 4.1). The act of modelling cells as agents means that the simulation is capable of outputting a variety of metrics relating to individual cell behaviours. The metrics I have devised revolve around the observable phenomenon I defined in the domain model - the heterogeneity in the formation of cellular aggregations in the liver. Since I hypothesise that this heterogeneity can be explained by early interactions between NKT and Kupffer cells, the metrics focus on the size of cellular aggregations in the simulation and the cumulative stimulation time received by KCs from NKT cells. Lastly, the dynamics of NKT cell migration in the sinusoids is captured by a metric for NKT cell time spent in the sinusoids and not interacting with KCs. The simulation metrics are summarised as follows:

PERCENTAGE INFLAMMATORY FOCI For a given simulation, this metric describes the percentage of infected KCs in a simulation that have a significant inflammatory focus, qualified at 4, 6 or 8 cells.

NKT HOURS The stimulation time received by individual Kupffer cells is cumulative and measured in NKT hours. Let us assume a single NKT cell stimulates a KC for an hour of simulated time, 3600 simulation iterations. In this case, the KC will be receiving 1 hour of stimulation per hour (1 NKT hour). Now let us assume that another KC has 3 NKT cells stimulating it, that KC is receiving 3 hours worth of stimulation per hour (3 NKT hours). This metric is illustrated in figure 37.

NKT CELL WANDER/PATROL TIME This metric measures the time spent by NKT cells patrolling the sinusoids and not interacting with KCs.

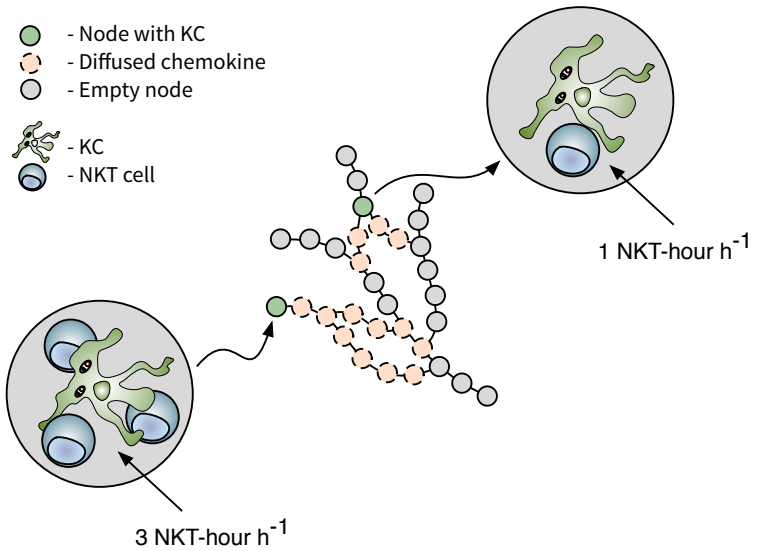


Figure 37: Cumulative Stimulation Time Metric - NKT Hours.

6.1.2 Aleatory Analysis

Simulations make use of seeds to initialise a random number generator that can be used to determine the outcomes of probabilistic events. In terms of simulation reproducibility it is important to keep a record of these seeds should one wish to replicate a set of simulation results, or should one wish to determine the outcome of different simulation scenarios whilst controlling for the stochastic processes underlying those experiments.

Aleatory uncertainty analysis is a technique that can be used to mitigate the effects of stochasticity on simulation output measures. Stochastic simulations will produce varying results when run with the identical initial parameters, and this stochasticity must be quantified and mitigated. Therefore, to calculate the minimum number of replicate simulation runs needed to generate results that cover a representative spectrum of possible system behaviours, statistical techniques need be used, such as those described by Alden et al. [2013]; Read et al. [2011].

Let us consider 20 sample subsets, and for each subset I run the simulation n times where $n \in \{10,50,100,300,500\}$, though a different set of sample sizes may be required. Median values are calculated for each simulation output measure across the runs, then subsequently a median calculated across the 20 subsets. By comparing 20 subsets of 10 runs against 20 subset of 50 and so forth, using a non-parametric statistical test (Vargha and Delaney [2000] A-Test), statistical significance can be determined between each subset.

Figure 38(**a**) illustrates the comparison between simulation output measures for each sample of 10 runs. Simulation results demonstrate that given an experiment performed with 10 simulation runs, a repeat simulation experiment (different subset) would produce large variation in all output measures. Therefore, any conclusions or predictions made from a simulation experiment with only 10 replicates are invalid. Figure 38(**a**)-(**e**) illustrates the same aleatory analysis for samples sizes 50, 100, 300 and 500.

From the summary figure 38(**f**) I can conclude that 300 simulation runs per parameter combination are enough to have acceptable uncertainty (small variance between identical sample subsets). Whilst statistically 300 simulation runs captures enough of the stochasticity to produce a set of representative results, a larger set of runs will further converge the A-test effect score on 0.5 (no difference between subsets and capturing all of the stochasticity). A balance must be struck between increasing the sample size and also maintaining tractable simulated-experiment execution times. Testing concluded that 500 simulation runs did not significantly increase execution times, and yet further converged the A-test scores for 2 out of our 3 measures. Therefore, 500 simulation runs were chosen for all of the simulated-experiments presented in this thesis.

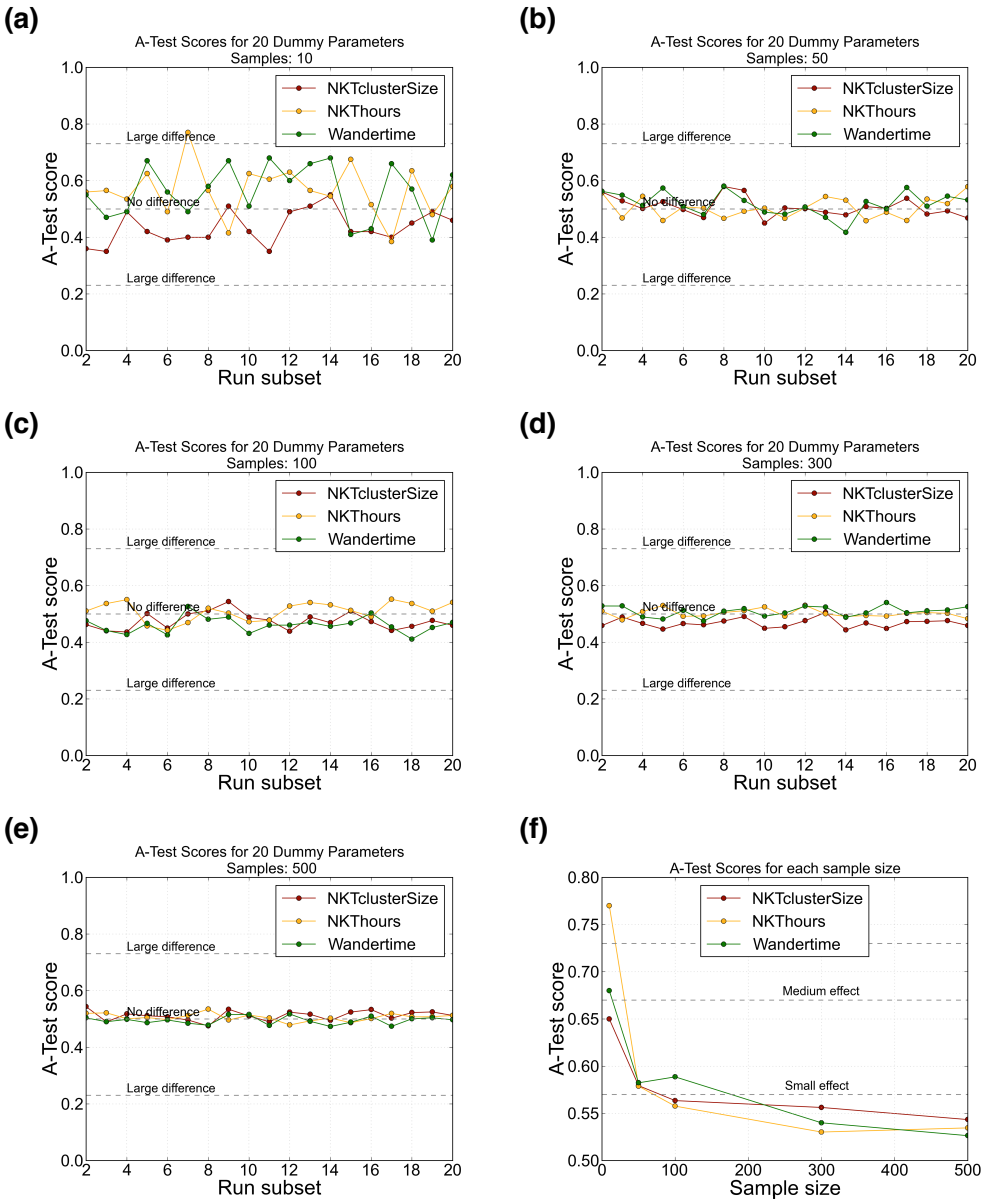


Figure 38: Aleatory Analysis Determines a Minimum of 300 Simulation Runs per Experiment. A-test (Vargha and Delaney [2000]) scores for three simulation output measures across sample sizes of 10 **(a)**, 50 **(b)**, 100 **(c)**, 300 **(d)** and 500 **(e)**, illustrating that greater than 300 simulation runs are needed to capture the variation in output results and mitigate uncertainty, summarised in **(f)**.

6.1.3 Latin-HyperCube Sensitivity Analysis

The computer simulation I have constructed contains multiple parameters with unknown values, or those with no specific biological equivalent, each responsible for different model components. Invariably, these parameters have an influence on the various responses observed by analysing the simulation output measures. In order to quantify simulation responses that result from specific variation in both single and multiple parameter values, a global parameter analysis is required. Statistical approaches such as Latin hypercube sampling (LHS), proposed by McKay et al. [1979], can be used to generate sets of parameter values within pre-defined value ranges, that allow one to determine the global sensitivity of a computer simulation to multivariate parameter perturbations [Marino et al., 2008; Read, 2011; Alden, 2012] (figure 39). Parameters that have compound effects on simulation output measures can then be identified, and if so required, the effect of those parameters could be mitigated.

Latin hypercube sampling techniques are available in a variety of computer programs and statistical tools, and this functionality is also present in a toolkit called *Spartan* - **S**imulation **P**arameter **A**nalysis **R** **T**oolkit **A**pplication [Alden et al., 2013]. Spartan is an R package with functionality to perform various analyses to quantify uncertainty and sensitivity in simulations, including but not limited to LHS sensitivity analysis and aleatory analysis (see section 6.1.2)).

An LHS analysis was performed for all of the unknown simulation parameters. Figures 40 to 42 provide various Partial Rank Correlation Coeffcients (PRCCoeffs) that highlight any relationships when analysing multiple data points from the simulation experiments using different sets of values for unknown parameters, sampled using LHS.

Analysing the influence of multiple parameter combinations on the formation of cellular aggregations reveals that only two parameters have statistically significant correlations, namely *leaveProb* (figure 40(**a**)) and *escapeItns* (figure 40(**b**)), both are negatively correlated. Whilst the parameters governing the production of stimulatory cytokine (*chemoIFN*) and stimulation required to reach maximal retention (*chemoRetain*) are both statistically insignificant when correlated with aggregation size, there is evidently a relationship in both cases (*chemoIFN* PRCCoeff=-0.15 figure 40(**c**), *chemoRetain* PRCCoeff=-0.17 figure 40(**f**)).

Parameters which most influence stimulation time received by KCs (*NKThours*) are *leaveProb* and *escapeItns*, both with strong negative correlations (figure 41(**a**) and figure 41(**b**) respectively). Several other parameters also approach significance: *chemoDist* (positive correlation figure 41(**g**)) and *anergicItns* (negative correlation figure 41(**j**)).

The NKT cell patrol or wandertime metric, as expected, sees positive correlation with both *leaveProb* and *escapeItns*. Similar to NKThours several parameters approach significance: *chemoDist* (negative correlation figure 42(**g**)) and *anergicItns* (positive correlation figure 42(**j**)).

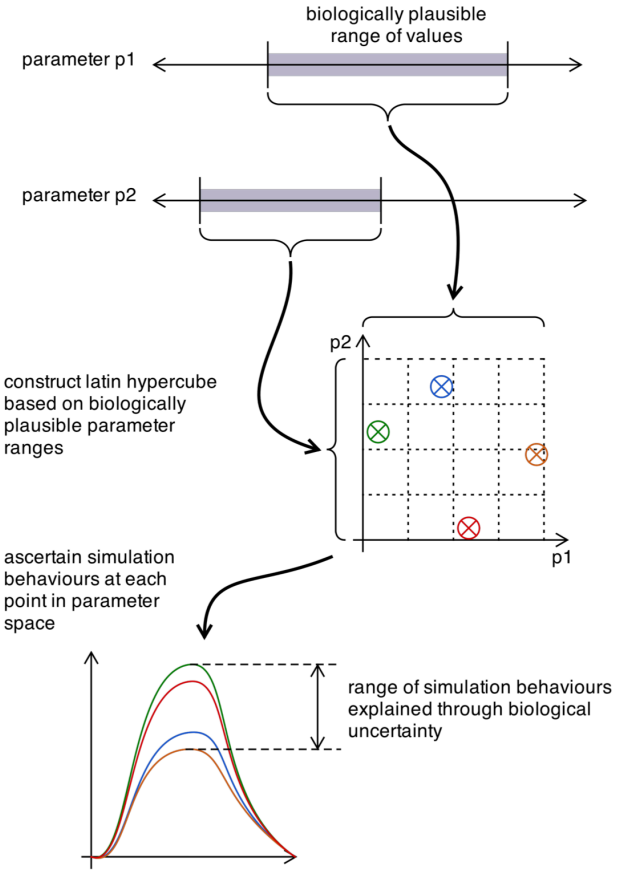


Figure 39: Latin-Hypercube Sampling of Parameter Ranges. Figure from [Read, 2011, Figure. 6.23].

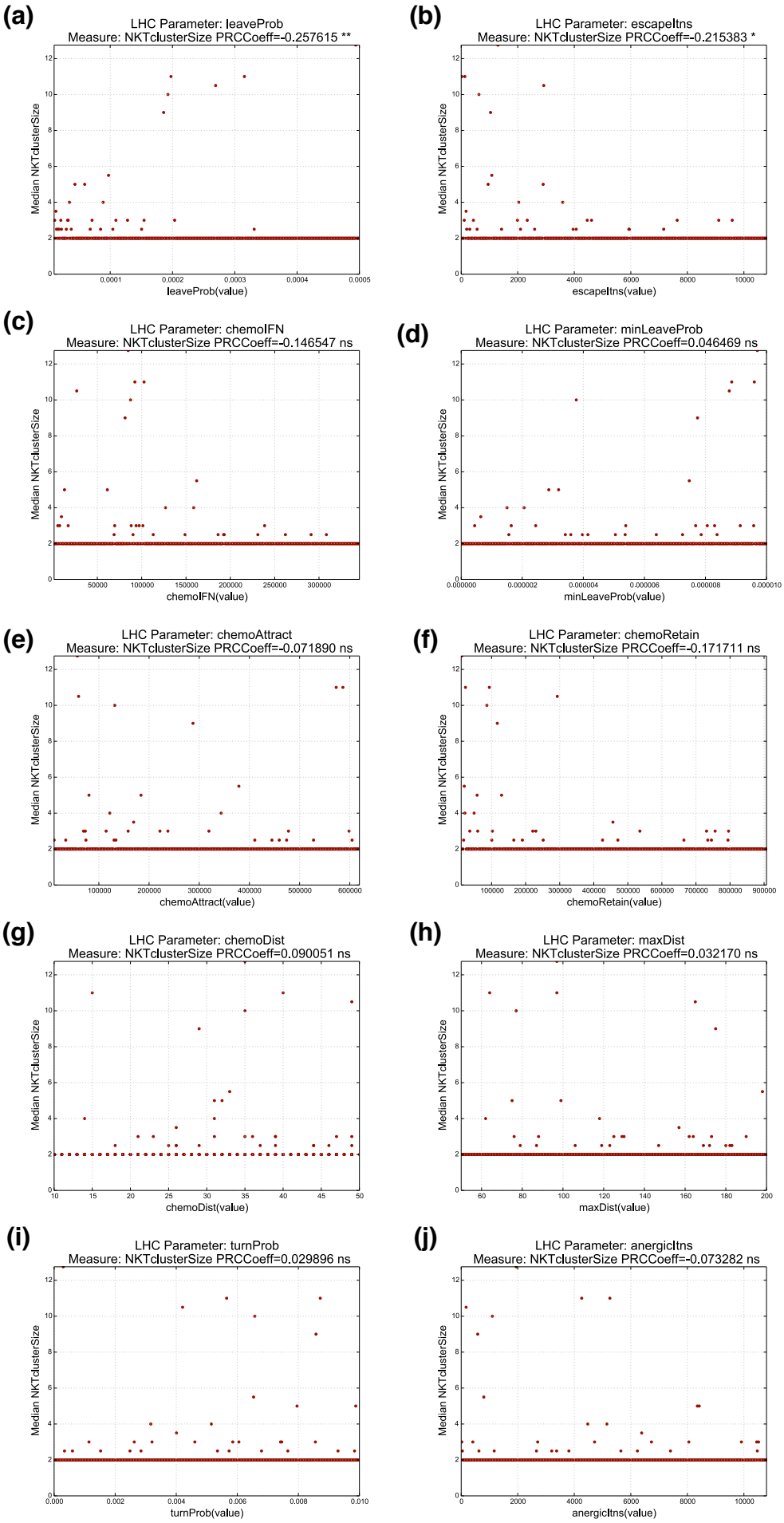


Figure 40: LHS Analysis - Clusters/Aggregations.

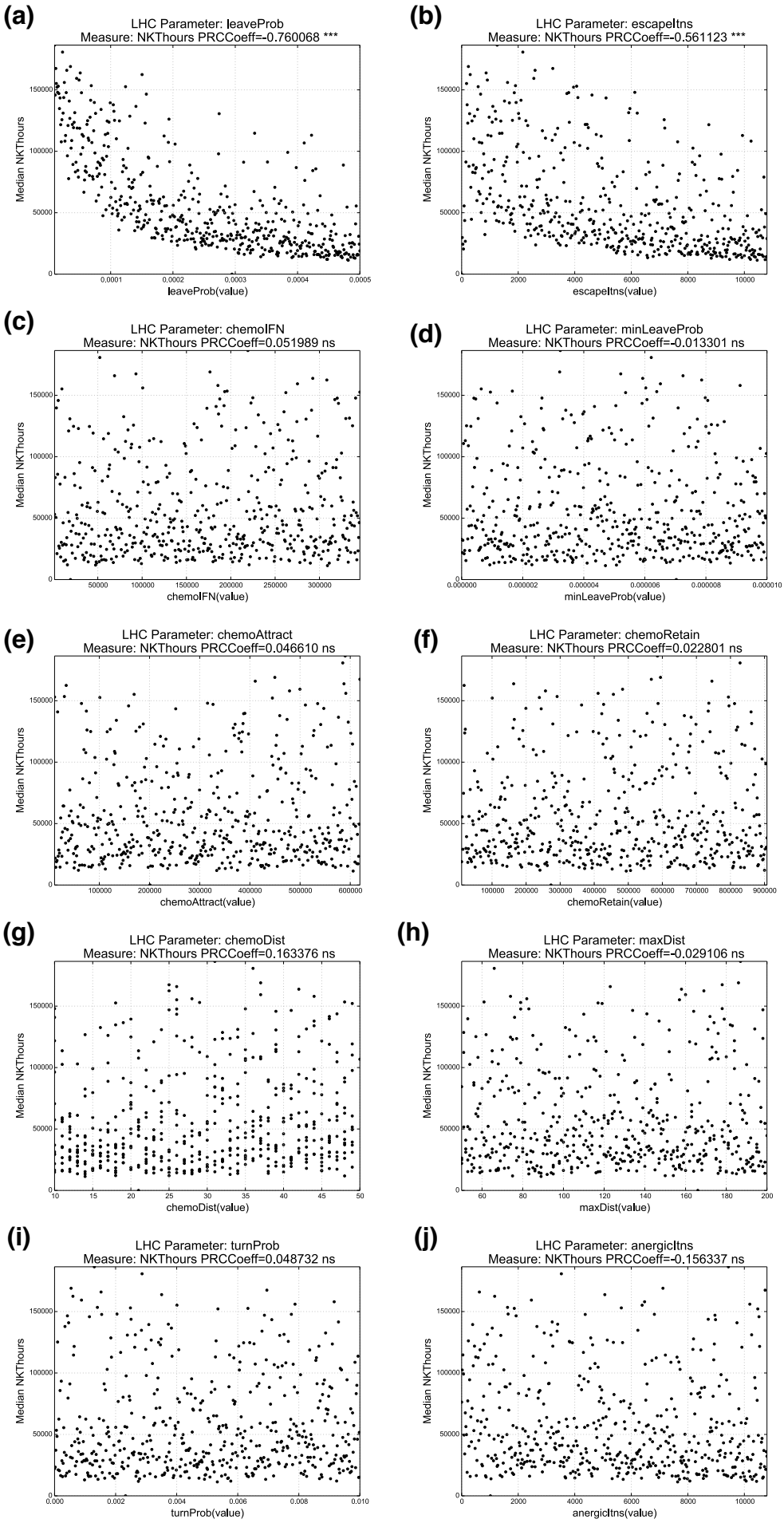


Figure 41: LHS Analysis - KC Stimulation Time.

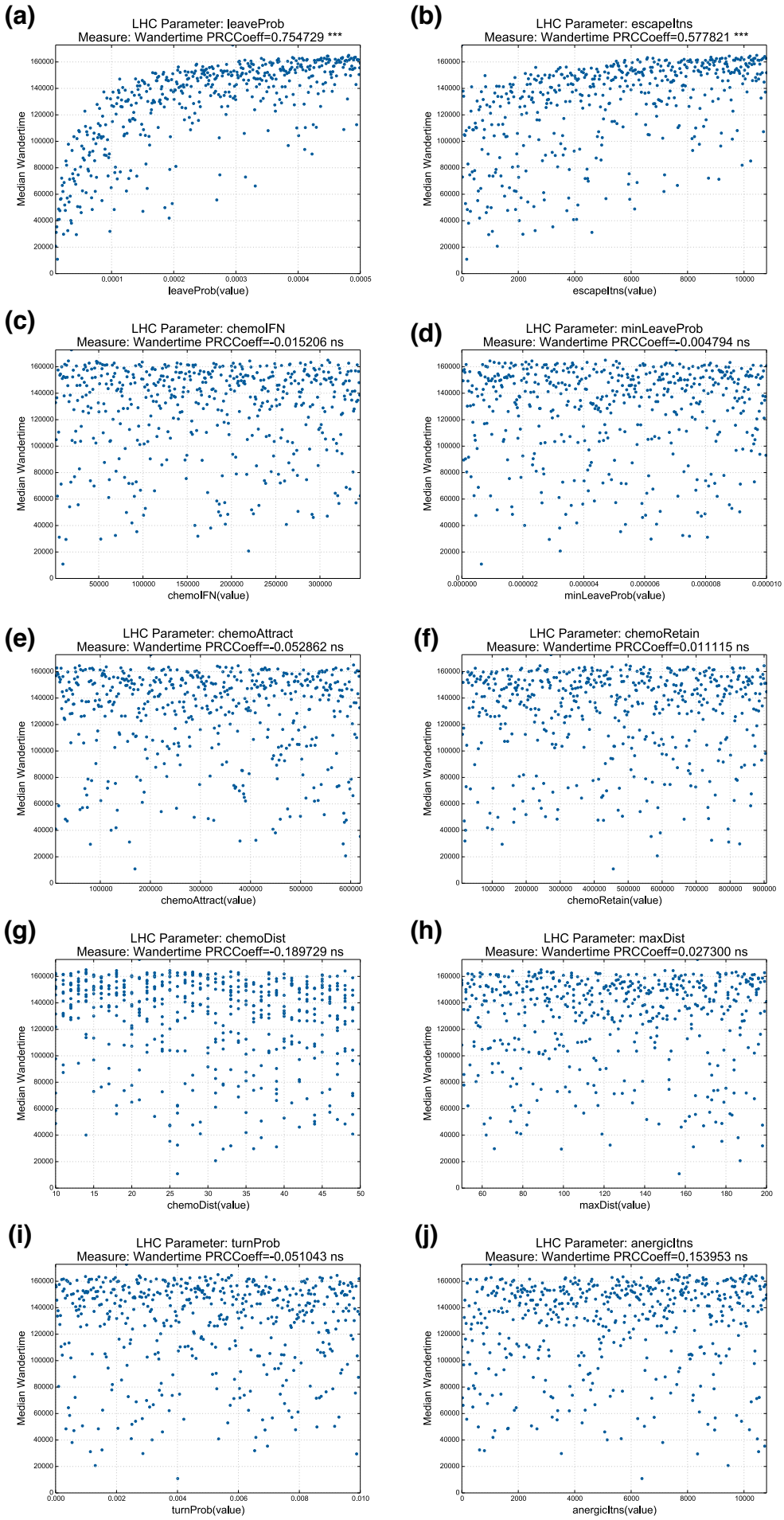


Figure 42: LHS Analysis - NKT Cell Patrol Time.

6.2 VARIANCE BETWEEN MULTIPLE STRUCTURES

This section addresses research questions 2 and 3, and forms the basis of thesis contribution 2, the determination of the role of the sinusoidal microenvironment in influencing inflammation. This section will also address the role of KC spatial location, by modifying the distribution of KCs throughout the liver lobule.

6.2.1 The Role of Sinusoid Structure

The objective of this simulation experiment was to quantify any differences in simulation results across multiple, computationally generated liver lobule morphologies. Firstly, 10 statistically and visually similar lobule sections (structures) were generated (figure 43). The structures were then used as environments for the baseline simulation, with 500 simulation runs performed on each (as per section 6.1.2).

Figure 44 illustrates simulation data when comparing the size of NKT cell clusters at infected KC locations versus the distances from the CV of those inflammatory foci. The pearson product-moment correlation coefficient was calculated to determine any linear relationship between the two variables across each structure (figure 44**(a)-(j)**), and data showed a modest relationship between distance from CV and cluster size (significant with N>1000, true N=16500). Variance between structures was then calculated by first collating, for each structure, the distribution of median distances for each cluster size (figure 44**(k)**), then comparing them with the Kolmogorov-Smirnov (KS) test of distribution equality (non-parametric). Results show that there is no statistically significant difference when comparing results across any structure, with the exception of a comparison between structures 8 and 6 (table 13). The Vargha-Delaney A-test [Vargha and Delaney, 2000] was then used to quantify the effect size of the difference and results show a large difference in effect size (table 14). Whilst the statistics demonstrate that there is a significant effect between the comparison of structures 8 and 6, from a practical perspective both have similar correlation coefficients; therefore, neither structure has a significantly greater, or lesser, influence on the relationship between cluster size and cluster distance from CV.

The data also showed that the larger spectrum of NKT clusters (inflammatory foci) are found further away from the central vein, at the peripheries of the liver lobule. Whilst there is a biologically measurable skewed distribution of KCs at the periphery of the lobule [Bouwens et al., 1986; Sleyster and Knook, 1982], large inflammatory foci predominantly at the peripheries may seem intuitive; however, what the data show is that there are few if any outliers in the midzonal or centrilobular regions that form large clusters. Alongside the predominantly periportal distribution of KCs, the imposed directionality of NKT cells entering the lobule sections from the peripheries demonstrates that larger clusters are more likely to be found at the peripheries. From a high level perspective, the liver microenvironment may have evolved to take advantage of this optimal means of KC uptake of pathogen leading to KC activation by early promoter cells.

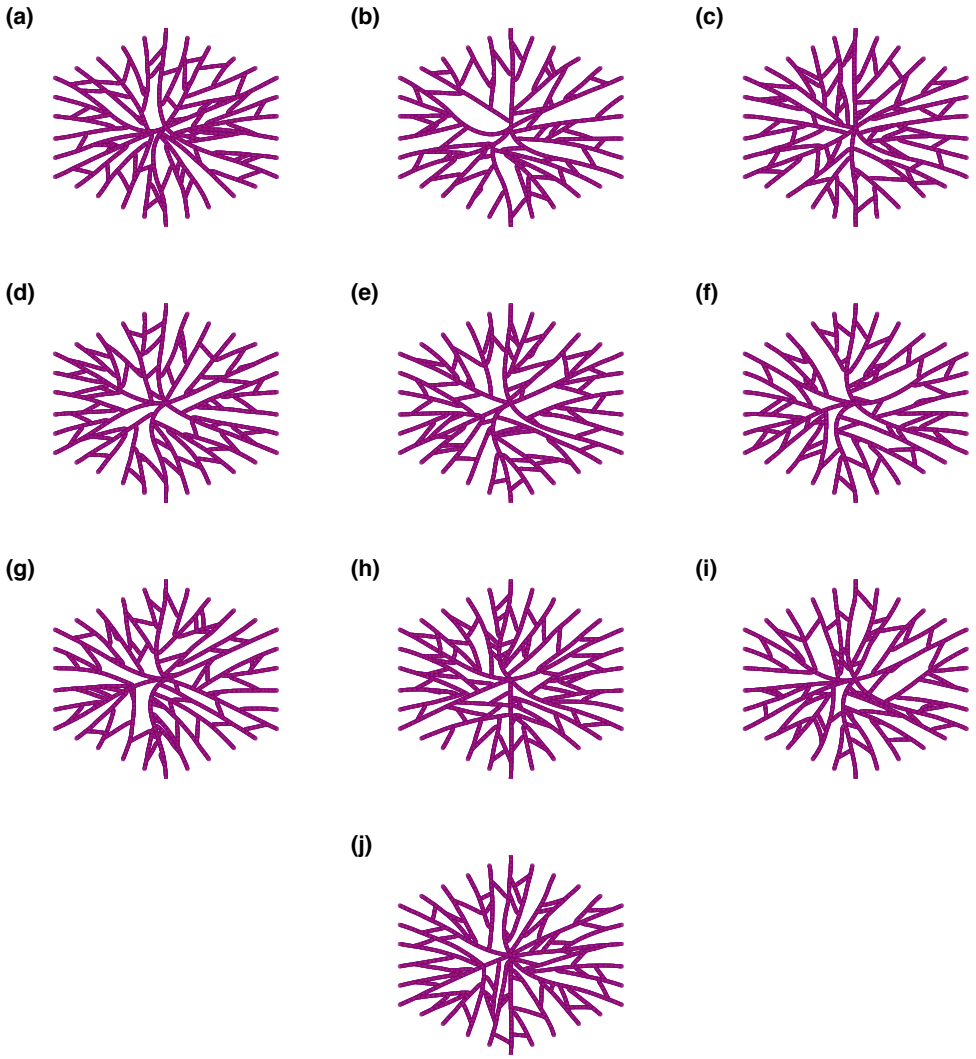


Figure 43: Compared Structures. Statistically and visually similar structures were generated to determine variance in simulation results between each (Structures 1-10 (a)-(j)).

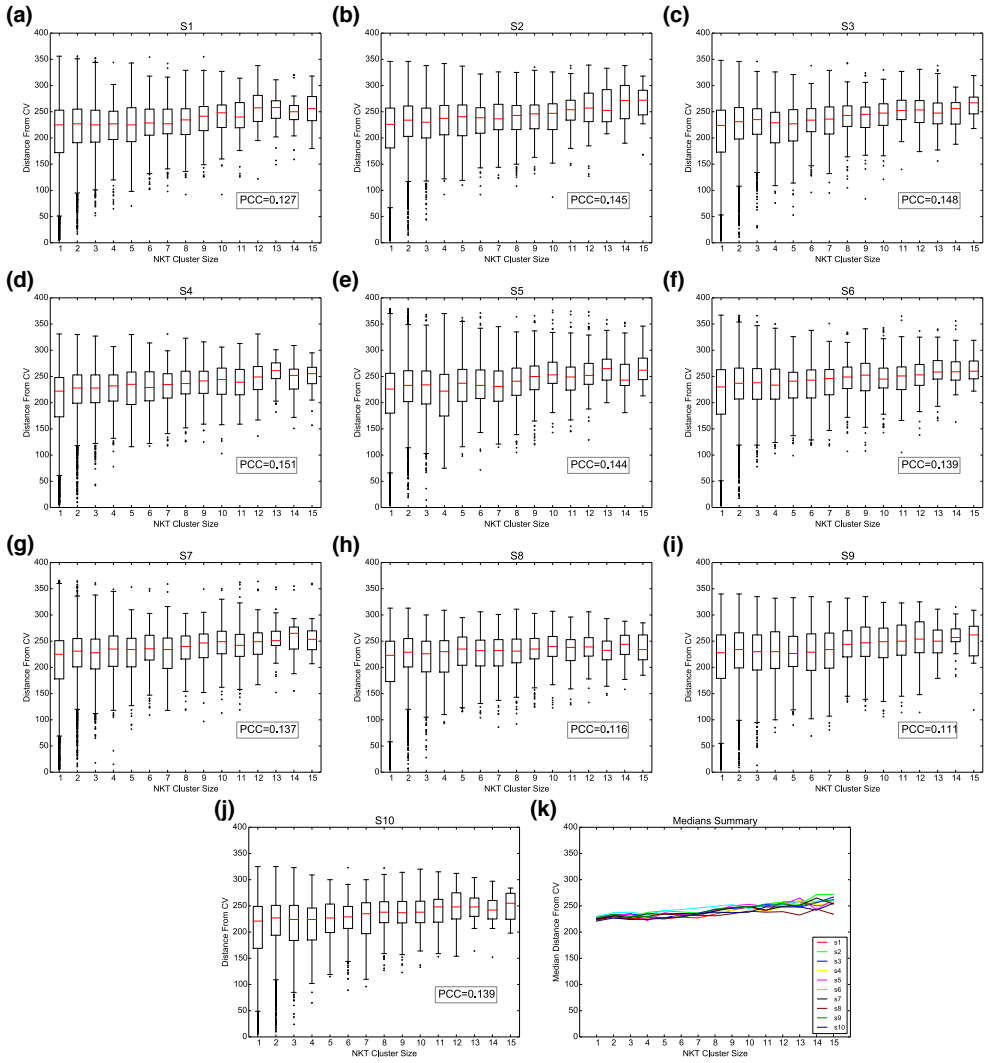


Figure 44: Cluster Size vs. Distance to CV. For 10 structures I compare the distribution of NKT cell cluster sizes (x-axis) versus the distance of those cluster sizes from the central vein (CV)(y-axis). Box and whisker plots depict the median and upper and lower quartiles. Structures 1-10 ((**a**)-(**j**)), data shows that larger cluster sizes are found further from the central vein (with Pearson correlation coefficients, N=16500 per structure experiment). (**k**), Comparison of median distances for each cluster size, see also table 13 for Kolmogorov-Smirnov statistics.

	1	2	3	4	5	6	7	8	9	10
1	**1**									
2	0.135863964	**1**								
3	0.307935289	0.588614411	**1**							
4	0.307935289	0.588614411	0.889899861	**1**						
5	0.307935289	0.889899861	0.998320005	0.998320005	**1**					
6	0.051467307	0.889899861	0.307935289	0.135863964	0.588614411	**1**				
7	0.307935289	0.588614411	0.998320005	0.998320005	0.998320005	0.307935289	**1**			
8	0.307935289	0.016786481	0.051467307	0.307935289	0.051467307	*0.001140698*	0.135863964	**1**		
9	0.307935289	0.588614411	0.998320005	0.889899861	0.889899861	0.307935289	0.889899861	0.051467307	**1**	
10	0.889899861	0.307935289	0.588614411	0.588614411	0.307935289	0.135863964	0.588614411	0.588614411	0.307935289	**1**

Table 13: Structures Kolmogorov-Smirnov Test Summaries (Validated KC Distribution). Kolmogorov-Smirnov statistics summary when comparing median distances vs NKT cell cluster size for biologically measured KC distributions (see figure 44).

	1	2	3	4	5	6	7	8	9	10
1	**0.5**									
2	0.653333333	**0.5**								
3	0.591111111	0.42	**0.5**							
4	0.564444444	0.357777778	0.44	**0.5**						
5	0.597777778	0.42	0.495555556	0.562222222	**0.5**					
6	0.715555556	0.551111111	0.624444444	0.7	0.615555556	**0.5**				
7	0.595555556	0.4	0.486666667	0.548888889	0.488888889	0.353333333	**0.5**			
8	0.455555556	0.197777778	0.295555556	0.362222222	0.291111111	*0.124444444*	0.275555556	**0.5**		
9	0.613333333	0.424444444	0.495555556	0.548888889	0.484444444	0.375555556	0.5	0.666666667	**0.5**	
10	0.451111111	0.306666667	0.382222222	0.417777778	0.366666667	0.233333333	0.368888889	0.562222222	0.353333333	**0.5**

Table 14: Structures A-Test Summaries (Validated KC Distribution). Structures A-Test Summaries for biologically measured KC distributions. Vargha-Delaney A-Test [Vargha and Delaney, 2000] effect sizes for simulation results. Effect sizes qualified at: no difference ($0.44 \leqslant a \leqslant 0.56$), small difference ($0.56 \leqslant a \leqslant 0.64$ or $0.36 \leqslant a \leqslant 0.44$), medium difference ($0.64 \leqslant a \leqslant 0.71$ or $0.29 \leqslant a \leqslant 0.36$), large difference ($a \geqslant 0.71$ or $a \leqslant 0.29$). Only sample comparisons that were significantly different when compared with the KS-test (table 13) are relevant, and those data are highlighted where they have a large effect size.

6.2.2 Altering the Distribution of KCs

After quantifying the role of sinusoid structure and determining the distributions of infected foci distances from the CV using the distribution of KCs published in the literature, I performed a simulation experiment to determine if a more even distribution of KCs throughout the lobule section would produce a similar pattern of foci distances. I hypothesised that:

An even distribution of KCs throughout the sinusoids will promote a more evenly distributed population of significant inflammatory foci.

Using the same structures as the previous simulation experiment (figure 43), pearson product-moment correlation coefficients were calculated. Across all structures coeffcient values were increased in comparison to the biologically validated KC distribution experiment (figure 45**(a)-(j)**). These simulation results show that there is a greater relationship between cluster size and mean distances of inflammatory foci from the central vein; this result was not intuitive. The expected result according to my hypothesis would have been for a decreased value for the correlation coefficient since the KCs are now evenly distributed between the three lobule regions. However, as in the previous simulation experiment, new cells entering the environment do so from the peripheries, making it more likely that they encounter peripheral KCs before they have the opportunity to navigate towards centrilobular KCs.

The distribution of medians distances across all structures was compared using the KS-test between the biologically validated KC distribution and the evenly distributed KC experiment (figures 44 and 45**(k)**). Results yield no significant differences between same structures with different KC distributions (table 17), despite there being a slight visible decrease in median distances for cluster sizes of 1-3 (figure 45**(k)**). For what I would consider as more significant inflammatory foci sizes of greater than 4 cells, the lack of significance is more pronounced when comparing the two simulation experiments (table 18).

As previously, I also compared each structure against the others using the KS-test, and I observed no significant differences for any pairing (table 15). A-test effect sizes are shown in table 16; however, as there were no significant differences in simulation results across structures, these effect sizes are redundant.

From these data I can conclude that an even spatial distribution of KCs does not promote a more evenly distributed population of significant inflammatory foci. I can also reaffirm that the biggest influencing factor on the formation of inflammatory foci is the imposed influx of cells from the peripheries.

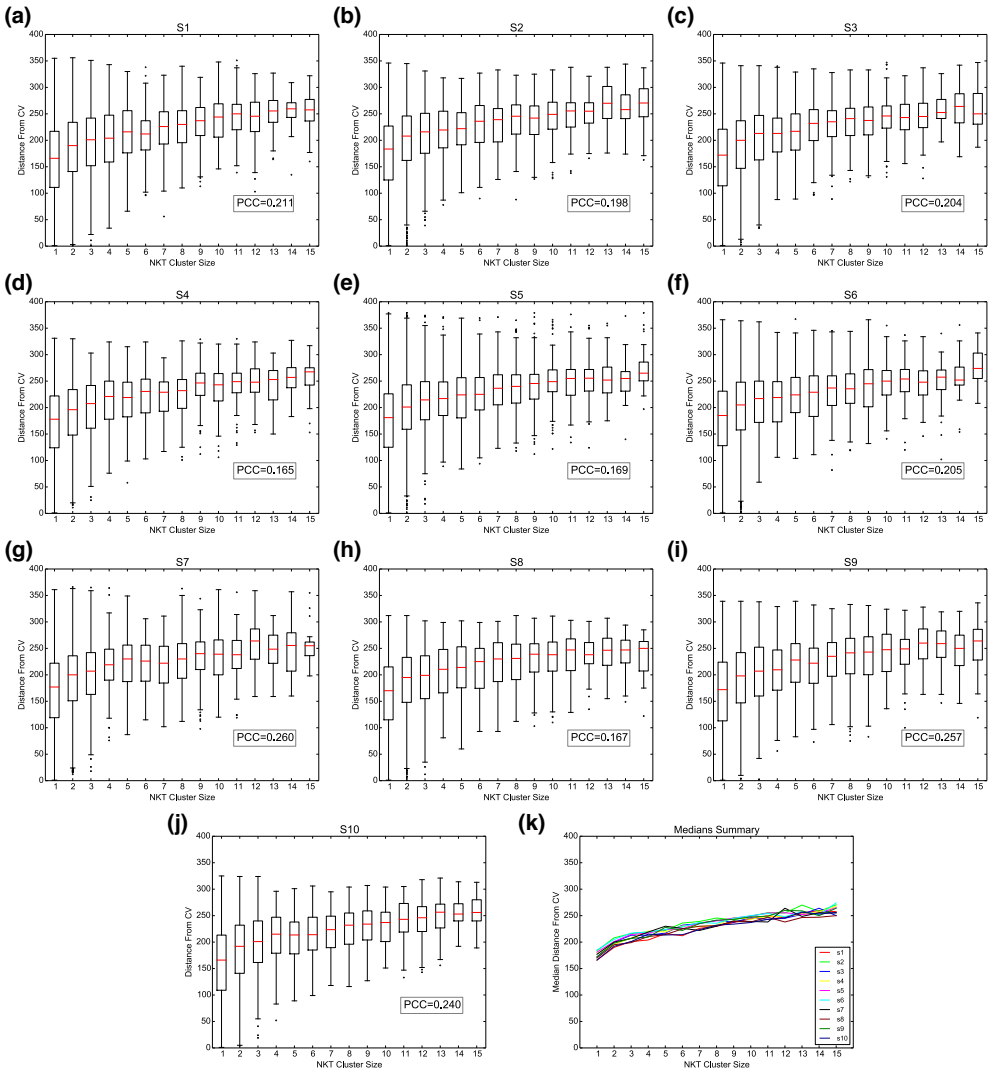


Figure 45: Cluster Size vs. Distance to CV (Even KC Distribution). For 10 structures I compare the distribution of NKT cell cluster sizes (x-axis) versus the distance of those cluster sizes from the central vein (CV)(y-axis) with evenly distributed Kupffer cells. Box and whisker plots depict the median and upper and lower quartiles. Structures 1-10 ((**a**)-(**j**)), data shows that larger cluster sizes are found further from the central vein (with Pearson correlation coefficients, N=16500 per structure experiment). (**k**), Comparison of median distances for each cluster size, see also table 15 for Kolmogorov-Smirnov statistics.

	1	2	3	4	5	6	7	8	9	10
1	**1**									
2	0.889899861	**1**								
3	0.889899861	0.588614411	**1**							
4	0.889899861	0.889899861	0.889899861	**1**						
5	0.889899861	0.998320005	0.889899861	0.998320005	**1**					
6	0.588614411	0.889899861	0.889899861	0.998320005	0.998320005	**1**				
7	0.889899861	0.588614411	0.889899861	0.889899861	0.889899861	0.889899861	**1**			
8	0.889899861	0.307935289	0.889899861	0.588614411	0.307935289	0.307935289	0.889899861	**1**		
9	0.998320005	0.998320005	0.889899861	0.998320005	0.998320005	0.998320005	0.588614411	0.307935289	**1**	
10	0.998320005	0.588614411	0.889899861	0.889899861	0.889899861	0.588614411	0.889899861	0.889899861	0.889899861	**1**

Table 15: Structures Kolmogorov-Smirnov Test Summaries (Even KC Distribution). Kolmogorov-Smirnov statistics summary when comparing median distances vs NKT cell cluster size for evenly distributed KCs (see figure 45).

	1	2	3	4	5	6	7	8	9	10
1	**0.5**									
2	0.615555556	**0.5**								
3	0.54	0.395555556	**0.5**							
4	0.551111111	0.42	0.524444444	**0.5**						
5	0.571111111	0.455555556	0.575555556	0.535555556	**0.5**					
6	0.584444444	0.457777778	0.566666667	0.54	0.495555556	**0.5**				
7	0.524444444	0.391111111	0.48	0.464444444	0.44	0.451111111	**0.5**			
8	0.475555556	0.34	0.424444444	0.411111111	0.362222222	0.375555556	0.442222222	**0.5**		
9	0.566666667	0.448888889	0.531111111	0.52	0.477777778	0.477777778	0.553333333	0.611111111	**0.5**	
10	0.482222222	0.355555556	0.451111111	0.428888889	0.393333333	0.384444444	0.457777778	0.504444444	0.411111111	**0.5**

Table 16: Structures A-Test Summaries (Even KC Distribution). Structures A-Test Summaries for evenly distributed KCs. Vargha-Delaney A-Test [Vargha and Delaney, 2000] effect sizes for simulation results. Effect sizes qualified at: no difference ($0.44 \leqslant a \leqslant 0.56$), small difference ($0.56 \leqslant a \leqslant 0.64$ or $0.36 \leqslant a \leqslant 0.44$), medium difference ($0.64 \leqslant a \leqslant 0.71$ or $0.29 \leqslant a \leqslant 0.36$), large difference ($a \geqslant 0.71$ or $a \leqslant 0.29$).

Structure	KS Test
1	0.1359
2	0.3079
3	0.3079
4	0.3079
5	0.3079
6	0.1359
7	0.1359
8	0.3079
9	0.3079
10	0.1359

Table 17: Comparison of KC distributions by structure. KS-test comparison of biologically validated and even KC distributions by structure.

Structure	KS Test
1	0.9852
2	0.9852
3	0.7358
4	0.7358
5	0.9852
6	0.3744
7	0.3744
8	0.3744
9	0.9852
10	0.7358

Table 18: Comparison of KC distributions by structure (>4 cells). KS-test comparison of biologically validated and even KC distributions by structure for inflammatory foci greater than 4 cells.

6.2.3 Summary

In summary, these data have answered two pertinent questions regarding the original research purpose. Firstly, I predict that variance in liver lobule morphology does not influence the formation of inflammatory foci in terms of the whole population of infected KCs, and secondly, the distribution of KCs throughout the lobule does not alter the spatial location of larger cluster sizes. I predict that though large foci could potentially form anywhere in the real system, were they to be quantified statistically, inflammatory foci are more likely to form at lobule peripheries, simply as a product of new cells entering from that region. This simulation experiment has also revealed that running each experiment on a single structure is sufficient to provide a representative set of simulation results.

6.3 ALTERING KC RETENTION OF NKTS

Since I model cell-mediated retention probabilistically, it is important to further understand how the parameters that influence the dynamics of KC retention of NKT cells can affect simulation output measures. Intuitively, if retention is too high NKT cells would never leave the location of the first KC they encounter. Equally, if retention is too low NKT cells would have fleeting interactions with KCs and never sustain interactions sufficient to activate them. I do not know what the distribution of NKT cell aggregations sizes would be at both extremes, nor in what quantities.

Simulation values involving NKT cell retention, i.e. NKT cell leave probability (δLeave), are calculated for every iteration an NKT cell interacts with a KC (1-second simulated time); therefore, across a simulation spanning 48 hours of simulated time, δLeave percentages of between 0-100 would lead to NKT cells satisfying their leave condition seconds after an initial KC interaction, and thus NKT cells would never be retained by KCs for any meaningful period of time. Indeed, the sensitivity analysis in section 6.1.3 demonstrated that parameter values greater than ~0.0005 for *leaveProb* (initial δLeave value), result in a median cluster size of no greater than 1 when excluding zero-inflated data (most KCs do not sustain aggregations).

For this simulation experiment a more focused factorial analysis was used to gather data, using two-at-a-time analysis to explore a range of values for both the initial (modifiable) and minimum δLeave probabilities. Figure 46 depicts the response landscapes for percentage inflammatory foci formed when modifying both initial and minimum δLeave values. Data show the majority of infected KCs do not form a significant inflammatory focus, and of the remaining KCs the maximum percentage foci formed is ~16.5%, ~11.8% and ~8.5% (figure 46**(a)**-**(c)** qualified at 4,6,8 cells), corresponding to ~5, ~3 and ~2 cells out of 33 per simulation respectively. These are small percentages but these data themselves help explain the observed heterogeneity in inflammatory foci *in vivo*, as multiple large inflammatory foci are the exception and not the norm (quantified experimentally in section 7.3).

Percentages of foci >4 NKT cells are highest when both initial and minimum values are low (figure 46**(a)**), this is mirrored in the observed mean total KC stimulation time per run (figure 47**(a)**). However, percentage foci >6 (figure 46**(b)**) and >8 NKT cells (figure 46**(c)**) are extremely low under the same conditions. This is explained by the complete cessation of NKT cell motility at such low δLeave probabilities (high retention). The KC population nearly receives the maximum possible stimulation time (49 NKT cells stimulating them for 48 hours = 2352 cumulative hours), since NKT cells spend all their time stimulating and not patrolling the sinusoids (figure 47**(b)**).

In summary, these data demonstrate that the formation of inflammatory foci when there are a finite number of NKT cells (the principal components of those foci), is a complex balance between KCs being able to optimally retain NKT cells whilst not totally arresting population wide NKT cell motility. The issue of finite NKT cells numbers is addressed in section 7.3.1.

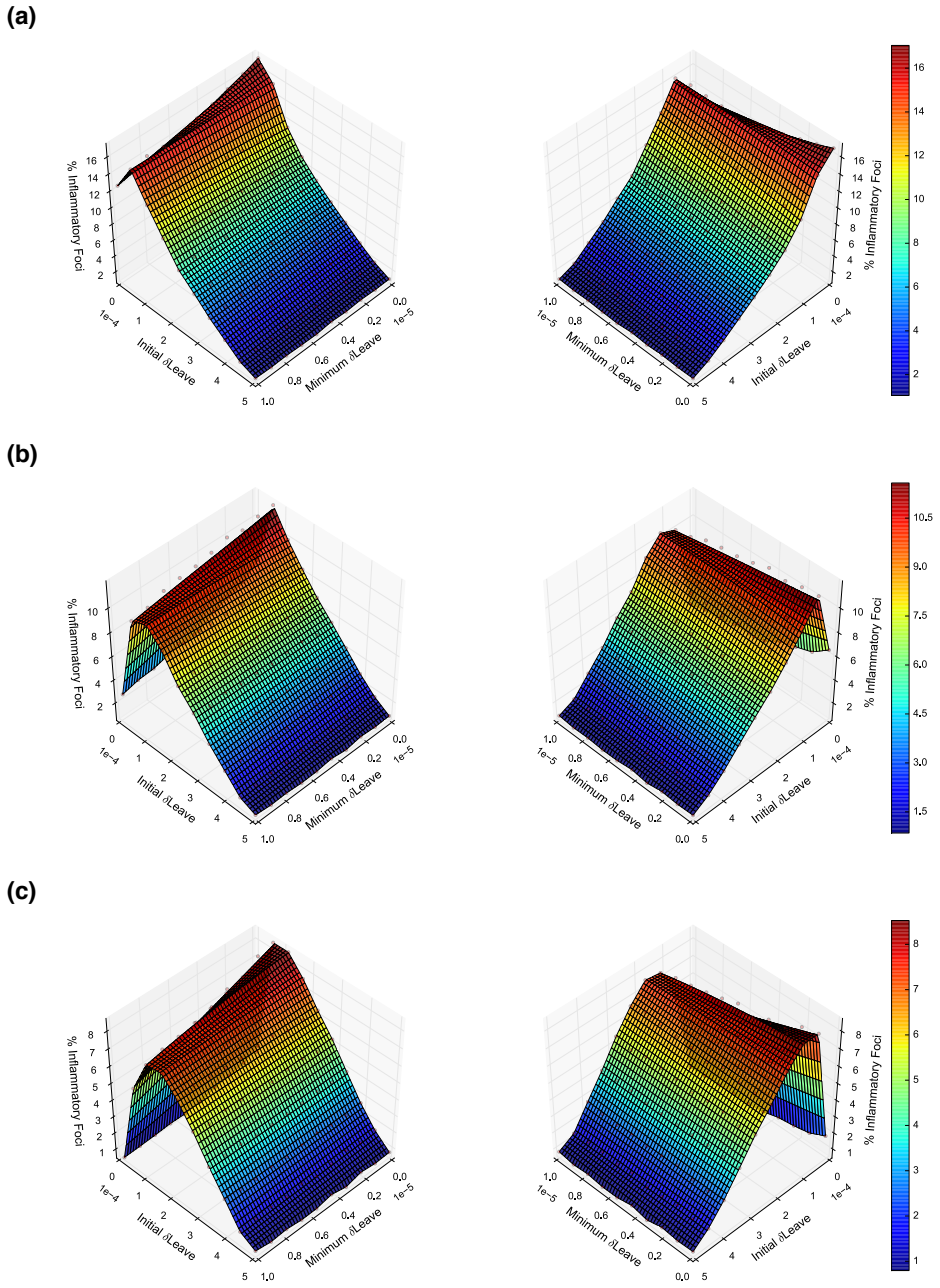


Figure 46: Probabilistic Retention Experiment. Two-at-a-time analysis to quantify the effects of minimum and starting retention probabilities on percentage foci formed when qualified at 4 - **(a)**, 6 - **(b)**, and 8 - **(c)** cells. 3D plots in each sub-figure are rotated to show multiple perspectives and reveal more detail of each surface profile.

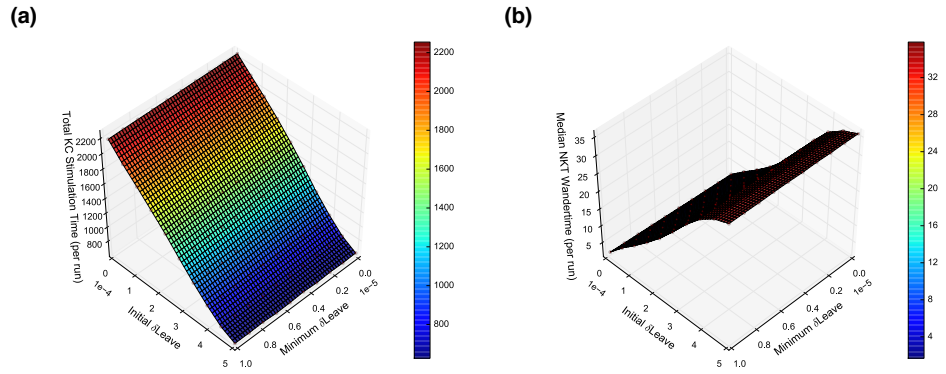


Figure 47: Probabilistic Retention Experiment Continued. Two-at-a-time analysis to quantify the effects of minimum and starting retention probabilities on **(a)** - mean total KC stimulation time per run and **(b)** - median NKT wander time per run.

6.4 SUMMARY

This chapter has described various simulation experiments which comprise the first part of a results model presented in thesis. I first detailed the simulation output metrics, I then employed aleatory analysis to determine the number of simulation runs required to generate. Secondly, I presented results of a latin-hypercube sensitivity analysis designed to determine the robustness of the simulation to parameter perturbation. Having established the means to interpret the simulation results, I then sought about answering several of my initial research questions. I explored the use of multiple sinusoids structures during simulation, and various distributions of KCs throughout the environment, and determined that variations in sinusoid structure and KC distribution are not sufficient to significantly modify neither the size or spatial distribution of aggregations (around infected KCs) throughout the network. Thirdly, seeing as the model utilises a probabilistic function to handle cell retention, I then performed a factorial analysis of the parameters that influence retention. These results highlight a delicate balance between KCs being able to optimally retain NKT cells whilst also preserving a degree of motility of NKT cells, allowing other cells to also receive a degree of activation and stimulation.

All of the results presented in this chapter were required for us to understand the dynamics of certain features that the abstracted model imposes on the system, namely: environment, probabilistic retention, and spatial distribution of KCs. Whilst these results are interesting and novel in this area, they are descriptive. Chapter 7 will demonstrate an iterative process of experimentation that was performed once I had established an understanding of the simulation platform using the results from this chapter.

7 USING SIMULATION TO EXPLORE MECHANISMS UNDERPINNING CELL RECRUITMENT IN THE LIVER MICROENVIRONMENT

Having implemented and calibrated a simulator (section 5.6.1 and chapter 6), it could then be used for novel computational experimentation. This chapter represents the body of that computational research, whilst also outlining some *in vivo* experiments used to validate the computational results. Where the simulator required modification, I explain how it remains fit-for-purpose at each stage of update.

Granulomas are associated with sub-clinical infection in humans, though much of the evidence is circumstantial. The literature relating to how granulomas confer protection in mouse models is well documented, with evidence suggesting they provide enhanced parasite removal [Murray, 2001], and therefore granulomas are traditionally regarded as being a host defence mechanism in VL. An exaggerated inflammatory response could lead to adverse tissue pathology and subsequent damage; therefore, a balance between the promoting a parasite clearing response whilst maintaining healthy tissue must be reached. This raises several questions; namely:

- What mechanisms might be more important in driving the granulomatous response?
- How can we perturb those mechanisms to dampen or promote that response?

To answer these questions I used the model and assumptions outlined previously in the domain model (chapter 4). Through a process of iterative hypothesis generation and predictive modelling, certain assumptions were modified based on newly acquired data, and subsequently a novel prediction was made regarding hepatic stimulation in trans. This chapter will detail the results from three experimental scenarios. Firstly I introduce a baseline model (Scenario A figure 48), where infected KCs can attract and retain NKT cells and uninfected cells call only attract. Secondly, the baseline model is extended to include an abstraction for parasite-induced activation of KCs, where infection induces upregulation of attractive mediators (Scenario B figure 52). Lastly, scenario B is extended to include upregulation of attractive mediators by uninfected KCs, through bystander activation in trans (Scenario C figure 53). Bystander activation in trans is the process by which the signals endogenous to the environment cause a response from cells within that environment.

7.1 SCENARIO A – EXPLORING A BASELINE MODEL

One of the original research hypotheses outlined in section 1.3 asks the question whether or not mediators of cellular recruitment can account for the varying size and distribution of granulomas observed *in vivo*. Little is known with regards to how production and area of influence of T cell-chemo-attractants affect cellular recruitment into inflammatory foci. In this section, I explore a range of factors that govern KC produced chemo-attractant *in silico*, and which could potentially play a role in the *in vivo* system.

7.1.1 Greater Area of Attractive Influence Promotes Inflammatory Foci

For the simulation experiment exploring the area of attractive influence, I work under the assumptions governed in experimental scenario A (figure 48). I assume that in terms of attraction, both uninfected and infected KCs function identically prior to activation of infected KCs; i.e. they both diffuse an attractive influence a minimum distance, and at the same concentration level, and that subsequently, only infected KCs can modify their attraction level and distance after activation and subsequent interactions - based on the KC-NKT amplification pathway highlighted earlier (chapter 4).

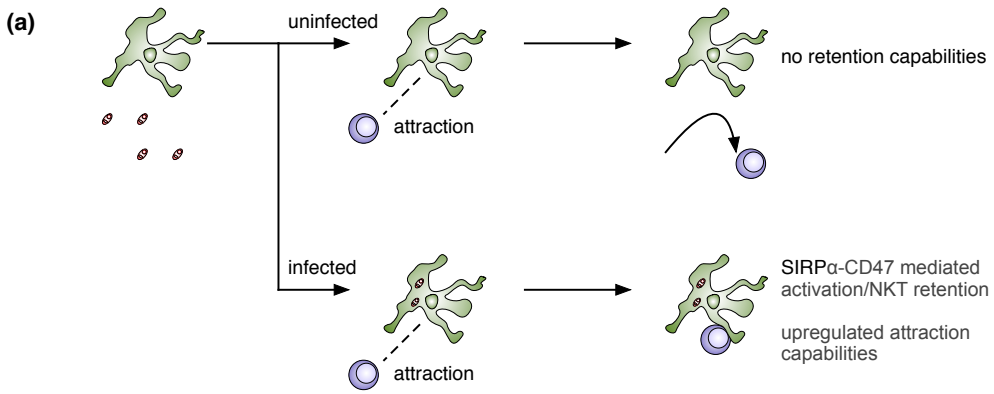


Figure 48: Baseline Model Scenario. **(a)** The baseline model.

To determine any interaction effects that might exist between parameters, I explore the area of influence using a range of maximum diffusion distances, whilst concurrently modifying one of the following:

- the total cumulative time required to activate KCs
- the total cumulative stimulation time KCs require reach maximal attractive chemokine production.

I posited the following hypothesis:

An increase in maximum diffusion distance of chemokine will lead to a greater percentage of inflammatory foci.

Two-at-a-time (TAT) analysis was first performed modifying both diffusion distance and the cumulative stimulation time required to activate KCs. Simulation data show that quicker activation of KCs does lead to increased percentages of significant inflammatory foci across all simulation runs, with a significant focus qualified at 4, 6 and 8 NKT cells at a KC location (figure 49(**a**)). The median stimulation time received by the KC population also increases with quicker activation times (figure 49(**c**)).

When performing TAT analysis modifying the maximum diffusion distance and the cumulative stimulation time required by KCs to reach maximal attraction level, the results were mirrored, i.e. increases in both percentage inflammatory foci and median stimulation time received by the KC population are observed (figure 49(**b**),(**d**)). This sort of experiment is not possible *in vivo* as there are many different known and unknown attractive mediators, and no current way of targeting their production and/or area of influence. Hence, this experiment has provided novel insight into how modifying these mechanisms might play a role in the formation of inflammatory foci.

After determining that quicker activation of KCs leads to greater formation of inflammatory foci, I revisited the simulation to determine the effect of pre-activation of KCs. I hypothesised that:

Pre-activation of KCs will lead to greater percentage of inflammatory foci

The function of KCs in the simulator was modified to change the logic dictating that KCs require a threshold of activation. A command line argument was used to enable this preactivation to be toggled. After implementation of this feature, all statistical analyses outlined in (chapter 6) where repeated to ensure the model was still fit-for-purpose.

Pre-activation did not significantly alter the percentages of inflammatory foci. The abundance of activated KCs leads to an environment where no individual KC can really have the opportunity to outperform others by having a 'head-start' after having being activated via NKT cell stimulation. This experiment was novel and could not have been performed experimentally, as it is currently not possible to instantaneously 'switch on' KC functions *in vivo*.

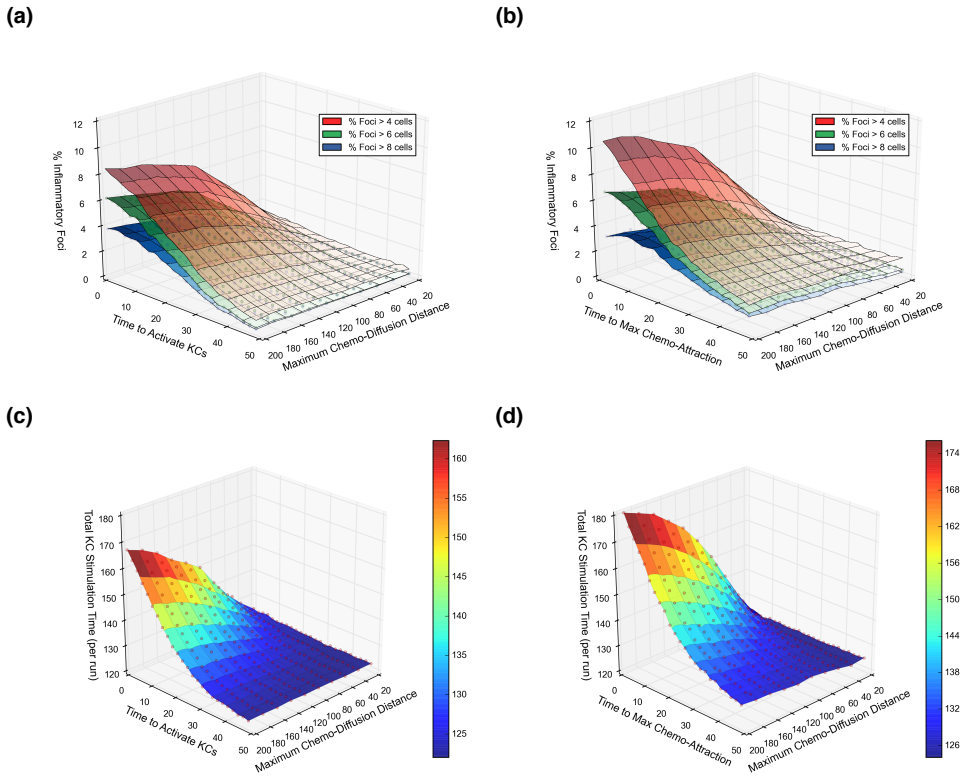


Figure 49: Attractive Chemokine Dynamics. Two-at-a-time (TAT) parameter analysis to quantify the percentage of inflammatory foci formed when modifying chemokine diffusion distance simultaneously with either **(a)** stimulation time required to activate KCs, or **(b)** stimulation time required for KCs to reach maximal attractive chemokine production. TAT analysis for the aforementioned parameter combinations, **(c)** and **(d)** respectively, to quantify the median, total stimulation time received by infected KCs.

7.1.2 Increasing Chemokine Attractive Area Has Diminishing Returns

A recent study to analyse chemokine gradients *in vivo* reported that functional chemokine gradients occur up to a distance of approximately 100µm [Weber et al., 2013]. My simulated data in section 7.1.1 predict diminishing returns when increasing maximum diffusion past a certain distance. Significant differences (P=⩽0.001) are observed when comparing distributions from each parameter increase with the previous value (e.g. 20µm-30µm: P=0.001216, 30µm-40µm: P=0.000019). When increasing diffusion distance from 120µm-130µm, the percentage foci increase subsequently becomes non significant (P=0.312 figure 50).

These data demonstrate that there is a tipping point, and that there is no functional benefit to having a greater attractive influence past ~120µm, similar to the distance reported by Weber et al. [2013]. The structure of the microenvironment, and the competition between cells essentially negates the greater area of influence.

Another finding of my results that is not clearly evident, since data is aggregated over multiple simulation runs, is that individual infected KCs, who perform very well initially (in terms of receiving a high degree of stimulation), often don't benefit in the long term. The diffusion of attraction throughout the network results in their attractive influence encompassing other KCs, many of which may be underperforming. This leads to what I would describe as indirect KC altruism, and often their downstream counterparts can go on to outperform them by the simulations end. Some of these findings may have broader implications for liver disease aetiology.

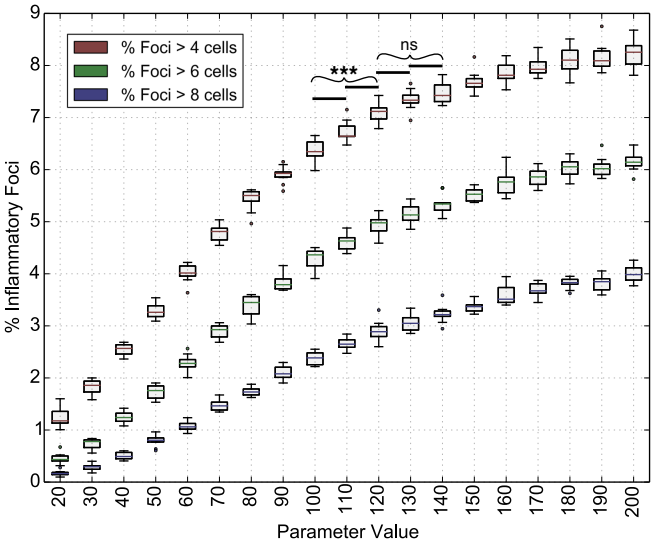


Figure 50: Diminishing Returns For Chemo-Attraction Distance. Effect on percentage inflammatory foci qualified at 4, 6 and 8 cells, when modifying maximum chemokine diffusion distance.

7.1.3 Implications For Manipulating Inflammatory Responses

From simulation data reported in sections 7.1.1 and 7.1.2 I predict that if it were possible to exclusively target the ability of infected KCs to expedite their up-regulation of chemo-attractants, and importantly, increase their area of influence (up to a tipping point), there would be an increased level of inflammation. Conversely, targeting these mechanisms in an inhibitory manner could abate the formation of inflammatory foci. Depending on the disease context, this could either be host-protective or host-detrimental. The direct manipulation of individual KC function could be a potential avenue for future research, though careful consideration would need to be made as to whether or not this is biologically achievable with any efficacy.

7.2 SCENARIOS B AND C – PARASITE-INDUCED ACTIVATION OF INFECTED KCS WITH/WITHOUT BYSTANDER ACTIVATION OF UNINFECTED KCS IN TRANS

Scenario A (figure 48) described in section 7.1 was developed on the assumption that only infected KCs received a signal to enhance chemo-attractant chemokine production during infection. However, this notion was developed from experimental data generated in the conventional manner of comparing chemokine production from infected vs. uninfected mice [Svensson et al., 2005]. However, when I examined chemokine gene expression in KCs from infected mice, but sorted to contain or not contain parasites [Beattie et al., 2013], I found that both uninfected and infected KCs are equally capable of up-regulating the production of T cell chemo-attractants including CXCL9/10 (figure 51). The mechanisms responsible for the uninfected KC chemokine production are unknown, though a number of mediators could be responsible, such as complement activation, platelets, or trans signalling - signalling perhaps caused by stress or other mediators due to the inflammatory environment.

I revised model scenario A (figure 48) to account for two additional experimental assumptions. Scenario B (figure 52) was constructed to determine the dynamics of parasite-induced activation of infected KCs only, and scenario C (figure 53) to investigate a model including both parasite-induced activation of infected KCs with the addition of uninfected KC up-regulation of chemo-attractants through bystander activation in trans. Scenario B was implemented using a time-dependent switch, with a factor analysis of both parasite-induced activation time (figure 54(**e**)), and parasite-induced activation with bystander activation time (figure 54(**f**)).

I explore three core simulation parameters that I hypothesise are most influential in driving hepatic inflammation: cumulative stimulation time required to activate infected KCs, cumulative stimulation time for KCs to reach maximal production of chemo-attraction, and maximum attraction diffusion distance. These parameters are compared across the three experimental scenarios, with 12 hours

(as this is the last time-point of the data figure 51) of both parasite-induced activation (scenario B) and bystander activation in trans (scenario C).

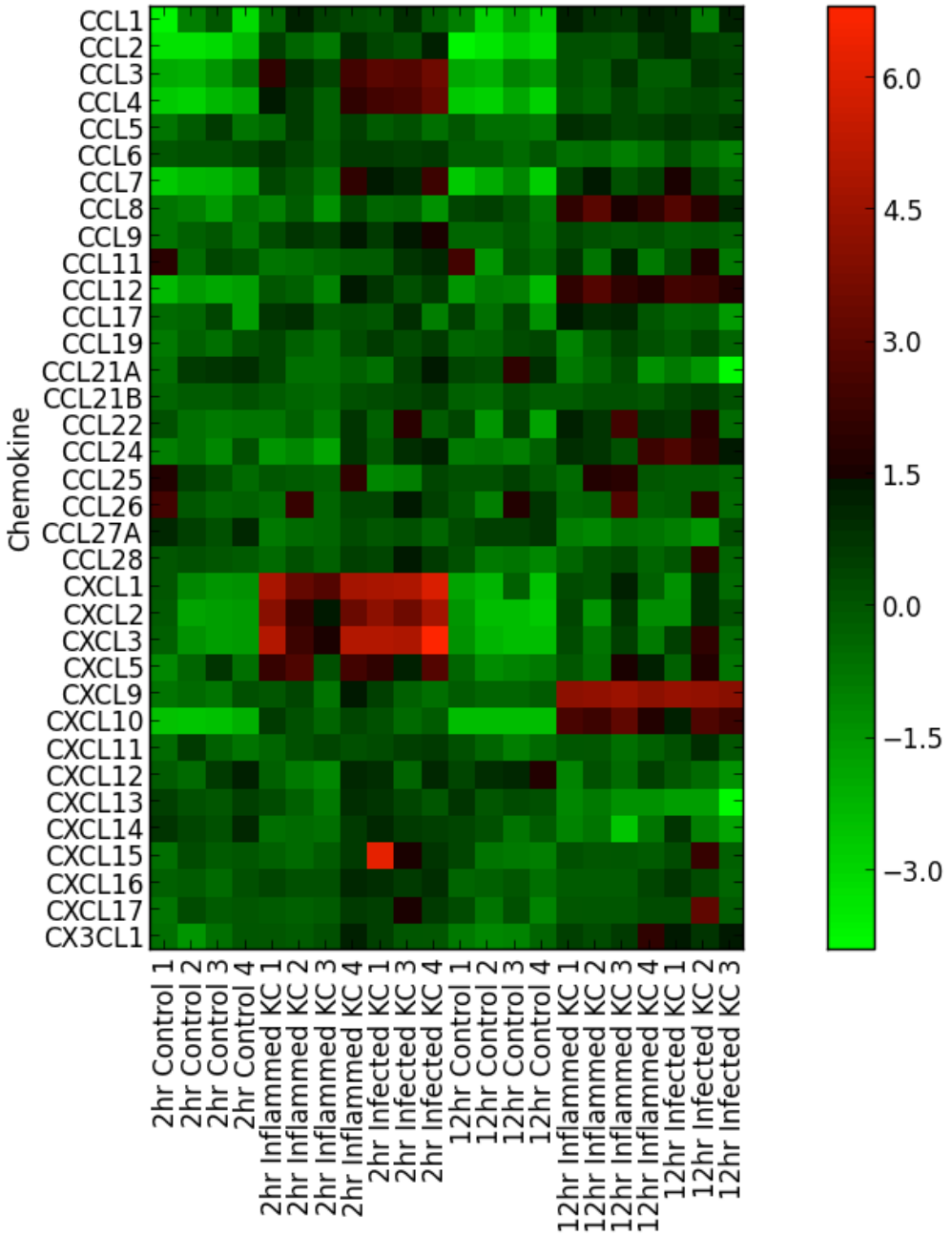


Figure 51: Chemokine Gene Expression. Fold change in chemokine gene expression in inflamed and infected KCs at 2 and 12 hours post-infection.

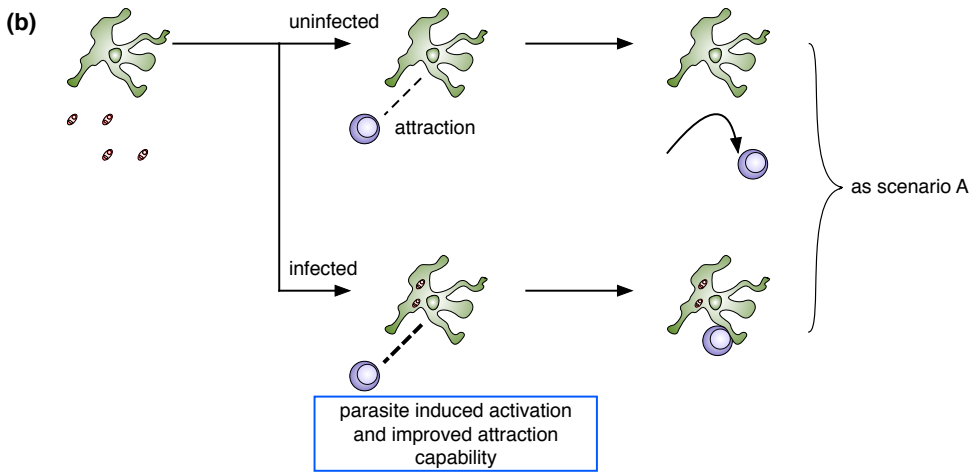


Figure 52: Parasite Induced Activation of Infected KCs. **(b)** Infected KCs have enhanced attraction capabilities and parasite induced activation.

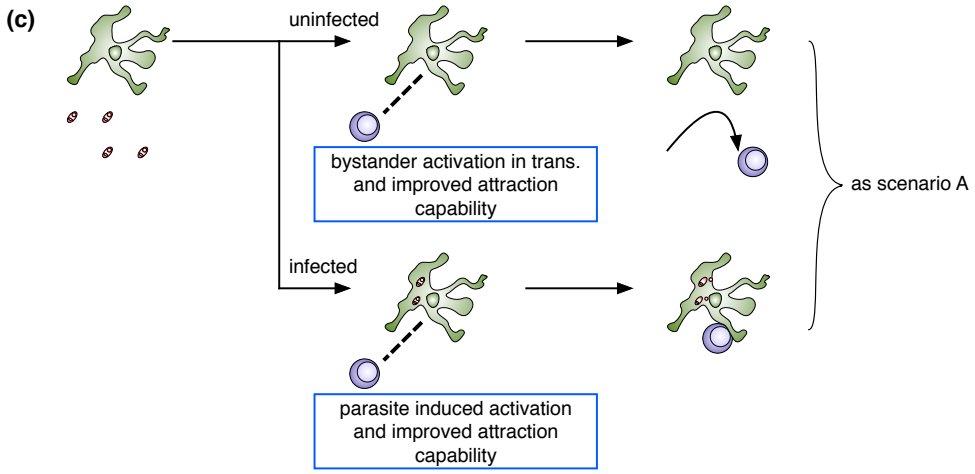


Figure 53: Parasite Induced Activation of Infected KCs With Bystander Activation. **(c)** As in scenario **(b)** but with bystander activation for uninfected KCs to enhance their production of chemo-attractant.

7.2.1 Response Landscapes for Parasite Induced Activation With/Without Bystander Activation In Trans

Two-at-a-time factorial analysis was used to determine the interaction effects on percentage inflammatory foci formed and KC cumulative stimulation time received when varying parameters governing:

- Cumulative time to activate KCs and maximum diffusion distance of chemokine (figure 54(**a**))
- Cumulative stimulation time required for KCs to reach maximum levels of chemoattraction and maximum diffusion distance of chemokine (figure 54(**b**))

Results demonstrate that quicker activation of KCs and greater chemokine diffusion distances lead to the greater proportions of inflammatory foci being formed (figure 54(**a**)), and stimulation time received by KCs (figure 54(**c**)). Similarly, quicker cumulative stimulation time required for KCs to reach maximal levels of chemoattraction and greater diffusion distances also lead to greater levels of inflammatory foci (figure 54(**b**)) and KC cumulative stimulation time (figure 54(**d**)).

I also chose to vary parasite induced activation time and bystander activation times concurrently with maximum chemokine diffusion distance. The response curves from my TAT analysis are in-line with previous data (section 7.1.1) showing that increasing diffusion distance leads to greater percentage inflammatory foci formed (figure 54(**e**)-(**f**)). However, if parasite induced activation and bystander activation persist, both have a negative effect on inflammatory foci formed as cells become uncompetitive because too many of them become activated (figure 54(**e**)-(**f**)).

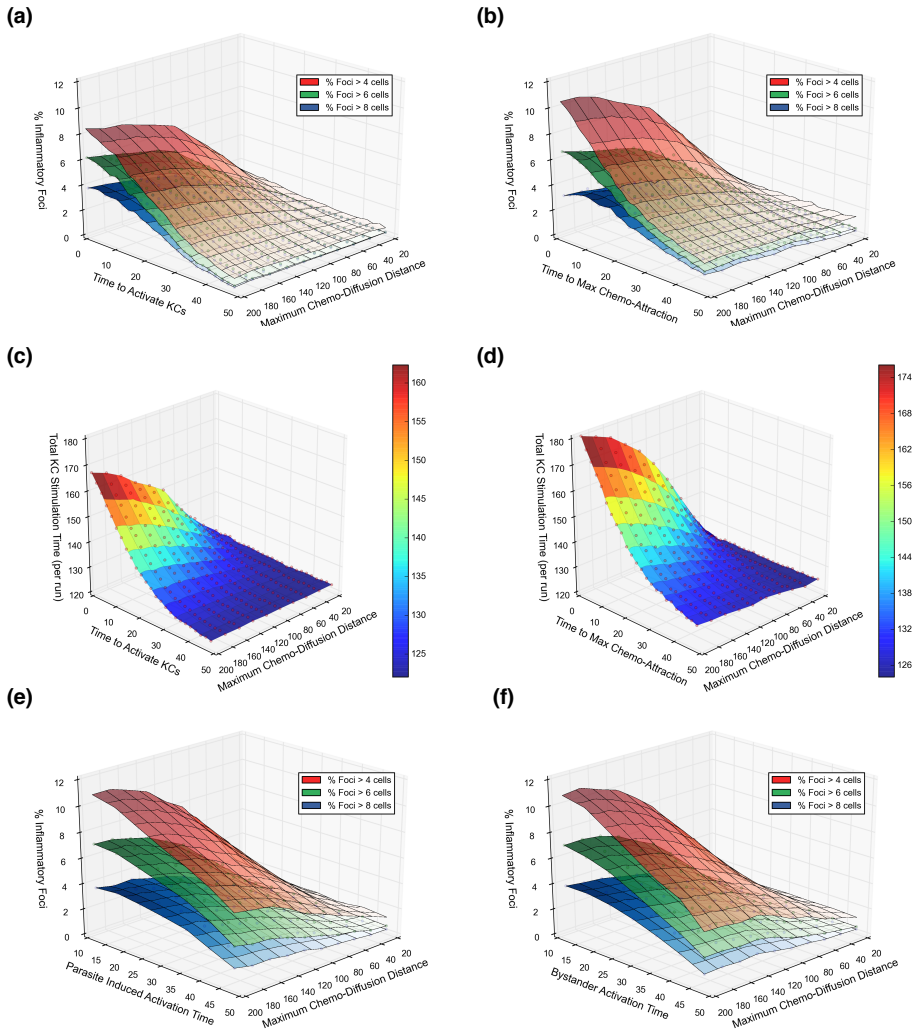


Figure 54: Dynamics of Parasite Induced Activation With/Without Bystander Activation In Trans. Two-at-a-time (TAT) parameter analysis modifying chemokine diffusion distance with ((a),(c)) Cumulative time to activate KCs, ((b),(d)) Cumulative time for KCs to reach maximal chemo-attraction, (e) Parasite induced activation time and (f) Bystander activation time.

7.2.2 12 Hours of Parasite Induced Activation With/Without Bystander Activation In Trans

The experiments in section 7.2.1 explored a range of parasite induced activation and bystander activation durations. Data shown previously in figure 51 only highlights that such mechanisms appear present at up to 12 hours post-infection. Focusing on that duration of both mechanisms, I statistically compare the data distributions observed across all three experimental scenarios previously conceptualised in figures 48, 52 and 53.

When modifying both the time to activate KCs and the time to reach maximal chemo-attractant production, results show that both scenarios B and C demonstrate a significant increase in percentage inflammatory foci over the baseline (figure 55**(a)-(b)**, scenario A vs. scenarios B/C $P \leqslant 0.0001$ for both).

Inflammatory foci sizes between B and C are technically statistically different when exploring times for activation (figure 55**(a)**, P=0.0491) and chemo-attractant production (figure 55**(b)**, P=0.0175), though in terms of effect, the difference is close to none for KC activation time (A-test score = 0.51), and small effect for chemo-attraction production time (A-test score = 0.39).

Whilst scenarios B and C have comparable levels of inflammatory foci, B results in a significantly greater level of stimulation time received by KCs (figure 55**(c)-(d)**, scenario B vs. scenario C $P \leqslant 0.0001$ for both), with the A-test effect size of these differences being medium (figure 55**(c)** A-test score = 0.64) and small (figure 55**(d)** A-test score = 0.61).

Analysis of nearest neighbor data also shows that both scenarios B and C have a significant difference in the formation of small cluster sizes (size<4) over the baseline model (figure 56: A vs. B, P=0.000549 N=1770; A vs. C, P= 0.0 N=1770). In terms of small clusters, scenario C is approaching significance over scenario B (B vs. C, P=0.137 N=1080).

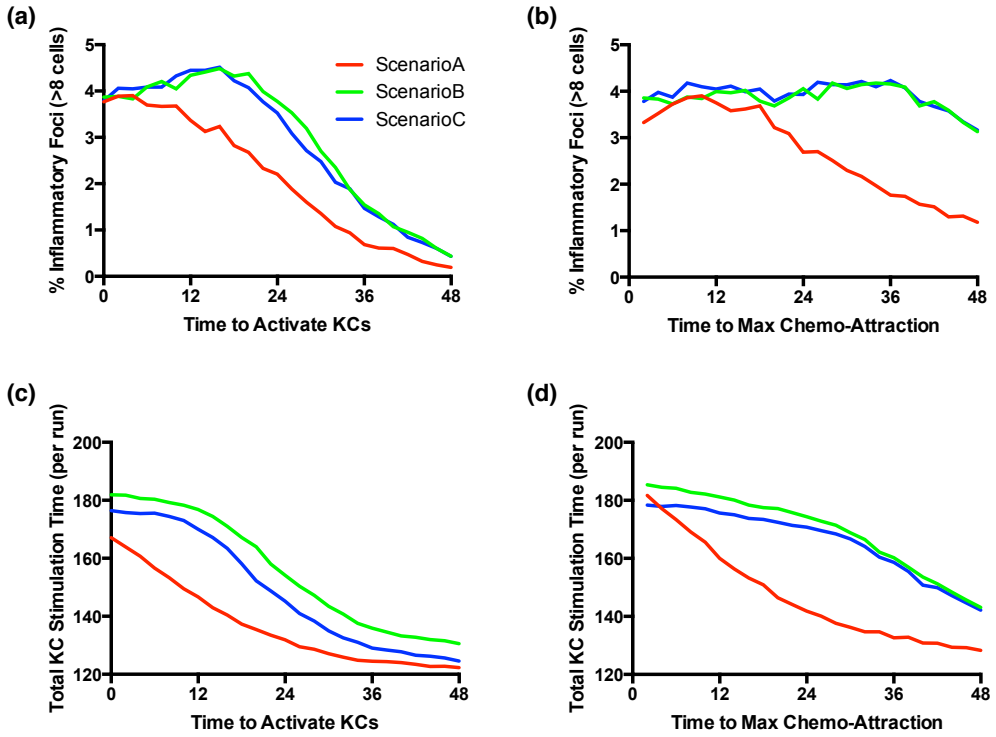


Figure 55: Dynamics of 12 Hours of Parasite Induced Activation With/Without Bystander Activation In Trans. Quantifying (a) the percentage of inflammatory foci formed and (c) mean, total stimulation time received by infected KCs, when varying the cumulative stimulation time required to activate infected KCs when comparing scenario A (baseline model), scenario B (model including parasite-induced activation of infected KCs) and scenario C (model including both parasite-induced activation of infected KCs and bystander activation in trans of uninfected KCs). For the same experimental scenarios, (b) Percentage inflammatory foci formed and (d) mean, total stimulation time received by infected KCs, when varying the cumulative stimulation time required for KCs to reach maximal attractive chemokine production.

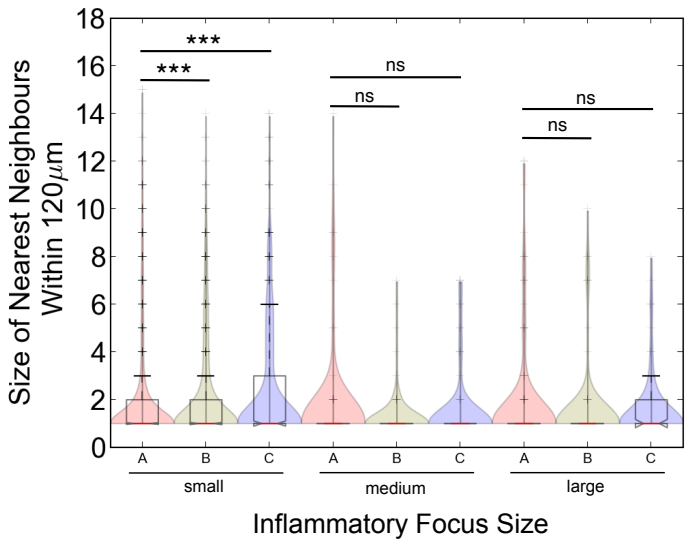


Figure 56: Nearest Neighbour Analysis. KC nearest neighbor network analysis. Clusters qualified at <4 small, 4-6 medium, >6 large.

7.2.3 Section Summary

Similar to the diffusion of attractive mediators and preactivation of KCs, the exploration of parasite-induced activation and bystander activation is not possible with current experimental techniques as these experiments inherently require some ability to directly target cell function at any point during an experiment. These experiments were thus ideal for experimentation through simulation.

Results from sections 7.2.1 and 7.2.1 together demonstrate that compared to only parasite-induced activation, the addition of both parasite-induced activation and also uninfected KC bystander activation in trans gives a less effective inflammatory response, with a lower median percentage foci and stimulation time received by infected KCs. This is due to the lobule sections being saturated with attraction from both uninfected and infected KCs in scenario C, and the competitiveness of infected KCs (which form the aggregations) in terms of recruitment, is diminished. These data highlight a potential mechanism of immune regulation, whereby the bystander activation of uninfected KCs creates a competitive environment in terms of attractive capability, thus making it more difficult for infected KCs to recruit NKT cells and have exacerbated inflammatory responses.

7.3 IL-15 MEDIATED IMMUNOTHERAPY PROMOTES INFLAMMATION

Given that my simulation, with its relevant modelling abstractions and assumptions, has NKT cells as the only promoter cell, I was interested to see if there is competition between infected KCs to attract the 49 baseline number of NKT cells, and whether NKT cells are a limiting factor in the formation of inflammatory foci. This section will detail the *in silico* experiment used to determine the relationship between availability of NKT cells and inflammatory aggregations, and the subsequent experimental validation of that relationship *in vivo*.

7.3.1 Increasing Availability of NKT Cells Promotes Inflammatory Aggregations In Silico

I first hypothesised the following:

An increase in NKT cells will results in an increased percentage of inflammatory foci and ***not*** *the same percentage of foci but with greater levels of cell infiltrate.*

This hypothesis is conceptualised in figure 58(**a**). Simulation results showed that increasing NKT cell numbers above the calibrated value does lead to significant increases in inflammatory foci, regardless of whether those foci are qualified at 4, 6 or even 8 cells (figure 58(**b**)).

Nearest neighbor analysis was then performed in order to predict whether increasing NKT cell numbers would cause extremely large clusters, or more distributed clustering throughout the infected KC population (illustrated in figure 58(**c**)). All KCs, in experiments with: baseline 49 NKT cells, 2-fold NKT cells, and 4-fold NKT cells, were compared to nearest neighbors within a network distance of 120μm. Zero-inflated data was ignored, as I was only interested in KCs that form aggregations of 1 or more, and focus sizes grouped into small (1-3 NKT cells), medium (4-6 NKT cells) and large (>6 NKT cells). Data show that the median neighbor size, and range, increases for focus sizes when increasing NKT cell numbers 2 and 4-fold (figure 58(**c**), table 19). These results were compelling evidence that increasing NKT cell numbers *in vivo* could increase the total percentage of hepatic inflammatory foci, as opposed to only increasing the size of foci and not the quantity.

7.3.2 Increasing Availability of NKT Cells Promotes Inflammation In Vivo

Given the simulation results in section 7.3.1, I hypothesised that:

An expansion of the NKT cell population in vivo *will lead to an increased level of inflammatory foci.*

To test whether my *in silico* prediction that increased NKT cell number would lead to an increase in inflammatory foci *in vivo*, the experimental scheme in figure 57 was devised.

To induce increased NKT cell proliferation and survival [Matsuda et al., 2002], mice were treated with 1mg recombinant IL-15 (rIL-15)(BioLegend) intravenously then infected 3 days later with *L. donovani*. Four days post-infection, mice livers were extracted, weighed then placed into 2% paraformaldehyde (PFA) in phosphate-buffered saline (PBS) for 2 hours, then 30% sucrose in PBS overnight. Tissues were then embedded in Optimal Cutting Temperature (OCT)(Sakura) and stored at -70°C until use. 10µm cryosections were fixed and labeled with Alexa647 or Alexa488 conjugated F480 (eBioscience) and DAPI (Invitrogen) to visualise Kupffer cells and cell nuclei respectively. Images were captured as 0.8-1µm optical slices using a LSM510 confocal microscope (Zeiss). Blinded slides were imaged to score the percentage of infected foci having formed a distinct inflammatory focus (greater than 15 cells), with imaging fields selected via td-Tomato expression.

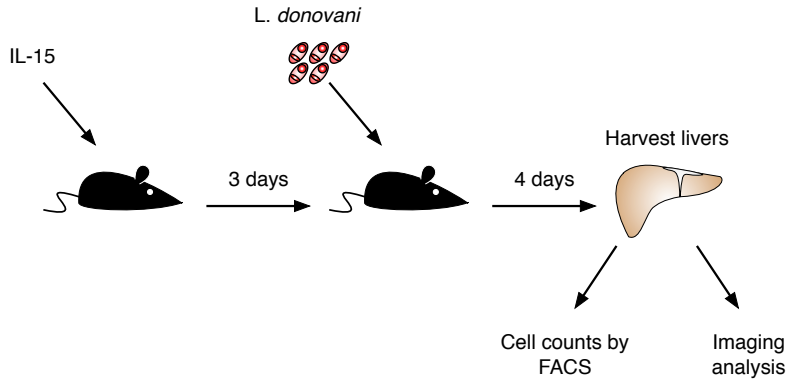


Figure 57: Experimental Procedure. Experimental scheme to determine whether NKT cell expansion leads to increased levels of inflammatory foci in the liver.

In uninfected mice, IL-15 treatment resulted in increases NKT cells (including CD1d restricted NKT cells), NK cells, and T cells (figure 58(**d**)-(**f**) and figure 59(**a**)-(**c**)). In infected mice, all cell types were already increased in number and the effect of IL-15 pre-treatment was limited to an increase in the number of NKT cells (figure 58(**d**)). Similarly, IL-15 pre-treatment had no effect on the relative frequency of NK cells and T cells (figure 60(**b**)-(**c**)), but resulted in an increase in the relative frequency of NKT cells (figure 60(**a**), from 15.0% ± 0.1 N=10 to 17.36% ± 0.8 N=10 per liver; P=0.0043).

I scored distinct inflammatory foci as accumulations of 15 or more cells around an infected KC, not discriminating between NKT cells or other mononuclear cells. There was significant heterogeneity in size within the clusters that qualified, even at the early time point examined (day 4, figure 58(**h**)-(**i**)). I found that the frequency of infected KCs that formed a distinct inflammatory focus was increased ~1.5 fold in mice pre-treated with IL-15 and which had a higher number of NKT cells in the liver at the time of infection (figure 58(**g**); P=0.0038). Thus, treatment of mice with rIL-15, even under conditions where the increase in NKT cell

number is relatively modest, leads to a significant enhancement of inflammatory foci. Dendritic cell-derived IL-15 has been shown in the literature to be capable of mediating the inflammatory response in the liver [Ohteki et al., 2006], my study demonstrates that host pre-treatment with rIL-15 also mediates inflammation, and there is strong evidence that NKT cells, particularly tetramer$^+$ NKT cells (figure 59(**d**)) are responsible for this [Robert-Gangneux et al., 2012; Wehr et al., 2013].

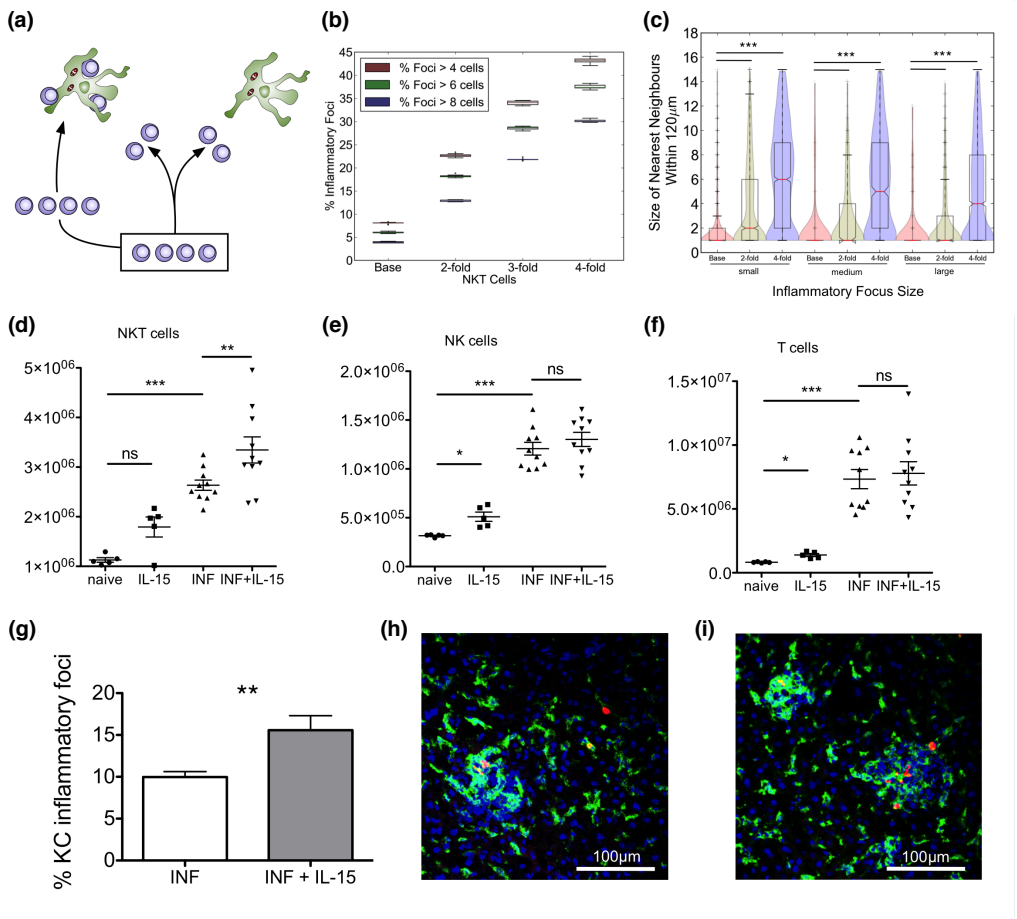


Figure 58: Expansion of NKT Cells Promotes Inflammation. (**a**) Predicting outcome of modifying NKT cell availability. (**b**) Increasing NKT cells *in silico* leads to greater percentages of KCs that form a significant inflammatory focus, when qualified at 4, 6 and 8 cells. (**c**) KC nearest neighbor network analysis. Absolute cell numbers for (**d**) - NKT, (**e**) - NKs and (**f**) - T cells for naïve, IL-15 treated naive (IL-15), infected (INF) and IL-15 pre-treated infected (INF+IL-15) mice. Results are depicted as mean ± SEM of two independent experiments, 5 mice per group per experiment. *P⩽0.05, **P⩽0.01, ***P⩽0.001 paired Students t-test. (**g**) Percentage of infected KCs having formed a significant inflammatory focus for infected and IL-15 pre-treated infected mice. Heterogeneity of inflammatory foci at d4 in both (**h**) infected and (**i**) IL-15 pre-treated + infected mice (F4/80 - green, parasite - red, DAPI - blue).

Cluster Size	Scenarios	P-value	N
small	A, B	0.0	1770
	A, C	0.0	1770
	B, C	0.0	2292
medium	A, B	0.00003	82
	A, C	0.0	82
	B, C	0.0	364
large	A, B	0.0006	140
	A, C	0.0	140
	B, C	0.0	684

Table 19: Statistics for figure 58C.

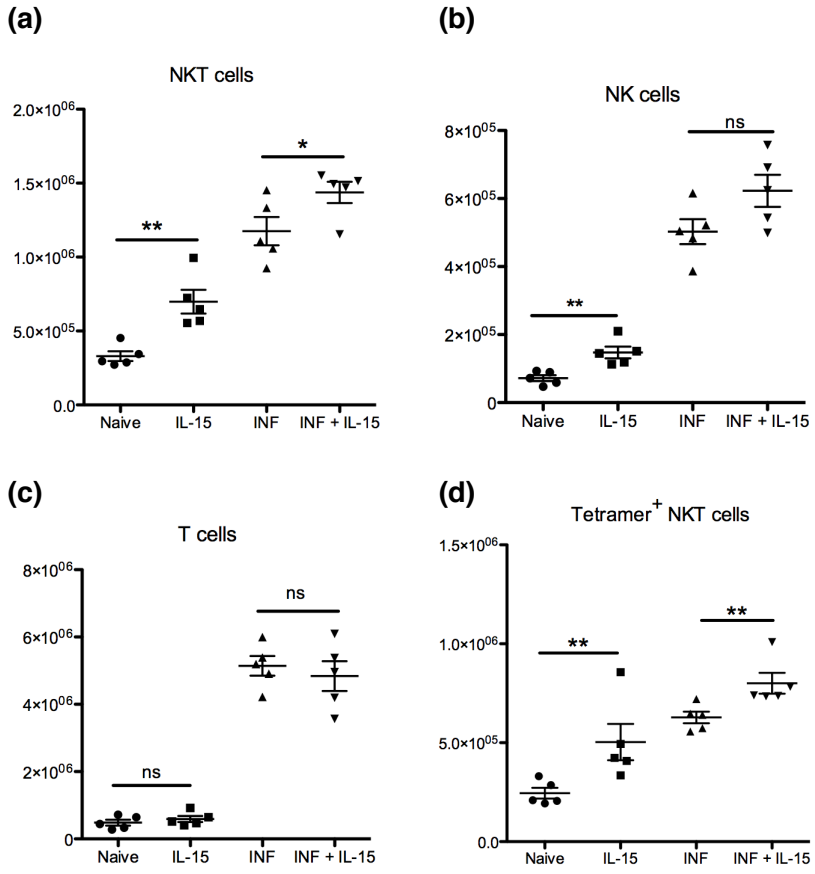


Figure 59: IL-15 Promotes NKT Cell Expansion. (a) Absolute cell numbers for NKT, (b) NK and (c) T cells, and (d) CD1d tetramer+ NKT cells for naïve, IL-15 treated naïve (IL-15), infected (INF) and IL-15 pre-treated infected (INF+IL-15) mice. Results are depicted as mean ± SEM of 5 mice per group. *P⩽0.05, **P⩽0.01 paired Students t-test.

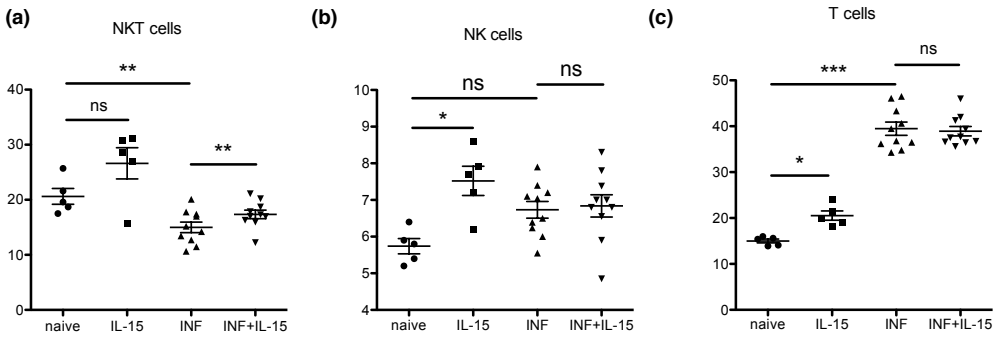


Figure 60: Relative Frequency of Mononuclear Cells Post IL-15 Treatment. **(a)** Relative frequency of NKT, **(b)** NK and **(c)** T cells for naïve, IL-15 treated naïve (IL-15), infected (INF) and IL-15 pre-treated infected (INF+IL-15) mice. Results are depicted as mean ± SEM of 5 mice per group. *P⩽0.05, **P⩽0.01 paired Students t-test.

7.4 RESULTS ARGUMENTATION

Section 5.6.7 detailed several argumentation efforts to argue that my simulation was built with accurate data and as a reasonable abstraction of the underlying biological system. This section will detail arguing the validity of the simulation results presented in this section.

From the simulation metrics described in section 6.1.1, I only have the capacity to experimentally measure the percentage of inflammatory foci *in vivo*, and by extension the size of those inflammatory foci. However, a caveat is that the *in vivo* inflammatory foci do not exclusively contain NKT cells, therefore a direct comparison to the simulation results cannot be made. I therefore utilise argumentation to claim that the simulated NKT cell only clusters are at least plausible representations of the inflammatory foci observed *in vivo*.

Figure 61 depicts ADV claim 2, which states that the simulation results, whilst not directly comparable to those *in vivo*, are plausible with respect to what I observe experimentally. This is done by arguing that heterogeneous cellular aggregations are both observed *in vivo* and also manifested in the *in silico* simulations, highlighting where the evidence for this exists within this thesis. Such an argument will need to be constructed in future to validate the simulation against the remaining simulation output measures, such validation is highlighted as potential future work in section 8.4.

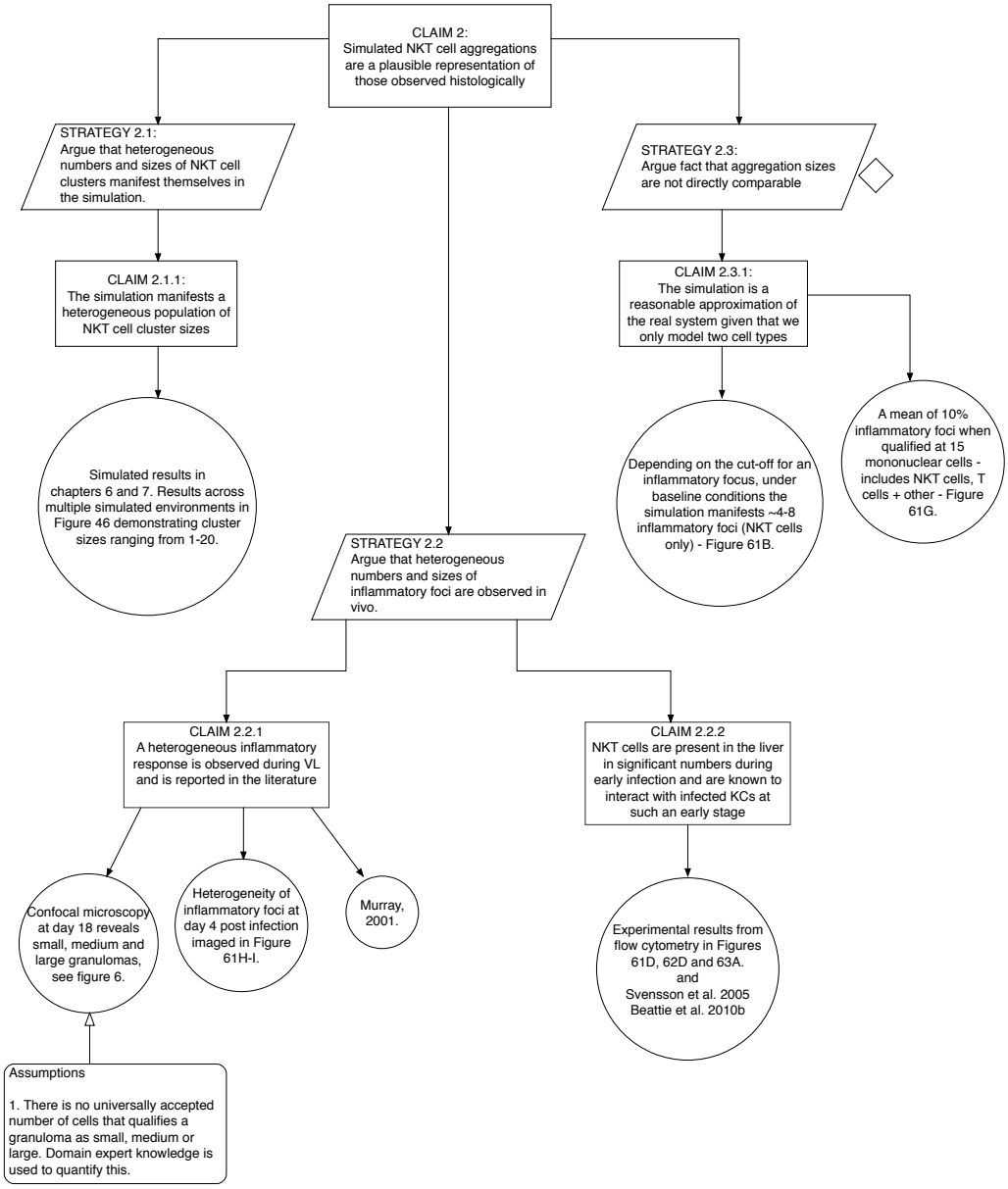


Figure 61: Argument Based Validation: Claim 2. Claim 2 argues that simulated NKT cell aggregations are a plausible representation of those observed histologically. Incomplete or unavailable data to substantiate a claim is denoted using a blank diamond.

7.5 SUMMARY

Transcriptional profiling elucidated a previously unidentified similarity between the capacity of both inflamed and infected KCs to produce a variety of inducible chemokines, including known chemo-attractants. Given the established canon that essential promoter cells are responsible for activation of macrophages such as KCs, and the results of the transcriptomic data, I explored through simulation several hypotheses relating to the dynamics of cellular migration, retention and attraction in the constrained hepatic microenvironment.

I first sought to quantify the influence of attractive chemokine area of influence. My results predicted that chemokine diffusion does play an important role in mediating the formation of inflammatory foci, though there are diminishing returns as a result of increased competition when lobules are flooded with an NKT cell attractive influence. Subsequently, I predicted a somewhat intuitive, yet previously unreported mechanism by which the production of attractive mediators by inflamed KCs dampens the overall inflammatory response in the liver microenvironment. I then demonstrated that any regulatory effect that might exist, could be overcome both *in silico* and *in vivo* (in an experimental model of Leishmania infection) with the expansion of the NKT cell population. Whilst it could be argued that this increase is intuitive, since in my simulation I was increasing a major functional contributor, the relationship between availability of NKT cells and percentage foci had not been quantified prior to my study. Further research is required to precisely identify and categorise the mechanisms responsible for inflamed KC bystander activation, and to modulate any such mechanisms effect.

Part IV

CONCLUSION

8 CONCLUSION

8.1 RESEARCH SUMMARY

As a whole, this thesis has formed a case study in the use of agent-based modelling to validate my research hypothesis, and answer the research questions stated in chapter 1.

My introductory chapter gave an overview of granulomas as a form of inflammation, highlighting Visceral Leishmaniasis (VL) as disease of interest. The limitations of current laboratory experimental techniques was highlighted, and I argued that this provided motivation for the development of computational approaches towards understanding the fundamental mechanisms underpinning inflammation in VL, which may have broader implications for inflammation in general (chapter 1).

Two literature review chapters were presented in this thesis. The first, chapter 2, outlined various modelling techniques that could be used to model biological processes, and I placed an emphasis on the agent-based modelling paradigm, which I argued as the most appropriate technique for spatial and temporal modelling of tissue-scale phenomena (section 2.1.3). In section 2.1.4 I covered the body of computational modelling literature relevant to granulomatous inflammation, and section 2.1.5 detailed previous studies that aimed to create computational representations of the liver microenvironment, where granulomas manifest in VL. My literature review highlighted several gaps in the literature (section 2.1.6), further motivating the research reported in this thesis. This chapter concluded with a discussion relating to the best practice for designing and developing models of complex systems (section 2.2), introducing CoSMoS as a framework that ensures the development process is performed in a principled manner.

The second literature review, chapter 3, presented the biological domain of interest, mainly liver granuloma formation in the context of VL. I described the composition of the liver (section 3.1), whose microenvironment plays host to Kupffer cells, which I describe as being central to uptake of *Leishmania* parasites and the subsequent formation of granulomas (section 3.2). Promoter responses from T and NKT cells were reviewed (section 3.3), and the known cytokine and signalling mechanisms pertinent to VL infection discussed (section 3.4).

This chapter will discuss the remaining thesis chapters, which were designed to answer the underlying research questions stated in section 1.3, and which contribute to the core research hypothesis. I will reiterate each research question and discuss the chapters that answered them, stating the relevant novel contributions. The limitations with the current body of research will then be discussed, and I will suggest future avenues of investigation.

8.2 RESEARCH QUESTIONS – AN EVALUATION

My first underlying research question was as follows:

Question: *Can a spatio-temporal agent-based simulation, incorporating the key cellular players and interaction dynamics into biologically plausible simulation environments, produce a heterogeneous population of granulomas?*

The research detailed in chapters 4 and 5 forms one of the primary contributions of this thesis, that being the development of a novel agent-based model of granuloma initiation in a spatially constrained representation of the liver microenvironment. The design and development of this model, and its presentation in this thesis, followed the format of the various CoSMoS framework stages described in section 2.2.1. The first stage of the modelling process was to describe the domain I wished to study, which formed the second literature review chapter (chapter 3).

The CoSMoS framework [Andrews et al., 2010] (described in section 2.2.1) provides no explicit instruction or recommendation as to how one should document the design and development of complex systems models, chapters 4 and 5 demonstrate a means of performing such activities. Chapter 4 represented a scoping study of the biological domain, which began by detailing several high-level observable phenomena from the domain. Using several diagrams from UML (see section 2.2.3), I then constructed a domain model. Variants of Class, State and Platform diagrams from UML allowed us to capture the relevant cells, cellular mechanisms, and interactions that I hypothesised as responsible at an individual level, for the observed population level responses, such as heterogeneous granuloma formation. The contribution of Chapter 4 represents a method of documenting a scoped model of an underlying biological domain utilising UML to graphically represent the proposed entities and interactions of that model. Chapter 5 worked through the development of a computational, agent-based model of my conceptual domain model. The construction of a computational representation of the liver microenvironment was detailed, followed by a systematic description of how the modelled cells, Kupffer and NKT cells, and the interactions between cells and environment were described in a logical form that could then be implemented as computer code. Subsequent to a process of calibration, and argumentation to establish confidence in the simulation as being representative of the underlying domain, I demonstrated that it was indeed possible to manifest heterogeneous populations of granulomas *in silico* chapter 6.

Having established baseline behaviours for my engineered simulation, I proceeded to answer the second underlying research question:

Question: *What role does the structure of the hepatic microenvironment play in granuloma heterogeneity?*

Section section 6.2 in my first results chapter, demonstrated that variance in inflammatory aggregations remained stable across multiple structures generated

using a novel algorithm. I then investigated whether or not different spatial distributions of KCs throughout the sinusoid structure, could alter the spatial distribution of aggregations. My results demonstrated that changes in KC spatial distribution do not significantly alter the spatial distribution of subsequent aggregations, and I concluded that the imposed entry of new cells into the environment, solely from the peripheries, is the biggest influencing factor in the formation of aggregations, which are predominantly found at the peripheries. The very fact that quantitative studies have shown that KCs are found in greater abundance at the peripheries [Sleyster and Knook, 1982; Bouwens et al., 1986], may mean that organisms have adapted to maximise contact with both cells and pathogens that enter at those locations. These results also begin to answer my final underlying research question:

Question: *Do KC spatial location and mediators of cellular recruitment, account for the varying size and distribution of granulomas observed* in vivo*?*

I have described my results relating to structure and KC spatial location. To investigate the dynamics of cellular recruitment, chapter 7 investigated how mechanisms of attraction and retention could be modified to perturb the distribution of cell aggregations, and also the stimulation time received by KCs, both of which I believe are relevant for long term development of granulomas. The contribution of my results were two fold. Firstly, I demonstrated *in silico* that competition from uninfected KCs, via attractive mediators, serves to dampen the overall inflammatory response provided by infected KCs, thus elucidating a potentially novel means of immune regulation in the liver. Secondly, based on a prediction made using my *in silico* data I demonstrated that the extent of inflammation in a murine model of VL could be increased with an expansion of the NKT cell population; this had not been demonstrated previously in this field.

Given that I have answered my underlying research questions, I now reiterate the core research hypothesis:

Core question: *A spatio-temporal agent-based model can be used to simulate heterogeneous inflammatory responses in the liver microenvironment, and provide new insight into the underlying mechanisms responsible for inflammation*

Computational and systems modelling approaches, though relatively established, are still not widely accepted in the field of biology [Macilwain, 2011], and a paper discussing the various social perspectives of interdisciplinary systems biology highlights many conflicting viewpoints from both computer scientists and biologists, with both groups having proponents and opponents of computational and mathematical approaches in the biosciences [Calvert and Fujimura, 2011]. Some opponents argue that inherent biological variability does not lend itself to imposing mathematical laws or descriptions about that biology [Calvert and Fujimura, 2011, p.160], or that there is often a disconnect when computer scientists and engineers drive systems biology, as they lack the in depth knowledge of the

field [Calvert and Fujimura, 2011, p.160]. Others take a far more critical view, and believe that many seemingly integrative systems approaches are actually just 'reductionism in the guise of entireness' [Huang, 2000]. I argue that this thesis has demonstrated that computational modelling can be of use within the field of biology.

Through the process of defining a set of biological cells and mechanisms believed pertinent to VL-induced granulomas, and through the principled conceptualisation and implementation of a model of that underlying biology, I constructed a spatio-temporal agent-based model of liver inflammation. Iterative hypothesis generation and experimentation, working as an interdisciplinary team, allowed us to investigate the role of environment, cell location, and cell attraction and retentive mediators in driving inflammation. Subsequent efforts at preliminary experimental validation have provided new insight into mice infected with VL, and assuming that my prediction of a novel immune regulatory mechanism holds true, I have demonstrated a means of overcoming that regulation by expanding a promoter cell population.

Although the focus of this thesis has revolved around the study of *Leishmania* granulomas, the body of work presented is also more widely applicable to scientists outside of this field. My research demonstrates how relatively minimalist models can still prove useful for hypothesis driven research, abstracting away from the underlying complexity and yet still capable of capturing key holistic, observable behaviours. There is a propensity for many modellers to adopt the 'kitchen-sink' approach, creating highly granular models that incorporate a raft of components and interactions. This approach has several disadvantages. Firstly, models can become intractable and be far more complex than they need be. Complexity can lead to erroneous models, and error can exhibit itself in many ways, such as incorrect assumptions, or problems with the implementation of a component and/or interactions; the more complex a model, the harder it is to demonstrate its validity. Also, communicating the purpose of your model, and arguing that your model functions as intended (verification), becomes more difficult as its complexity increases. I believe I have struck a balance in this respect, and assert that my model in it's current state can be used to further investigate VL as a biological domain, with future validation and modification adding greater predictive power to the model.

8.3 LIMITATIONS OF THE RESEARCH

Whilst I have explored dynamics relating to NKT cell recruitment and retention by Kupffer cells, I reiterate that my research has only been concerned with early stage granuloma formation, within the first 48 hours of simulated infection. Whilst my assumptions are based on current understanding in the field [Moore et al., 2013] and my simulated results are plausible, my model only determines conditions that give either maximum levels of cellular aggregation, or maximum levels of stimulation received by KCs. I cannot categorically state that a large aggregations of early promoter NKT cells, whose KC focus has received a high-

level of stimulation, will develop into a fully formed granuloma. There may be an intermediary step between the early initiation of granulomas and the later recruitment of additional mononuclear cells which make up the structure of a mature granuloma.

A limitation of the current simulation is that the structures I utilise are all based upon idealised hexagonal sections of liver lobule. As I explain in Appendix 9.3, mice liver lobules vary in a variety of degrees from this hexagonal structure. Future work in this regard is described in section 8.4.3.3.

In the context of the underlying biology, extremely large cluster sizes (>15) observed in simulation may not be plausible, though this will only be addressed with further imaging studies of NKT cell clustering *in vivo*. My model doesn't account for distention of the sinusoids, nor do I restrict cell numbers at infected nodes as I cannot quantify these restrictions in the real system. Therefore, I must make these types of assumptions, and in any case the data showed that these clusters only exist in very small percentages in my simulations (chapter 7). Future work will look too investigate these issues further.

8.4 FUTURE WORK

This section will outline the future work which aims to further validate my model.

8.4.1 Quantifying Granuloma Location Experimentally

In an attempt to quantify granuloma location *in vivo*, and identify centrilobular regions as a point of reference to measure the distance of granulomas from, I used immunofluorescence staining with anti-glutamine synthetase (AGS) antibody to identify glutamine synthetase, an enzyme present constitutively in centrilobularly located hepatocytes [Dane et al., 2009]. The method and protocol used to extract liver sections and stain them are described in the appendix (see Appendix 9.3). Data proved inconclusive for a variety of reasons explained in Appendix 9.3. This experiment proved inconclusive, even after investing a significant period of time - primarily on the confocal microscopy. However, new technologies such as the Axio Scan Z1 slide scanner (Zeiss), are now allowing researchers to load a significant number of slides, drastically improving the imaging time. This mass, automated scanning, would make it simpler for a researcher to quickly identify usable images with easily identifiable lobule structures, facilitating the quantification of granuloma location within them.

8.4.2 Validating Simulated NKT Cell Behaviour

To properly validate the migration behaviour of NKT cells in my simulation, I will require the use of 3D plus time imaging techniques such as 2-photon intravital microscopy in order to capture the *in vivo* behaviour of these cells, tracking NKT cell movement over time. Such techniques require a particular strain of

mice, with fluorescent markers that allow NKT cells to be identified and tracked independent of other mononuclear cells. This is possible with the CXCR6^{gfp+} mice, which have been used in other studies [Geissmann et al., 2005; Wehr et al., 2013]; however, requires a significant investment in both financial and human resource as well as time, hence why this contribution is not available in this thesis.

8.4.3 Model Extensions and Refinements

8.4.3.1 *Simulated Cells and Time Period*

The abstraction that my simulation represents only attempts to capture events up to the first 48 hours subsequent to infection. To assess the functional contribution of the various lymphocytes, in addition to NKT cells, and other mononuclear cells, further iterations of the model might benefit from their inclusion. However, those cell populations are only believed to play a role in granuloma maturation after the initial amplification stage [Moore et al., 2013]. Careful consideration of the cascading effects of additional cells, and their influence on the environment, must be made prior to their implementation within my model.

8.4.3.2 *KC Cytokine Production and Parasite Dynamics*

A recent computational study highlighted a key role for the KC produced (autocrine) IL-10 cytokine in regulating KC ability to control parasite burden within granulomas [Albergante et al., 2013]. My model doesn't simulate any parasite dynamics, though implementing those dynamics would be possible given the modular nature of the simulator. This type of extension would allow us to further investigate various aspects of KC activation and subsequent cytokine and chemokine production, which ultimately has an influence on the microenvironment and both lymphocyte and myeloid cell populations. The model by Albergante et al. [2013] represents a population level view of the same system explored in this thesis, I approached the system from an individual level perspective. There may be the potential for a great level of insight to be gleaned from utilising both models to investigate future hypotheses relating to hepatic inflammation, with population level findings being validated by my individual level simulation, and vice versa.

8.4.3.3 *Lobule Variation and Interconnectivity*

Future iterations of the model and structure generation algorithm will perturb the location of the vertices of the generated sinusoid sections. Lobule section morphology varies from the idealised hexagons in the literature, therefore modifying those vertices will allow us to generate ever more realistic lobule sections that account for that heterogeneity. Lobule sections may also be combined in future, to investigate inter-migration of cells between lobule sections, and how that influences inflammatory responses.

8.4.3.4 *Potential For Multi-Scale Modelling*

My simulation captures events at the cellular level. Many of the processes dictating cell behaviour, and the factors influencing cells and the environment, are abstractions. Extensions to the model could include modelling at multiple scales, for example including an intra-cellular model of Kupffer cells, allowing the model to incorporate the influence that parasites have on KC function, and the dynamics of parasite survival and the implications this has on downstream infection.

There is also potential for the model to be extended to include dynamics at the organ-level. For example, the simulation currently has a fixed population of NKT cells in the micro-environment. Little is known at the micro-level about the influx rates of cells into lobules. Data can be acquired on cell numbers at the organ-level, and population-level models of the dynamics of cell quantities and infiltration into the liver could be used as exogenous input for my simulation. Ultimately, a balance between models at different scales will be beneficial to investigate all pertinent aspects of hepatic inflammation in the context of VL.

Lastly, *Cell* objects, and the *View* objects that represent them when utilising visualisation, use a three dimensional Cartesian coordinate system, therefore should there be any benefit to moving towards a spatially continuous 3D representation of the environment, this could be achieved through modification of the *structure* package, though this would have implications for how cells will then interact. This extension may be of interest should future research be more focused on how cellular interactions and forces in three dimensions influences inflammatory foci, as in some tumor modelling.

8.5 AN AGILE APPROACH TO CREATING A MODEL OF INFLAMMATION

8.5.1 On Scrum

In the context of the research presented in this thesis, the developer was a lone individual taking on the roles of stakeholder, developer, and scrum master. In order to adhere to the letter of scrum, a project and developer team consisting of several individuals is more sensible; however, the principles that scrum advocates are more than valuable to a lone developer. Scrum provides a means of prioritising software features to be agreed by stakeholders, managing progress, and incremental development. These values are ultimately as beneficial to an individual as they are to a team.

Within this CoSMoS software project, I have no customers, only stakeholders, and indeed my research has been undertaken with immunologists very much involved in every stage of the development process. Requirements evolved over time, which were then scheduled for implementation in subsequent software iterations. Agile methods were invaluable when experimental data emerged from the laboratory unexpectedly. Domain experts appreciated the flexibility of discussing model changes in order to add or remove functionality to reflect new understand-

ing acquired from the laboratory or the literature. I feel that by adopting an agile development methodology researchers can increase productivity and stakeholder engagement, which will ultimately improve the quality of the final software product.

8.5.2 On Documentation

Agile methods generally promote a 'lean' approach to software documentation; one of the core values of The-Agile-Manifesto [2001] is 'working software over comprehensive documentation'. That isn't to say that one should not document a project, but what the methodology deems more important is intricate knowledge of the operation of software through close team cooperation during its' development. Knowledge of the software product can then be communicated to new team members by those who have intricate knowledge of its creation. Pitt-Francis et al. [2008] echo these values, and use practices from the XP variant of agile, such as pair-programming (two-developers working in unison), to integrate new team members into a project. However, academic projects may go through periods of inactivity, for example someone might be employed to build upon the research of a previous PhD student or post-doc. If the original developers are not members of that particular institute anymore, the new developer can only rely on whatever software documentation exists to use, maintain and extend the software.

In the context of the software developed for this thesis, there were no provisions for long term staffing resources on the project. In order to ensure the future viability of any research stemming from the project, some level of documentation was required. Chapters 4 and 5 represent that documentation, and although there were significant resource overheads in updating that documentation throughout the project, those chapters provide a means for future developers to understand the principles underlying my simulation software.

8.6 SUMMARY

This chapter has summarised the research contributions in this thesis, and explained how a computational model, developed in a principled manner (chapters 4 and 5), and the subsequent simulated experiments performed to answer my underlying research questions (chapters 6 and 7), have evidenced the validity of my core research hypothesis (stated in chapter 1). I feel that my model and results represent a novel leap in spatio-temporal modelling of *Leishmania* induced liver inflammation, and should also prove of interest to those studying and modelling inflammatory responses in other tissues. The future work outlined in section 8.4 is achievable, and will serve to further highlight the contribution to the field that this thesis represents.

Part V

APPENDIX

9 APPENDIX MATERIAL

9.1 MODEL IMPLEMENTATION

9.1.1 Initialisation and Update Cycle

The *Main* class in the *Simulation* package handles the command line arguments used to provide parameterisation information for the simulation. One of the core functions of this class is to instantiate the main simulation object. To make use of various functionalities provided by MASON, including scheduling, the simulation class needs to make use of the *Steppable* class in the *sim.engine* package. Class *EVLCore* provides an interface that extends the *Steppable* class. Both *EVLSim* and *EVLSimWithUI* implement the *EVLCore* interface. My main simulation object is represented in the *Environment* class, which extends the MASON *SimState* class, which contains an *EVLSim* simulation object that is steppable, and handles all simulation updates. The simulation has an update cycle which is time-stepped, with each iteration representing one second of simulated time. The MASON class *Schedule* within the *sim.engine* package has pre-written libraries for dealing with simulation scheduling (see also Appendix 9.1.3 for how this relates to cells and cytokine management). Listing 3 details the overriden *start* method contained in the *Environment* class.

Listing 3: Simulation Scheduling

```
    @Override
2   public void start() {
        super.start();
4       duration = System.currentTimeMillis();
        setEVLParameters();
6       setup();
        schedule.scheduleRepeating(Schedule.EPOCH,1,evl);
8   }
```

9.1.1.1 *With Visualisation*

The model with visualisation adopts the Model-View-Controller design pattern. The model is represented within the *Environment* class. When visualisation is employed, the *EVLSimWithUI* controller class constructor sets up a number of *Portrayal* objects, which are a type of MASON object that deals with the display of 2D and 3D view objects. The constructor is also provided with the *Environment* object so that the *EVLSimWithUI*, which extends the *GUIState* class can receive model updates and relay them to the GUI.

Listing 4: Visualisation Controller Constructor

```
    EVLStartWithUI(SimState state) {
        super(state);
        Environment env = (Environment) state;

        KupfferP = new ContinuousPortrayal3D();
        KupfferP.setField(env.view.KupfferField);
        NKTP = new ContinuousPortrayal3D();
        NKTP.setField(env.view.NKTField);
        sinusoidP = new ContinuousPortrayal3D();
        sinusoidP.setField(env.view.SinusoidMap);
        sinusoidP.setDirtyField(true);
        chemoP = new ContinuousPortrayal3D();
        chemoP.setField(env.view.ChemoMap);

        // Build the box
        wireFrameP = new WireFrameBoxPortrayal3D(-EXTRA_SPACE,
                                                 -EXTRA_SPACE,
                                                 -EXTRA_SPACE,
                                                 Environment.XMAX + 2*EXTRA_SPACE,
                                                 Environment.YMAX + 2*EXTRA_SPACE,
                                                 Environment.ZMAX);
    }
```

9.1.2 Structure Creation

The algorithm, created in collaboration with Dr. Paul Andrews, used to generate representative sinusoid structures are detailed in algorithms 3 and 4. Structure files are represented using the XML file format (listing 5). Each file contains a *Structure*, which is comprised of *Node* and *Edge* elements. Nodes have several attributes; a unique identifier, a type (regular node n, edge node e, drain node d) and individual cartesian coordinate values. The structures generated all have a zero value for the z coordinate component. Edge attributes include a unique identifier, two node identifiers (the nodes which make up this edge), and an identifier which states the direction of flow for this edge (towards the drain node/central vein).

Listing 5: XML Structure File

```
<Structure>
<Nodes>
<Node id="0" type="n" x="0.662198" y="0.111683" z="0.000000"/>
...
</Nodes>
<Edges>
<Edge id="0" nodeA="6789" nodeB="8410" flow="8410"/>
...
</Edges>
</Structure>
```

A *Structure* package was written to represent the network of nodes and edges contained within a structure file. The package contains class representations for the structure, its nodes and edges (figure 64). An XML DocumentBuilder is constructed to read the structure parameter file, and the specific structure file

provided via command line argument to the simulation. A single structure object is a component of the overall simulation *Environment* object.

Listing 6: Structure Creation

```
private void createStructure(int NORMALISE, Parameters params)
    {
        try
        {
            /* Using factory get an instance of document builder */
            System.out.println("    Loading "+params.structureFile);
            DocumentBuilder db = dbf.newDocumentBuilder();
            File in = new File(params.directory+"/src/networks/"+params.structureFile);

            /* Parse using builder to get DOM representation of the XML file */
            dom = db.parse(in);

            System.out.println("Creating nodes");
            setNodes(dom, NORMALISE, params);

            System.out.println("Creating edges");
            setEdges(dom);

        }catch(ParserConfigurationException pce) {
                pce.printStackTrace();
        }
        catch(SAXException se) {
                se.printStackTrace();
        }catch(IOException ioe) {
                ioe.printStackTrace();
        }

    }
```

9.1.3 Initialising and Scheduling Cells and the Cytokine Manager

Subsequent to the program and structure initialisation, the cytokine manager and cells are initialised before adding the simulation object to the scheduler (listing 7). Firstly, the cytokine manager is instantiated, followed by a conditional statement which reads from the command line parameters to determine which KC distribution algorithm has been selected, then creates the population of KCs (placement is performed using algorithms 1 and 2). Secondly, the population of NKT cells are instantiated. Lastly, the *schedule()* method adds the cytokine manager and cells to the scheduler.

The MASON scheduler provides a threadsafe means of queuing and scheduling events to occur. All objects within the simulation are initialised using the *Schedule.EPOCH* field from the *Schedule* class, which schedules them to be stepped at the first possible time.

Listing 7: Simulation Setup

```
@Override
public void start() {
    super.start();
    duration = System.currentTimeMillis();
    setEVLParameters();
    setup();
```

```
    schedule.scheduleRepeating(Schedule.EPOCH,1,evl);
}

private void setEVLParameters() {
    EVLSim.iterations = params.simIterations;
    EVLSim.logIterations = params.logEveryIterations;
}

private void setup() {
    cytokineManager = new CytokineManager(params);
    if (params.distribute.equals("bouwens")) createKCsBouwens();
    else createKCsEvenly();
    createNKTs();
    for (Kupffer k : kupffers) {
        Initialise.kupfferParameters(this,k);
    }
    for (NKT nkt : nkts) Initialise.nktParams(this,nkt);
    schedule();
}
```

Cell states defined in the various state diagrams in the platform model are implemented by attributing specific cell objects, objects of class *NKT* and *Kupffer*, with an *enum* type. Transitions between states are governed by various conditional statements within the relevant class.

9.1.3.1 *KC Cells*

The instance variable *kState* is an object of enum type *KupfferState* (figure 67). Only the *aware* and *engaged* constant values are used, although additional values have been declared to facilitate the addition of further functionality.

The *Kupffer* class contains most of the core functionality for KCs, however, two child classes have been created to differentiate additional functions between infected and uninfected KCs. Kupffer class functions are then overridden as required. Within the *step* function, a switch statement is used to determine the current cell functionality conditional on its current state (listing 8).

Listing 8: Infected KC Step Function

```
@Override
public void step( SimState state ) {
    env = (Environment)state;

    switch (kState)
    {
        case aware:
            aware();
            break;
        case engaged:
            engaged();
            break;
        ...
    }
}
```

The transition between states, in this case the *aware* and *engaged* state is achieved with a conditional statement within the *aware()* method. As the state diagram for infected KCs dictates (figure 19), the aware to engaged transition will occur if the current node IFN-γ level is greater than a threshold. This is

implemented using a conditional statement that compares the double value of the current nodes IFN-γ with the *ifnThreshold* value parameterised when KCs are initialised.

Listing 9: Infected Aware Method

```
@Override
protected void aware() {
    /* Check if the IFN threshold has been reached on the node
     * If so, become engaged. */
    if (Double.compare(env.getIFN(node),ifnThreshold) >= 0) {
        kState = KupfferState.engaged;
    }
}
```

The engaged functionality for KCs is implemented in the *increment()* method (listing 10).

Listing 10: KC Increment Method

```
/**
 * Increment attraction and if there multiple cells co-localised
 * then increment retention.
 */
private void increment() {
    /* Increment attractive chemokine function only if we're after
     * the chemoSwitch limit. Until then, all attraction increments are
     * dealth with by the cytokine manager */
    if (env.evl.iteration > env.params.chemoSwitchIteration) env.incrementAttract(node, env.
        cellsAtNode());
    /* Cells at the node is a count of NKT cells only
     * only increment if there are cells
     * - Retention dealt with normally*/
    if(env.cellsAtNode(node) >= 1)
        env.incrementRetain(node, env.cellsAtNode(node));
}
```

9.1.3.2 *NKT Cells*

The more complex array of states, in comparison to KCs, of NKT cells requires two types of enum to implement the cell states. NKT cell objects have instance variables for their *MoveState* (governing movement states) and their *ActionState* (governing action states). Switch statements within the NKT cell *step()* method are used to determine which functions the NKT cell performs depending on its current state (listing 11). ActionState function is determined first, followed by MoveState function, though these are evaluated sequentially within the same simulation iteration.

Transitions between states are also implemented using a variety of conditional statements. An example is when NKT cells begin the simulation, they are in the *inactive* ActionState and the *patrolling* MoveState. The *sense()* method is invoked whilst patrolling, and this method provides functionality for the cell to query the network environment (listing 12).

Listing 11: NKT Step Method

```
@Override
```

```
public void step ( SimState state ) {
    env = (Environment) state;

    switch(action_state)
    {
        case inactive:
            break;
        case producing:
            produce();
            break;
        case recovering:
            recover();
            break;
    }
    switch(move_state)
    {
        case patrolling:
            sense();
            break;
        case chemotaxing:
            chemotax();
            break;
        case aggregating:
            action_state = ActionState.producing;
            break;
    }
}
```

First, the sense method checks to see if the NKT is currently recovering after aggregating, if this is the case then the only function of the NKT cell is to move around the network. If that condition is not satisfied, the NKT cell will then determine if it is on a network node with diffused chemokine and, assuming there is no KC on that node (which results in NKT cell aggregation), the NKT cell will transition into the *chemotaxing* MoveState. As the platform model doesn't implement interactions with uninfected KC, the method also ensures that KCs do not aggregate if a co-located KC is of type KCUn (the child class of Kupffer representing an uninfected KC).

Listing 12: NKT Sense Method

```
private void sense()
{
if (action_state == ActionState.recovering) move();
else {
    if (env.getNode(node).getChemotaxNodes().size() > 0 && env.structure.getNode(node).getKC()
          == null) move_state = MoveState.chemotaxing;
    else if (env.structure.getNode(node).getKC() != null)
    {
        if (env.structure.getNode(node).getKC() instanceof KCUn) forceOut();
        else bind_to_kc();
    }
    else move();
    }
}
```

The movement code for NKT cells is detailed in listing 13. As reported in section 5.6.3.2, the *move()* method is first used to determine when cells should move subject to a probabilistic calculation of speed within pre-defined bounds. When the cell satisfies the move condition, a three stage movement method is invoked.

MOVE_STAGE_A There are 3 possible scenarios for NKT cell movement. Either a) the cell is on an edge node, where there is only one possible edge to move along, b) there are more than one edges connected to the current node (a branch point or a segment of sinusoid), or c) the NKT cell will probabilistically reverse direction and traverse the previous edge and return to the previous node. Method *move_stage_a* has several conditional statements to satisfy these scenarios, and conditional on the outcome of those statements a relevant option is passed to *move_stage_b* with the *edge* on which the cell will move.

MOVE_STAGE_B This method receives an *edge* and an integer *option*. This method extracts the required edge information for the next move and relays them to move_stage_c to finalise the move with the environment. The parent cell class for NKT cells contains instance variables which store the last edge and node, and these are used to set the current NKT cell information if the NKT cell is probabilistically reversing in the network.

MOVE_STAGE_C This method utilises the accessor/mutator methods in the environment class to finalise the move.

Listing 13: NKT Cell Movement Code

```
private void move() {
    move -= 1;
    if (move == 0) {
        /* Recalculate the next cell velocity based on the parameter bounds */
        int diff = moveMax - moveMin;
        if (diff == 0) {
            move = moveMin;
        }
        else {
            move = env.random.nextInt(moveMax - moveMin) + moveMin;
        }
        lastEdge = move_stage_a();
        distance += 1; /* TEST DISTANCE CODE */
        if (env.getNode(node).isDrain() == true)
        {
            /* If the cell has moved to a drain, reenter elsewhere */
            node = env.reenter();
        }
    }
}

/**
 * Method to move an NKT cell to a new node
 * @return Returns the NKT cells last edge after this movement cycle
 */
private Edge move_stage_a( ) {
    ArrayList<Edge> last = new ArrayList<Edge>();
    last.add(lastEdge);
    ArrayList<Edge> nedges = new ArrayList<Edge>(env.getNode(node).getEdges());
    nedges.removeAll(last);
    ArrayList<Edge> edges = new ArrayList<Edge>(nedges);
    Edge e = new Edge();
    if (edges.size() > 1) {
        /* There are two or more edges to choose from */
        int size = edges.size();
        int item = env.random.nextInt(size);
```

```
        e = edges.get(item);
        return move_stage_b(e, 2);
    }
    else if (env.random.nextDouble() < turnProb && lastEdge != null) {
        /* Probabilistic change in direction */
        /* Option 0 to ensure the lastEdge is switched */
        return move_stage_b(e, 0);
    }
    else {
        /* There is only one edge to choose from */
        try {
            return move_stage_b(edges.get(0), 1);
        }
        catch (Exception except){
            return null;
        }
    }
}
/**
 * Sets the NKT Cells new node id based on the fetching the
 * other connected node.
 * @param e The relevant edge to move along
 * @param option Option to ensure the lastEdge is set correctly
 * @return Returns the current edge
 */
private Edge move_stage_b(Edge e, int option) {
    int a_id,b_id;

    if (option == 0)
    {
        a_id = lastEdge.getNodeA().id;
        b_id= lastEdge.getNodeB().id;
    }
    else
    {
        a_id = e.getNodeA().id;
        b_id = e.getNodeB().id;
    }
    if (node == a_id) { move_stage_c(b_id, a_id); }
    else if (node == b_id) { move_stage_c(a_id,b_id); }

    if (option == 0) { return lastEdge; }
    else { return e; }
}

/**
 * Set the new node details for this Cell.
 * @param a The id of the relevant nodeA
 * @param b The id of the relevant nodeB
 */
private void move_stage_c(int nodeA, int nodeB)
{
    node = env.getNode(nodeA).id;
    location = env.getNode(node).getLocation();
    lastNode = env.getNode(nodeB).id;
    env.removeCellFromNode(nodeB, nktid);
    /* add and remove cell to the "cells" arraylist of the node */
    env.addCellToNode(nodeA, nktid);
}

/**
 * Set the probability to turn in the sinusoids for this cell.
 * @param turnProb
 */
public void setTurnProb(double turnProb) {
    this.turnProb = turnProb;
```

```
103 }
```

9.1.3.3 *Cytokine Manager*

The *CytokineManager* class is used to handle all chemokine functionality, both attractive and retentive chemokine. The constructor takes values for various chemokine constants from the simulation input parameters object (listing 14). As detailed in section 5.3.4, chemokines are modelled as a sigmoid function, and the method in listing 15 is used to calculate the constant c for equation (1).

Listing 14: CytokineManager Constructor

```
1 public CytokineManager(Parameters params) {
      chemoIFNConstant = calculateConstant(params.chemoIFN);
3     chemoAttractConstant = calculateConstant(params.chemoAttract);
      chemoRetainConstant = calculateConstant(params.chemoRetain);
5     linearDiffusionTick = calculateLinearDiffusionTick(params.chemoDist,params.maxDist);
  }
```

Listing 15: Calculate constant

```
      double calculateConstant(double thresholdValue, int increments) {
2             return Math.abs((Math.log((1/thresholdValue)-1)-6)/increments);
      }
```

The CytokineManager *step* function involves multiple procedures. Firstly, for each KC the *updateComponents* method (*updateComponentsUninfected* for uninfected KCs) updates the level of IFN-γ, attractive chemokine and retentive chemokine at the respective nodes. The individual cells increment the variable of the sigmoid function representing time (see section 5.3.4.1), and the updateComponents method evaluates the specific cytokine or chemokine value based on the sigmoid function (using the *sigmoid* method).

Once source cytokine and chemokine levels have been calculated, the diffusion algorithm for attraction is initiated. Firstly, each node has an ArrayList which stores the id of any KC that has chemokine diffusing up to that node's location. That ArrayList is cleared and is ready to be updated. Secondly, for each KC the current level of attractive chemokine is extracted from that KC's current node. The method *calculateLinearDiffusionTick* is then used to evaluate how many nodes the current level of chemokine will diffuse (see section 5.3.4.2). A conditional statement is then used to ensure that this distance (in nodes) is less than the maximum parameterised diffusion distance. Lastly, the *walk_nodes_kc* function is invoked and recursively sets the chemotax nodes for every node that will now be influenced by the current KC's chemokine.

Listing 16: CytokineManager Step Function

```
1 @Override
      public void step(SimState state) {
3         env = (Environment)state;
          /* Update all the chemokines components at infected KC nodes */
5         for (Kupffer k : env.getKupffers()) {
              if (k instanceof KCInf) {
```

```
                updateComponents(k.getNode());
            }
            else {
                updateComponentsUninfected(k.getNode()); /* Actually update uninfected attract
                      */
            }
        }

        /* Diffuse every iteration - this may need optimised at a later date for performance
             */
        if (EVLSim.iteration % 1 == 0) {
            /* Remove all attractive chemokine nodes */
            for (Node n : env.structure.getNodes()) {
                if (n.getKC()==null) {
                 env.delChemotaxNodes(n.id);
                }
            }

            /* Recalculate all attractive chemokine and diffuse */
            for (Kupffer k : env.getKupffers()) {
                int chemoDist;
                double currentLevel = env.getAttract(k.getNode());
                /* Now calculate how many extra ticks to add to minimum diffusion distance */
                int ticks = (int)(Math.round(currentLevel/linearDiffusionTick));
                chemoDist = Environment.params.chemoDist + ticks;
                if (chemoDist > Environment.params.maxDist) chemoDist = Environment.params.
                     maxDist;

                /* Perform the diffusion */
                simulation.Functions.walk_nodes_kc(k.getNode(),
                    k.getNode(),
                    0,
                    chemoDist,
                    env,
                    k.getNode());
            }
        }
    }
```

9.1.4 UML Class Diagrams

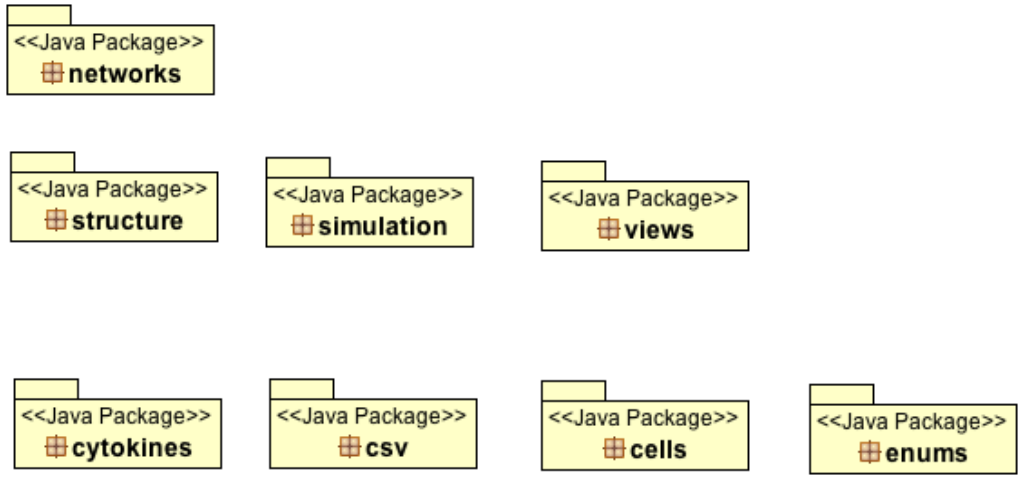


Figure 62: Simulator Packages

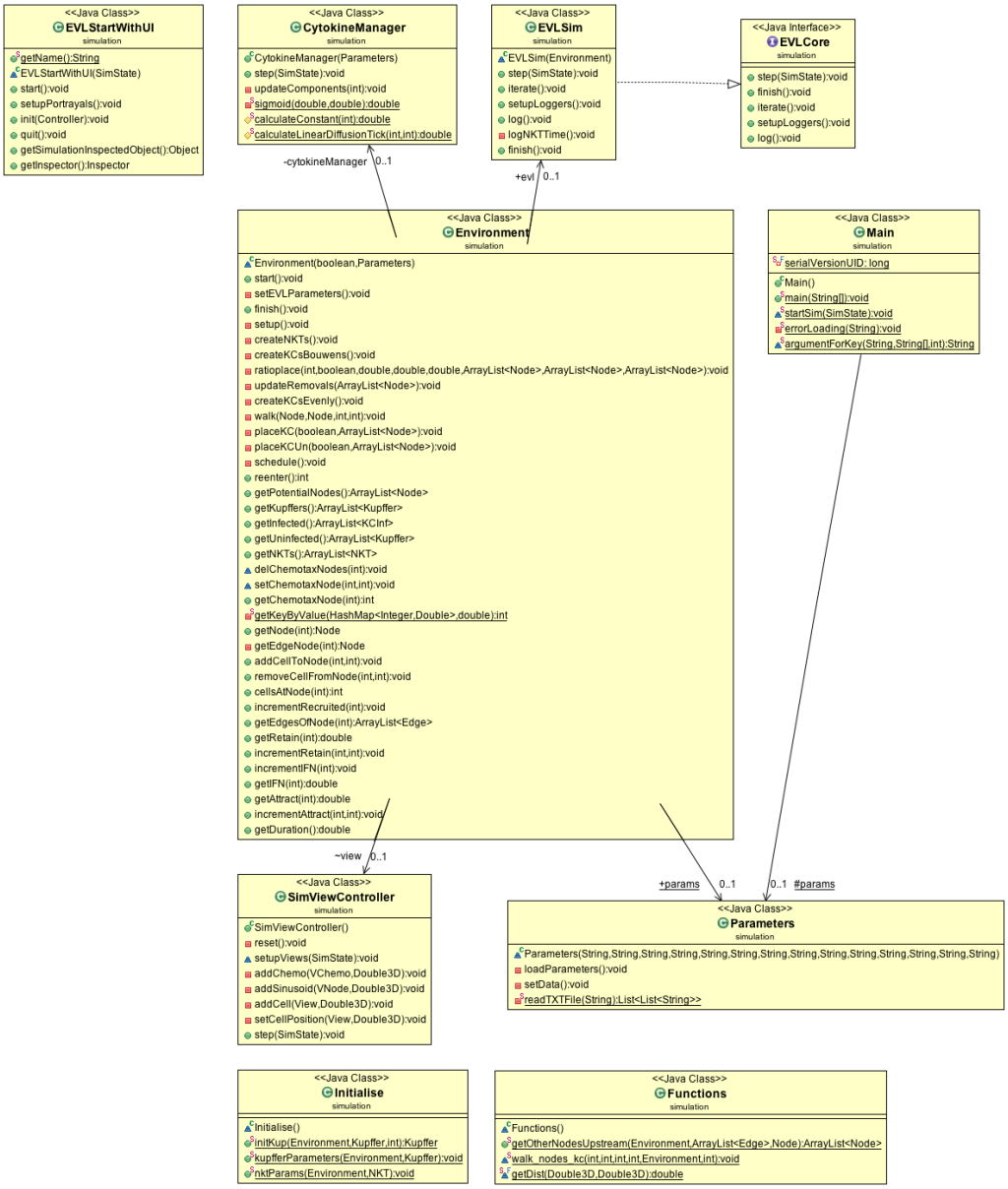


Figure 63: Package: Simulation

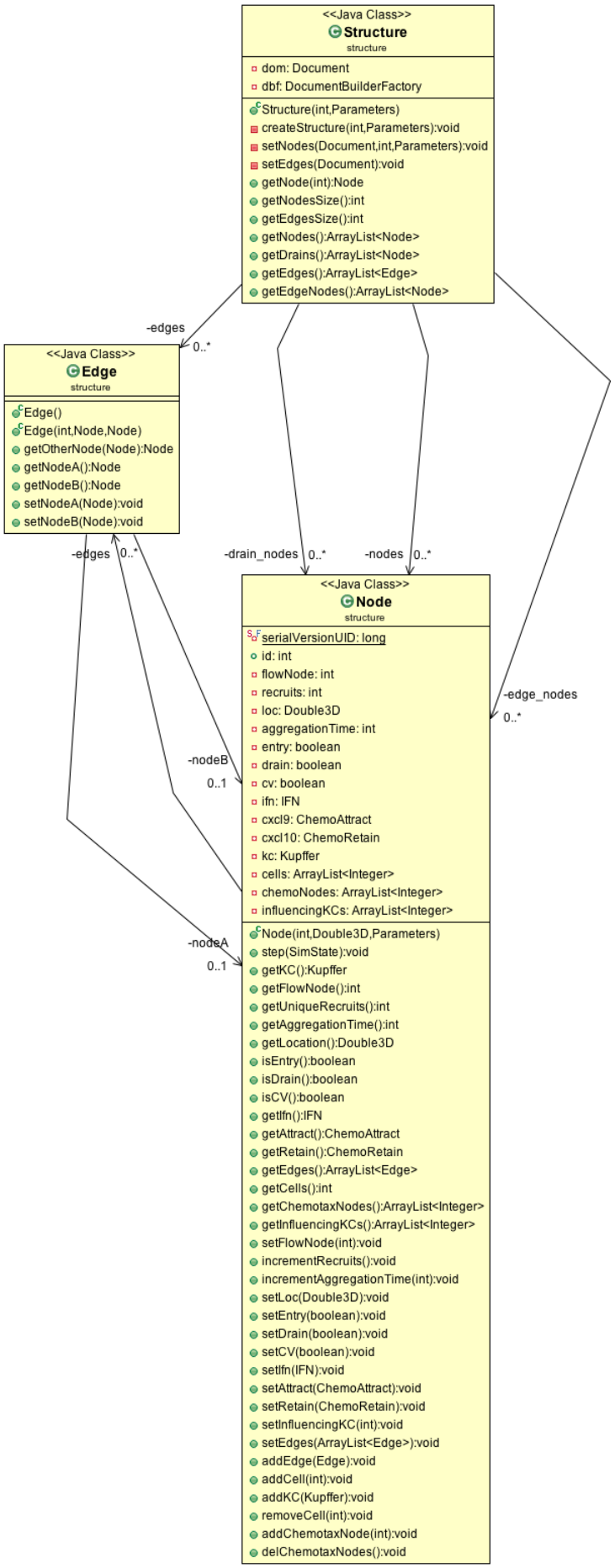


Figure 64: Package: Structure

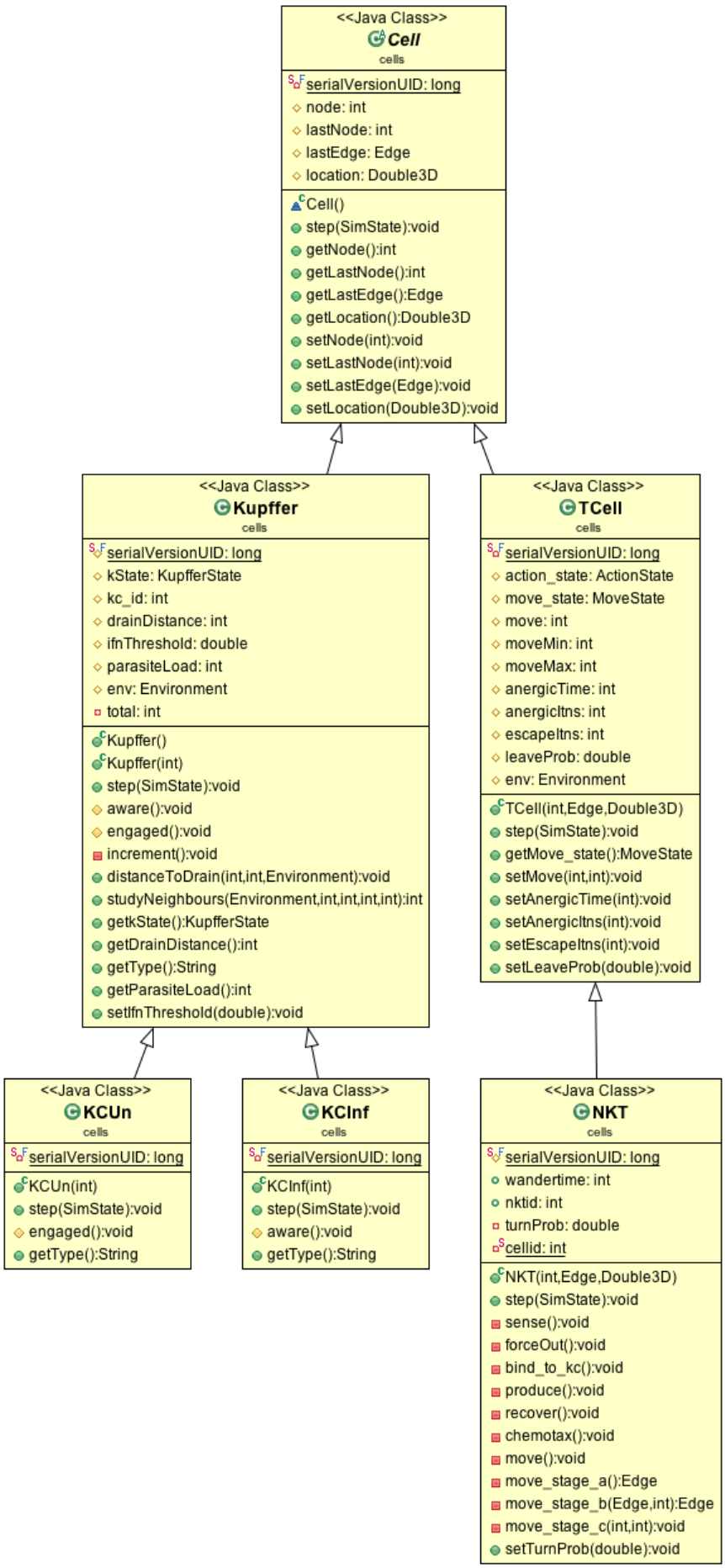


Figure 65: Package: Cells

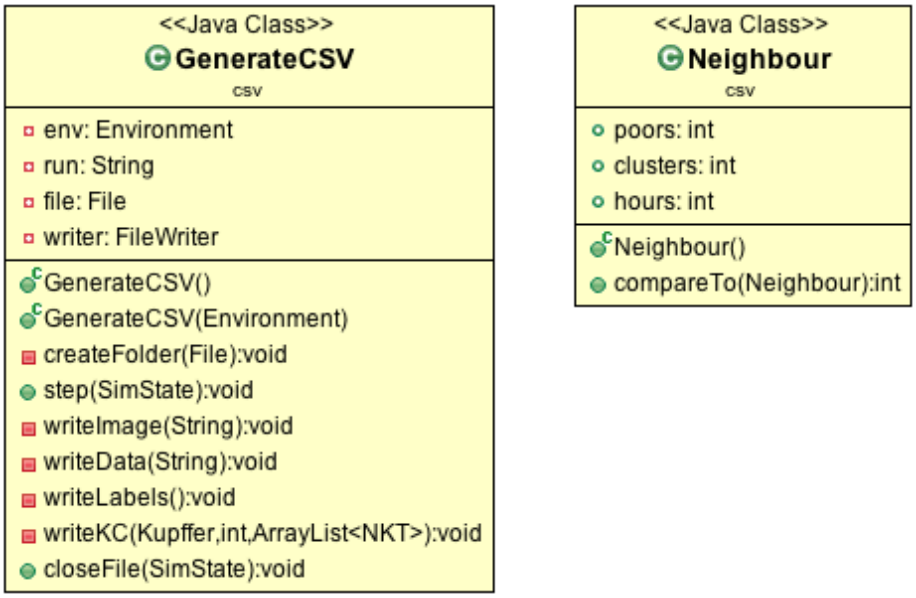


Figure 66: Package: CSV

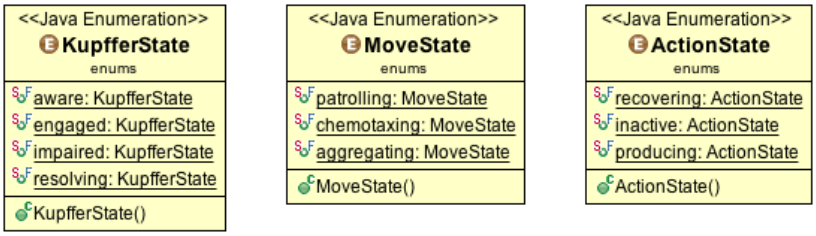


Figure 67: Package: Enums

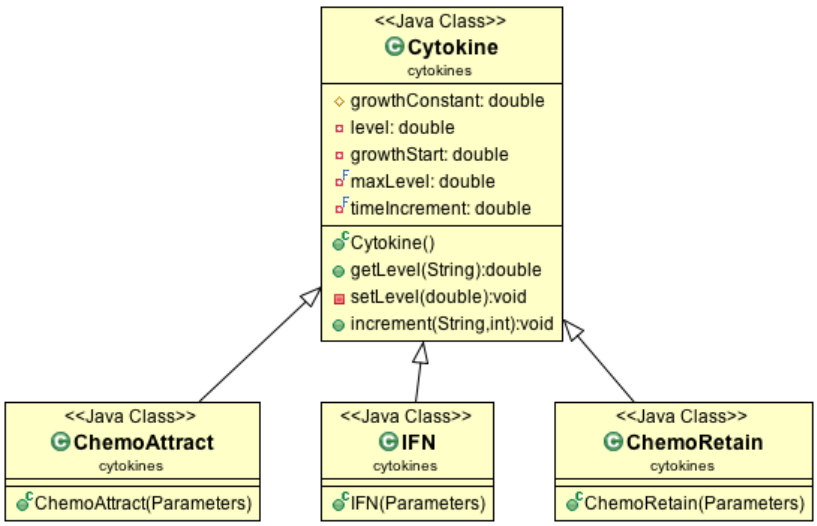


Figure 68: Package: Cytokines

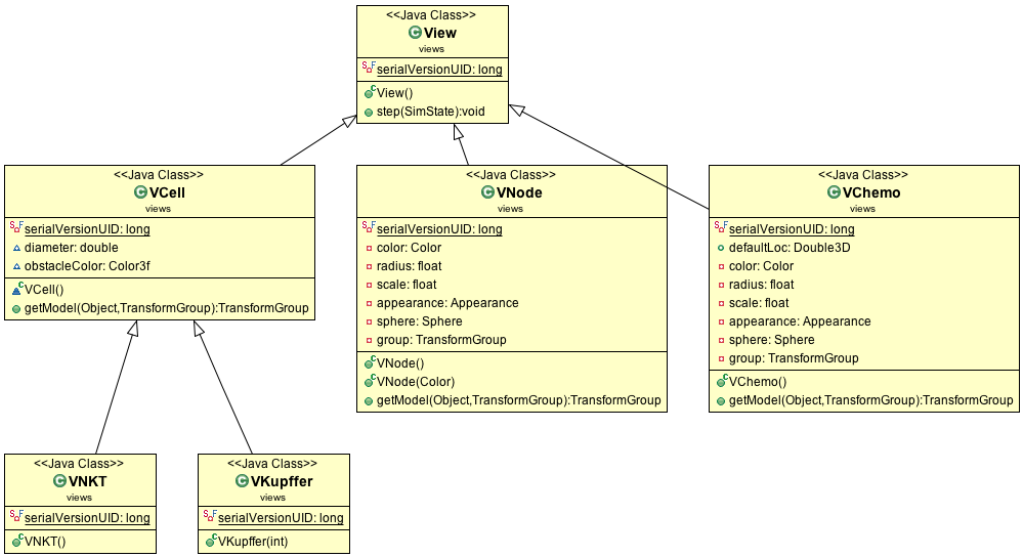


Figure 69: Package: Views

9.1.5 Algorithms

```
Input : A set A of candidate nodes
Input : Number of infected KCs numInfected
Input : Number of uninfected KCs numUninfected
viableCentrilobular ← []
viableMidzonal ← []
viablePeriportal ← []
for node in A do
    if distanceFromCentralVein(node) < 100 then
        viableCentilobular.append(node)
    else if distanceFromCentralVein(node) < 200 then
        viableMidzonal.append(node)
    else
        else viablePeriportal.append(node)
    end
end
forCV ← (int)Math.round(numInfected * ratioCV)
forMZ ← (int)Math.round(numInfected * ratioMZ)
forPP ← numInfected − forCV − forMZ for i in forCV do
    /* Place KC on a random viable node */
    removeNode ← placedKC(viableCentrilobular)
end
for i in forMZ do
    /* Place KC on a random viable node */ removeNode ← placedKC(viableMidzonal)
end
for i in forPP do
    /* Place KC on a random viable node */
    removeNode ← placedKC(viablePeriportal)
end
```

Algorithm 1 : KC placement algorithm.

```
Input : A set S of viable nodes
Input : A candidate node α
Input : Network tree N
Input : Minimum separation distance μ
T ← []
Recurse N up to distance μ from node α
        Add currentNode to T
Calculate set difference removing proximal nodes
Return to placement algorithm to generate a new candidate node
```

Algorithm 2 : KC placement algorithm (2).

1. Generate a potential node (μ) 1 μm from the current node (δ):
if *If we are within range of an attracting node (α)* **then**
| μ is generated in the direction of α (see figure 17b).
else
| μ is generated based on our current direction with a small random adjustment (see figure 17a).
end
2. Create a new segment between δ and another node:
if *The line between δ and μ intersects another sinusoid segment in the sinusoid network* **then**
| Connect δ to the existing node closest to μ (see figure 17d).
else
| Connect δ and μ (see figure 17c).
end
Repeat Steps 1 and 2 until an intersection is detected.

Algorithm 3 : Sinusoid branch generation

A drain node (representing the central vein) is placed in the centre of the 2D space, surrounded by six entry nodes (representing the locations of portal triads) in an irregular hexagon formation (see figure 17a). The exact location of the entry nodes is determined stochastically.
for *each of the six entry nodes* **do**
| grow a sinusoid branch (algorithm 3)) from the entry node towards the attracting drain node (see figure 17f).
end
An additional set of entry nodes is created for each original entry node and aligned with the original node. These additional nodes represent additional sources of blood supply coming out of the portal triad (see figure 17g).
while *mean sinusoid length of the entire structure reaches 43.1μm.* **do**
| Additional sinusoids are created to connect existing sinusoids (see figure 17i).
| Select the longest sinusoid in the structure.
| Select a node in the longest sinusoid and grow a sinusoid (see algorithm 1) to either the left or right at an angle drawn from a normal distribution with a mean of 32.5°.
end

Algorithm 4 : Sinusoid branch generation

9.2 APPENDIX DATA

The data in table 20 was acquired by our collaborators Amy Sawtell and Lynette Beattie, then processed to include the infection rates. I later analysed the data to produce the number of KCs per image section, required to parameterise the number of KCs in our simulated environment.

Experiment 1	2 hour infection											
	Mouse 1				Mouse 2				Mouse 3			
	Motile	Sessile	Infected M	Infected S	Motile	Sessile	Infected M	Infected S	Motile	Sessile	Infected M	Infected S
	8	4	3	1	3	4	0	1	9	2	2	0
	12	0	2	0	9	1	3	0	14	5	2	2
	15	1	3	0	0	3	0	1	22	6	7	1
	12	1	4	0	10	3	1	1	24	8	5	4
	5	10	0	2	10	2	1	2	16	3	8	0
	3	4	1	1	8	3	3	0	6	3	3	1
	3	4	1	0	1	11	0	2	10	4	2	2
	11	1	2	0	15	2	5	0	15	0	4	0
	9	5	0	2	6	4	1	2	18	13	4	3
	4	11	0	1	0	14	0	2	30	2	7	1
Totals	82	41	16	7	62	47	14	11	164	46	44	14
Percentage M/S			19.5	17.1			22.6	23.4			26.8	30.4
Total infected				23.0				25.0				58.0
Total infected %				18.7				22.9				27.6
Average infected %				23.1								
Motile KCs per image section				102.7								
Sessile KCs per image section				44.7								

Experiment 2	2 hour infection											
	Mouse 1				Mouse 2				Mouse 3			
	Motile	Sessile	Infected M	Infected S	Motile	Sessile	Infected M	Infected S	Motile	Sessile	Infected M	Infected S
	4	6	1	0	9	3	1	1	0	18	0	4
	5	9	3	3	6	5	1	1	3	11	1	1
	1	8	0	1	7	2	2	0	3	20	1	3
	3	8	0	2	9	4	1	1	2	10	0	3
	3	10	0	1	7	1	3	0	2	17	0	5
	2	6	0	1	5	4	1	3	0	20	0	4
	2	13	1	3	2	14	1	6	2	14	0	2
	4	10	1	3	8	9	3	2	5	16	0	4
	3	7	1	2	6	11	2	4	0	14	0	4
	2	1	0	2	10	3	1	0	0	14	0	3
Totals	29	78	7	18	69	56	16	18	17	154	2	33
Percentage M/S			24.1	23.1			23.2	32.1			11.8	21.4
Total infected				25.0				34.0				35.0
Total infected %				23.4				27.2				20.5
Average infected %				23.7								
Motile KCs per image section				38.3								
Sessile KCs per image section				96.0								

Table 20: Quantifying KC Numbers and Infection Rates.

9.3 QUANTIFYING GRANULOMA LOCATION IN VIVO

To quantify granuloma location *in vivo*, and to ascertain whether granulomas are more likely to form in the peripheries, I first needed to define the point of reference within the liver lobule. The obvious choice was the central vein of the lobule; however, there are challenges in accurate identification of central veins from mouse liver sections, which visually, are far removed from the idealised hexagonal structures illustrated in the literature. Without access to an expert histologist, I experimented to ascertain whether or not antibody staining of liver sections could clearly distinguish central veins from the likes of portal veins.

Mice were infected intravenously with $3x10^7$ *Leishmania donovani* amastigotes from the tandom Tomato fluorescent protein expressing Ethiopian strain (tdTom.LV9). After 18 days, livers were extracted, weighed and placed into 2% PFA in PBS for 2 hours, then 30% surcrose in PBS overnight. Tissues were then embedded in Optimal Cutting Temperature (OCT)(Sakura) and stored at -70°C until use. 20μm cryosections were fixed and labelled (see protocol in Appendix 9.3.1) anti-glutamine synthetase to identify centrilobular hepatocytes, and DAPI (Invitrogen) to visualise cell nuclei. Images were captured as 1μm optical slices using a LSM510 confocal microscope (Zeiss).

Whilst images such as figure 7 were acquired, which clearly identify a lobule section with Kupffer cells (green), leishmania parasites (red), cell nuclei (blue) and the AGS antibody stained central vein (white), other images proved harder to definitively categorise. The primary reason for this was the inconsistency in mice liver lobule structure. Often, central veins (assuming AGS identifies them correctly), can be found near side by side, distorting any imposed hexagonal structure, and thus making them difficult to use as a reference point for quantifying granuloma location.

9.3.1 AGS Staining Protocol

Immunofluorescent Labelling Protocol - Anti-glutamine Synthetase

- After cutting, allow slides to air dry for >30min - preferably overnight
- Encircle sections with 'Immedge pen' and allow to dry
- Wash slides x 3 in PBS - 5 min per wash
- Block slides in 5% serum, 30 minutes rest
  - Make up 1900 μl PBS to 100 μl Donkey serum
  - Vortex serum
  - Apply to slides
  - Wait 30 minutes
- Add primary antibody (anti-glutamine synthetase) at predeterminded dilution, diluted in 5% serum in PBS - 1 hour rest
  - Take 1 μl AGS and dilute into 999 μl Donkey serum
  - Take 200 μl from (a) and dilute into 800 μl Donkey serum
  - Take 100 μl from (b) and dilute in 100 μl Donkey serum
- Wash slides x3 in PBS - 5 min per wash
- Add secondary antibody (Donkey anti rabbit)
  - Take 1 μl of Donkey anti-rabbit and dilute in 499 μl PBS
  - Apply to slides
  - Wait 40 minutes
- Wash slides x3 in PBS - 5 min per wash
- Incubate sections in 1 μg/ml DAPI for 5min rest
- Wash slides x3 PBS
- Mount slides in Pro-long gold (molecular probes) and seal coverslips with nail polish the following day
- Store slides at 4◦C in the dark

10 GLOSSARY AND DEFINITIONS

ABM - Agent-based Modelling
APC - Antigen presenting cell
DAPI - 4',6-diamidino-2-phenylindole
Ex vivo - Latin for "out of the living"
EVL - Experimental Visceral Leishmaniasis
FACS - Fluorescence-activated cell sorting
IFN-γ - Interferon gamma
In silico - Performed using computer simulation
In situ - Latin for "in position"
In vivo - Latin for "within the living"
In vitro - Latin for "in glass"
KC - Kupffer Cell
LHS - Latin Hypercube Sampling
NKT - Natural Killer T cell
OCT - Optimal Cutting Temperature (Sakura)
ROI - Reactive oxygyon intermediates
PFA - Paraformaldehyde
PBS - Phosphate-buffered saline
rIL-15 - Recombinant Interleukin-15
SIRPα - Signal-regulatory protein alpha
TCR - T cell receptor
UML - Unified Modeling Language
VL - Visceral Leishmaniasis

BIBLIOGRAPHY

S. Adra, T. Sun, S. MacNeil, M. Holcombe, and R. Smallwood. Development of a Three Dimensional Multiscale Computational Model of the Human Epidermis. *PLoS ONE*, 5(1):e8511+, Jan 2010. doi: 10.1371/journal.pone.0008511.

B. D. Aguda, C. B. Marsh, M. Thacker, and E. D. Crouser. An In Silico Modeling Approach to Understanding the Dynamics of Sarcoidosis. *PLoS ONE*, 6(5): e19544, 2011.

L. Albergante. *A Petri Net Model of Liver Response to Visceral Leishmaniasis*. PhD thesis, Dipartimento di Matematica, Universita degli Studi di Milano, 2010.

L. Albergante, J. Timmis, P. S. Andrews, L. Beattie, and P. M. Kaye. A Petri Net Model of Granulomatous Inflammation. In E. Hart, C. McEwan, J. Timmis, and A. Hone, editors, *ICARIS*, volume 6209 of *Lecture Notes in Computer Science*, pages 1–3. Springer, 2010.

L. Albergante, J Timmis, L Beattie, , and P. M. Kaye. A Petri Net Model of Granulomatous Inflammation: Implications for IL-10-Mediated Control of Leishmania Donovani Infection. *PLoS Comput Biol*, 9(11), 2013. doi: 10.1371/journal.pcbi.1003334.

K. Alden, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. C. Coles. Pairing Experimentation and Computational Modeling to Understand the Role of Tissue Inducer Cells in the Development of Lymphoid Organs. *Frontiers in Immunology*, 3(July):172, 2012.

K. Alden, M. Read, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. Coles. Spartan: A Comprehensive Tool for Understanding Uncertainty in Simulations of Biological Systems. *PLoS Comput Biol*, 9(2):e1002916, 02 2013. doi: 10.1371/journal.pcbi.1002916.

K. J. Alden. *Simulation and Statistical Techniques to Explore Lymphoid Tissue Organogenesis*. PhD thesis, Department of Biology, University of York, Aug 2012.

J. Alvar, I. D. Vélez, C. Bern, M. Herrero, P. Desjeux, J. Cano, J. Jannin, and M. Den Boer. Leishmaniasis Worldwide and Global Estimates of its Incidence. *PLoS ONE*, 7(5):e35671, 2012.

G. An. Concepts for Developing a Collaborative In Silico Model of the Acute Inflammatory Response Using Agent-Based Modeling. *Journal of Critical Care*, 21(1):105–110; discussion 110–111, 2006.

G. An. Translational Systems Biology Using an Agent-Based Approach for Dynamic Knowledge Representation: An Evolutionary Paradigm for Biomedical Research. *Wound Repair and Regeneration*, 18(1):8–12, 2010.

G. An and S. Christley. Addressing the Translational Dilemma: Dynamic Knowledge Representation of Inflammation Using Agent-Based Modeling. *Crit Rev Biomed Eng*, 40(4):323–340, 2012.

P. S. Andrews, F. Polack, A. T. Sampson, J. Timmis, and M. Coles. Simulating Biology: Towards Understanding What the Simulation Shows. In *In: Proceedings of the 2008 Workshop on Complex Systems Modelling and Simulation*, pages 93–123. Luniver Press, 2008.

P. S. Andrews, F. A. C. Polack, A. T. Sampson, S. Stepney, and J. Timmis. The CoSMoS Process, Version 0.1: A Process for the Modelling and Simulation of Complex Systems. Technical report, Department of Computer Science, University of York, 2010.

P. S. Andrews, S. Stepney, Tim Hoverd, F. A. C. Polack, A. T Sampson, and J. Timmis. *CoSMoS Process, Models, and Metamodels*, pages 1–13. Luniver Press, 2011.

G. E. Arteel, R. G. Thurman, J. M. Yates, and J. A. Raleigh. Evidence that Hypoxia Markers Detect Oxygen Gradients in Liver: Pimonidazole and Retrograde Perfusion of Rat Liver. *Br. J. Cancer*, 72(4):889–895, Oct 1995.

R. Axelrod. *The Evolution of Cooperation.* Basic Books, Oct 1985. ISBN 0465021212.

G. C. Balan, C. Cioffi-Revilla, S. Luke, L. Panait, and S. Paus. MASON: A Java Multi-Agent Simulation Library. In *Proceedings of the Agent 2003 Conference*, 2003.

A. L. Bauer, C. A. Beauchemin, and A. S. Perelson. Agent-Based Modeling of Host-Pathogen Systems: The Successes and Challenges. *Inf Sci (Ny)*, 179(10): 1379–1389, Apr 2009.

B Bauer and J. Odell. UML 2.0 and Agents: How to Build Agent-based Systems With the New UML Standard. *Eng. Appl. of AI*, 18(2):141–157, 2005.

B. Bauer, J. P. Müller, J. Odell, Z. T. Ik, and D. München. Agent UML : A Formalism for Specifying Multiagent Interaction. *Agent Oriented Software Engineering*, 1957:91–103, 2001.

L. Beattie, A. Peltan, A. Maroof, A. Kirby, N. Brown, M. Coles, D. F. Smith, and P. M. Kaye. Dynamic Imaging of Experimental Leishmania donovani-Induced Hepatic Granulomas Detects Kupffer cell-Restricted Antigen Presentation to Antigen-Specific CD8+ T Cells. *PLoS Pathogens*, 6(3):17, 2010a.

L. Beattie, M. Svensson, A. Bune, N. Brown, A. Maroof, S. Zubairi, K. R. Smith, and P. M. Kaye. Leishmania Donovani-Induced Expression of Signal Regulatory

Protein α on Kupffer Cells Enhances Hepatic Invariant NKT-cell Activation. *European Journal of Immunology*, 40(1):117–123, 2010b.

L. Beattie, M. D'El-Rei Hermida, J. W. Moore, A. Maroof, N. Brown, D. Lagos, and P. M. Kaye. A Transcriptomic network Identified in Uninfected Macrophages Responding to Inflammation Controls Intracellular Pathogen Survival. *Cell Host Microbe*, 14(3):357–368, Sep 2013.

K. Beck. Embracing Change with Extreme Programming. *Computer*, 32(10): 70–77, 1999.

R. K. Benninger and D. W. Piston. Two-Photon Excitation Microscopy for the Study of Living Cells and Tissues. *Curr Protoc Cell Biol*, Chapter 4:Unit4.11, Jun 2013.

F. Bergmann, B. E. Shapiro, and M. Hucka. SBML Software Matrix. *http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix, Accessed 30 January 2013,*, Jun 2011.

H. Bersini. *Immune System Modeling: The OO Way*, volume 4163, pages 150–163. Springer, 2006.

H. Bersini. UML for ABM. *Journal of Artificial Societies and Social Simulation*, 15(1):9, 2012. ISSN 1460-7425.

H. Bersini, D. Klatzmann, A. Six, and V. Thomas-Vaslin. State-Transition Diagrams for Biologists. *PLoS ONE*, 7(7):e41165, 2012.

S. Bhattacharya, L. K. Shoda, Q. Zhang, C. G. Woods, B. A. Howell, S. Q. Siler, J. L. Woodhead, Y. Yang, P. McMullen, P. B. Watkins, and M. E. Andersen. Modeling Drug- and Chemical-Induced Hepatotoxicity with Systems Biology Approaches. *Front Physiol*, 3:462, 2012.

F. Biancuzzi and S. Warden. *Masterminds of Programming: Conversations with the Creators of Major Programming Languages*. O'Reilly, Beijing, 2009. ISBN 978-0-596-51517-1.

E. Bonabeau. Agent-Based Modeling: Methods and Techniques for Simulating Human Systems. *Proceedings of the National Academy of Sciences*, 99(3): 7280–7287, May 2002.

L. Bouwens, M. Baekeland, R. De Zanger, and E. Wisse. Quantitation, Tissue Distribution and Proliferation Kinetics of Kupffer Cells in Normal Rat Liver. *Hepatology*, 6(4):718–722, 1986.

J. Bown, P. S. Andrews, Y. Deeni, A. Goltsov, M. Idowu, F. A. Polack, A. T. Sampson, M. Shovman, and S. Stepney. Engineering Simulations for Cancer Systems Biology. *Curr Drug Targets*, 13(12):1560–1574, Nov 2012.

B. N. Brown, I. M. Price, F. R. Toapanta, D. R. DeAlmeida, C. A. Wiley, T. M. Ross, T. D. Oury, and Y. Vodovotz. An Agent-Based Model of Inflammation

and Fibrosis Following Particulate Exposure in the Lung. *Mathematical Biosciences*, 231(2):186–196, 2011.

J. Calvert and J. H. Fujimura. Calculating Life? Duelling Discourses in Interdisciplinary Systems Biology. *Stud Hist Philos Biol Biomed Sci*, 42(2):155–163, Jun 2011.

C. Carnaud, D. Lee, O. Donnars, S. H. Park, A. Beavis, Y. Koezuka, and A. Bendelac. Cutting Edge: Cross-Talk Between Cells of the Innate Immune System: NKT Cells Rapidly Activate NK Cells. *J. Immunol.*, 163(9):4647–4650, Nov 1999.

A. Chackerian, J. Alt, V. Perera, and S. M. Behar. Activation of NKT Cells Protects Mice from Tuberculosis. *Infect. Immun.*, 70(11):6302–6309, Nov 2002.

S. Coakley, R. Smallwood, and M. Holcombe. Using X-Machines as a Formal Basis for Describing Agents in Agent-Based Modelling. *Proceedings of the 2006 Agent-Directed Simulation Conference*, 2006.

A. Cockburn and J. Highsmith. Agile Software Development, The People Factor. *Computer*, 34(11):131–133, 2001.

I. R. Cohen and D. Harel. Explaining a Complex Living System: Dynamics, Multi-Scaling and Emergence. *Journal of the Royal Society Interface the Royal Society*, 4(13):175–182, 2007.

N. Collier and M. North. Parallel Agent-Based Simulation with Repast for High Performance Computing. *Simulation*, Nov 2012. doi: 10.1177/0037549712462620.

P. R. Crocker, J. M. Blackwell, and D. J. Bradley. Expression of the Natural Resistance Gene Lsh in Resident Liver Macrophages. *Infection and Immunity*, 43(3):1033–1040, 1984.

C. C. Cunningham and C. G. Van Horn. Energy Availability and Alcohol-Related Liver Pathology. *Alcohol Res Health*, 27(4):291–299, 2003.

G. M. Dancik, D. E. Jones, and K. S. Dorman. Parameter Estimation and Sensitivity Analysis in an Agent-Based Model of Leishmania Major Infection. *Journal of theoretical biology*, 262(3):398–412, Mar 2010. doi: 10.1016/j.jtbi.2009.10.007.

A. P. Dane, S. C. Cunningham, N. S. Graf, and I. E. Alexander. Sexually Dimorphic Patterns of Episomal rAAV Genome Persistence in the Adult Mouse Liver and Correlation with Hepatocellular Proliferation. *Mol. Ther.*, 17(9):1548–1554, Sep 2009.

C. Deissenberg, S. Van Der Hoog, and H. Dawid. EURACE: A Massively Parallel Agent-Based Model of the European Economy. *Applied Mathematics and Computation*, 204(2):541–552, Oct 2008. doi: 10.1016/j.amc.2008.05.116.

P. Desjeux. The Increase in Risk Factors for Leishmaniasis Worldwide. *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 95(3):239–243, 2001.

P. Desjeux. Leishmaniasis: Current Situation and New Perspectives. *Comp. Immunol. Microbiol. Infect. Dis.*, 27(5):305–318, Sep 2004.

P. Dhar, T. C. Meng, S. Somani, L. Ye, A. Sairam, M. Chitre, Z. Hao, and K. Sakharkar. Cellware–A Multi-Algorithmic Software for Computational Systems Biology. *Bioinformatics*, 20(8):1319–21, May 2004.

X. Dong, P. T. Foteinou, S. E. Calvano, S. F. Lowry, and I. P. Androulakis. Agent-Based Modeling of Endotoxin-Induced Acute Inflammatory Response in Human Blood Leukocytes. *PLoS ONE*, 5(2):13, 2010.

T. Dyba and T. Dingsoyr. Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, 50(9-10):833–859, 2008.

S. Efroni, D. Harel, and I. R. Cohen. Reactive Animation: Realistic Modeling of Complex Dynamic Systems. *Computer*, 38(1):38–47, 2005a.

S. Efroni, D. Harel, and I. R. Cohen. A Theory for Complex Systems: Reactive Animation. *Multidisciplinary Approaches to Theory in Medicine*, 3:309–324, 2005b.

J. G. Egen, A. G. Rothfuchs, C. G. Feng, N. Winter, A. Sher, and R. N. Germain. Macrophage and T cell Dynamics During the Development and Disintegration of Mycobacterial Granulomas. *Immunity*, 28(2):271–284, 2008.

J. G. Egen, A. G. Rothfuchs, C. G. Feng, M. A. Horwitz, A. Sher, and R. N. Germain. Intravital Imaging Reveals Limited Antigen Presentation and T cell Effector Function in Mycobacterial Granulomas. *Immunity*, 34(5):807–819, 2011.

C. R. Engwerda, M. Ato, S. Stäger, C. E. Alexander, A. C. Stanley, and P. M. Kaye. Distinct Roles for Lymphotoxin-α and Tumor Necrosis Factor in the Control of Leishmania Donovani Infection. *The American journal of pathology*, 165(6):2123–2133, 2004.

M. Fallahi-Sichani, M. A. Schaller, D. E. Kirschner, S. L. Kunkel, and J. J. Linderman. Identification of Key Processes that Control Tumor Necrosis Factor Availability in a Tuberculosis Granuloma. *PLoS Computational Biology*, 6 (5):19, 2010.

M. Fallahi-Sichani, M. El-Kebir, S. Marino, D. E. Kirschner, and J. J. Linderman. Multiscale Computational Modeling Reveals a Critical Role for TNF-α Receptor 1 Dynamics in Tuberculosis Granuloma Formation. *J. Immunol.*, 186(6): 3472–3483, Mar 2011.

M. Fallahi-Sichani, D. E. Kirschner, and J. J. Linderman. NF-κb Signaling Dynamics Play a Key Role in Infection Control in Tuberculosis. *Frontiers in physiology*, 3(June):170, 2012.

A. Finney, M. Hucka, B. Bornstein, S. Keating, B. E. Shapiro, J. Matthews, B. Kovitz, M. Schilstra, A. Funahashi, J. Doyle, and et al. Software Infrastructure for Effective Communication and Reuse of Computational Models. *System Modeling in Cellular Biology*, 2006.

A. J. Flügge, J. Timmis, P. Andrews, J. W. J. Moore, and P. Kaye. Modelling and Simulation of Granuloma Formation in Visceral Leishmaniasis. *2009 IEEE Congress on Evolutionary Computation*, pages 3052–3059, 2009.

S. Forrest and C. Beauchemin. Computer Immunology. *Immunological Reviews*, 216(10):176–97, 2007.

P. Garnett, S. Stepney, F. Day, and O. Leyser. Using the CoSMoS Process to Enhance an Executable Model of Auxin Transport Canalisation. In *Proceedings of the 2010 Workshop on Complex Systems Modelling and Simulation*, pages 9–32. Luniver Press, 2010.

F. Geissmann, T. O. Cameron, S. Sidobre, N. Manlongat, M. Kronenberg, M. J. Briskin, M. L. Dustin, and D. R. Littman. Intravascular Immune Surveillance by CXCR6+ NKT Cells Patrolling Liver Sinusoids. *PLoS Biol.*, 3(4):e113, Apr 2005.

R. N. Germain, M. Meier-Schellersheim, A. Nita-Lazar, and I. A. C. Fraser. Systems Biology in Immunology: A Computational Modeling Perspective. *Annual Review of Immunology*, 29(1):527–585, 2011.

T. Ghetiu, R. D. Alex, P. S. Andrews, F. A. C. Polack, and J. Bown. Equivalence Arguments for Complex Systems Simulations - A Case-Study. *CoSMoS Workshop, Luniver Press*, 2009.

T. Ghetiu, F. A. C. Polack, and J. Bown. Argument-Driven Validation of Computer Simulations - A Necessity Rather Than an Option. In *VALID 2010: The Second International Conference on Advances in System Testing and Validation Lifecycle*, pages 1–4. IEEE Press, 2010.

D. I. Godfrey, K. J. Hammond, L. D. Poulton, M. J. Smyth, and A. G. Baxter. NKT Cells: Facts, Functions and Fallacies. *Immunology today*, 21(11):573–83, 2000.

G. Gonzalez-Aseguinolaza, C. de Oliveira, M. Tomaska, S. Hong, O. Bruna-Romero, T. Nakayama, M. Taniguchi, A. Bendelac, L. Van Kaer, Y. Koezuka, and M. Tsuji. Alpha-Galactosylceramide-Activated Valpha-14 Natural Killer T Cells Mediate Protection Against Murine Malaria. *Proc. Natl. Acad. Sci. U.S.A.*, 97(15):8461–8466, Jul 2000.

R. B. Greaves, M. Read, J. Timmis, P. S. Andrews, J. A. Butler, B. O. Gerckens, and V. Kumar. In Silico Investigation of Novel Biological Pathways: The

Role of CD200 in Regulation of T cell Priming in Experimental Autoimmune Encephalomyelitis. *BioSystems*, 112(2):107–121, May 2013.

V. Grimm and S. F. Railsback. *Individual-Based Modeling and Ecology*. Princeton University Press, 2013.

V. Grimm, U. Berger, F. Bastiansen, S. Eliassen, V. Ginot, J. Giske, J. Goss-Custard, T. Grand, S. K. Heinz, G. Huse, and et al. A Standard Protocol for Describing Individual-based and Agent-based Models. *Ecological modelling*, 198(1):115–126, 2006.

V. Grimm, U. Berger, D. L. DeAngelis, J. G. Polhill, J. Giske, and S. F. Railsback. The ODD Protocol: A Review and First Update. *Ecological Modelling*, 221 (23):2760–2768, 2010.

A. B. Hailegiorgis, W. G. Kennedy, M. Rouleau, J. K. Bassett, M. Coletti, G. C. Balan, and T. Gulden. An Agent Based Model of Climate Change and Conflict Among Pastoralists in East Africa. *Conflict*, 2010.

D. Harel and E Gery. Executable Object Modeling with Statecharts. In *Proceedings of the 18th international conference on Software engineering*, pages 246–257. IEEE Computer Society, 1996.

D. Harel and H. Kugler. *The Rhapsody Semantics of Statecharts (or, On The Executable Core of the UML)*, volume 3147, pages 325–354. Springer, 2004.

R. Heiland, A. Shirinifard, M. Swat, G.L. Thomas, J. Sluka, A. Lumsdaine, B. Zaitlen, and J.A. Glazier. Visualizing Cells and Their Connectivity Graphs for CompuCell3D. In *Biological Data Visualization (BioVis), 2012 IEEE Symposium on*, pages 85–90, 2012.

S. Höhme, M. Brulport, A. Bauer, E. Bedawy, W. Schormann, M. Hermes, V. Puppe, R. Gebhardt, S. Zellmer, M. Schwarz, and et al. Prediction and Validation of Cell Alignment Along Microvessels as Order Principle to Restore Tissue Architecture in Liver Regeneration. *Proceedings of the National Academy of Sciences of the United States of America*, 107(23):10371–10376, 2010.

M. Holcombe, S. Adra, M. Bicak, S. Chin, S. Coakley, A. I. Graham, J. Green, C. Greenough, D. Jackson, M. Kiran, S. MacNeil, A. Maleki-Dizaji, P. McMinn, M. Pogson, R. Poole, E. Qwarnstrom, F. Ratnieks, M. D. Rolfe, R. Smallwood, T. Sun, and D. Worth. Modelling Complex Biological Systems Using an Agent-Based Approach. *Integr Biol (Camb)*, 4(1):53–64, Jan 2012.

M. Holub, C-W. Cheng, S. Mott, P. Wintermeyer, N. Van Rooijen, and S. H. Gregory. Neutrophils Sequestered in the Liver Suppress the Proinflammatory Response of Kupffer Cells to Systemic Bacterial Infection. *The Journal of Immunology*, 183(5):3309–3316, 2009.

M. Hommel. Visceral Leishmaniasis: Biology of the Parasite. *J. Infect.*, 39(2): 101–111, Sep 1999.

S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer. COPASI–A COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–74, Dec 2006.

S. Huang. The Practical Problems of Post-Genomic Biology. *Nat. Biotechnol.*, 18(5):471–472, May 2000.

M. Hucka. The Systems Biology Markup Language (SBML): A Medium for Representation and Exchange of Biochemical Network Models. *Bioinformatics*, 19(4):524–531, Mar 2003.

D. A. Hume, D. Halpin, H. Charlton, and S. Gordon. The Mononuclear Phagocyte System of the Mouse Defined by Immunohistochemical Localization of Antigen F4/80: Macrophages of Endocrine Organs. *Proceedings of the National Academy of Sciences of the United States of America*, 81(13):4174–4177, 1984.

T. Ideker, T. Galitski, and L. Hood. A New Approach to Decoding Life: Systems Biology. *Annu Rev Genomics Hum Genet*, 2:343–372, 2001.

H. Ishibashi, M. Nakamura, A. Komori, K. Migita, and S. Shimoda. Liver Architecture, Cell Function, and Disease. *Semin Immunopathol*, 31(3):399–409, Sep 2009.

H. Ishikawa, H. Hisaeda, M. Taniguchi, T. Nakayama, T. Sakai, Y. Maekawa, Y. Nakano, M. Zhang, T. Zhang, M. Nishitani, and et al. CD4(+) V(alpha)14 NKT Cells Play a Crucial Role in an Early Stage of Protective Immunity Against Infection with Leishmania Major. *International Immunology*, 12(9): 1267–1274, 2000.

T. Iyoda, M. Ushida, Y. Kimura, K. Minamino, A. Hayuka, S. Yokohata, H. Ehara, and K. Inaba. Invariant NKT cell Anergy is Induced by a Strong TCR-Mediated Signal Plus Co-stimulation. *Int. Immunol.*, 22(11):905–913, Nov 2010.

K. Jungermann. Metabolic Zonation of Liver Parenchyma. *Semin. Liver Dis.*, 8(4):329–341, Nov 1988.

K. Jungermann and N. Katz. Functional Specialization of Different Hepatocyte Populations. *Physiol. Rev.*, 69(3):708–764, Jul 1989.

J. A. Juno, Y. Keynan, and K. R. Fowke. Invariant NKT Cells: Regulation and Function During Viral Infection. *PLoS Pathog.*, 8(8):e1002838, 2012.

P. Kaye and P. Scott. Leishmaniasis: Complexity at the Host–Pathogen Interface. *Nature Reviews Microbiology*, 9(8):604–615, 2011.

T. P. Kelly. *Arguing Safety – A Systematic Approach to Managing Safety Cases*. PhD thesis, Department of Computer Science, University of York, 1998.

Y. Kim and P. Rajagopalan. 3D Hepatic Cultures Simultaneously Maintain Primary Hepatocyte and Liver Sinusoidal Endothelial cell Phenotypes. *PLoS ONE*, 5(11):e15456, 2010.

H. Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, 2002. doi: 10.1126/science.1069492.

H. Kitano. A Graphical Notation for Biochemical Networks. *Biosilico*, 1(5): 169–176, Nov 2003.

I. Klein, J. C. Cornejo, N. K. Polakos, B. John, S. A. Wuensch, D. J. Topham, R. H. Pierce, and I. N. Crispe. Kupffer cell Heterogeneity: Functional Properties of Bone Marrow–Derived and Sessile Hepatic Macrophages. *Blood*, 110 (12):4077–4085, 2007.

K. W. Kohn, M. I. Aladjem, J. N. Weinstein, and Y. Pommier. Molecular Interaction Maps of Bioregulatory Networks: A General Rubric for Systems Biology. *Molecular Biology of the Cell*, 17(1):1–13, Jan 2006.

F. Kolpakov and M. Puzanov. BioUML: Visual Modeling, Automated Code Generation and Simulation of Biological Systems. In *Proceedings of the 5th international conference on Bioinformatics of Genome Regulation and Structure (BGRS)*, pages 281–284, 2006.

M. Kronenberg and L. Gapin. The Unconventional Lifestyle of NKT Cells. *Nat. Rev. Immunol.*, 2(8):557–568, Aug 2002.

D. Kucharavy and R. De Guio. Application of s-shaped curves. pages 81–88, 2007.

C. Larman and V. R. Basili. Iterative and Incremental Developments. A Brief History. *Computer*, 36(6):47–56, 2003.

W. Y. Lee, T. J. Moriarty, C. H. Wong, H. Zhou, R. M. Strieter, N. van Rooijen, G. Chaconas, and P. Kubes. An Intravascular Immune Response to Borrelia Burgdorferi Involves Kupffer Cells and iNKT Cells. *Nat. Immunol.*, 11(4): 295–302, Apr 2010.

D. A. Lepay, C. F. Nathan, R. M. Steinman, H. W. Murray, and Z. A. Cohn. Murine Kupffer Cells. Mononuclear Phagocytes Deficient in the Generation of Reactive Oxygen Intermediates. *The Journal of Experimental Medicine*, 161 (5):1079–1096, 1985a.

D. A. Lepay, R. M. Steinman, C. F. Nathan, H. W. Murray, and Z. A. Cohn. Liver Macrophages in Murine Listeriosis. Cell-Mediated Immunity is Correlated with an Influx of Macrophages Capable of Generating Reactive Oxygen Intermediates. *The Journal of Experimental Medicine*, 161(6):1503–1512, 1985b.

N. Y. Li, K. Verdolini, G. Clermont, Q. Mi, E. N. Rubinstein, P. A. Hebda, and Y. Vodovotz. A Patient-Specific In Silico Model of Inflammation and Healing Tested in Acute Vocal Fold Injury. *PLoS ONE*, 3(7):e2789, 2008.

Y. Li, A. Sampson, J. Bown, and Y. Deeni. Understanding Tissue Morphology: Model Repurposing Using the CoSMoS Process. In S. Stepney and P. S. Andrews, editors, *Proceedings of the 2013 Workshop on Complex Systems Modelling and Simulation*, pages 73–91. Luniver Press, Jul 2013. ISBN 978-1-905986-39-2.

S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. MASON: A New Multi-Agent Simulation Toolkit. *Proceedings of the 2004 SwarmFest Workshop*, 8 (2):316–327, 2004.

C. Macilwain. Systems Biology: Evolving into the Mainstream. *Cell*, 144(6): 839–841, Mar 2011.

S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner. A Methodology for Performing Global Uncertainty and Sensitivity Analysis in Systems Biology. *J. Theor. Biol.*, 254(1):178–196, Sep 2008.

S. Marino, A. Myers, J. L. Flynn, and D. E. Kirschner. TNF and IL-10 are Major Factors in Modulation of the Phagocytic cell Environment in Lung and Lymph Node in Tuberculosis: A Next-Generation Two-Compartmental Model. *Journal of Theoretical Biology*, 265(4):586–598, 2010.

A. Maroof, L. Beattie, S. Zubairi, M. Svensson, S. Stager, and P. M. Kaye. Post-transcriptional Regulation of IL-10 Gene Expression Allows Natural Killer Cells to Express Immunoregulatory Function. *Immunity*, 29(2):295–305, Aug 2008.

J. L. Matsuda, L. Gapin, S. Sidobre, W. C. Kieper, J. T. Tan, R. Ceredig, C. D. Surh, and M. Kronenberg. Homeostasis of V alpha 14i NKT Cells. *Nat. Immunol.*, 3(10):966–974, Oct 2002.

S. R. Mattarollo, A. C. West, K. Steegh, H. Duret, C. Paget, B. Martin, G. M. Matthews, J. Shortt, M. Chesi, P. L. Bergsagel, M. Bots, J. Zuber, S. W. Lowe, R. W. Johnstone, and M. J. Smyth. NKT cell Adjuvant-Based Tumor Vaccine for Treatment of Myconcogene-Driven mouse B-cell Lymphoma. *Blood*, 120 (15):3019–3029, Oct 2012.

M. J. McElrath, H. W. Murray, and Z. A. Cohn. The Dynamics of Granuloma Formation in Experimental Visceral Leishmaniasis. *J. Exp. Med.*, 167(6): 1927–1937, Jun 1988.

E. McFarlane, K. C. Carter, A. N. McKenzie, P. M. Kaye, F. Brombacher, and J. Alexander. Endogenous IL-13 Plays a Crucial Role in Liver Granuloma Maturation During Leishmania Donovani Infection, Independent of IL-4Rα-Responsive Macrophages and Neutrophils. *J. Infect. Dis.*, 204(1):36–43, Jul 2011.

M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):pp. 239–245, 1979.

C. L. Meier, M. Svensson, and P. M. Kaye. Leishmania-Induced Inhibition of Macrophage Antigen Presentation Analyzed at the Single-cell Level. *The Journal of Immunology*, 171(12):6706–6713, 2003.

Q. Mi, B. Rivière, G. Clermont, D. L. Steed, and Y. Vodovotz. Agent-Based Model of Inflammation and Wound Healing: Insights into Diabetic Foot Ulcer Pathology and the Role of Transforming Growth Factor-beta1. *Wound repair and regeneration official publication of the Wound Healing Society and the European Tissue Repair Society*, 15(5):671–82, 2007.

Q. Mi, N. Y. Li, C. Ziraldo, A. Ghuma, M. Mikheev, R. Squires, D. O. Okonkwo, K. Verdolini-Abbott, G. Constantine, G. An, and Y. Vodovotz. Translational Systems Biology of Inflammation: Potential Applications to Personalized Medicine. *Per Med*, 7(5):549–559, Sep 2010.

N. Minar, R. Burkhart, C. Langton, and M. Askenazi. The Swarm Simulation System: A Toolkit for Building Multi-Agent Simulations. Santa Fe Institute, 1996.

S. Mochida, I. Ogata, Y. Ohta, S. Yamada, and K. Fujiwara. In Situ Evaluation of the Stimulatory State of Hepatic Macrophages Based on their Ability to Produce Superoxide Anions in Rats. *The Journal of pathology*, 158(1):67–71, 1989.

J. W. Moore, D. Moyo, L. Beattie, P. S. Andrews, J. Timmis, and P. M. Kaye. Functional Complexity of the Leishmania Granuloma and the Potential of In Silico Modeling. *Front Immunol*, 4:35, 2013.

J. W. J. Moore, L. Beattie, J. E. Dalton, B. M. J. Owens, A. Maroof, M. Coles, and P. M. Kaye. B cell: T Cell Interactions Occur Within Hepatic Granulomas During Experimental Visceral Leishmaniasis. *PLoS ONE*, 7(3):e34143, 2012.

D. M. Mosser and J. P. Edwards. Exploring the Full Spectrum of Macrophage Activation. *Nature Reviews Immunology*, 8(12):958–969, 2008.

K. Murphy, P. Travers, M. Walport, and C. Janeway. *Janeway's Immunobiology*. Garland Science, New York, 2008.

H. W. Murray. Tissue Granuloma Structure-Function in Experimental Visceral Leishmaniasis. *International Journal of Experimental Pathology*, 82(5):249–267, 2001.

H. W. Murray, A. Jungbluth, E. Ritter, C. Montelibano, and M. W. Marino. Visceral Leishmaniasis in Mice Devoid of Tumor Necrosis Factor and Response to Treatment. *Infect. Immun.*, 68(11):6289–6293, Nov 2000.

F. Mussai, C. De Santo, and V. Cerundolo. Interaction Between Invariant NKT Cells and Myeloid-Derived Suppressor Cells in Cancer Patients: Evidence and Therapeutic Opportunities. *J. Immunother.*, 35(6):449–459, Jul 2012.

M. Naito, G. Hasegawa, Y. Ebe, and T. Yamamoto. Differentiation and Function of Kupffer Cells. *Medical electron microscopy official journal of the Clinical Electron Microscopy Society of Japan*, 37(1):16–28, 2004.

R. E. Nance and R. G. Sargent. Perspectives on the Evolution of Simulation. *Operations Research*, 50(1):161–172, 2002.

NC3Rs. What are the 3rs?, Jun Accessed 29th July 2013. URL http://www.nc3rs.org.uk/page.asp?id=7.

S. Nerur, R. Mahapatra, and G. Mangalaraj. Challenges of Migrating to Agile Methodologies. *Communications of the ACM*, 48(5):72–78, 2005.

C. Nikolai and G. Madey. Tools of the Trade : A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 12(2), 2009.

M. J. North, N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. Complex Adaptive Systems Modeling with Repast Simphony. *Complex Adaptive Systems Modeling*, (1):3, 2013. doi: 10.1186/2194-3206-1-3.

M. Oheim, D. J. Michael, M. Geisbauer, D. Madsen, and R. H. Chow. Principles of Two-Photon Excitation Fluorescence Microscopy and Other Nonlinear Imaging Approaches. *Adv. Drug Deliv. Rev.*, 58(7):788–808, Sep 2006.

T. Ohteki, H. Tada, K. Ishida, T. Sato, C. Maki, T. Yamada, J. Hamuro, and S. Koyasu. Essential Roles of DC-Derived IL-15 as a Mediator of Inflammatory Responses In Vivo. *The Journal of Experimental Medicine*, 203(10):2329–2338, 2006.

OMG®. Unified Modeling Language Infrastructure Specification Version 2.4, November 2010. URL http://www.omg.org/spec/UML/2.4.1/.

OMG®. Introduction to OMG's Unified Modeling Language TM, Apr 2012. URL http://www.omg.org/gettingstarted/what_is_{uml}.htm.

S. R. Palmer and J. M. Felsing. *A Practical Guide to Feature-Driven Development*. The Coad Series. Prentice Hall PTR, Feb 2002.

L. Panait and S. Luke. A Pheromone-Based Utility Model for Collaborative Foraging. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 36–43. IEEE Computer Society, 2004.

A. Patel, N. Harker, L. Moreira-Santos, M. Ferreira, K. Alden, J. Timmis, K. Foster, A. Garefalaki, P. Pachnis, P. Andrews, H. Enomoto, J. Milbrandt, V. Pachnis, M. C. Coles, D. Kioussis, and H. Veiga-Fernandes. Differential RET Signaling Pathways Drive Development of the Enteric Lymphoid and Nervous Systems. *Sci Signal*, 5(235):ra55, 2012.

R. D. Pearson and A. d. Q. Sousa. Clinical Spectrum of Leishmaniasis. *Clin Infect Dis.*, 22:1–13, Jun 1996.

D. Phan and F. Varenne. Agent-Based Models and Simulations in Economics and Social Sciences: From Conceptual Exploration to Distinct Ways of Experimenting. *Journal of Artificial Societies and Social Simulation*, 13(1):5, 2010.

K. A. Pilones, J. Aryankalayil, and S. Demaria. Invariant NKT Cells as Novel Targets for Immunotherapy in Solid Tumors. *Clin. Dev. Immunol.*, 2012: 720803, 2012.

D. W. Piston. Imaging Living Cells and Tissues by Two-Photon Excitation Microscopy. *Trends Cell Biol.*, 9(2):66–69, Feb 1999.

J. Pitt-Francis, M. O. Bernabeu, J. Cooper, A. Garny, L. Momtahan, J. Osborne, P. Pathmanathan, B. Rodriguez, J. P. Whiteley, and D. J. Gavaghan. Chaste: Using Agile Programming Techniques to Develop Computational Biology Software. *Philos Trans A Math Phys Eng Sci*, 366(1878):3111–3136, Sep 2008.

M. Pogson, R. Smallwood, E. Qwarnstrom, and M. Holcombe. Formal Agent-Based Modelling of Intracellular Chemical Interactions. *BioSystems*, 85(1): 37–45, Jul 2006.

M. Pogson, M. Holcombe, R. Smallwood, and E. Qwarnstrom. Introducing Spatial Information into Predictive NF-kappaB Modelling–An Agent-Based Approach. *PLoS ONE*, 3(6):e2367, 2008.

F. Polack. Argumentation and the Design of Emergent Systems. *Department of Computer Science, University of York, working paper, available at www-users.cs.york.ac.uk/fiona/PUBS/Arguments.pdf*, 2008.

F. Polack, A. Droop, P. Garnett, T. Ghetiu, and S. Stepney. Simulation Validation: Exploring the Suitability of a Simulation of Cell Division and Differentiation in the Prostate. In S. Stepney, P. S. Andrews, C. Ritson, and P. Weltch, editors, *CoSMoS Workshop*. Luniver Press, 2011.

F. A. C. Polack. Arguing Validation of Simulations in Science. In *Workshop on Complex Systems Modelling and Simulation*, pages 51–74, 2010.

F. A. C. Polack, P. S. Andrews, and A. T. Sampson. The Engineering of Concurrent Simulations of Complex Systems. *2009 IEEE Congress on Evolutionary Computation*, (3098):217–224, 2009.

F. A. C. Polack, P. S. Andrews, T. Ghetiu, M. Read, S. Stepney, J. Timmis, and A. T. Sampson. Reflections on the Simulation of Complex Systems for Science. In *ICECCS 2010: Fifteenth IEEE International Conference on Engineering of Complex Computer Systems*, pages 276–285. IEEE Press, Mar 2010. ISBN 978-0-7695-4015-3.

J. J. Pothen, M. E. Poynter, and J. H. Bates. The Inflammatory Twitch as a General Strategy for Controlling the Host Response. *J. Immunol.*, 190(7): 3510–3516, Apr 2013.

S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent Based Simulation Platforms: Review and Development Recommendations. *Simulation*, 8(9):609–623, 2005.

L. Ramakrishnan. Revisiting the Role of the Granuloma in Tuberculosis. *Nat. Rev. Immunol.*, 12(5):352–366, May 2012.

J. C. J. Ray, J. L. Flynn, and D. E. Kirschner. Synergy Between Individual TNF-Dependent Functions Determines Granuloma Performance for Controlling Mycobacterium Tuberculosis Infection. *The Journal of Immunology*, 182(6):3706–3717, 2009.

M. Read, P. S. Andrews, J. Timmis, and V. Kumar. A Domain Model of Experimental Autoimmune Encephalomyelitis. In *2nd Workshop on Complex Systems Modelling and Simulation*, pages 9–44, 2009a.

M. Read, J. Timmis, P. S. Andrews, and V. Kumar. Using UML to Model EAE and its Regulatory Network. In P. S. Andrews, J. Timmis, N. D. L. Owens, U. Aickelin, E. Hart, A. Hone, and A. M. Editors Tyrrell, editors, *ICARIS*, volume 5666 of *Lecture Notes in Computer Science*, pages 4–6. Springer, 2009b. ISBN 978-3-642-03245-5.

M. Read, P. S. Andrews, J. Timmis, and V. Kumar. Techniques for Grounding Agent-Based Simulations in the Real Domain : A Case Study in Experimental Autoimmune Encephalomyelitis. *Mathematical and Computer Modelling of Dynamical Systems*, 18(1):67–86, 2011.

M. N. Read. *Statistical and Modelling Techniques to Build Confidence in the Investigation of Immunology Through Agent-Based Simulation.* PhD thesis, Department of Computer Science, University of York, Sep 2011.

T. Reenskaug. Models - Views - Controllers. *Technical note Xerox PARC*, (December):1–2, 1979.

T. Reenskaug. The Model-View-Controller (MVC) Its Past and Present. *University of Oslo Draft*, (Mvc):1–16, 2003.

A. Reynolds, J. Rubin, G. Clermont, J. Day, Y. Vodovotz, and G. B. Ermentrout. A Reduced Mathematical Model of the Acute Inflammatory Response: I. Derivation of Model and Analysis of Anti-inflammation. *Journal of Theoretical Biology*, 242(1):220 – 236, 2006. doi: http://dx.doi.org/10.1016/j.jtbi.2006.02.016.

P. Richmond and D. Romano. Agent Based GPU, A Real-Time 3D Simulation and Interactive Visualisation Framework for Massive Agent Based Modelling on the GPU. *Proceedings of International Workshop on Supervisualisation 2008 (IWSV08, Part of ICS08) Kos Island, Greece*, 2008.

P. Richmond, S. Coakley, and D. Romano. A High Performance Agent Based Modelling Framework on Graphics Card Hardware With CUDA. *Proc. of*

8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009), May, 10-15, 2009, Budapest, Hungary, 2009.

F. Robert-Gangneux, A. S. Drogoul, O. Rostan, C. Piquet-Pellorce, J. Cayon, M. Lisbonne, A. Herbelin, H. Gascan, C. Guiguen, M. Samson, and J. P. Gangneux. Invariant NKT Cells Drive Hepatic Cytokinic Microenvironment Favoring Efficient Granuloma Formation and Early Control of Leishmania Donovani Infection. *PLoS ONE*, 7(3):e33413, 2012.

W. M. S. Russel and R. L. Burch. *The Principles of Humane Experimental Technique*. Methuen London, 1959.

M. A. Sanchez, N. L. Diaz, O. Zerpa, E. Negron, J. Convit, and F. J. Tapia. Organ-Specific Immunity in Canine Visceral Leishmaniasis: Analysis of Symptomatic and Asymptomatic Dogs Naturally Infected with Leishmania Chagasi. *Am. J. Trop. Med. Hyg.*, 70(6):618–624, Jun 2004.

R. G. Sargent. Verification and validation of Simulation Models. In *Proceedings of the 30th conference on Winter simulation*, pages 121–130. IEEE Computer Society Press, 1998.

R. G. Sargent. Verification and Validation of Simulation Models. In *Simulation Conference, 2007 Winter*, pages 124–137, 2007.

T. Sato, H. Thorlacius, B. Johnston, T. L. Staton, W. Xiang, D. R. Littman, and E. C. Butcher. Role for CXCR6 in recruitment of activated CD8+ lymphocytes to inflamed liver. *J. Immunol.*, 174(1):277–283, Jan 2005.

R. Saxena, N. D. Theise, and J. M. Crawford. Microanatomy of the Human Liver-Exploring the Hidden Interfaces. *Hepatology*, 30(6):1339–1346, Dec 1999.

W. Schroeder, K. Martin, and B. Lorensen. The Visualization Toolkit: An Object Oriented Approach to 3d Graphics 3rd Edition, Kitware. *Inc. Publisher*, 2003.

K. Schwaber. Scrum Development Process. In *Business Object Design and Implementation*, pages 117–134. Springer, 1997.

K. Schwaber and M. Beedle. *Agile Software Development with Scrum*, volume 18. Prentice Hall, 2001.

SDG. (swarm development group) swarm homepage, accessed september, 2013. URL http://www.swarm.org/.

J. L. Segovia-Juarez, S. Ganguli, and D. Kirschner. Identifying Control Mechanisms of Granuloma Formation During M. Tuberculosis Infection Using an Agent-Based Model. *Journal of Theoretical Biology*, 231(3):357–376, 2004.

K. Seino and M. Taniguchi. Functionally Distinct NKT cell Subsets and Subtypes. *J. Exp. Med.*, 202(12):1623–1626, Dec 2005.

E. C. Sleyster and D. L. Knook. Relation Between Localization and Function of Rat Liver Kupffer Cells. *Laboratory investigation a journal of technical methods and pathology*, 47(5):484–490, 1982.

I. Sommerville. *Software Engineering: (9th Edition)*. Addison Wesley, 8 edition, Jun 2009.

S. Stäger, J. Alexander, K. C. Carter, F. Brombacher, and P. M. Kaye. Both Interleukin-4 (IL-4) and IL-4 Receptor Alpha Signaling Contribute to the Development of Hepatic Granulomas with Optimal Antileishmanial Activity. *Infection and Immunity*, 71(8):4804–4807, 2003.

A. C. Stanley and C. R. Engwerda. Balancing Immunity and Pathology in Visceral Leishmaniasis. *Immunology and Cell Biology*, 85(2):138–147, 2007.

A. C. Stanley, Y. Zhou, F. H. Amante, L. M. Randall, A. Haque, D. G. Pellicci, G. R. Hill, M. J. Smyth, D. I. Godfrey, and C. R. Engwerda. Activation of Invariant NKT Cells Exacerbates Experimental Visceral Leishmaniasis. *PLoS Pathogens*, 4(2):14, 2008.

J. Stapleton. *DSDM: Business Focused Development 2nd Edition*. Pearson Education, 2nd edition, 2003. ISBN 978-0321112248.

J. J. Stern, M. J. Oca, B. Y. Rubin, S. L. Anderson, and H. W. Murray. Role of L3T4+ and LyT-2+ Cells in Experimental Visceral Leishmaniasis. *J. Immunol.*, 140(11):3971–3977, Jun 1988.

J. R. Stern, S. Christley, O. Zaborina, J. C. Alverdy, and G. An. Integration of TGF-β and EGFR-Based Signaling Pathways Using an Agent-Based Model of Epithelial Restitution. *Wound Repair Regen*, 20(6):862–871, 2012.

J. R. Stern, A. D. Olivas, V. Valuckaite, O. Zaborina, J. C. Alverdy, and G. An. Agent-Based Model of Epithelial Host-Pathogen Interactions in Anastomotic Leak. *J. Surg. Res.*, Jan 2013.

T. Sun, P. McMinn, S. Coakley, M. Holcombe, R. Smallwood, and S. Macneil. An integrated Systems Biology Approach to Understanding the Rules of Keratinocyte Colony Formation. *J R Soc Interface*, 4(17):1077–1092, Dec 2007.

M. Svensson, S. Zubairi, A. Maroof, F. Kazi, M. Taniguchi, and P. M. Kaye. Invariant NKT Cells are Essential for the Regulation of Hepatic CXCL10 Gene Expression During Leishmania Donovani Infection. *Infection and Immunity*, 73(11):7541–7547, 2005.

The-Agile-Manifesto. Manifesto for Agile Software Development, 2001. URL http://www.agilemanifesto.org/.

S. Tisue and U. Wilensky. Netlogo : A Simple Environment for Modeling Complexity. *International Conference on Complex Systems*, pages 1–10, 2004.

A. Vargha and H. D. Delaney. A Critique and Improvement of the "CL" Common Language Effect Size Statistics of McGraw and Wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000. doi: 10.2307/1165329.

Y. Vodovotz, G. Clermont, C. Chow, and G. An. Mathematical Models of the Acute Inflammatory Response. *Curr Opin Crit Care*, 10(5):383–390, Oct 2004.

D. C. Walker, G. Hill, S. M. Wood, R. H. Smallwood, and J. Southgate. Agent-Based Computational Modeling of Wounded Epithelial cell Monolayers. *IEEE Trans Nanobioscience*, 3(3):153–163, Sep 2004a.

D. C. Walker, J. Southgate, G. Hill, M. Holcombe, D. R. Hose, S. M. Wood, S. Mac Neil, and R. H. Smallwood. The Epitheliome: Agent-Based Modelling of the Social Behaviour of Cells. *BioSystems*, 76(1-3):89–100, 2004b.

D. C. Walker, N. T. Georgopoulos, and J. Southgate. From Pathway to Population–A Multiscale Model of Juxtacrine EGFR-MAPK Signalling. *BMC Syst Biol*, 2:102, 2008.

J. Wambaugh and I. Shah. Simulating Microdosimetry in a Virtual Hepatic Lobule. *PLoS Computational Biology*, 6(4):16, 2010.

R. Weaver, G. Despotou, G. Kelly, and J. McDermid. Combining Software Evidence: Arguments and Assurance. *ACM SIGSOFT Software Engineering Notes*, 30(4):1–7, 2005.

K. Webb and T. White. UML as a cell and Biochemistry Modeling Language. *Biosystems*, 2005:283–302, 2005.

M. Weber, R. Hauschild, J. Schwarz, C. Moussion, I. De Vries, D. F. Legler, S. A. Luther, T. Bollenbach, and M. Sixt. Interstitial Dendritic Cell Guidance by Haptotactic Chemokine Gradients. *Science*, 339(6117):328–332, Jan 2013.

A. Wehr, C. Baeck, F. Heymann, P. M. Niemietz, L. Hammerich, C. Martin, H. W. Zimmermann, O. Pack, N. Gassler, K. Hittatiya, A. Ludwig, T. Luedde, C. Trautwein, and F. Tacke. Chemokine Receptor CXCR6-Dependent Hepatic NKT cell Accumulation Promotes Inflammation and Liver Fibrosis. *J. Immunol.*, 190(10):5226–5236, May 2013.

E. Whitley, J. Ball, and et. al. Statistics Review 6: Nonparametric Methods. *Critical Care London*, 6(6):509–513, 2002.

U. Wilensky. NetLogo. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. http://ccl.northwestern.edu/netlogo/*, 1999.

S. Wolf, J-P. Bouchaud, F. Cecconi, S. Cincotti, H. Dawid, H. Gintis, S. Van Der Hoog, C. C. Jaeger, D. V. Kovalevsky, A. Mandel, and et al. Describing Economic Agent-based Models – Dahlem ABM Documentation Guidelines. *Complexity Economics*, 2(1), 2013.

World Health Organization. Control of leishmaniasis a60/10. Technical report, 2007a. URL http://apps.who.int/gb/ebwha/pdf_files/WHA60/A60_10-en.pdf.

World Health Organization. Wha60.13 control of leishmaniasis. Technical report, 2007b. URL http://www.who.int/neglected_diseases/mediacentre/WHA_60.13_Eng.pdf?ua=1.

World Health Organization. Leishmaniasis. Technical report, 2014. URL http://www.who.int/leishmaniasis/clinical_forms_leishmaniases/en/.

M. Zeremski, L. M. Petrovic, L. Chiriboga, Q. B. Brown, H. T. Yee, M. Kinkhabwala, I. M. Jacobson, R. Dimova, M. Markatou, and A. H. Talal. Intrahepatic Levels of CXCR3-Associated Chemokines Correlate With Liver Inflammation and Fibrosis in Chronic Hepatitis C. *Hepatology*, 48(5):1440–1450, Nov 2008.

S. Zubairi, S. L. Sanos, S. Hill, and P. M. Kaye. Immunotherapy with OX40L-Fc or Anti-CTLA-4 Enhances Local Tissue Responses and Killing of Leishmania Donovani. *Eur. J. Immunol.*, 34(5):1433–1440, May 2004.