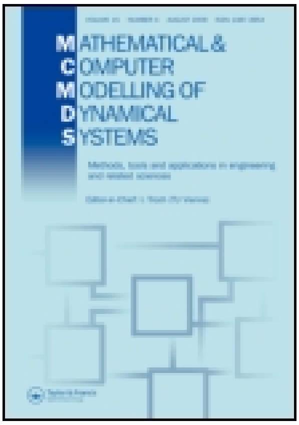

Mathematical and Computer Modelling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/nmcm20

Techniques for grounding agent-based simulations in the real domain: a case study in experimental autoimmune encephalomyelitis

Mark Read [a] , Paul S. Andrews [b] , Jon Timmis [a b] & Vipin Kumar [c]

[a] Department of Electronics , University of York , York , North Yorkshire , YO10 5DD , UK

[b] Department of Computer Science , University of York , York , North Yorkshire , YO10 5DD , UK

[c] Laboratory of Autoimmunity , Torrey Pines Institute of Molecular Studies , San Diego , CA , USA

Published online: 01 Aug 2011.

Mathematical and Computer Modelling of Dynamical Systems
Vol. 18, No. 1, February 2012, 67–86

Taylor & Francis
Taylor & Francis Group

Techniques for grounding agent-based simulations in the real domain: a case study in experimental autoimmune encephalomyelitis

Mark Read[a]*, Paul S. Andrews[b], Jon Timmis[a,b] and Vipin Kumar[c]

[a]*Department of Electronics, University of York, York, North Yorkshire YO10 5DD, UK;* [b]*Department of Computer Science, University of York, York, North Yorkshire YO10 5DD, UK;* [c]*Laboratory of Autoimmunity, Torrey Pines Institute of Molecular Studies, San Diego, CA, USA*

(*Received 3 May 2011; final version received 6 May 2011*)

For computational agent-based simulation, to become a serious tool for investigating biological systems requires the implications of simulation-derived results to be appreciated in terms of the original system. However, epistemic uncertainty regarding the exact nature of biological systems can complicate the calibration of models and simulations that attempt to capture their structure and behaviour, and can obscure the interpretation of simulation-derived experimental results with respect to the real domain. We present an approach to the calibration of an agent-based model of experimental autoimmune encephalomyelitis (EAE), a mouse proxy for multiple sclerosis (MS), which harnesses interaction between a modeller and domain expert in mitigating uncertainty in the data derived from the real domain. A novel uncertainty analysis technique is presented that, in conjunction with a latin hypercube-based global sensitivity analysis, can indicate the implications of epistemic uncertainty in the real domain. These analyses may be considered in the context of domain-specific knowledge to qualify the certainty placed on the results of *in silico* experimentation.

Keywords: *in silico* experimentation; agent-based simulation; sensitivity analysis; uncertainty analysis; calibration; stochasticity; computational immunology; experimental autoimmune encophalomyelitis; interpretation of simulation results

1. Introduction

Computational agent-based techniques are finding increasing application in the modelling and simulation of complex systems. For biological research they offer a complement to traditional wet-lab research techniques, enabling experimentation that is impractical or even impossible in the real domain [1]. Within agent-based models and simulations (ABMS), individual elements of a system can be explicitly represented and carry their own state [2]. For example, an agent-based model of an infection in a body compartment might explicitly represent individual immune system cells as discrete elements in the model, rather than capturing entire cell populations as single model elements. Agent-based models often explicitly represent the environment (such as spatial orderings) in which agents are placed, which determine the movement and interaction dynamics of the agents [3]. The agent-based simulations that we consider in this article are *stochastic* in nature; repeated simulation runs with the same parameters will not necessarily yield identical dynamics, as

*Corresponding author. Email: mnr101@ohm.york.ac.uk

ISSN 1387-3954 print/ISSN 1744-5051 online

http://dx.doi.org/10.1080/13873954.2011.601419
http://www.tandfonline.com

the determination of an agent's behaviour (such as movement) is subject to the generation of pseudo-random numbers from a given distribution.

Our work concerns *computational immunology*: the modelling and simulation of the immune system. Advances in traditional wet-lab techniques are resulting in vast quantities of data pertaining to the behaviour of very specific system elements under very specific conditions [4]. Modelling and simulation attempts to integrate these data into a coherent whole [5,6], indicating inconsistencies within the data and areas where understanding is lacking. This can often feed back into the wet-lab by directing avenues for experimentation [7]. It allows for the formulation and evaluation of hypotheses concerning system behaviour and provides a means through which these hypotheses can be evaluated in the context of established knowledge [8,9]. Owing to the flexibility of computer code, simulation can facilitate experiments that are impossible to perform in the real domain as a result of either ethical considerations or issues in accessibility [3,10].

The work presented here is conducted in the context of the CoSMoS project,[1] which seeks to build capacity in complex system simulation construction and analysis. It is developing the *CoSMoS process*, an approach investigating complex domains through simulation that places emphasis on capturing the domain, and subsequent transitions from explicit models of a domain to executable simulations [11]. Though agent-based modelling and simulation offers great potential in assisting scientific experimentation, its success ultimately depends on the ability to interpret simulation-based results in terms of the original domain. Simulations, however, are artificial and abstract representations of the real domains on which they are based, and hence results do not necessarily translate directly from one domain into the other.

The goal of this article is to explore the problem of relating simulation results back to domain reality, and by way of a case study provide examples of how *calibration* and statistical techniques can be used to explore *uncertainty* in simulation results. In doing this, we provide an example of how to qualify the significance of simulation-derived results in terms of the original domain. Like any type of model, ABMS are simplifications; it is intractable to represent every aspect of the real domain in a model, both computationally and because the domain is often not sufficiently well understood. There are many aspects of biological systems for which there exists no consensus in the literature or that remain to be investigated. For example, in 2000 the journal *Seminars in Immunology* [12] dedicated an entire issue to a debate among leading immunologists regarding the function of the immune system in directing bio-destructive activities towards pathogenic invaders, and not the host [12]. Highlighted was a lack of consensus among leaders in immunology with regard to this fundamental and essential aspect of immune system function. In the context of modelling and simulation, the lack of knowledge regarding a particular aspect of the domain is referred to as *epistemic uncertainty* [13]. Epistemic uncertainty presents a challenge to the construction, calibration and interpretation of simulations, since the exact nature of a phenomenon of interest may be unclear.

Taken together, abstraction (which dictates that all aspects of the domain that are represented in the simulation must compensate for the actions of those that are not) and epistemic uncertainty complicate the relationship between the real domain and the simulation. Understanding this relationship is critical to relating predictive results[2] arising from ABMS back to the real domain, hence it is important that the implications of epistemic uncertainty and abstraction be appreciated. If the validity of AMBS-generated predictions hinges on aspects of the domain that are not well understood, then caution must be exercised in the interpretation of results.

In the vast majority of cases where simulation results are reported in the computational immunology literature, either the results are assumed to be absolutely representative of the target domain, or no effort to indicate the significance of simulation-derived results in terms of the original domain is made. To the best knowledge of the authors, we present here an original approach to qualify the significance of simulation-derived results in terms of the real domain through the use of uncertainty and sensitivity analyses [14], statistical techniques that elucidate the relationship between a system's inputs and its output. Uncertainty and sensitivity analyses have found recent application in exploring immune system models and simulations, indicating parameters that are influential in dictating simulation behaviours [3,15,16]; however, the use of these techniques in linking simulation results back into the original domain is novel.

Our work is grounded in an immunological case study for the modelling and simulation of experimental autoimmune encephalomyelitis (EAE), a mouse proxy for multiple sclerosis (MS). We make extensive use of a domain expert in mitigating uncertainty in data derived from the real domain. A novel uncertainty analysis technique that examines the simulation's robustness to parameter perturbation is presented, indicating how far a parameter may be perturbed before a scientifically significant change in simulation behaviour is observed. A latin hypercube global sensitivity analysis that determines simulation sensitivity with respect to its various parameters is presented. When considered in the context of domain-specific knowledge, these two analyses can qualify the implications of epistemic uncertainty on simulation-derived predictive results and will contribute to further work in identifying the confidence that may be placed on simulation-derived predictions.

The article is organized as follows: Section 2 describes EAE, a mouse proxy for MS, which constitutes the domain for this case study. Section 3 describes our calibration technique, along with examples of real domain data that demonstrate epistemic uncertainty and motivate our approach. Section 4 presents the uncertainty and sensitivity analyses employed in this work. Section 5 concludes this article.

2. Experimental autoimmune encephalomyelitis

EAE is an autoimmune disease in mice that serves as a proxy for MS in humans [17,18]. The disease results in the stripping of the insulatory myelin sheath from the neurons of the central nervous system (CNS). An abstract representation of the key cells involved in EAE, and their relationship to one another is presented in Figure 1. EAE is induced in mice by injecting myelin basic protein (MBP – a myelin derivative). MBP is ingested by dendritic cells (DCs) which induce a population of MBP-specific autoimmune CD4 T helper 1 (CD4Th1) cells. The CD4Th1 cell population infiltrates the CNS and secretes molecules that prompt local microglia cells to kill neurons. Neurons killed in this manner are ingested by DCs resident in the CNS. DCs then derive and present MBP from neurons and further induce and activate MBP-specific CD4Th1 cells. In this manner autoimmunity self-perpetuates.

The physiological death of CD4Th1 cells, which occurs some time after their activation, leads to their digestion by DCs. Such DCs are then able to induce and activate CD4 regulatory T cells (CD4Treg) and CD8 regulatory T cells (CD8Treg), the former being required for the activation of the latter. Activated CD8Treg cells kill MBP-specific CD4Th1 cells upon direct cell contact. The population-level deletion of CD4Th1 cells permits the expansion of a competing, but normally suppressed, CD4 T helper 2 (CD4Th2) cell population [19], which does not promote autoimmunity.

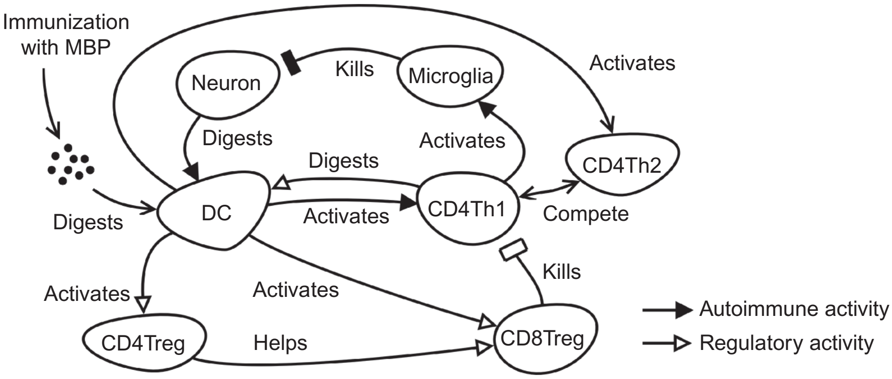

Figure 1. An abstract depiction of the major cells involved in EAE and its associated regulatory recovery. MBP is the target of autoimmunity and is a protein expressed in the CNS. Adapted from [20].

Note: EAE, experimental autoimmune encephalomyelitis.

EAE is a complex disease, characterized by the interaction of two coupled feedback mechanisms: self-perpetuating autoimmunity and the regulation that it promotes. Disease and recovery arise as emergent properties resulting from the interactions of millions of cells of many different types. These interactions between cells occur across several bodily compartments. The complexities of this disease make reasoning about its nature and predicting the result of a particular intervention challenging. EAE is amenable to computational modelling and simulation techniques, where established knowledge and data can be integrated with hypotheses of system operation and executed in the presence of different interventions to indicate how the real system might respond.

3. Calibration

Calibration is an important aspect of *in silico* experimentation, it seeks to align a simulation's behaviour with that of the target domain, by adjusting parameters and manipulating the simulation's mechanics. However, there exists significant uncertainty and variance in data from the real domain. The stochastic nature of the immune system results in individual experimental animals experiencing vastly different progressions of EAE. Identifying immunological data in a format that a simulation can be calibrated against, and with sufficient samples to constitute a fine-grained representation, can be a challenging task.

Figure 2 highlights the type of data typically derived from EAE experiments with real mice in the laboratory. Figure 2(a) shows the mean severity of EAE, measured on a five point scale [21], experienced by groups of 6–8 mice of different strains following the same induction of EAE administered in each group. There is considerable variation in EAE progression experienced by each species of mouse. Figure 2(b) shows the severity of EAE experienced by groups of five mice, with each group undergoing a different intervention. It can be seen that within groups of exactly the same experimental animal undergoing exactly the same intervention, there is considerable variation in the severities of EAE experienced. Figure 2(c) indicates the number of CD4Th1 cells found in different bodily compartments at various times following immunization in several experimental animals. It can be seen that there is once again significant spread in the data, with several examples where there are no data points lying on the calculated mean values. Acquiring data such as these can require the killing of an experimental animal, and since genetically identical animals undergoing exactly the same EAE induction can experience significantly different responses (Figure 2(b)), it is difficult to compile a representative progression of EAE in terms of individual cell population number.

Data of this nature can be challenging to calibrate a simulation against. A stochastic computer simulation can be run many hundreds or thousands of times in order to obtain highly representative averaged values for a particular metric of interest. The same fidelity of data is not available in the immunological literature, where the number of samples obtained in acquiring an average rarely exceeds 10. Whereas a computational simulation can readily provide exact numbers of a particular cell type over time, the data of Figure 2(c) only *indicate* their number, there is not an easily established exact mapping between the metric used and the actual number of cells observed by the instrument.

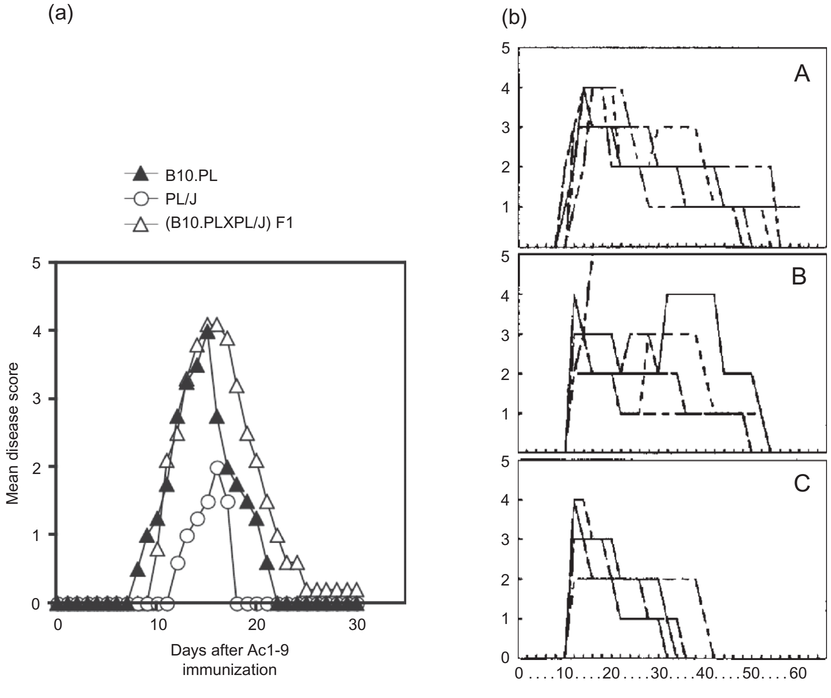

Figure 2. Examples of clinical data pertaining to the progression of EAE in mice. EAE severity is scored on a scale of 0–5, '1, flaccid tail; 2, hind limb weakness; 3, hind limb paralysis; 4, whole body paralysis; 5, death' [21]. (a) The mean EAE severity experienced by groups of 6–8 mice of different strains following the same induction of EAE administered in each group. Adapted from [22]. (b) EAE severity experienced by each of 5 mice under different interventions (boxes A–C), over time (in days). Taken from [21]. The interventions involve injecting mice experiencing EAE with varying quantities of Treg cells. (c) An indication of the number of CD4Th1 cells (EI on the *y*-axis) residing in a selection of organs at various times following the induction of EAE from [23]. Organs examined were lymph nodes (LN), spleen (Sp), blood (Bl), lungs (Lu), bone marrow (BM), liver(Li), mesenteric lymph nodes (mLN) and thymus (Thy).

Note: EAE, experimental autoimmune encephalomyelitis.

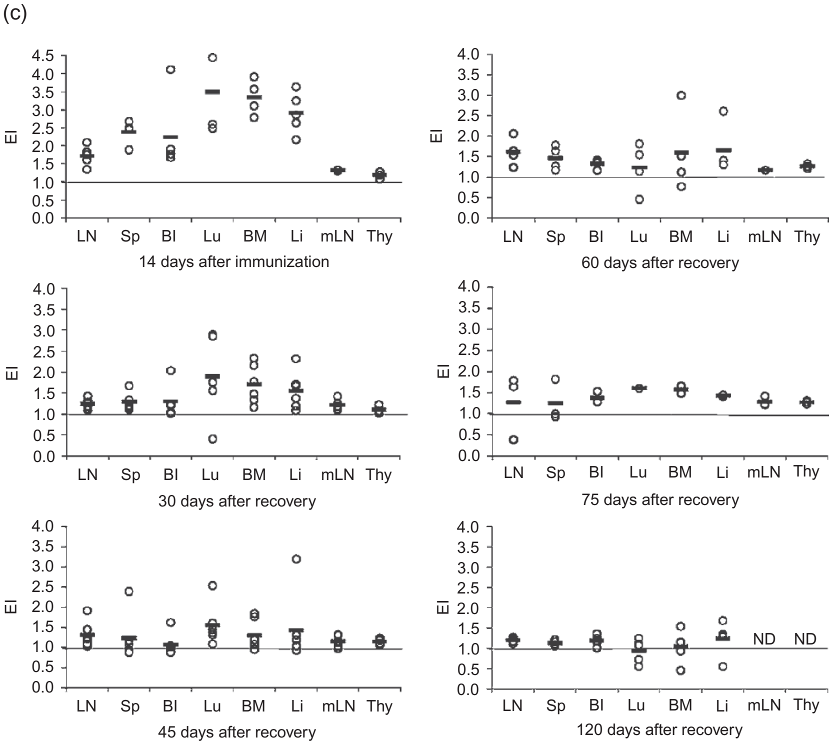

Figure 2. (Continued).

3.1. *Calibration process*

Calibration of the EAE simulator was undertaken as a collaborative effort between the modeller (Mark Read) and the domain expert (Vipin Kumar). The domain expert provides a consistent and comprehensive single source of data against which to calibrate, which helps mitigate the uncertainty and variation in the data surrounding EAE. In this manner, the simulation is calibrated against the domain expert's understanding of EAE. The input of the domain expert in this calibration procedure serves to keep the simulation well grounded in the domain.

The process through which the simulation was calibrated is as follows: The simulation is executed with a 'best guess' set of parameters and representative median results are collected. The domain expert identifies aspects of simulation dynamics that deviate from his perspective of the real system. Both domain expert and modeller discuss to identify the source of the deviation in the simulation. Both parties' input are invaluable: the domain expert brings a wealth of domain-specific understanding, whilst the modeller, having built the simulation, has a detailed understanding and intuition of how this information has been abstracted and how the simulation operates. Potential avenues of model and simulation amendment and development are identified, and each one is independently integrated into the simulation in turn. In each case the simulation is then executed once more to obtain representative median results. Subsequent interactions between domain expert and modeller re-examine the results and decide upon which amendments are to be permanently adopted into the simulation. As such, the calibration process is iterative.

Throughout this calibration process, the simulation's behaviour is examined under two circumstances: that of normal EAE progression and recovery following induction of EAE; and the progression of EAE with the regulatory network disabled.[3] It is important to calibrate the simulation against these two different circumstances. An incorrect model can be fitted to the data derived from a single biological circumstance, such that its dynamics reflect that observed *in vivo*. We believe it is less likely that an incorrect model can replicate *in vivo* dynamics of several different biological circumstances in turn without having to drastically manipulate its parameters or change the underlying model. The parameter manipulations would reflect a need to compensate for the model's incomplete capture of the real domain. If the complexities and mechanics of the real domain are adequately represented in the simulation, then both biological circumstances (presence and absence of regulatory activity) should be replicated without having to tweak parameters. The end goal of simulating EAE is to use the simulation to perform experimentation that are intractable in the real domain. More confidence can be held in the results of predictive *in silico* experimentation if the simulation correctly represents multiple disparate real-world experimental circumstances.

3.2. Baseline behaviour

The calibration process results in the simulation being brought into a state that both domain expert and modeller accept as representing a normal, baseline behaviour. This baseline behaviour, shown for both the presence and absence of regulatory activity in Figure 3, is generated by both a set of parameter values and the simulation mechanics for which they are appropriate. As explained in Section 1, the simulation is an abstract representation of the real domain; all elements present in the simulation compensate for the action of elements of the real domain that are not represented. Were the simulation's mechanics to be altered in some way, then the simulation's parameters may not represent exactly the same aspects as they did previously. Their role in compensating for model abstractions would be changed, and as a result, it may be necessary to recalibrate the simulation.

The baseline simulation serves as an accepted 'normal' state against which to contrast the results of experimentation with the simulation. The aspects of the system that *in silico* experiments investigate are usually represented as parameters in the system.

4. Exploring the EAE simulation with uncertainty and sensitivity analyses

4.1. Aleatory and epistemic uncertainty

There are two sources of uncertainty in the simulation that must be analysed before the significance of predictive results arising from the simulation can be appreciated: aleatory and epistemic uncertainty. Aleatory uncertainty results from inherent stochasticity in a system [13] and is present in both the biological system (as can be observed in Figure 2(b) and (c)) and the simulation. In order to acquire representative predictive results from a stochastic simulation, it is necessary to acquire many samples in the form of simulation executions. Appreciation of the spread of data is important for understanding when some experiment in the simulation has generated scientifically significant results.

Epistemic uncertainty results from a lack of knowledge about the value that a particular parameter should be assigned [13]. There are many immunological details that are not well understood, such as the decay rates of certain molecules, or the exact average time that a particular cell type might remain in a particular state. Further to this, the simulation's abstract nature dictates that there is not a direct translation from these details into

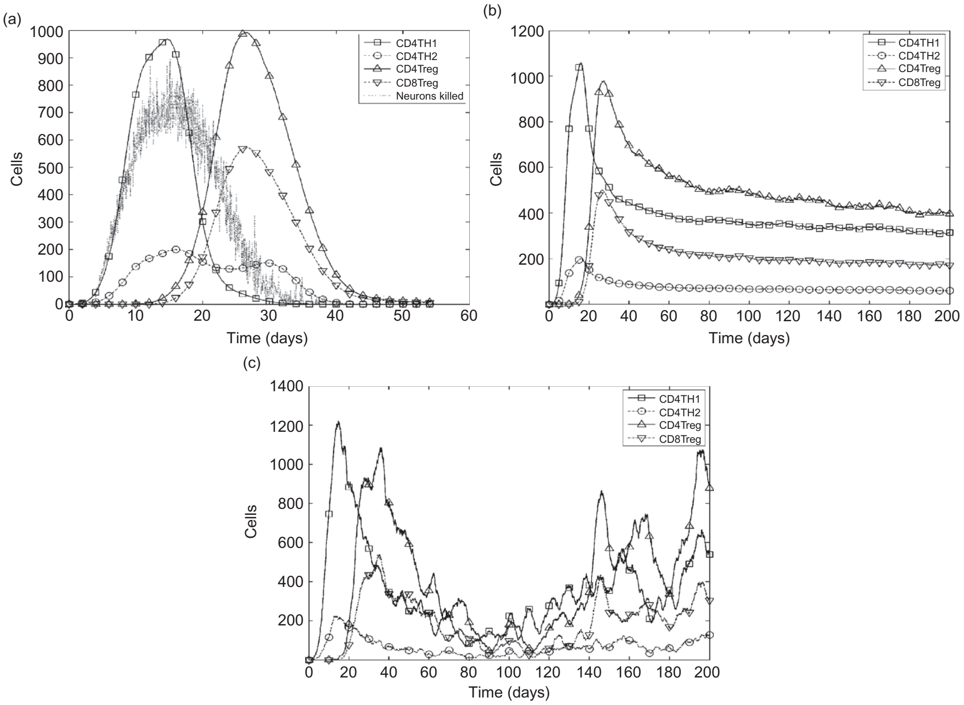

Figure 3. The baseline behaviour of the EAE simulation. (a) The baseline behaviour of the EAE simulation, following calibration, in the presence of regulation. Graph shows the number of effector T cells in the system over time and the number of neurons killed per hour (×10 for clarity). This is the median of 1000 simulation runs. (b) The baseline behaviour of the EAE simulation, following calibration, with regulatory activity disabled by preventing CD8Tregs from killing CD4Th1 cells. This represents the median behaviour of 1000 individual runs such as that depicted here. (c) An example of a single simulation run, in the absence of regulatory activity. The number of CD4Th1 cells declines heavily to a minimum at around day 80, but rises again thereafter. This reflects the relapsing nature of EAE seen in absence of regulation in some mice. The median progressions shown in Figure 3(a) and (b) are compiled from unique individual executions such as that of (c).

Note: EAE, experimental autoimmune encephalomyelitis.

simulation parameters, since, as previously stated, everything present in the simulation must compensate for all those aspects of the real domain that are absent.

In order to fully appreciate the results of experimentation with the simulation, it is necessary to investigate the simulation's robustness to parameter alterations. The results of *in silico* experimentation can be considered to be of greater significance if the predictions that they delineate hold under perturbation of the simulation's parameters. If a prediction relies on a parameter holding a very specific value within its biologically plausible range, then the prediction cannot be held with great confidence. If a prediction breaks only when parameter values are perturbed to outside their biologically accepted range, then more confidence can be held that the prediction is representative of the real system.

Biologically plausible domains for parameter values are not clearcut and are accepted with some informal degree of confidence. This can arise from the different experimental set-ups through which results are obtained. For example, results arising from *in vitro* experiments conducted in test tubes may not be completely representative of the real system since the elements under investigation have been removed from their natural environment, yet it may be impossible to perform such experiments *in vivo*. Additional information useful in qualifying the certainty of simulation predictions is the relative contribution that each parameter makes to the simulation's system-wide dynamics. Epistemic uncertainty surrounding parameters that are of little consequence to simulation behaviour is less of an issue than uncertainty surrounding parameters of greater influence.

Uncertainty and sensitivity analysis techniques are closely related statistical techniques that attribute variation in a system's outputs to variation in its inputs [14]. Uncertainty analysis investigates the effect of uncertainty in parameter values on output behaviour, whereas sensitivity analysis investigates the relative sensitivity of a simulation's output to its individual inputs. We have employed two uncertainty and sensitivity analysis techniques, one that qualifies the robustness of the simulation to parameter alteration, and one that determines the relative influence of parameters on simulation behaviour, discussed in Sections 4.3 and 4.4, respectively.

4.2. Uncertainty and sensitivity analysis responses

Uncertainty and sensitivity analysis techniques perform their analysis on a single variable that is assumed as the output of a system [14]. The results of analyses can only be appreciated in terms of this variable, typically termed the *response*. For the purposes of analysing the EAE simulation, nine responses are assumed, and the uncertainty and sensitivity analysis techniques are run on each. Figure 4 shows these nine responses, namely, the maximum number of effector T cells reached over the simulation's execution for each of the four

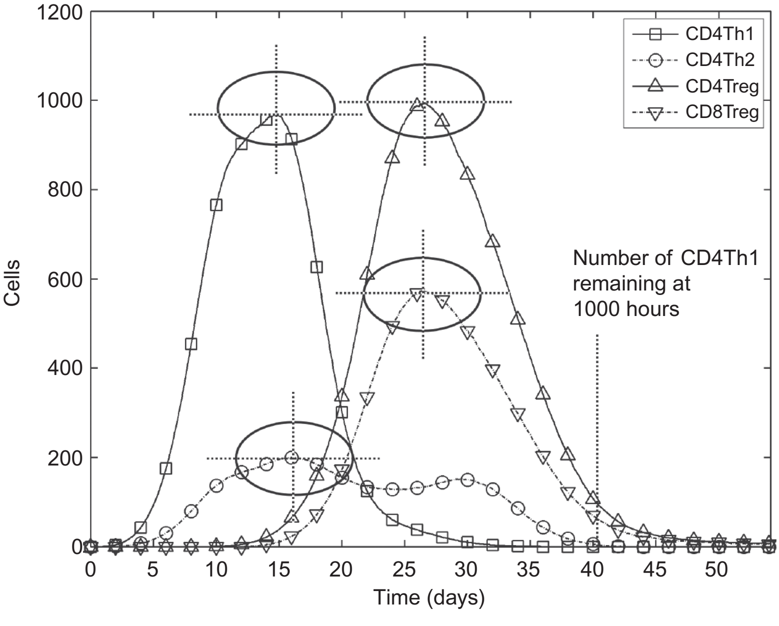

Figure 4. The nine responses adopted to analyse the EAE simulation, shown on a graph of the number of effector T cells over time. For each of the four T cell species, both the maximum number reached and the time at which that number was reached are assumed as a separate response (these are indicated as crosshairs on the diagram). The ninth response is the number of autoimmune CD4Th1 effector cells remaining at 1000 hours.

Note: EAE, experimental autoimmune encephalomyelitis.

species of T cell, the times at which those four peaks occurred and finally the number of effector CD4Th1 cells that remain at 1000 hours.

Analysing the simulation through these nine responses yields detailed information on how its parameters affect each of the major T cell populations independently, rather than attempting to condense a complex disease into a single variable. The choice of the nine responses was approved by the domain expert as being of biological relevance and interest in the context of EAE. For example, the last response, the number of CD4Th1 cells remaining at 1000 hours, is an effective means of determining whether efficient regulation of the CD4Th1 population has taken place. This cannot other eight responses.

4.3. Robustness analysis

In order to assess the robustness of the simulation to parameter perturbation, we have devised a *one at a time* uncertainty analysis technique, whereby each parameter is adjusted independently of the others, which remain at their baseline values. This baseline may be the result of calibration, as is the case for the examples presented here, or may be a point in parameter space that reflects a point of interest in making a prediction of the system; in this manner our technique would entail deliberately breaking that prediction. When considered in the context of biologically accepted ranges for parameter values, the results of this analysis help to qualify the validity of simulation results.

We employ the Vargha–Delaney *A* test [24], which is a non-parametric effect magnitude test, to determine when a parameter adjustment has resulted in a *scientifically significant* change in simulation behaviour from the baseline. The *A* test compares two population distributions and returns a value in the range [0.0,1.0] that represents the probability that a randomly chosen sample taken from population A is larger than a randomly chosen sample from population B. A value of 0.5 indicates no difference, whereas values above 0.71 and below 0.29 indicate a ‘large’ difference in the distributions [24]. Table 1 details which *A* test scores relate to various magnitudes of difference between two populations.

Our choice in using a non-parametric test is to avoid the assumption that underlying distributions are normally distributed. We use an effect magnitude test to determine scientific significance in place of statistical significance since a sufficiently large number of samples can always reveal a statistical significance, unless the variable of interest has no effect.

4.3.1. Experimental methodology

The baseline assumed in these examples is the result of the calibration process described in Section 3.1, though any point of interest in parameter space may be used. Each parameter is considered in turn and is perturbed around its baseline value. For each perturbation, the simulation is executed *n* number of times. The nine responses are calculated for each of the *n* simulation runs, and together form nine distributions. These can each be compared using

Table 1. The magnitude of effect size indicated by *A* test score [24].

Difference	Large	Medium	Small	None	Small	Medium	Large
A test score	0.29	0.36	0.44	0.50	0.56	0.64	0.71

the *A* test to similar distributions obtained from executing the simulation using baseline values *n* times.

Figure 5(a) shows a graph depicting the robustness analysis on a particular parameter of the simulation. The effect magnitude of the parameter being assigned various values around the default value can be observed from the nine lines representing each of the nine responses. The *y*-axis shows the *A* test scores for each response under each parameter value, when compared to the default. The default value for this parameter is 100 and is where all the response lines converge. The *A* test score here is 0.5, as the default parameter value distribution data are being compared to themselves. Horizontal lines are drawn at *A* test score values of 0.71 and 0.29. These are the boundaries for differences in distributions that are considered 'large' [24]. If a response line exceeds or falls below either of these respective boundaries, then a scientifically significant change in that response is observed for the particular parameter value that generated it.

Figure 5(b) shows the effect of the parameter adjusted in Figure 5(a) on a particular response, as a box and whisker plot. The distribution of response values that arise from the *n* simulation executions for each parameter value is indicated by the boxes.

Data illustrated in this manner give a clear indication of exactly how the parameter of interest influences each response.

4.3.2. Mitigating aleatory uncertainty

The intention of this uncertainty analysis technique is to determine the robustness of the simulation to parameter perturbations. When considered in the context of epistemic uncertainty, a robust simulation may produce predictive results in which a relatively high degree of confidence may be placed. If the simulation's behaviour is shown to be fragile when parameters are perturbed to specific values that lie within the range of values reported in the domain's literature, then predictions arising from the simulation cannot be held with a high degree of confidence; the prediction may[4] rely on the simulation holding a parameter value more specific than what the literature can dictate as appropriate. The simulation results might be an artefact of underspecified parameter values.

When performing this uncertainty analysis technique, it is important to obtain sufficient samples such that the response distributions obtained through parameter perturbation may be considered representative of the simulation's nature, rather than the result of aleatory uncertainty arising from the stochastic nature of the simulation. Uncertainty and sensitivity analyses techniques may be applied to any system, including systems that are costly to run, such as the computational expense of the present EAE simulation. Finding the relationship between sample size and the effect of aleatory uncertainty is important for balancing requirements (e.g. desired fidelity of data) and resources (e.g. computational resource required).

The following procedure is used to investigate the relationship between the number of samples obtained and the effect of aleatory uncertainty. The baseline parameter set-up is run 20 times, forming 20 dummy parameter permutations. These dummy permutations have no effect on the simulation's behaviour, since no parameter values are changing. This procedure can quantify the effect of aleatory uncertainty on *A* test scores, by examining the scores between the first and remaining 19 distributions generated. By repeating the experiment for different sample sizes, the number of samples required to reduce the noise in *A* test scores that arise from aleatory uncertainty to an acceptable level can be determined. Figure 7(a)–(c) shows the *A* test scores across 20 dummy parameters using sample sizes of 5, 50 and 500, respectively. Table 2 shows how the maximum and median *A* test scores

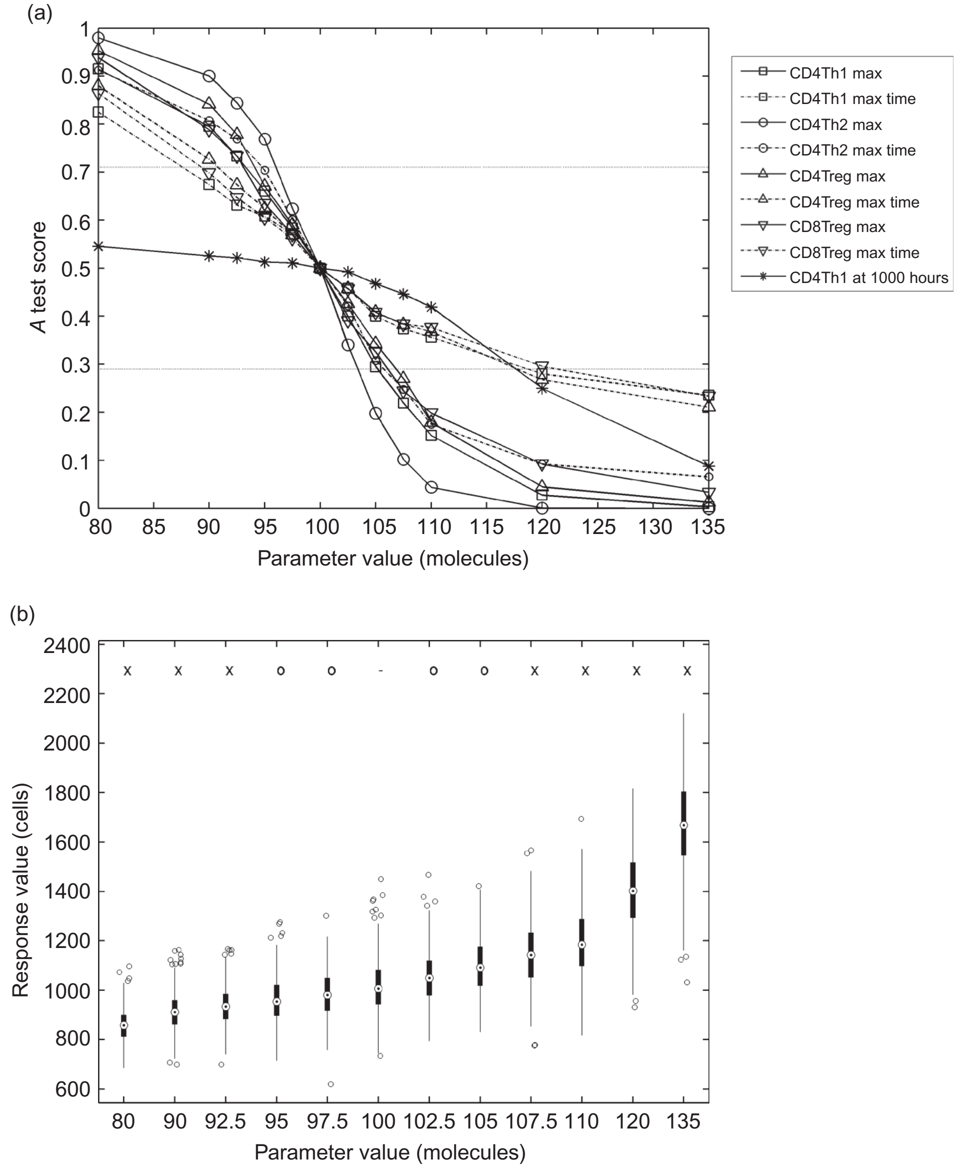

Figure 5. Example robustness analysis of a parameter in the EAE simulation. Perturbation of this parameter has a large effect on all the responses. The parameter investigated here dictates the rate of TNF-α, a substance harmful to neurons in sufficient concentration, secretion by activated microglia cells in the CNS. (a) *A* test scores for the response distributions arising from parameter perturbations. The default value for this parameter is 100 and is where all lines on the graph converge. (b) The distribution of response data (maximum number of CD4Th1 cells reached) for each parameter perturbation. A dash, a circle and a cross above each box respectively indicate the default value, a change in distribution not considered large and a change that is considered large.

Note: EAE, experimental autoimmune encephalomyelitis.

derived from these 20 dummy parameters are affected by the sample size used, for the maximum number of CD4Th1 cells reached during simulation execution response. Figure 6 plots how the sample size affects the maximum *A* test scores achieved for each response. For both Table 2 and Figure 6, scores below 0.5 have been assigned their corresponding value above 0.5.

Figure 6 indicates that a sample size of at least 350 is required to reduce the magnitude of aleatory uncertainty to an effect size less than 'small' (see Table 1), for all responses.

Table 2. An example of how sample size can reduce aleatory uncertainty, as measured by *A* test scores.

Sample size	Maximum *A* test score	Median *A* test score
1	1.0000	1.0000
5	0.9200	0.6800
10	0.7900	0.6300
50	0.5826	0.5228*
100	0.5746	0.5228*
200	0.5519*	0.5290*
500	0.5261*	0.5161*
1000	0.5199*	0.5074*

Notes: *Denotes the scores corresponding to an effect magnitude of less than 'small' (0.56).
All scores below 0.5 are assigned corresponding values above 0.5 before maximum and median calculations are performed; we are interested only in the magnitude of effect, not its direction above or below 0.5. These example scores relate to distributions of the maximum number of CD4Th1 cells occurring at any point in simulation execution response.

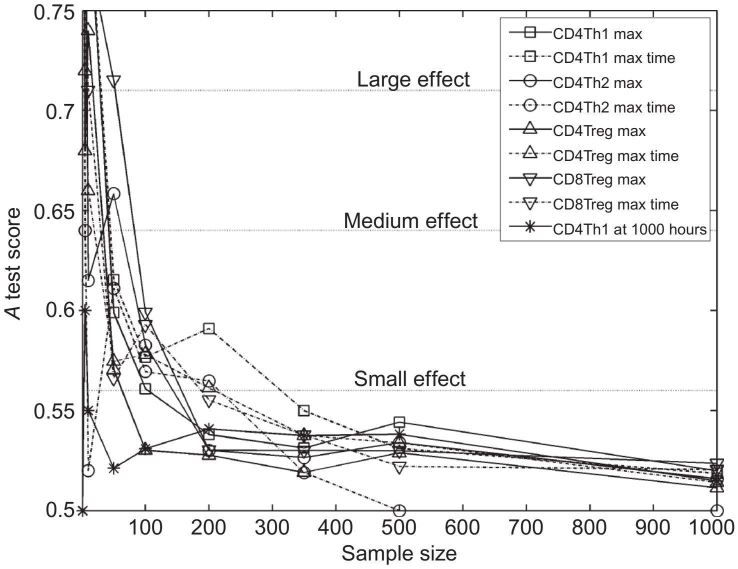

Figure 6. The maximum *A* test score achieved for each response over 20 dummy parameter experiments. The sample size represents the number of simulation runs used in compiling the response distributions for each dummy parameter experiment. Note that all scores below 0.5 are assigned corresponding values above 0.5 before maximum are performed; we are interested only in the magnitude of effect, not its direction above or below 0.5. The three effect magnitude boundaries for the *A* test are indicated.

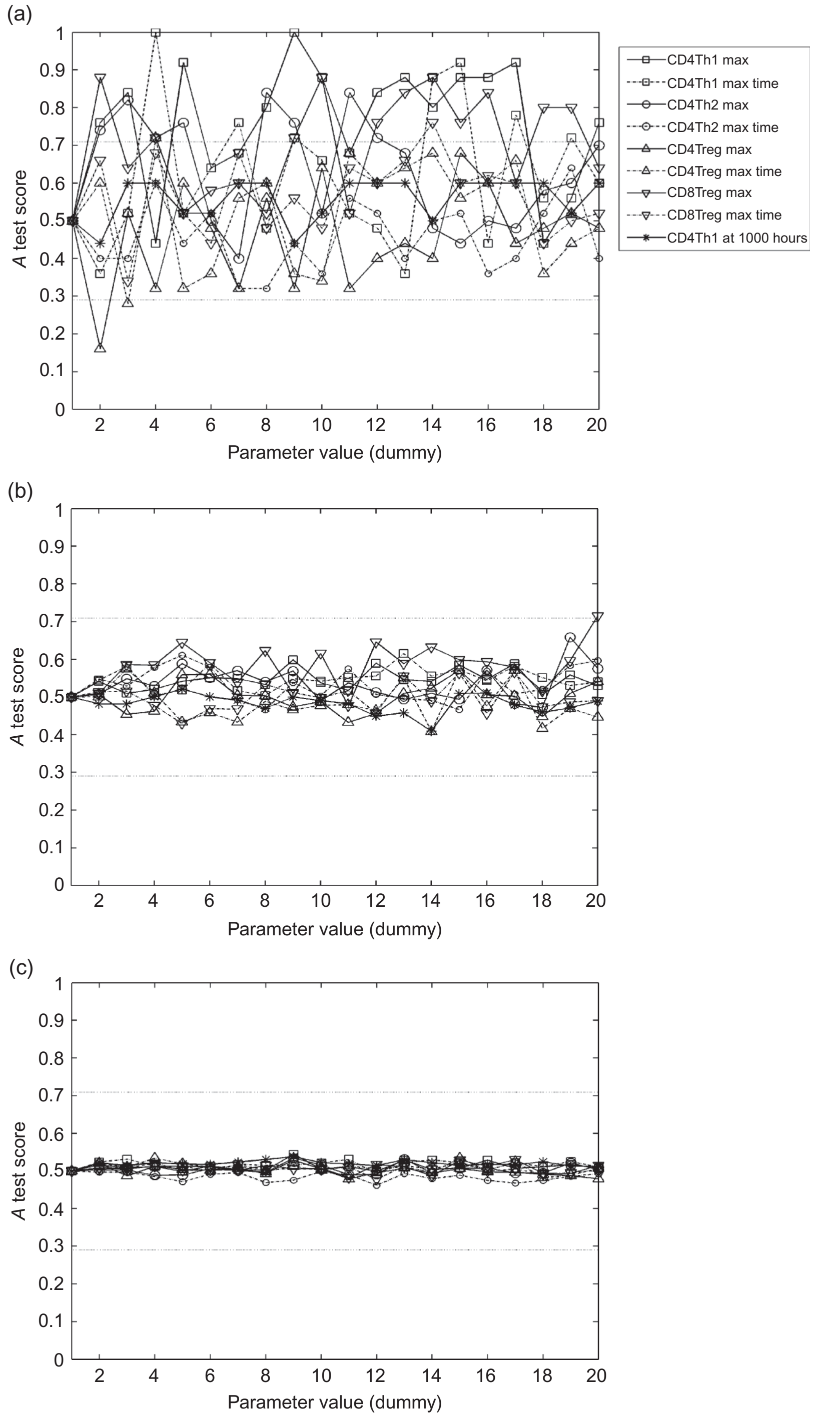

Figure 7. The effect of aleatory uncertainty on the results of the robustness analysis, for various sample sizes used when obtaining representative results from the simulation. The *x*-axes are labelled ‘dummy parameter’ as no parameters are actually being changed. The tests are designed to ascertain how the number of samples (simulation executions) from which median data are compiled affects the consistency of results. *A* test scores for 20 dummy parameter permutations, using sample sizes of (a) 5, (b) 50 and (c) 500.

Hence, when using sample sizes of 350 simulation executions or more, results that deliver effect magnitudes less than or equal to 'small' should be discarded; one cannot determine whether the results are genuinely representative of the simulation's behaviour, or the result of aleatory uncertainty.

4.4. Determining the influence of parameters

The robustness analysis technique reported above is effective at providing an indication for how far a parameter may be adjusted before some significant deviation in simulation behaviour occurs. A shortcoming of the technique is that it does not elucidate non-linear effects between parameter values that are revealed only by adjusting two or more simultaneously; a particular parameter's influence on simulation behaviour may vary in magnitude depending on the value held by another parameter.

A global sensitivity analysis technique is used to provide a more representative indication of the relative influence of parameters on simulation behaviour, since it does highlight non-linear effects. *Global* sensitivity analysis refers to an experimental set-up where all parameters are perturbed simultaneously, as opposed to *one at a time* methods. Identifying which parameters exhibit the greatest influence over simulation behaviour can help qualify the implications of epistemic uncertainty for prediction validity. Significant uncertainty in a parameter's value is of greater consequence if that parameter has a large effect on simulation dynamics than if it is found to have a small effect.

4.4.1. Experimental methodology

A latin hypercube design [25] is used to select *k* number of samples from the simulation's parameter space. Figure 8(a) shows an example latin hypercube design over two parameters, taking 10 samples. The domain for each parameter over which samples are to be taken is defined and divided into *k* sections. In the example the 10 sections are uniformly distributed, but that need not be the case if information from the real domain dictates otherwise [14]. For systems that are costly to run, latin hypercube design ensures an efficient coverage of parameter space using a minimal number of samples [14]. A single sample exists in each segment of each parameter's domain.

The simulation is run *n* times at each of the *k* sample points, and each of the nine responses is calculated for each run. Median values for each response at each sample point are calculated. The samples are ordered according to their value for a particular parameter of interest, and for each response a plot of median response value against each sample's value for the parameter of interest is generated. If the latin hypercube is of good design, having chosen samples such that correlations between parameters are minimized, then this ordering of samples should leave a minimum correlation between any other parameters. Any correlation between median response values and parameter values for each plot can be attributed to the effect of that parameter. Parameters that have a significant influence on simulation behaviour will yield bigger correlations than those that have less influence.

Figure 8(b) and (c) illustrates this technique with two examples from the EAE simulator. The samples are ordered according to the parameter of interest, and the median values for the time at which the maximum number of CD4Treg cells is reached are plotted. There are 300 samples, and the simulation was run 300 times at each sample point, to minimize the effect of aleatory uncertainty.[5] The variation in the data arises from the pseudo-random

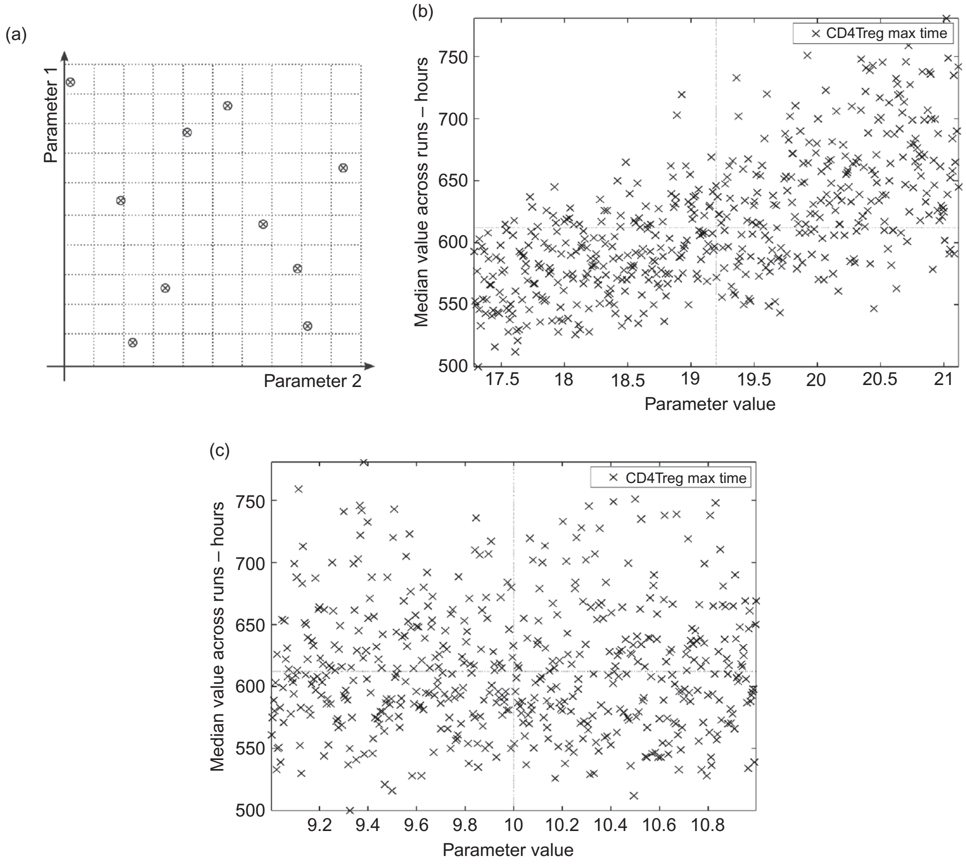

Figure 8. Examples of latin hypercube design approach to determining relative parameter influence on simulation behaviour. The horizontal and vertical lines on Figure 8(b) and (c) indicate the default parameter value and its associated response value. (a) An example latin hypercube design for two parameters. Ten samples are taken from the parameter space, within the specified domain of each parameter. Each of the 10 sub-domains (indicated by dotted lines) contains a sample, ensuring that the full domain of each parameter is explored. (b) This simulation parameter has a large influence on the time at which the maximum number of CD4Treg effector cells is reached during simulation execution. The rank correlation coefficient is 0.65. The parameter analysed here dictates the mean time that a T cell spends in a proliferative state before differentiating into an effector cell. (c) This simulation parameter has a small influence on the time at which the maximum number of CD4Treg effector cells is reached during simulation execution. The rank correlation coefficient is −0.001. The parameter analysed here is the rate of type 1 cytokine (an immune system messenger molecule) secretion by activated CD8Treg cells.

values assigned to all other parameters but the one of interest, once the samples have been ordered. In Figure 8(b), the parameter being examined is of such influence that despite all this movement in other simulation parameters, a clear trend does emerge. This is not the case for the parameter analysed in Figure 8(c). As such, large epistemic uncertainty in the domain literature relating to the parameter of Figure 8(b) must be given more consideration when interpreting simulation-derived results into the original domain than epistemic uncertainty concerning the parameter of Figure 8(c).

5. Discussion

While computational modelling and simulation techniques such as ABMS have the ability to assist and complement more traditional wet-lab experimentation, there are several challenges that must be addressed before this potential can be fully realized. The complexities of biological systems dictate that models and simulations must adopt simplifying assumptions and abstractions in their representations. It is usually intractable to fully represent a complex system in a model or simulation, and there typically exist many aspects of the real domain that are not well understood. Epistemic uncertainty in the real domain can prevent the exact determination of simulation parameter values. It is critical in relating the results of *in silico* experimentation back to the real domain that the implications of these uncertainties, abstractions and assumptions be appreciated. To address this within the context of an EAE case study, we describe a method for calibrating the simulation with input from a domain expert and explore suitable statistical techniques to tackle uncertainty in simulation results.

The value of input from a domain expert in calibration cannot be understated, as it helps mitigate the uncertainties and inconsistencies found in data pertaining to the domain. We also note the importance of considering simulation dynamics under multiple circumstances when calibrating. The different circumstances arise from interventions applied to the real domain that expose the complexity of the system, which the simulation intends to capture. Comparing simulation dynamics under these different interventions with the dynamics of the real system helps avoid the pitfall of fitting an incorrect model to a single set of biological data.

Our novel sensitivity analysis technique makes use of the Vargha–Delaney *A* test in assessing a simulation's robustness to parameter perturbation. In addition, a global sensitivity analysis technique that makes use of latin hypercube design quantifies the relative influence of each parameter on simulation dynamics. When considered in the context of information from the real domain (not shown in this article), these analyses can help qualify the certainty of predictions arising from *in silico* experimentation. Predictions that rest on a highly influential parameter holding a value within a range more specific than the biological literature can attest to undermine the certainty of that prediction.

Though Vargha and Delaney present guidelines for the *A* test scores that correspond to large differences between two sets of results, these scores are not well justified in the real domain of EAE investigated here. Ongoing work is examining how to assign an EAE severity score, a measure used in the wet-lab, to the execution of the simulation. Such an analysis will allow results from *in silico* experimentation to be better related to activities being conducted in the wet-lab. This is challenging because the EAE severity score is a subjective analysis of the level of autoimmunity being experienced, and results from observing the entire experimental animal, many aspects of which are not represented in the simulation.

6. Materials and methods

6.1. EAE simulation and statistical analysis

The EAE simulation used in this article[6] is coded in Java and makes use of the MASON [26,27] library to provide simulation infrastructure. Random number seeds for simulation execution are supplied such that each simulation execution makes use of a unique seed. All statistical analyses are performed using Matlab. Non-parametric statistics are used throughout.

6.2. Compilation of median-averaged graphs

The EAE simulation provides data on specific metrics of interest for every hour of simulated time. Unless otherwise stated averaged graphs are compiled from 500 simulation executions. Each sample in simulated time for each metric is treated as a dependent statistical variable. The 500 simulation runs form a distribution of 500 sample points for each of these variables, and it is from here that median values are extracted.

6.3. Compilation of response distributions

The nine responses identified in Figure 4 and Section 4.2 are calculated for each individual simulation execution. Analyses of experimentation making use of multiple (*n*) simulation executions treats each of the nine responses as a dependent variable, drawing a population of *n* samples for each response from the *n* simulation executions. Note that the median response value as drawn from a population of *n* simulation runs will not necessarily equal the single response value derived from compiling the median execution of *n* simulation executions (as described above, in creating averaged graphs).

6.4. Calculating the A test

The following Matlab code is used to generate *A* test scores in this article. Note that it is only valid if the two distributions being compared contain the exact same number of samples.

```
function A = Atest(X,Y)
[p,h,st] = ranksum(X,Y,'alpha',0.05);
N = size(X,1); M = size(Y,1);
A = (st.ranksum/N – (N + 1)/2)/M;
```

Acknowledgements

Paul Andrews is funded by EPSRC grant EP/E053505/1. Work in Dr. Kumar's laboratory has been supported by grants from the National Institutes of Health, USA.

Notes

1. The EPSRC funded *Complex System Modelling and Simulation infrastructure* (CoSMoS) project. http://www.cosmos-research.org/.
2. With regard to simulation, predictive results are those which highlight *possible* scenarios that could arise in the real domain being modelled.
3. This is achievable in the real domain through a number of different interventions [21], whilst in the simulation it is trivial to revoke the ability of CD8Tregs to kill CD4Th1 cells, hence disabling regulatory activity.
4. One may not be certain that the prediction definitely *does* rely on this, since abstraction dictates that the relationship between simulation parameters and domain parameters is not exact. The analysis can, however, indicate that caution must be exercised when interpreting simulation results.
5. Samples from parameter space obtained using latin hypercube sampling can be computationally expensive to execute in the simulation. For example, parameters might be chosen that generate a lot of cells that do not die quickly. As such, only 300 simulation executions are performed here.
6. Available upon request from the corresponding author.

References

[1] D.E. Kirschner and J.J. Linderman, *Mathematical and computational approaches can complement experimental studies of host-pathogen interactions*, Cell. Microbiol. 11 (2009), pp. 531–539.

[2] A.L. Bauer, C.A. Beauchemin, and A.S. Perelson, *Agent based modeling of host-pathogen systems: The successes and challenges*, Inf. Sci. 179 (2009), pp. 1379–1389.

[3] 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*, J. Theor. Biol. 182 (2009), pp. 3706–3717.

[4] I.R. Cohen, *Modeling immune behavior for experimentalists*, Immunol. Rev. 216 (2007), pp. 232–236.

[5] S. Efroni, D. Harel, and I.R. Cohen, *Toward rigorous comprehension of biological complexity: Modeling, execution, and visualization of thymic T-cell maturation*, Genome Res. 13 (2003), pp. 2485–2497.

[6] I.R. Cohen, *Real and artificial immune systems: Computing the state of the body*, Nat. Rev. Immunol. 7 (2007), pp. 569–574.

[7] D. Young, J. Stark, and D. Kirschner, *Systems biology of persistent infection: Tuberculosis as a case study*, Nat. Rev. Microbiol. 6 (2008), pp. 520–528.

[8] N. Swerdlin, I.R. Cohen, and D. Harel, *The lymph node B cell immune response: Dynamic analysis in silico*, Proc. IEEE 96 (2008), pp. 1421–1443.

[9] A.K. Chakraborty and J. Das, *Pairing computation with experimentation: A powerful coupling for understanding T cell signalling*, Nat. Rev. Immunol. 10 (2010), pp. 59–71.

[10] H. Kitano, *Computational systems biology*, Nature 420 (2002), pp. 206–210.

[11] 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*, YCS-2010-453, Department of Computer Science, University of York, York, 2010.

[12] R.E. Langman and M. Cohn, *Editorial introduction*, Semin. Immunol. 12 (2000), pp. 159–162.

[13] J.C. Helton, *Uncertainty and sensitivity analyss for models of complex systems*, in *Computational Methods in Transport: Verification and Validation*, T.J. Barth, M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, and T. Schlick, eds., Springer, Berlin, Heidelberg, 2008, pp. 207–228.

[14] A. Saltelli, K. Chan, and E.M. Scott (eds.), *Sensitivity Analysis, Wiley Series in Probability and Statistics*, Wiley, New York, 2000.

[15] S. Marino, I.B. Hogue, C.J. Ray, and D.E. Kirschner, *A methodology for performing global uncertainty analysis and sensitivity analysis in systems biology*, J. Theor. Biol. 254 (2008), pp. 178–196.

[16] C. Beauchemin, J. Samuel, and J. Tuszynski, *A simple cellular automaton model for inuenza A viral infections*, J. Theor. Biol. 232 (2005), pp. 223–234.

[17] V. Kumar, *Homeostatic control of immunity by TCR peptide-specific Tregs*, J. Clin. Invest. 114 (2004), pp. 1222–1226.

[18] V. Kumar and E. Sercarz, *An integrative model of regulation centered on recognition of TCR peptide/MHC complexes*, Immunol. Rev. 182 (2001), pp. 113–121.

[19] T.R.F. Smith, I. Maricic, F. Ria, S. Schneider, and V. Kumar, *CD8α^+ dendritic cells prime TCR-peptide-reactive regulatory CD4$^+$FoxP3$^-$ T cells*, Eur. J. Immunol. 40 (7) (2010), pp. 1906–1915.

[20] M. Read, P.S. Andrews, J. Timmis, and V. Kumar, *A Domain Model of Experimental Autoimmune Encephalomyelitis*, Workshop on Complex Systems Modelling and Simulation, S. Stepney, P. Welch, P.S. Andrews, and J. Timmis, eds., Luniver Press, Beckington, UK, August 2009, pp. 9–44.

[21] V. Kumar, K. Stellrecht, and E. Sercarz, *Inactivation of T cell receptor peptide-specific CD4 regulatory T cells induces chronic experimental autoimmune encephalomyelitis*, J. Exp. Med. 184 (1996), pp. 1609–1617.

[22] L.T. Madakamutil, I. Maricic, E.E. Sercarz, and V. Kumar, *Immunodominance in the TCR repertoire of a TCR peptide-specific CD4+ Treg population that controls experimental autoimmune encephalomyelitis*, J. Immunol. 180 (2008), pp. 4577–4585.

[23] J.S. Menezes, P. Elzenvan den, J. Thornes, D. Huffman, N.M. Droin, E. Maverakis, and E.E. Sercarz, *A public T cell clonotype within a heterogeneous autoreactive repertoire is dominant in driving EAE*, J. Clin. Invest. 117 (2007), pp. 2176–2185.

[24] A. Vargha and H.D. Delaney, *A critique and improvement of the CL common language effect size statistics of McGraw and Wong*, J. Edu. Behav. Stat. 25 (2000), pp. 101–132.

[25] 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 (1979), pp. 239–245.

[26] G.C. Balan, C. Cioffi-Revilla, S. Luke, L. Panait, and S. Paus, *MASON: A Java Multi-Agent Simulation Library*, Proceedings of the Agent 2003 Conference, Argonne National Laboratory, Argonne, IL, 2003.

[27] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan, *MASON: A New Multi-Agent Simulation Toolkit*, Proceedings of the 2004 Swarmfest Workshop, George Mason University, Fairfax, VA, 2004.