Article:
Alden, Kieran James, Cosgrove, Jason, Coles, Mark Christopher et al. (2018) Using Emulation to Engineer and Understand Simulations of Biological Systems. IEEE/ACM Transactions on Computational Biology and Bioinformatics. ISSN: 1545-5963

https://doi.org/10.1109/TCBB.2018.2843339

Using Emulation to Engineer and Understand Simulations of Biological Systems

Kieran Alden * *Member, IEEE,* Jason Cosgrove *, Mark Coles, Jon Timmis *Senior Member, IEEE*

***Abstract*—Modeling and simulation techniques have demonstrated success in studying biological systems. As the drive to better capture biological complexity leads to more sophisticated simulators, it becomes challenging to perform statistical analyses that help translate predictions into increased understanding. These analyses may require repeated executions and extensive sampling of high-dimensional parameter spaces: analyses that may become intractable due to time and resource limitations. Significant reduction in these requirements can be obtained using surrogate models, or emulators, that can rapidly and accurately predict the output of an existing simulator. We apply emulation to evaluate and enrich understanding of a previously published agent-based simulator of lymphoid tissue organogenesis, showing an ensemble of machine learning techniques can reproduce results obtained using a suite of statistical analyses within seconds. This performance improvement permits incorporation of previously intractable analyses, including multi-objective optimization to obtain parameter sets that yield a desired response, and Approximate Bayesian Computation to assess parametric uncertainty. To facilitate exploitation of emulation in simulation-focused studies, we extend our open source statistical package, *spartan*, to provide a suite of tools for emulator development, validation, and application. Overcoming resource limitations permits enriched evaluation and refinement, easing translation of simulator insights into increased biological understanding.**

***Index Terms*—Emulation, Ensemble, Mechanistic Modeling, Sensitivity Analysis, Multi-Objective Optimization, Approximate Bayesian Computation, Machine Learning.**

I. Introduction

THE objective driving simulation-focused biological research is to generate novel predictions that increase our understanding of biological systems and inform laboratory studies. As simulations become more sophisticated, capturing complex diseases [1] and large-scale metabolic networks [2], this objective becomes more challenging. In addition, key research-led policy areas that exploit the benefits of simulation are seeing a focus shift, from a desire to understand average population behaviors to appreciating the range of behaviors observed within a population. This approach benefits applications such as person-centered healthcare [3], where a provision may be better suited to some individuals than others. Capturing increased complexity and individual heterogeneity can give rise to models that are time and resource intensive, and thus difficult to parameterize and evaluate. This in turn impacts the confidence one has in simulation-derived predictions, limiting the translation of these insights into further laboratory or clinical studies.

K. Alden, J.Cosgrove, and J.Timmis are with the Department of Electronic Engineering and York Computational Immunology Lab, University of York, Heslington, York. YO10 5DD, United Kingdom. M.Coles is with the Kennedy Institute of Rheumatology at the University of Oxford. J.Cosgrove and M.Coles are also with the Centre for Immunology and Infection at the Department of Biology and Hull York Medical School, University of York, Heslington, York. YO10 5DD, United Kingdom. J.Timmis and M.Coles are also with SimOmics Ltd, York, UK. * These authors contributed equally to this work. Correspondence e-mail: kieran.alden@york.ac.uk.

A. Performance Issues in Analyzing Simulations

Significant insights are being generated from non-deterministic models designed to incorporate stochasticity and heterogeneity observed in real life systems. In applications such as target evaluation for drug discovery and understanding emergence of disease dynamics from individual cellular interactions, the incorporation of stochastic molecular, cellular, and environmental processes is desired to ground the model in the domain being explored [4], [5]. Although the composition of non-deterministic models may themselves not be that complex or computationally intensive, diverse sets of outputs may be produced for a fixed parameter input [6]: a factor usually mitigated by summarizing replicate executions. Ensuring enough replicates are performed such that this summary is representative of the parameter input is critical for statistical analyses, specifically sensitivity analyses, that permit systematic exploration of the parameter space and elucidation of the pathways impacting simulation response [7]. An increase in model complexity gives rise to high-dimensional parameter spaces, that require significant computational infrastructure to explore, especially if a large number of replicate executions are required per parameter set. As it is common to simulate biological systems for which our understanding remains incomplete, there may be significant uncertainty around a subset of these parameters: their value range may remain unknown or poorly constrained [8]. This parametric uncertainty impacts the calibration process used to align simulation behaviors to a desired or expected response, complicating both the formation of a baseline state to which subsequent perturbations are compared [9], and understanding the range of parameter values that produce that desired response. The latter is of critical importance when considering model selection, or in capturing individual heterogeneity by performing executions where heterogeneous individuals within a population are represented by simulation executions of different parameter sets.

A range of statistical analysis techniques can be applied to understand and mitigate the factors above. Yet as the execution time for a simulation increases, it becomes less tractable to perform these analyses in a time-frame that can run parallel to laboratory or clinical studies. We have previously described techniques that aid in quantifying the number of replicate executions required to ensure a result is representative

of a specified parameter set [6], [10], mitigating *aleatory* uncertainty. We have shown that agent-based simulations that capture both stochasticity and heterogeneity can require hundreds of replicates to generate a representative output for a single parameter set [10], [6]. Sensitivity analyses may incorporate both a local parameter analysis that assesses the uncertainty around the value of each parameter individually, and global analyses that can reveal non-linear relationships between model parameters. For the latter, adequate sampling of the parameter space is crucial. Often a latin-hypercube (LHC) sampling scheme is adopted [11], where a number of model parameter sets are generated and a Partial Rank Correlation Coefficient (PRCC) calculated to quantify any effect between a parameter and model response [6]. However, summarizing parameter sensitivities through a PRCC may not capture the magnitude or non-monotonic relationships between parameter inputs and emergent outputs of the simulator. Alternative parameter sampling approaches include the extended Fourier amplitude sampling test (eFAST) [12], where parameter samples are selected from sinusoidal curves through the parameter space, with each parameter taken in turn as that of interest and sampled at a significantly different frequency than its complementary set. Statistical analyses of simulation executions under these conditions provides a partition of the observed variance in response between the parameters of interest, indicating those having significant impact on behaviors. Although a powerful technique, the characteristics of this sampling approach give rise to a significant number of parameter sets. For a simulator of six parameters, taking 65 samples from each sinusoidal curve, with three curve phase shifts introduced to mitigate selection of near identical parameter sets [12], a total of 1,170 parameter sets is generated. In our previous application of this technique, where a simulator required 500 executions to mitigate aleatory uncertainty, 585,000 simulation executions were required [10]. Even with the availability of high-performance computing resources, such resource-intensive analysis become intractable for simulators with a long execution time.

A range of additional techniques have shown similar promise in understanding parametric uncertainties and optimizing parameter configurations with respect to a desired output, automating the calibration process. Approximate Bayesian Computation (ABC) techniques provide a means of understanding the uncertainty around each parameter value by generating posterior distributions for each [13], [14]. This makes it possible to sample parameters from a distribution predicted to replicate behaviors that align well to a desired response, rather than calibrate parameters to an individual value. Such sampling may be a powerful approach to adopt in person-centered studies where each patient can be represented as a parameter configuration sampled from the posterior distributions. Similarly, multi-objective evolutionary algorithms (MOEAs) have shown promise in addressing problems such as parameter calibration [9]. There may be several simulation responses that should be matched against experimentally observed data: it may be the case that the accuracy of one simulation response to the observation cannot be improved without compromising other responses [15]. Whereas ABC gives a distribution of values in which a parameter may lie, MOEA techniques permit identification of the optimal trade off between those simulation responses and the associated parameter configurations under which that outcome is achieved [16]. In optimization routines, an MOEA approach could thus be used to find a set of parameter configurations for an alternative desired outcome. Both ABC and MOEA adopt a heuristic approach where parameter sets are iteratively generated, executed, and evaluated until a convergence criterion is met. It is thus difficult to be aware of the execution time required for both analyses prior to execution, limiting the application of these analysis in time and resource intensive projects.

B. Addressing Performance Issues using Machine Learning

A surrogate tool, or emulator, that is capable of converting a set of parameter values into a prediction of the simulation response that is representative of a high number of replicates, is an attractive option for reducing resource requirements [17]. In saving resources, emulation can serve as a useful adjunct to the original simulator, providing insights where complex analyses were previously intractable. This could have a significant impact on the outcomes of a model-informed biological research project. We have previously noted that for any simulation result to be meaningful in the context of the real biological system, it is critical that the relationship between the model and the real-world it captures is understood [18], [19]. We note that producing an emulator that captures a simulation does add a further layer of abstraction from the real biological system, and does not make the simulator itself entirely redundant. However, if the accuracy of that emulator can be quantified and understood, a useful tool is be produced that provides a means of complementing existing simulation analyses while enhancing the range of potential analyses that could be performed. This approach could be applied at all phases of simulator development, from highlighting potential coding errors prior to running complex analyses, refining model design by gaining an initial understanding of influential simulated pathways, and informing analyses to be performed using the simulator.

Emulation has primarily been achieved through a Bayesian approach where a statistical model, usually a Gaussian process, is used to estimate simulator outputs. Such emulators have been applied to aid parameter estimation in a stochastic model of mitochondrial DNA population dynamics [20], an epidemic model of influenza [21], and models of hormonal crosstalk in Arabidopsis root development [22]. Machine learning approaches, powered by recent technical advances in computation and increased availability of large datasets, have also shown promise in identifying complex non-linear relationships within multivariate datasets [23]. Using supervised learning approaches, a machine learning algorithm can learn the behaviors of a simulator, to quickly and accurately predict the simulation response for parameter sets the algorithm has yet to observe. This attribute makes machine learning algorithms well placed to emulate simulators of biological systems, as illustrated by the use of support vector machines to emulate models of haemorrhage and renal denervation, resulting in a 6-fold decrease in computation time [24].

C. Emulation to Understand Models of Biological Systems

Previously we developed an agent-based model of the pre-natal development of Peyer's Patches (PP), a secondary lymphoid tissue that triggers adaptive immune responses to infection [25], [10], [26]. This simulator, described in the cited works and introduced briefly in Figure 1(B) and the Methods, was applied within a sensitivity analysis routine, to determine the key biological mechanisms that influence cell behavior during the process of tissue development. This routine utilized our previously published sensitivity analysis tool, *spartan* [5], [27], [28]. This published study utilized local and global sensitivity analyses to: reveal how robust parameters for which a value was unknown are to perturbation; reveal non-linear interactions between parameters; and to partition the variance between those parameters. These analyses produced the hypothesis that lymphoid organ development may be biphasic: one that has since been verified in the laboratory [25]. As this simulation captures the emergent behaviors from interactions of hundreds of heterogeneous individual cells, there is a high level of stochasticity. A substantial amount computer and time resources (Table 1) (on the order of months) were required to perform these sensitivity analyses, limiting application of additional analysis techniques such as ABC and MOEA.

A range of different machine learning approaches have been developed [29], [30], [31], [32], each with their own set of advantages and limitations, with performance of each specific to the data on which is is trained [33]. In this paper we explore the relative performance of a range of these techniques in predicting outputs obtained from the agent-based model. We show that one technique may have poorer predictive power on a section of the parameter space than another, yet outperform other techniques for an alternative region. To mitigate this effect, we combine different algorithms into a hybrid tool, or ensemble, that is capable of outperforming each technique in isolation. Using the ensemble, we replicate previously published statistical analyses in the order of seconds rather than months. With strong performance assured, additional analysis routines that enrich our understanding of the simulator yet were previously intractable have now been conducted using the ensemble. These results provide a strong argument for the use of machine learning approaches in supporting the engineering and enriched analysis of simulations of biological systems (Figure 1(A)). To promote the adoption of emulation in the systems biology community and aid others in evaluating the approaches described herein, we extend existing functionality within *spartan*, to permit the generation, validation, and application of emulators and ensembles. The extended tool is available from the Comprehensive R Archive Network (CRAN), and supported by tutorials and example simulation data available from the *spartan* website (http://www.york.ac.uk/ycil/software).

In the description of our Results, gained using the additional functionality in *spartan*, section A details the application of a range of supervised machine learning approaches to generate emulators of a simulation, each trained using a latin-hypercube sample of the parameter space. Section B examines the performance of each machine learning technique in isolation. In Sections C and D we detail the generation and performance of combining the emulators into an ensemble capable of rapidly and accurately reproducing simulator behaviors. Section E replicates the previously conducted sensitivity analyses for multiple simulation time-points, with results consistent with published simulator results. The significant improvement in performance facilitated enriched analyses, specifically Approximate Bayesian Computation and Multi-Objective Optimization, detailed in Section F. A discussion then follows on the role that machine learning techniques and our extended *spartan* tool could have within a process of developing and understanding models of biological systems.

II. Methods

A. Case Study Simulation

Given PP emerge through interactions between two populations of hematopoietic and non-hematopoietic stromal cells, mediated by expressed chemoattractant factors within the developing tissue's local environment and factors that aid cell adhesion in that locality, our model adopts an agent-based approach. Each cell involved in PP development is explicitly captured in the model as an individual entity, each possessing their own attributes, and interacts with other cellular and environmental actors in accordance with a specified set of rules [10]. The total number of cells modelled is set to match the estimated counts of each cell population estimated from flow cytometry experiments. Expression and response to adhesion factors and chemokines in the environment is modelled using mathematical constructs, controlled by the parameters identified in Figure 1(A). In the laboratory cell velocity and displacement behavior responses have been established by observing cells using an *ex vivo* cell culture system [25] for a one-hour period, providing a baseline through which to parameterize the simulation and suggest the values to which the mathematical constructs must be set in order to capture observed behaviors. Thus the simulation outputs cell velocity and displacement for all agents over the same one hour period and twelve-hour intervals that follow, as well as a calculation of the size of the cell aggregations that develop. Sensitivity analysis techniques were applied that perturbed the values of these parameters in order to examine how cell velocity and displacement alters under different physiological conditions [5].

B. spartan

Open source and supported by multiple publications and tutorials, *spartan* comprises a suite of statistical analyses that aim to help understand how simulation-derived predictions could be interpreted in the context of the biological system being captured. The datasets originally released with *spartan* have been used in this study, providing an accessible set of data for demonstrating application of emulation and easing reproduction of the presented analyses. *spartan* has been extended to offer four additional techniques: (i) Generation of emulations using five machine learning techniques; (ii) Generation of an ensemble that combines these emulators into one single predictive tool; (iii) Provision of a software

wrapper that permits the use of an ensemble for performing Approximate Bayesian Computation; and (iv) Provision of a software wrapper permitting the application of a multi-objective optimization algorithm, through which the ensemble is used to locate parameters that lead to a desired emulated outcome. The latest version exploits the functionality in a number of additional R packages, namely *randomForest* [34], *mlegp* [35], *neuralnet* [36], *e1071* [37], *mco* [38], and *plotrix* [39].

C. Specification of Computer Resources

The simulation runs were performed on the York Advanced Research Computing Cluster, a resource of 70 nodes, 138 processors, 1462 cores, and 10.2TB RAM. The emulators and ensembles were generated and used for experimentation on an Apple MacBook Pro, 2.5GHz Intel four core i7, 16GB RAM.

D. Emulator Creation

1) Generation of Training, Testing and Validation Datasets: The *spartan* tutorial dataset for demonstrating performance of a sampling-based global sensitivity analysis using LHC sampling consists of 500 parameter sets. Each set was executed 500 times to mitigate aleatory uncertainty, and median responses calculated to summarize simulator performance under those conditions [26]. Spartan divided the data set into training (75%), testing (15%) and validation (10%) sets (percentages can be changed), which were used to create and assess the performance of emulators generated using five machine learning algorithms. One emulation was generated for each simulation response (cell velocity and displacement), such that the performance of one response does not bias another.

2) Neural Networks (ANN): ANN's are inspired by the neuronal circuits in the brain, with computations structured in terms of an interconnected group of artificial neurons. During the learning phase, the weighting of connections between neurons are adjusted in such a way that the network can convert a set of inputs (simulation parameters) into a set of desired outputs (simulation responses). Neural networks were developed in *spartan* using the *neuralnet* package [36] with supervised learning of the data achieved through backpropagation. To determine optimal hyperparameters of the network we performed ten-fold cross validation (default value, but can be altered) on a selection of structures with five inputs (the parameters) and two outputs (velocity and displacement), with one to four hidden layers (the specific details are covered in the software tutorial). The number of generations defaults to 800,000, but can be modified by the user. The accuracy of each fold was determined to be the root mean squared error (RMSE) between the predicted cell behavior responses and those observed in the simulation, and the accuracy of the network structure determined to be the average of the ten fold RMSE. The network structure with the minimum average RMSE was selected as the structure that would be used in creation of the emulator.

3) Random Forest (RF): A decision tree is structured to convert inputs (parameters) into a set of predicted outputs, and comprises root, internal and leaf nodes. Each internal node represents a decision with two branches leading to stratification of the training data, in this case for the purpose of regression. A RF is an ensemble of decision trees, trained on different parts of the same training set, with the goal of avoiding issues of overfitting [29], [40]. The RF was generated within *spartan* using the *randomForest* package [34] with supervised learning achieved by creating a forest with 500 trees and no limitation on tree depth or maximum number of terminal nodes (as default in the *randomForest* package.

4) Gaussian Process (GP): A GP model is a non-parametric approach that finds a distribution over the possible functions that are consistent with the observed data facilitating supervised learning of simulator outputs. A Gaussian process model was created in *spartan* using the *mlegp* package [35] with default parameter settings.

5) Generalized Linear Model (GLM): A GLM is a generalized form of ordinary linear regression, allowing for predictions of simulator outputs without assuming that the error distributions follow a normal distribution. A GLM was created in *spartan* using the *glm* method in the *base* R package, with default parameter settings.

6) Support Vector Machine (SVM): A support vector machine constructs a hyperplane, or set of hyperplanes within a feature space to facilitate classification and regression predictions [30]. The svm model was generated within *spartan* using the *e1071* package [37] using a radial basis kernel. The parameter epsilon, which controls the threshold error for fitting the hyperplane, and the cost parameter, the penalty for violating a constraint that can be adjusted to deal with overfitting left at default values of 0.1 and 1 respectively.

7) Evaluating Emulator Performance: Emulator performance was evaluated by calculating the RMSE between the set of emulator predictions for unseen parameter values in the test data with simulator responses observed under those parameter conditions.

E. Ensemble Creation

Each individual emulator is used to make predictions of the simulator output responses for the parameter values in the test set. The predictions from each emulator form input nodes to a neural network, with the output nodes being the actual observed responses from the test set. A network consisting of one hidden layer with a single node is used to calculate weightings of each algorithm's performance. The relative weighting of each algorithm is then used to combine emulator responses to form an ensemble. It may not be the case that an ensemble of all five machine learning techniques provides better accuracy than by combining a subset of emulators. As such, we assessed all combinations of emulators, determining the optimal ensemble structure that provided the lowest RMSE, averaged across all simulation responses, between predicted values and those observed from the original simulator. Consequently, the total time taken to generate an ensemble (shown in Table 1) will be dependent on the emulators which the ensemble includes.

F. Sensitivity Analysis Using Ensemble

1) Sampling-Based Sensitivity Analysis: A new list of 500 parameter sets was created for the parameters identified in

Figure 1 using the LHC sampling method in *spartan*. The generated CSV file of parameter values and the optimal ensemble was specified as input to a new *spartan* method designed to generate responses for each parameter set using an ensemble. This produces a CSV file summarizing ensemble response for each parameter set. Creation of this file permits result analysis using the pre-existing techniques within *spartan* [5], [27]. This analysis produces a Partial Rank Correlation Coefficient (PRCC) for each parameter value, that quantifies the relationship between a parameter and an output response, providing an indication as to the influence of that parameter, although the values of the complementary set are also being perturbed. PRCC values were generated for all parameter-measure pairings, for all simulation time-points (hours 12-72, in 12 hour increments), permitting direct comparison to analyses previously conducted using the simulator [25], [5], [26].

2) Variance-Based Sensitivity Analysis: A new list of parameter value sets for performing an analysis using eFAST were obtained using the parameter sampling method in *spartan*. A dummy parameter was introduced to the sampling, giving seven parameters. In accordance with guidance concerning eFAST sampling frequency [7], 65 values were sampled from the sinusoidal curves that cover the value space for each parameter, generating 390 (65*7) value sets. Due to the properties of sigmoidal sampling and a high chance of repeated values, repeated sampling after a frequency shift is suggested. Applying three frequency shifts (curves) generated a total of 1,170 parameter sets for analysis. Similarly to the sampling-based analysis above, a new method has been included in *spartan* that processes the CSV value files generated for each resampling curve, generating output predictions for each parameter set using the ensemble. Again these can be analysed using the pre-existing techniques within *spartan*. For each simulation response, variance in output was attributed to each of the seven parameters (Si value). The Si values were calculated for both cell velocity and displacement at both hours 12 and 72, to permit comparison with previously published eFAST results obtained using the simulator.

G. Enriched Analysis Using Ensemble

1) Approximate Bayesian Computation: The R package EasyABC [41] provides a number of algorithms through which parameter posterior distributions can be predicted. In the extended version of *spartan* we provide a wrapper that normalizes the parameter sets generated by the ABC algorithm and inputs these into the ensemble, before re-scaling the predictions and returning those values back for assessment of fit against a specified tolerance level. In the analyses presented in this paper, we ran the Delmoral implementation of the sequential ABC algorithm [42], with the target summary statistics of cell velocity and displacement being the medians of the cell behavior measures observed in *ex vivo* culture and published previously. The algorithm parameters were set at the default values given in the EasyABC documentation.

2) Multi-Objective Optimization (MOO): MOO was used to find parameter sets that met three objectives at hour 72: minimize the RMSE between emulator and simulator responses for cell velocity; minimize the RMSE between emulator and simulator responses for cell displacement; and maximize the area of the cell aggregation that develops (the PP). These sets were derived using the non-dominated sorting genetic algorithm II (NSGA-II) [43] using the mco R package [38]. With a population size of 300, values for generation number (400), mutation (0.8) and crossover probabilities (0.4) (**table 2**) were determined by sensitivity analysis, choosing parameters that performed well on all three objectives and maximized the variance of the parameter inputs. As we wished to replicate the cell behaviour responses, the parameter values were constrained to match the predicted posterior distributions observed using the EasyABC package: distributions observed in Figures 6 and S6.

III. Results

A. Emulation Generation

Our approach utilized the *spartan* tutorial dataset as described in the Methods to generate emulators using five machine learning algorithms, with generation time of each shown in Table 1. To indicate success of the training process, and provide a comparison of performance with the test set, the RMSE obtained in training each algorithm is compared in Figure S4. To emulate and replicate previously published temporal sensitivity analyses [5], [28], where simulation behaviors were assessed at twelve-hour intervals, emulators were generated at twelve hour intervals to hour 72.

B. Emulator Performance

Emulator performance data for cell velocity at hour 12 is shown in Figure 2(A), with performance comparison of cell displacement available in Supplementary Figure S1. The hour 12 dataset facilitated a comparison of how each algorithm can learn a highly skewed dataset (kurtosis: 6.353, Figure 2(B)) with fewer examples towards the lower end of the distribution. This artifact impacted the performance of the support vector machine, random forest, and generalized linear model algorithms, with less of an impact observed for Gaussian process and neural network derived predictions. Emulator performance on both cell velocity and displacement responses at hour 72 can be seen in Supplementary Figures S2-S3.

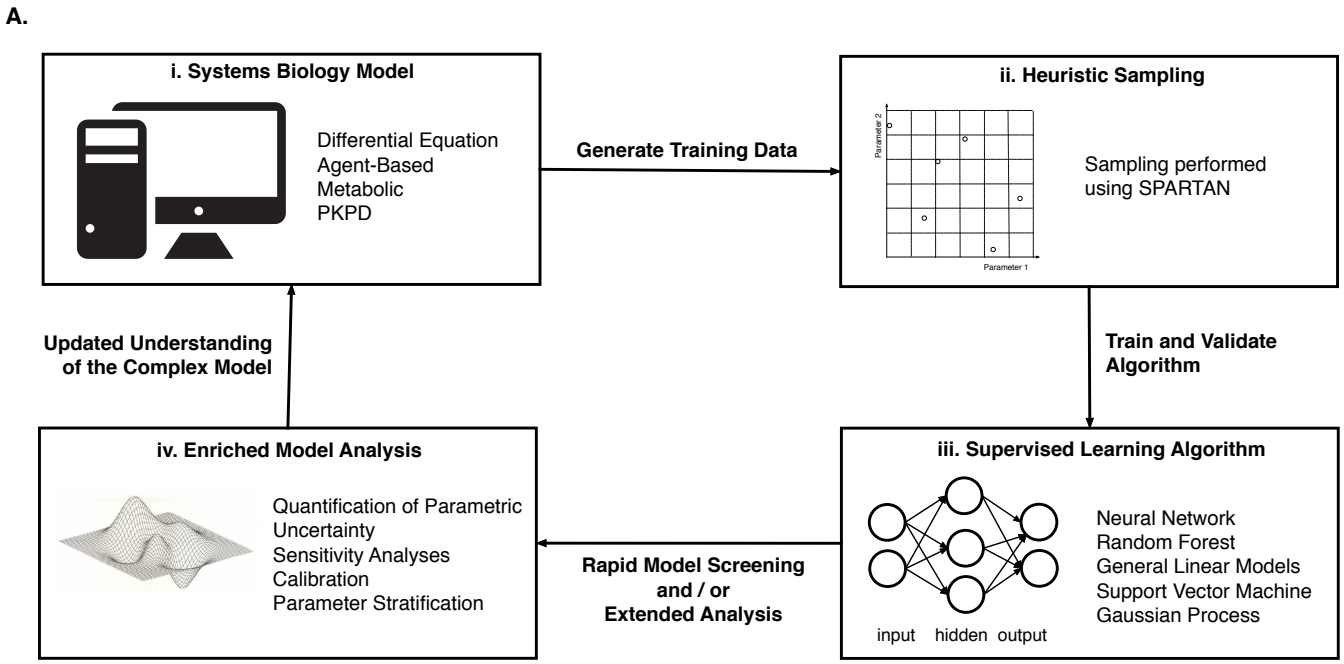
C. Ensemble Generation

From the respective emulators at each time-point and each response, *spartan* was used to create ensemble models, through combining emulators into one predictive tool. Ensemble generation times for both hours 12 and 72 are listed in Table 1.

D. Ensemble Performance

As the test subset of the partition data was used to weight performance of each emulator and thus derive the best performing ensemble, performance of the ensemble itself was assessed by comparing response predictions for the parameter values in the validation set with those observed from the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2843339, IEEE/ACM Transactions on Computational Biology and Bioinformatics



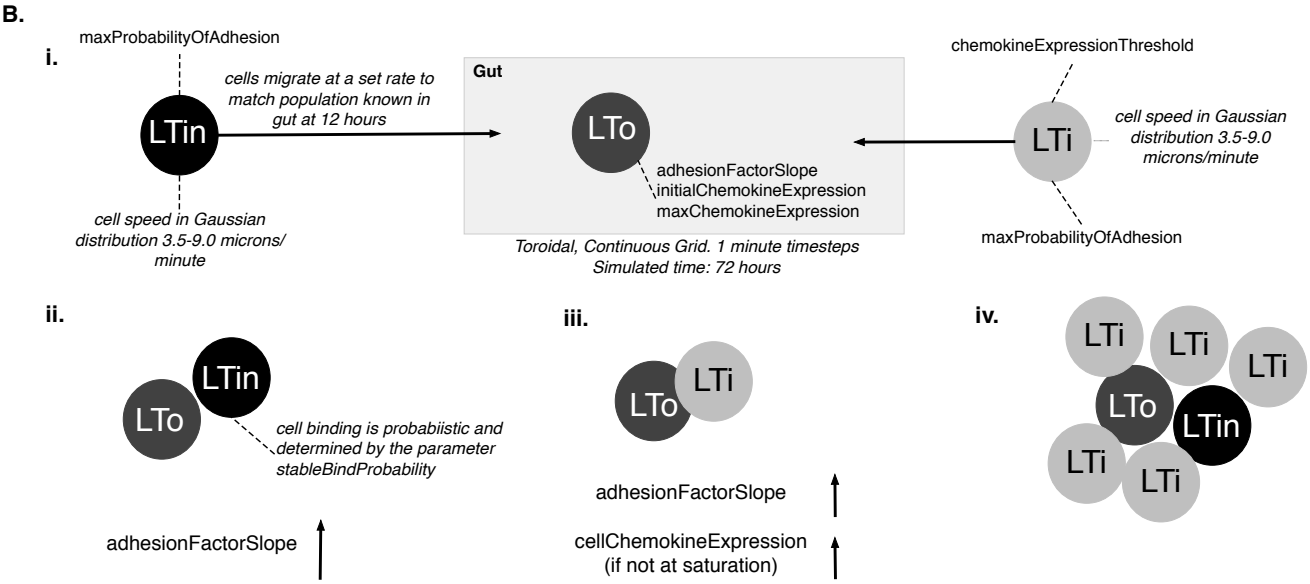


Fig. 1. A: A framework to emulate simulations of biological systems. The behavior of a systems biology model (i) is summarized by applying latin-hypercube sampling (ii), with simulation results under those conditions generating a dataset used to inform the training and validation of an emulator using machine learning techniques (iii). The emulator is then used in place of the systems biology model to accurately and rapidly predict responses to conduct a suite of analyses that may be intractable using the original simulator (iv). Emulator development, validation, and analysis techniques have all been incorporated within the *spartan* R package. B: Schematic overview of the case study model of Peyer's Patch (PP) development. Full implementation detail can be found in our previously published work [10]. (i) The model captures the migration and aggregation of Lymphoid Tissue Initiator (LTin) and Lymphoid Tissue Inducer (LTi) cells into the developing gastrointestinal tract, and their interaction with Lymphoid Tissue Organiser (LTo) cells, modeled using six key parameters. Both LTin and LTi cells express adhesion receptors, modelled using a mathematical construct controlled by parameter *maxProbabilityOfAdhesion*, to model the probability the receptor binds to adhesion factors expressed by the LTo . LTi cells express chemokine receptors that are controlled by the parameter *chemokineExpressionThreshold* to determine whether an LTi cell responds to chemokine expression in it's vicinity. Adhesion factor expression by an LTo cell is represented using a linear model function that is adjusted with each stable cell contact by increasing the parameter *adhesionFactorSlope*. Chemokine expression across the environment is varied between *initialChemokineExpression* and *maxChemokineExpression*. (ii) LTin and LTo cell contact causes LTo differentiation, increasing adhesion factor expression. Success of receptor binding is captured using probability parameter *stableBindProbability*. (iii): LTi and LTo contact causes further LTo differentiation and increased expression of adhesion factors, in addition to increased expression of chemokines. (iv) This processes give rise to the emergence of cell aggregates that become PP. The simulator outputs the area of this aggregation at hour 72.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2843339, IEEE/ACM Transactions on Computational Biology and Bioinformatics

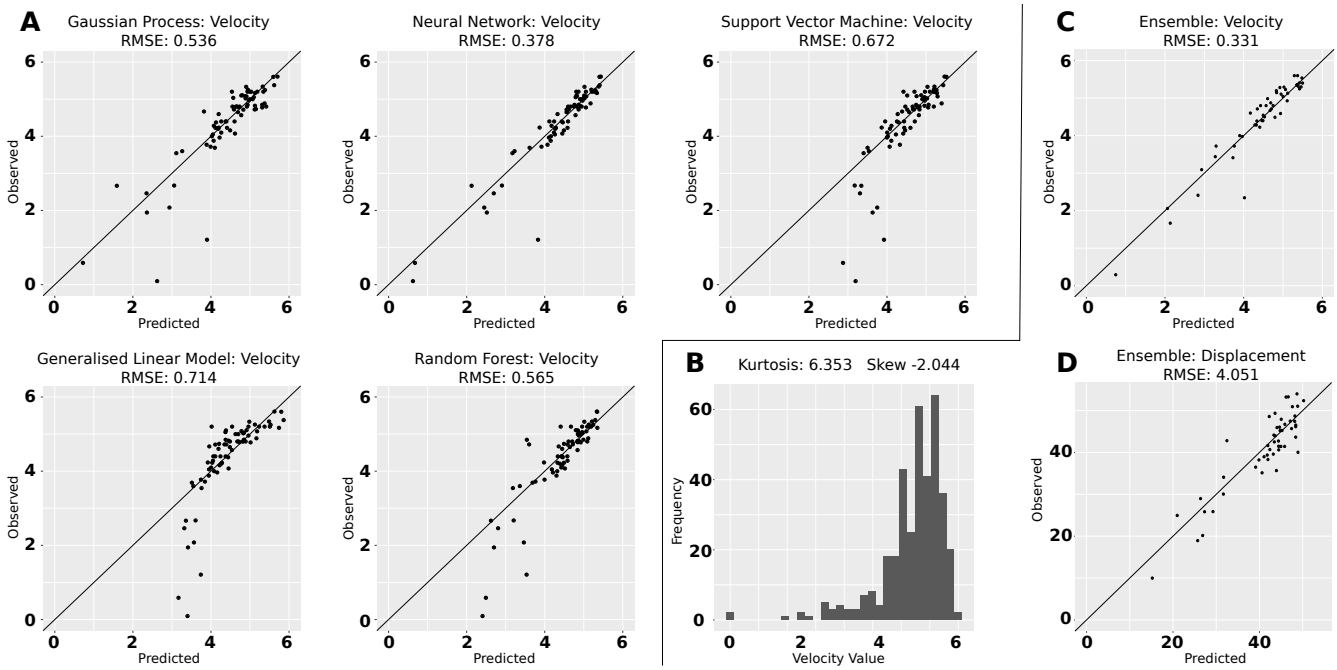


Fig. 2. Emulation performance, integration, and comparison. A: Performance of five machine learning techniques (stated in the graph header) in predicting simulator responses, in this case cell velocity at the twelve-hour time-point of the case study simulation. B: The distribution of the training dataset for the Velocity response at the twelve-hour timepoint. C-D: An ensemble approach that combines multiple machine learning techniques out-performs each technique in isolation. Both responses at hour 72 are shown in Supplementary Material (Fig. S2-S3).

published simulator. This comparison is shown for both cell velocity and displacement at hour 12 in Figure 2(C-D). We observed a decrease in the RMSE for both measures. For velocity an RMSE of $0.331\,\mu m$/minute is observed, an improvement of the lowest RMSE found when using a single emulation approach: the $0.378\,\mu m$/minute obtained using a neural network. For displacement, an RMSE of $4.051\,\mu m$, again an improvement on the $5.223\,\mu m$ observed using a single machine learning approach. We present ensemble performance results for both cell responses at hour 72 in Supplementary Figures S2-S3. Our results suggest that an ensemble of machine-learning approaches does outperform each technique in isolation, and is capable of mitigating characteristics of the training dataset, such as the skew mentioned previously.

E. Sensitivity Analysis

With the ensemble generated and performance assured, we replicated the sensitivity analyses that had previously been conducted at hours 12 and 72 [10], [5], [28], using the ensemble in place of the original simulator. We contrast performance both against the original analysis results and in terms of time and resource requirements necessary to perform these analyses. The computing resources used, from which the wall time statistics were generated, are specified in section C of the Methods.

1) Sampling-Based Global Analysis (LHC): Partial Rank Correlation Coefficients were calculated for each parameter-response pairing using the latin-hypercube sampling and analysis technique described in the Methods. This analysis took 6.45 seconds for hour 12 and 3.49 seconds for hour 72. Note that this analysis was conducted for a new set of 500 parameter sets, derived using *spartan*, as the ensemble had been trained on the parameter values used in the published analyses. The results for two parameters, controlling the probability of cell adhesion and response to chemokine expression, are shown for cell velocity in Figure 3(A), the results obtained using the ensemble on the left against the original simulator analysis on the right. For *maxProbabilityOfAdhesion*, both results show a clear trend in the data, supported by a high correlation coefficient. The ensemble has replicated the original analysis hypothesis that the probability of cellular adhesion significantly influences cell behavior, although another five parameters are also being perturbed. This provided confidence that the emulator could capture complex interactions between parameters. For the parameter *chemokineExpressionThreshold*, the original analysis found no correlation between parameter value and output response [10], a finding supported by additional local sensitivity analyses that suggested a perturbation in this parameter had little impact on cell behaviour [5]. This result is again replicated using the ensemble.

Figure 3(B&C) ease comparison of the results generated by the ensemble with those of the original analysis by presenting the PRCC values as a polar plot, one for each cell behavior response. In the published simulator analysis, a significant negative correlation is suggested between the probability of cell adhesion and cell displacement, contradicting the accepted hypothesis that chemokine expression is the critical pathway in PP development [44]. The emulator replicates that suggestion for both cell velocity and displacement. When considering cell velocity for hour 12 (Figure 3(B)), the emulator produces PRCC values that are quantitatively very similar to those in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2843339, IEEE/ACM Transactions on Computational Biology and Bioinformatics

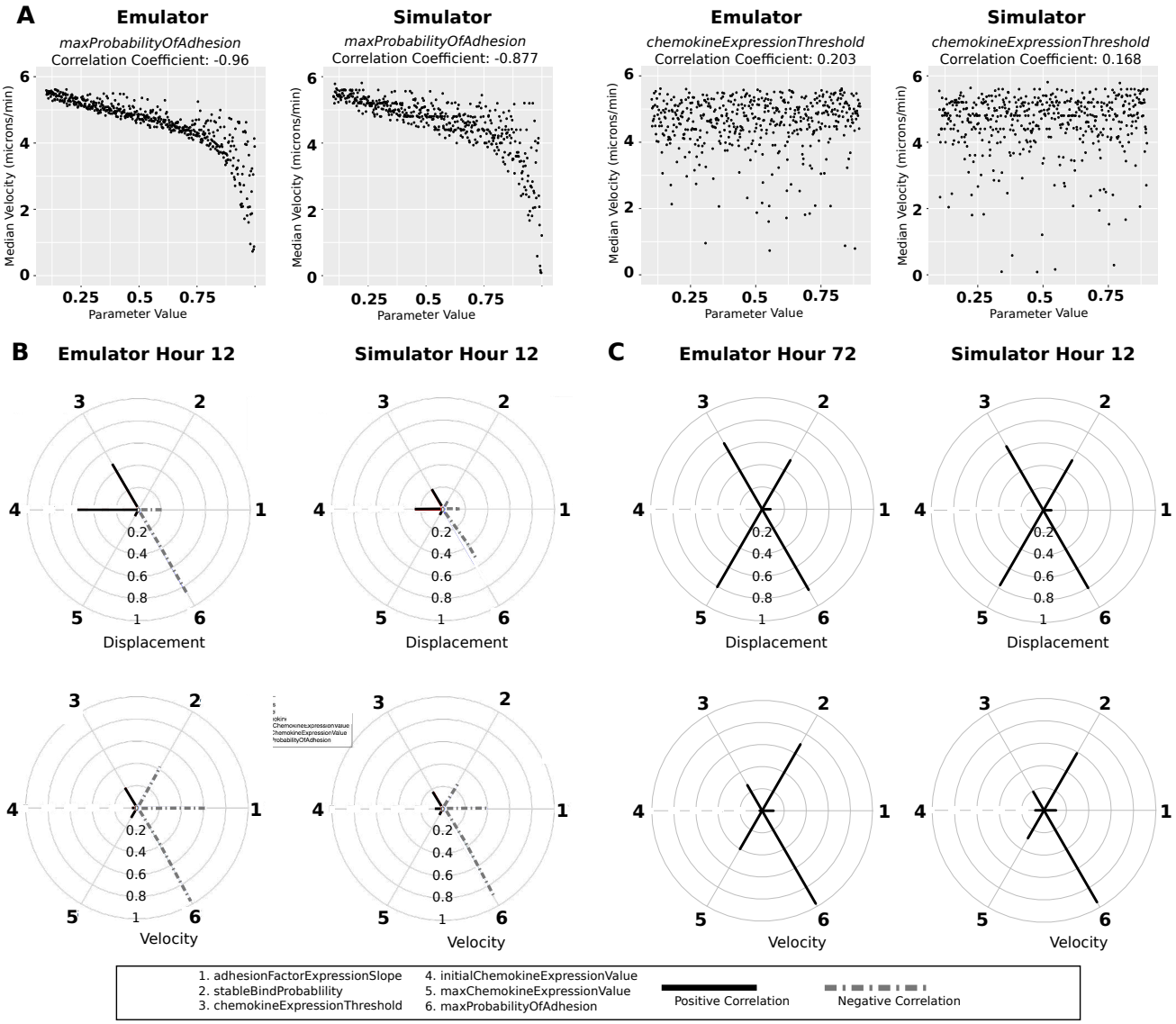


Fig. 3. Replicating simulator sensitivity analysis using an emulator. A: The emulator is capable of capturing the key behaviours observed in a global sensitivity analysis, using latin-hypercube sampling and calculation of summary Partial Rank Correlation Coefficients (PRCC), illustrated here for parameters *maxProbabilityOfAdhesion* and *chemokineExpressionThreshold*. B: Comparison of the PRCC values for all parameters at the twelve-hour time-point obtained using both the simulator and emulator. C: Comparison of a the PRCC values obtained at the 72 hour time-point

the original time intensive analysis. For cell displacement, the ensemble does suggest a stronger correlation between cell displacement and both chemokine response and initial chemokine expression parameters than that suggested by the simulator analysis at hour 12. The analysis, constructed to mimic that conducted in the laboratory [25], examines the behaviors of cells within $50\,\mu m$ of a developing PP. At this early time-point, measures of cell displacement are sensitive to the number of cells that are located in this vicinity: a number that can be very low in some cases. A low number of examples thus impacts the ability for the machine learning algorithm to predict this response. As simulation time progresses and additional cells enter this vicinity, a higher number of cells provides more data on which to train the ensemble, and accuracy for the cell displacement measure improves (Figure 3(C)).

2) Variance-Based Global Analysis (eFAST): The original application of the eFAST analysis using *spartan*, described in detail in the Methods, required 682,500 executions of the simulator: an intensive analysis that is potentially intractable for studies with a greater number of parameters than that of the presented case study. This analysis was repeated using the ensemble, and the calculated variance in simulation response that can be attributed to each parameter (the Si value) presented in Figure 4. This analysis took 14.67 seconds for hour 12 and 6.00 seconds for hour 72. When considering cell displacement at hour 12, the original simulator analysis found the maximum probability of cell adhesion accounted for more variance than the complementary set (Figures 4(A-B)). Again the emulator reproduces this finding, but assigns much more of the variance to this one parameter. However the performance of the emulator is much more comparable

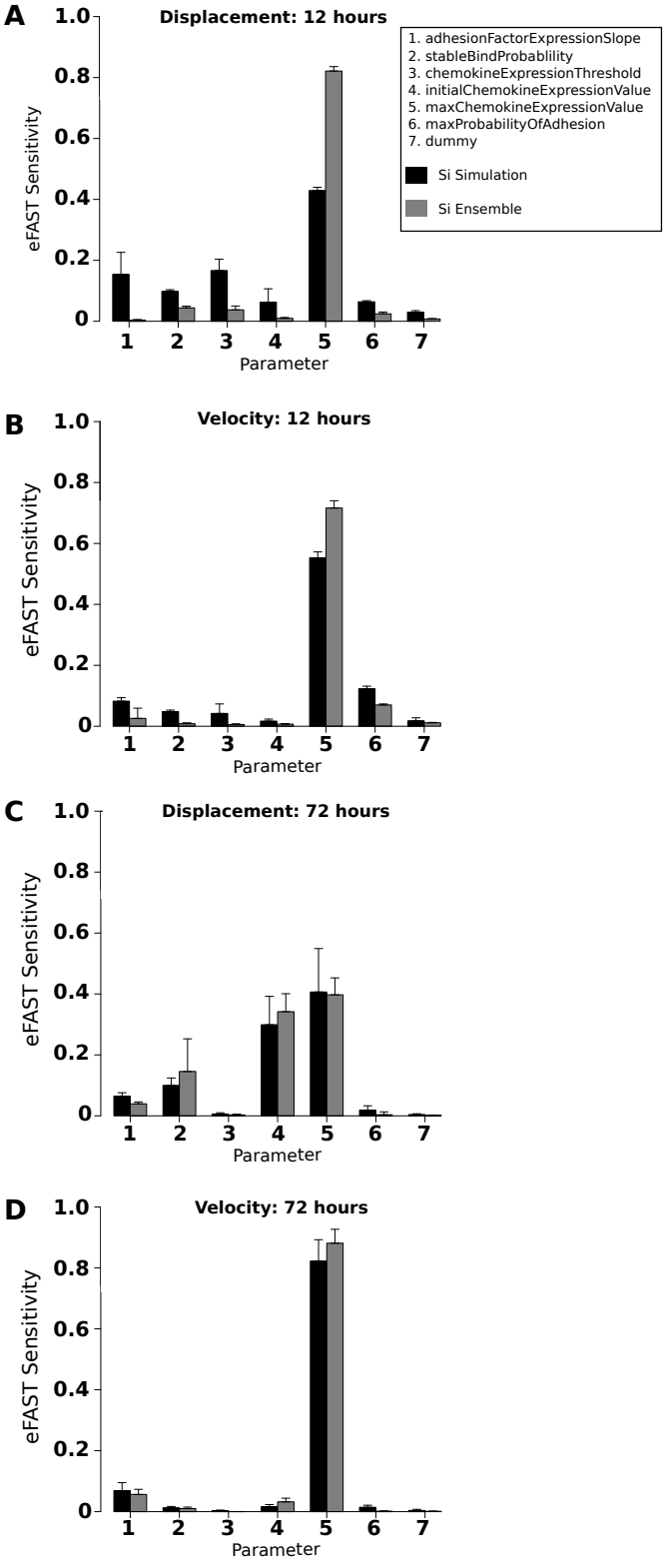


Fig. 4. Reproducing an extended Fourier amplitude sampling test, comparing the assigned partition of variance values (Si) obtained for the simulator with those obtained using the emulator. Si values for cell displacement at 12 hours compared in A, 72 hours in C. The same comparison for cell velocity is shown in B and D.

to that of the simulator analysis at hour 72 (Figure 4(C-D)). This difference supports conclusions made previously when examining the PRCC values, that predictive power may be impacted by a lower number of examples at an early time-point in PP development. This affect is also observed when contrasting the amount of variance accounted for by each parameter and higher-order interactions with others: a comparison made in Supplementary Figure S5. Here it can be observed that the ensemble is capable of predicting these higher-order interactions, with predictive power again increasing throughout the simulation timecourse.

3) Temporal Sensitivity Using the Ensemble: Previously we applied the case study simulator and sensitivity analysis methods in *spartan* to suggest that PP development may be biphasic: dependent on adhesion factor expression at hour 12 yet highly influenced by chemoattractant expression and response at hour 72 [28]. By contrasting PRCC values for the six parameters obtained at twelve hour intervals, we were able to suggest that a change in the influence of a subset of the simulator parameters occurs between hours 24 and 36. Using the approach described in the methods we created an ensemble for each twelve hour interval, permitting a replication of this temporal analysis (Figure 5). Using each ensemble and *spartan*, PRCC values were calculated for each parameter-response pairing at each interval. It is clear that the ensemble has captured the performance of the simulator over the time-course for all parameters and simulation responses. Some deviation is observed at hour twelve (Figure 5(B,C,E)), as observed for the previous sampling and variance based sensitivity analyses.

F. Enriched Analyses

1) Approximate Bayesian Computation (ABC): To determine posterior distributions for each of the case study parameters, an ABC approach was adopted as described in the Methods. Predicted posterior distributions for parameters *chemokineExpressionThreshold* (A), *maxProbabilityOfAdhesion* (B), and adhesionFactorExpressionSlope (C) for hour 12 are presented in Figure 6, with the remaining parameters presented in Supplementary Figure S6. For adhesion factor expression, the posterior is positively skewed, only including parameters that are less than 1.2. Conversely for maximum probability of cellular adhesion, the distribution is negatively skewed, suggesting larger values of the parameter lead to cell responses that replicate those observed in the laboratory. In both cases, the original simulator's calibrated values of 1.0 and 0.65 respectively fall within the predicted posterior distributions [10]. For chemokine expression threshold, the posterior distribution is normally distributed across the parameter value range, suggesting a high level of uncertainty in the value that should be assigned to this parameter. This supports our previously published sensitivity analyses for this parameter [10], [5], that determined a perturbation in parameter value to have no effect on simulated responses.

2) Multi-Objective Optimization (MOO): To align cell behaviors to experimental data while maximizing the emergent area of produced PP, we performed automated calibration

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2843339, IEEE/ACM Transactions on Computational Biology and Bioinformatics

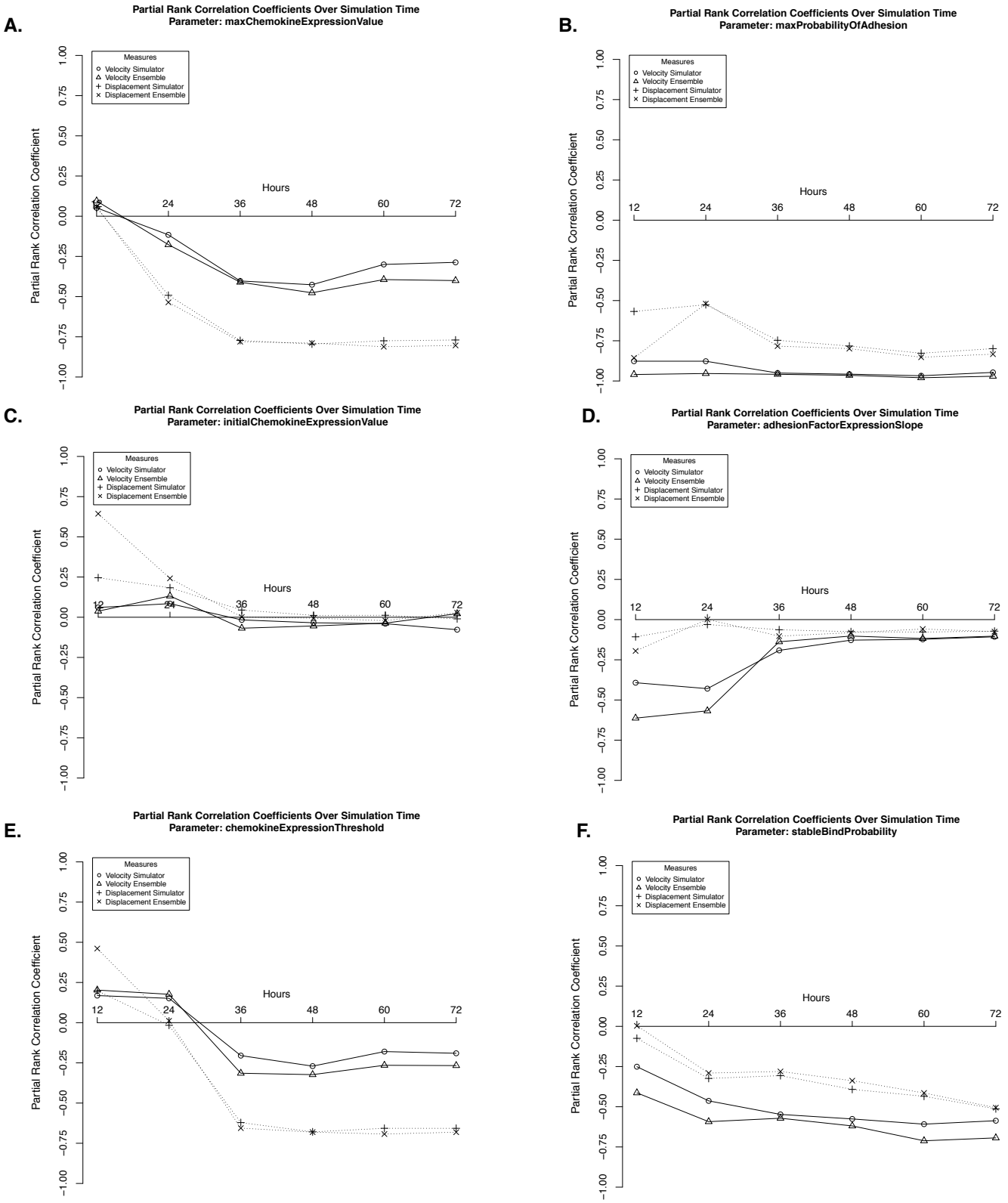


Fig. 5. Replicating a temporal sensitivity analysis of parameter influence, published in [28], using latin-hypercube sampling. Partial Rank Correlation Coefficients for each parameter and measure pair were calculated at six discrete time-points, for both the simulator and emulator

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCBB.2018.2843339, IEEE/ACM Transactions on Computational Biology and Bioinformatics

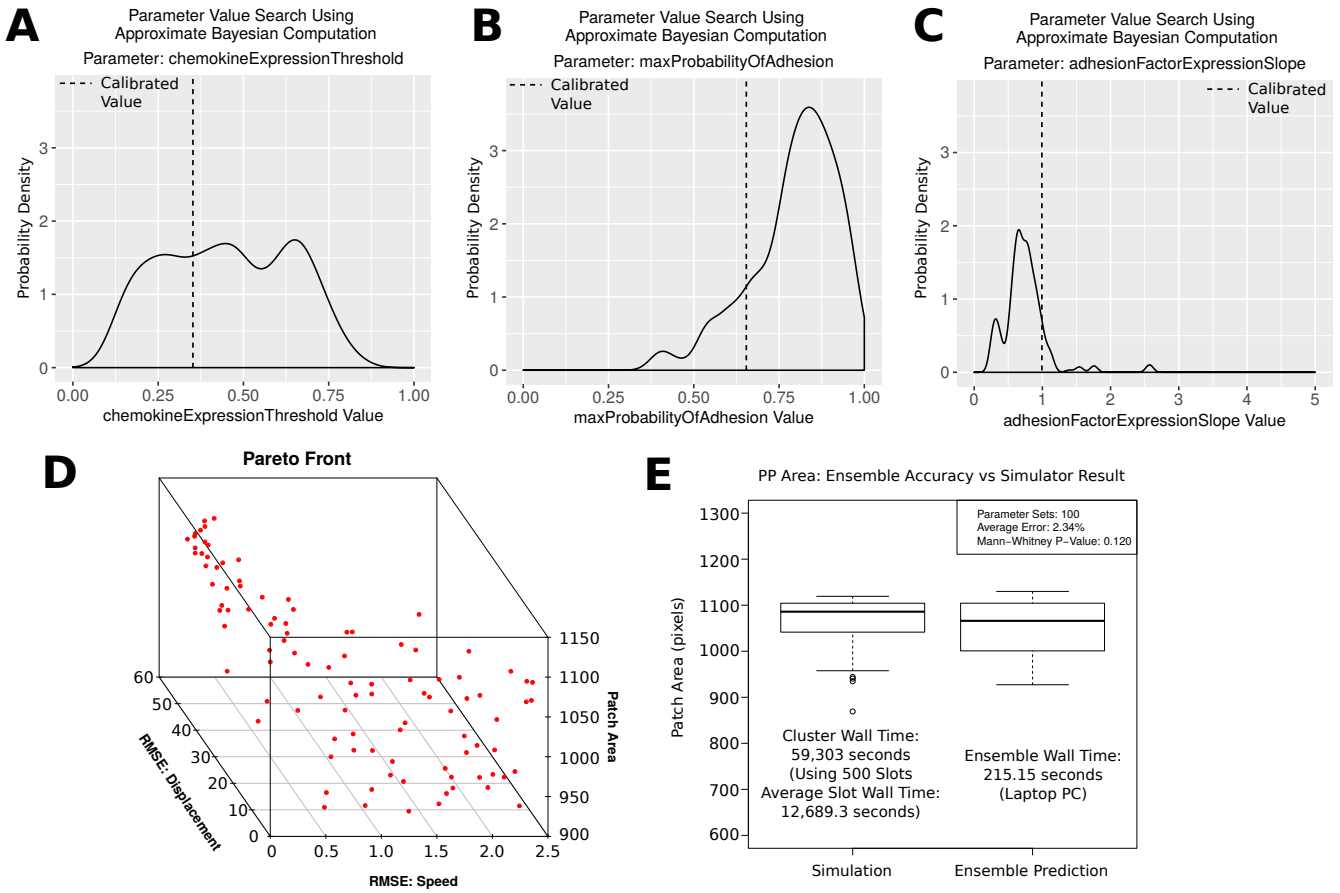


Fig. 6. Enriched Analysis. A-C: Using the emulator to perform an ABC analysis to obtain posterior distributions for parameters *chemokineExpressionThreshold*, *maxProbabilityOfAdhesion*, and adhesionFactorExpressionSlope, that align cell behaviors with laboratory measures. D: A Pareto front of solutions representing the optimal trade off in performance between cell behaviors and patch area, using NSGA-II. E: Comparison of simulator observations and emulator predictions of patch area for parameter inputs from Pareto front in D. Wall times stated are using the resources detailed in the Methods.

using the MOEA, NSGA-II. To ensure cell behaviours are preserved, the value space for each parameter was restricted to that of the posterior distribution predicted using ABC. Using NSGA-II in conjunction with our emulator we found 100 parameter sets that represent the optimal solutions evolved by the NSGA-II algorithm. Figure 6(D) is a Pareto optimal front showing the optimal trade-off between the three objectives. As patch area exceeds a value of $900\,\mu m^2$, the accuracy of the cell behavior measures decreases, suggesting $900\,\mu m^2$ is the largest patch area obtainable under baseline conditions. To verify the accuracy of those Pareto optimal solutions, parameter inputs were assessed using the simulator, with no statistically significant difference between emulator predictions and simulator observations (Figure 6(E)).

IV. Discussion

Sophisticated statistical analysis techniques are required to facilitate translation of simulation outputs into increased biological understanding. For many biomedical research applications, simulators may require significant time and computational infrastructure to evaluate. This resource requirement not only limits the use of certain statistical analysis techniques, but is also a significant obstacle in the embedding of a simulation as a key decision making platform to complement an ongoing laboratory or clinical study.

We illustrate the use of machine learning approaches to construct emulator tools that rapidly and accurately replicate previously published intensive statistical analyses of an agent-based simulator of lymphoid tissue formation. To ease wider application, we extended the functionality of *spartan* [5], [28] to permit the emulation of biological simulators. Using this extended tool and the computing resources specified in the Methods, we replicated a sampling-based sensitivity analysis that previously required 250,000 simulation executions (each execution taking at least 94 seconds) in 3.49 seconds, and a variance-based sensitivity analysis, requiring 682,500 simulation executions, in 6 seconds. Further, a temporal sensitivity analysis was reproduced that is consistent with that published previously for all simulated emergent cell behaviors [28].

Including five different machine learning algorithms permits us to contrast performance for this specific case study and demonstrate the benefits of combining these into an ensemble. It can be noted in Figures 2 and S1-S3, that in this case the neural network is the top performing algorithm, for both velocity and displacement, yielding the lowest RMSE in all cases. As such this algorithm was given the highest weighting

Simulator Performance	
Calibrated Simulator Execution Time (seconds)	94.265
Replicate executions Required Per Parameter Set to Mitigate Aleatory Uncertainty	500
Executions Required for 500 Sample LHC	250,000
Executions Required for eFAST	682,500

Emulator Performance		
	Time (seconds)	
Emulator Training Time	**12Hr**	**72Hr**
GLM	0.197	0.209
SVM	0.245	0.244
RF	0.693	0.651
NN	280.35	246.45
GP	484.357	500.704
Ensemble Generation Time	281.771	747.913
Emulated LHC Analysis	6.45	3.49
Emulated eFAST Analysis	14.67	6.00

TABLE I
PERFORMANCE STATISTICS FOR BOTH USE OF SIMULATOR, INDIVIDUAL EMULATION, AND ENSEMBLE, USING THE COMPUTING RESOURCES SPECIFIED IN SECTION C OF THE METHODS. WHEREAS ONE EXECUTION OF THE ORIGINAL SIMULATOR AT CALIBRATED VALUES MAY TAKE 94 SECONDS. BOTH SENSITIVITY ANALYSES WERE PERFORMED IN A FRACTION OF THE TIME TAKEN TO PERFORM ONE EXECUTION OF THE ORIGINAL SIMULATOR.

each time an ensemble was generated. Notable from Figures 2 and S1 is that the neural network and gaussian process models are more accurate over the entire output range for both velocity and displacement, than the general linear model, SVM, and random forest, where prediction accuracy is decreased for lower output values. Through combining the five algorithms into an ensemble, the RMSE is lower for both output measures than the neural network in isolation. The weighting of the stronger algorithms corrects those that have made poorer predictions at the lower end of the output scale, while better agreement increases the accuracy at the upper end.

We note that emulator performance in comparison with the previously published results was improved at hour 72 in comparison with hour 12. At each time-point, responses are analyzed for cells that are located within $50\,\mu m$ of a developing PP. Early in development, at hour 12, there are fewer immune cells within that vicinity than at hour 72, skewing the output distributions. A comparison of the performance at both time-points for displacement can be drawn from Figures 3,4, S1 and S3. The lower number of examples at hour 12 can impact the machine learning algorithm's ability to learn the response for the complete parameter range, in particular for the Generalised Linear Model, Random Forest, and Support Vector Machine algorithms. One of the key strengths of generating an ensemble is that the predictions obtained using a combination of weighted emulators was found to mitigate this artifact of the training dataset (Figure 2C), without the need for an increased number of training data points or adaptive sampling schemes. As PP development progresses over time, a greater number of cells fall within this range, providing a larger training data set and a wider variety of behaviors, improving accuracy of predicted cell displacement for the aforementioned algorithms. Although mitigated in this case, it remains important to be aware of how the training data characteristics may impact predictive performance.

In this application, we generated emulators for each simulation output response, for each time-point of PP development. Given the strong performance statistics in Table 1, this was sensible, as each emulator could be generated relatively efficiently while ensuring the prediction of one output was not impacted by the other. Further work could consider the accuracy of emulators that are trained to predict multiple output responses, to determine if there is a balance between the level of accuracy such an approach could achieve and the time taken to generate an emulator for each response. We also recognize the potential issue to overfit each algorithm, and provide the user with training statistics to aid assessment of the performance over both the training and test sets (Figure S4), as well as apply cross fold validation to aim to reduce that risk. It can be noted from Figure S4 that the RMSE observed in training is lower for gaussian process models than the other algorithms, which does suggest some overfitting, although the performance on the test set is comparable to the algorithm's complementary set. In addition, we also recognize there could be an interesting challenge in creating one emulator that accurately predicts cellular responses across the time period, rather than training one for each time-point. Given the insights that can be gained from temporal sensitivity analyses (Figure 5), building one emulator/ensemble rather than several may yield further performance benefits.

The generation of rapid predictions of simulator output facilitated the use of heuristic approaches that sequentially run, evaluate and adapt parameter inputs to yield a desired set of simulation outputs. For complex models such as the case study, traditional Bayesian approaches to generate likelihood distributions for each parameter become intractable, necessitating posterior prediction using approximate Bayesian computation approaches. The generated posterior distribution provides capacity to sample parameter values from a distribution that leads to a desired response, rather than fix a single value to each parameter. Such an approach could see an ensemble used in place of an original simulator in assessing what kind of variability might occur within a patient cohort, informing the statistical design of a trial, or assessing what proportion of patients may respond favorably to a therapeutic intervention. It may then be possible to infer summary population characteristics and responses via the outputs of several ensembles each representing one individual. Here our ABC analysis highlighted a high level of uncertainty in the parameter chemokine expression threshold, suggesting that the parameter is poorly constrained. The distribution of the parameter adhesion factor expression was tightly constrained across a narrow range of values while the distribution for maximum probability was positively skewed. All three results are consistent with results from previously published sensitivity analyses that suggested the influence of each parameter value at this time-point [25], [10]. In those previous analyses, only a local analysis indicated the extent to which a parameter could be perturbed before simulator behavior was significantly changed [10]. However a local analysis holds all other parameters to a fixed value,

failing to account for non-linear interactions between a parameter and it's complementary set. A posterior distribution now indicates the range over which each parameter may exist, taking all other parameters into account.

Through multi-objective optimization we obtained a population of parameter configurations that gave rise to a desired simulator output. In our previous studies we focused on calibrating the simulator such that the emergent cell behavior properties of velocity and displacement were consistent with those observed in the laboratory [25]. The ensemble provides capacity to address further interesting research questions that may not have been possible previously. Here we were interested in determining features of the parameter space that give rise to those cell behavior responses, while maximizing the area of the PP that develop. This reveals the optimal trade off between obtaining a large patch area and decreasing the accuracy of simulated cell motility. This method is useful in determining how well a simulation captures each output response, and how it may be necessary to compromise on the accuracy of some output responses to improve the accuracy of others. Aside from calibration, MOEA can be employed evaluate competing models, with the advantage that it can assess several output metrics simultaneously, identifying the optimal trade-off in performance against each [15].

Emulation can provide significant added value to simulation-focused biomedical research programmes. Through rapid identification of key mechanisms and pathways, emulators can inform experiments to quantify sensitive parameters, and identify sections of the simulator that are highly influential and may require refinement. In the presented case study presented, we examined cell behaviors in *ex vivo* culture at hour 12 [25]. If an emulation approach had been used to perform the temporal sensitivity analyses earlier, this may have directed additional experiments towards later time-points, where the analyses suggest a switch from an adhesion driven to chemokine mediated process. The application of emulation may expedite simulator development by permitting rapid prototyping and identification of errors in model design, parameterization, and software infrastructure. Testing an emulation of a simulator avoids identification of errors late in the development process that could incur significant time penalties, especially when running time-intensive statistical analyses.

V. Conclusion

Issues of time and resource limitations incurred in simulator analysis can be addressed by integrating machine learning approaches within the process of simulator development, analysis, refinement, and translation. We illustrate the exploitation of five machine learning algorithms in developing emulators that rapidly and accurately replicate intensive statistical analyses performed previously, and through generation of an ensemble permit enriched understanding of behaviors through performance of additional analysis routines. An extended software tool, spartan (https://www.york.ac.uk/ycil/software/spartan/) is provided capable of expediting the translation of simulator-derived insights into a better understanding of the design, organization, dynamics, and function of biological systems.

Acknowledgment

JC is supported by Wellcome Trust 4-year PhD programme studentship: Combating Infectious Disease: Computational Approaches in Translation Science (WT095024MA). JC and KA are part-funded by the Wellcome Trust (204829) through the Centre for Future Health at the University of York. MC is funded in part by the Medical Research Council (G0601156 and MR/K021125/1), NC3Rs (NC/K999527/1) and the Human Frontiers Science Program (RGP0006/2009-C). JT is supported by the EPSRC (EP/K040820/1).

References

[1] G. Fousteri, J. R. Chan, Y. Zheng, C. Whiting, A. Dave, D. Bresson, M. Croft, and M. von Herrath, "Virtual optimization of nasal insulin therapy predicts immunization frequency to be crucial for diabetes protection," vol. 59, no. 12, pp. 3148–3158, 2010.

[2] F. Gatto, H. Miess, A. Schulze, and J. Nielsen, "Flux balance analysis predicts essential genes in clear cell renal cell carcinoma metabolism," vol. 5, pp. 10 738 EP –, 06 2015.

[3] G. Blau and S. Orcun, "A bayesian pharmacometric approach for personalized medicine: A proof of concept study with simulated data," in *Proceedings of the 2009 Winter Simulation Conference (WSC)*, Dec 2009, pp. 1969–1976.

[4] C. Gong, J. T. Mattila, M. Miller, J. L. Flynn, J. J. Linderman, and D. Kirschner, "Predicting lymph node output efficiency using systems biology," *Journal of Theoretical Biology*, vol. 335, pp. 169 – 184, 2013.

[5] K. Alden, M. Read, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. C. Coles, "Spartan: A Comprehensive Tool for Understanding Uncertainty in Simulations of Biological Systems," *PLoS computational biology*, vol. 9, no. 2, 2013.

[6] 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*, vol. 18, no. 1, pp. 67–86, 2012.

[7] S. Marino, I. B. Hogue, C. J. Ray, and D. E. Kirschner, "A methodology for performing global uncertainty and sensitivity analysis in systems biology." *Journal of theoretical biology*, vol. 254, no. 1, pp. 178–96, sep 2008.

[8] R. N. Gutenkunst, J. J. Waterfall, F. P. Casey, K. S. Brown, C. R. Myers, and J. P. Sethna, "Universally sloppy parameter sensitivities in systems biology models," *PLOS Computational Biology*, vol. 3, no. 10, pp. 1–8, 10 2007.

[9] M. N. Read, K. Alden, L. M. Rose, and J. Timmis, "Automated multi-objective calibration of biological agent-based simulations," vol. 13, no. 122, 2016.

[10] K. Alden, J. Timmis, P. S. Andrews, H. Veiga-Fernandes, and M. C. Coles, "Pairing experimentation and computational modelling to understand the role of tissue inducer cells in the development of lymphoid organs," *Frontiers in Immunology*, vol. 3, pp. 1–20, 2012.

[11] 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," *Techometrics*, vol. 21, pp. 239–245, 1979.

[12] A. Saltelli and R. Bollardo, "An alternative way to compute Fourier amplitude sensitivity test (FAST)," *Comput. Stat. Data Anal.*, vol. 26, no. 4, pp. 445–460, 1998.

[13] K. Csillery, M. G. B. Blum, O. E. Gaggiotti, and O. Francois, "Approximate Bayesian Computation (ABC) in practice," *Trends in Ecology and Evolution*, vol. 25, no. 7, pp. 410–418, 2010.

[14] M. A. Beaumont, "Approximate Bayesian Computation in Evolution and Ecology," *Annual Review of Ecology, Evolution, and Systematics*, vol. 41, no. 1, pp. 379–406, 2011.

[15] M. N. Read, J. Bailey, J. Timmis, and T. Chtanova, "Leukocyte motility models assessed through simulation and multi-objective optimization-based model selection," *PLOS Computational Biology*, vol. 12, no. 9, pp. 1–34, 09 2016.

[16] J. Fiege, B. McCurdy, P. Potrebko, H. Champion, and A. Cull, "Pareto: A novel evolutionary optimization approach to multiobjective imrt planning," *Medical Physics*, vol. 38, no. 9, pp. 5217–5229, 2011.

[17] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001.

[18] P. S. Andrews, F. Polack, A. T. Sampson, J. Timmis, L. Scott, and M. Coles, "Simulating biology : towards understanding what the simulation shows," in *Proceedings of the 2008 Workshop on Complex Systems Modelling and Simulation*, 2008, pp. 93–123.

[19] K. Alden, P. S. Andrews, F. A. C. Polack, H. Veiga-Fernandes, J. Timmis, and M. C. Coles, "Using Argument Notation to Engineer Biological Simulations with Increased Confidence," *Journal of the Royal Society Interface*, vol. 12, no. 105, 2015.

[20] D. A. Henderson, R. J. Boys, K. J. Krishnan, C. Lawless, and D. J. Wilkinson, "Bayesian emulation and calibration of a stochastic computer model of mitochondrial dna deletions in substantia nigra neurons," *Journal of the American Statistical Association*, vol. 104, no. 485, pp. 76–87, 2009.

[21] M. Farah, P. Birrell, S. Conti, and D. D. Angelis, "Bayesian emulation and calibration of a dynamic epidemic model for a/h1n1 influenza," *Journal of the American Statistical Association*, vol. 109, no. 508, pp. 1398–1411, 2014.

[22] I. Vernon, J. Liu, M. Goldstein, J. Rowe, J. Topping, and K. Lindsey, "Bayesian uncertainty analysis for complex systems biology models: emulation, global parameter searches and evaluation of gene functions," *pre-print*, 2016.

[23] I. Inza, B. Calvo, R. Armañanzas, E. Bengoetxea, P. Larrañaga, and J. A. Lozano, *Machine Learning: An Indispensable Tool in Bioinformatics*. Totowa, NJ: Humana Press, 2010, pp. 25–48.

[24] W. A. Pruett and R. L. Hester, "The creation of surrogate models for fast estimation of complex model outcomes," *PLOS ONE*, vol. 11, no. 6, pp. 1–11, 06 2016.

[25] A. Patel, N. Harker, L. Moreira-Santos, M. Ferreira, K. Alden, J. Timmis, K. E. Foster, A. Garefalaki, P. Pachnis, P. S. Andrews, H. Enomoto, J. Milbrandt, V. Pachnis, M. C. Coles, D. Kioussis, and H. Veiga-Fernandes, "Differential RET responses orchestrate lymphoid and nervous enteric system development," *Science Signalling*, vol. 5, no. 235, 2012.

[26] K. Alden, P. S. Andrews, H. Veiga-Fernandes, J. Timmis, and M. Coles, "Utilising a simulation platform to understand the effect of domain model assumptions," *Natural Computing*, jun 2014.

[27] K. Alden, M. Read, P. S. Andrews, J. Timmis, and M. Coles, "Applying spartan to understand parameter uncertainty in simulations," *R Journal*, vol. 6, no. 2, pp. 63–80, 2014.

[28] K. Alden, J. Timmis, P. Andrews, H. Veiga-Fernandes, and M. Coles, "Extending and applying spartan to perform temporal sensitivity analyses for predicting changes in influential biological pathways in computational models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 2, pp. 431–442, March 2017.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep 1995.

[31] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 1 2006.

[32] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[33] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 161–168.

[34] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: http://CRAN.R-project.org/doc/Rnews/

[35] G. M. Dancik, *mlegp: Maximum Likelihood Estimates of Gaussian Processes*, 2013, r package version 3.1.4. [Online]. Available: https://CRAN.R-project.org/package=mlegp

[36] S. Fritsch and F. Guenther, *neuralnet: Training of Neural Networks*, 2016, r package version 1.33. [Online]. Available: https://CRAN.R-project.org/package=neuralnet

[37] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2017, r package version 1.6-8. [Online]. Available: https://CRAN.R-project.org/package=e1071

[38] O. Mersmann, *mco: Multiple Criteria Optimization Algorithms and Related Functions*, 2014, r package version 1.0-15.1. [Online]. Available: https://CRAN.R-project.org/package=mco

[39] L. J, "Plotrix: a package in the red light district of r," *R-News*, vol. 6, no. 4, pp. 8–12, 2006.

[40] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, Aug 1998.

[41] F. Jabot, T. Faure, and N. Dumoulin, "Easyabc: performing efficient approximate bayesian computation sampling schemes using r," *Methods in Ecology and Evolution*, vol. 4, no. 7, pp. 684–687, 2013.

[42] P. Del Moral, A. Doucet, and A. Jasra, "An adaptive sequential monte carlo method for approximate bayesian computation," *Statistics and Computing*, vol. 22, no. 5, pp. 1009–1020, Sep 2012.

[43] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[44] T. D. Randall, D. M. Carragher, and J. Rangel-Moreno, "Development of secondary lymphoid organs," *Annual Review Immunology*, vol. 26, pp. 627–650, 2008.

Kieran Alden is a Research Fellow in Intelligent and Adaptive Systems in the Department of Electronic Engineering at the University of York. He conducts interdisciplinary research that aims to increase confidence in predictions generated by computer models of biological systems, through the development and application of novel techniques that understand and quantify the relationship between the model and the biological system that model is designed to capture. He is a member of the IEEE.

Jason Cosgrove is a PhD student on the Combating Infectious Disease: Computational Approaches in Translational Science (CIDCATS) Doctoral Training Programme at the University of York. His interdisciplinary research focuses on combining experimental and theoretical approaches to understand how molecular, cellular and tissue level processes coordinate functional immune responses.

Mark Coles is a Professor of Immunology at the Kennedy Institute of Rheumatology, University of Oxford, and Honorary Professor of Immunology at the University of York. He conducts research on experimental and systems immunology. He has developed a program of research to understand cellular, molecular, and biophysical mechanisms regulating immune tissue development and function. To model complex dynamical immune environments, his work has focused on the development, application, and analysis of multi-scale agent-based models.

Jon Timmis is Professor of Intelligent and Adaptive Systems at the Department of Electronic Engineering, University of York. He is a previous holder of both a Royal Society-Wolfson Research Merit Award and Royal Academy of Engineering Enterprise Fellowship. His research interests are interdisciplinary in nature, and focus on the modelling and simulation of the immune system, the development of evidence-based simulations, and fault tolerance in biologically-inspired systems. He is a Senior Member of the IEEE.