A Tool to Construct Agent Based Models of BioChemical Cascades

C.W. Liew[1] and Robert Kurt[2]
[1]Department of Computer Science, Lafayette College, Easton, PA 18042
[2]Department of Biology, Lafayette College, Easton, PA 18042
liewc@lafayette.edu

Abstract

Simulation models are extremely useful but can be difficult to construct. There are numerous biochemical cascades in the human body and simulation models would help our understanding of how they work. We have developed a new approach for constructing agent based simulation models of biochemical cascades that is both powerful and general. The approach has been implemented in the **CASCADE** system and has been used to develop simulation models for 12 biochemical cascades including one (TLR4-Myd88) that has two feedback mechanisms. Early results indicate that the tool can greatly help in students understanding how to model biological systems (biochemical cascades in particular) using agent based techniques. This paper describes our approach, the use of the system and our evaluation of the generality, power and use of the system. In addition, the simulation model for the TLR4-Myd88 cascade predicted a novel result that was subsequently validated with a lab experiment.

Introduction

Simulation models can be difficult to construct, especially for those who are not computer scientists. Their utility is not in doubt but the effort and knowledge required may be beyond the scope of many biologists. We have developed a new tool for creating agent based model biochemical cascades where the cascades are represented as directed cyclic graphs. The tool (the **CASCADE** system) accepts user specifications of cascades and then constructs the corresponding computational models. The computational models are written in **Netlogo**(Wilensky and Rand (2015)) and can then be run by the user. This paper describes the tool, the process by which the model is specified and constructed and finally our evaluations of the tool by (1) a class of undergraduate biology students to test its generality and (2) our research group creating a complex model to test its functionality.

Biological systems are complex and consists of many complex interactions involving many components. Understanding these interactions is key to the development of tools, drugs, policies etc., to build a sustainable environment. The complexity of these systems makes it difficult to comprehend and determine the impact of a change in any one component in the system. In addition, because of the interdependent interactions it is difficult to conduct the "ideal" laboratory experiments of only changing one variable. One approach is to construct computer models of such systems and to study the behavior in these virtual worlds. Having a computer model allows a scientist to isolate changes and study the effects of such changes, something that frequently cannot be easily performed in the lab. Once a model has been constructed it can be used repeatedly to determine an "averaged" behavior, again something that is difficult to do in the lab in many cases. Broadly speaking two common approaches are (1) mathematical models and (2) simulation models. Mathematical models are very powerful and can be used to determine concentrations of the participating components necessary to generate desired behaviors. Biological systems are described with a system of interdependent mathematical equations and modelers can explore the effects of changing the constants for each term, in effect changing either the relative importance or concentration of the various components (*e.g.,* Klipp and Liebermeister (2006); Bray et al. (1993)). Simulation models (*e.g.,* Yang et al. (2010)) are used to explore biological systems by "putting components together" and running them to see what happens. Simulation models are less formal and require less domain knowledge to construct than mathematical models. On the other hand, simulation models are less precise and may require much more time to acquire results. They are also more commonly used to explore stochastic processes. Agent based modeling Bonabeau (2002) is a technique for developing simulation models and the technique can be used when we do not have the necessary information to develop a full set of differential equations, each equation describing the the dynamics of one constituent of the system.

Our approach has been quite successful. We have designed and implemented a tool that can create simulation models for biochemical cascades, a much more productive approach than to construct models piecemeal. This paper describes (1) how we model biochemical cascades in general, (2) the construction of the TLR4-Myd88 simulation model, and (3) the evaluation of the **CASCADE** system. The main

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

claim of this paper is a general approach to constructing agent based models of biochemical cascades. The approach has been implemented in a computer program (**CASCADE**) that has been used by groups of students and faculty to instantiate simulation models of several biochemical cascades. We have used the **CASCADE** system to successfully construct several biochemical cascades as well as to construct a complex cascade (TLR4-Myd88). The results validate our claims as to the generality and the functionality of the system. This is a significant step forward in increasing our understanding of how we might create simulation tools and systems for complex biological systems.

Modeling A BioChemical Cascade

A biochemical cascade (or a signaling pathway) is a series of chemical reactions which are initiated by a stimulus (first messenger) acting on a receptor that is transduced to the cell interior through second messengers (which amplify the initial signal) and ultimately to effector molecules, resulting in a cell response to the initial stimulus. At each step of the signaling cascade, various controlling factors are involved to regulate cellular actions, responding effectively to cues about their changing internal and external environments. Cascades have been studied extensively in the biological domain and we now have a good idea of the components involved in each cascade and the interactions that they participate in. The underlying structure and framework of these models is a directed network graph. Cascades are composed of a complex series of biochemical reactions/interactions. Ignoring spontaneous reactions, each interaction involves two or more components (proteins, ligands, phosphates etc.,). Biologically, it is difficult if not impossible to isolate a single component to determine its role and relative importance in a cascade. This is where a computer model is useful. We can use computer models to simulate the behavior of a cascade and then experiment with different concentrations of components to determine the relative importance of each and also their impact on each other.

The Modeling Process

We view the process of constructing a computational model of a biochemical cascade to be equivalent to that of building a directed (cyclic) graph (also known as a network graph). The graph represents the possible interactions where each node is either a component or the result of an interaction between two components. The result is a combination of (1) the production of a new component, and/or (2) changing the state of the participating components. Biochemical cascades usually have some form of feedback so the corresponding graph is cyclic. The process of constructing the model requires the specification of all components and interactions. We have constrained the graph by requiring that all interactions only involve two components. This does not change the functionality of the model in that all multi-component interactions (more than 2 participating components) can be converted to binary interactions by adding intermediate nodes (interactions).

The steps in constructing an agent based computational model of a signaling cascade using the **CASCADE** system are:

1. identify the components participating in the cascade,
2. for each component, specify the name, size, color, location, shape and initial population,
3. identify the interactions in the cascade,
4. for each interaction, specify the name, components, enabling interactions, disabling interactions, products, number of products,
5. check for correctness of the model,
6. generate the Netlogo implementation,

The user first starts with identifying the components that are involved in the cascade. She then specifies the visual characteristics (size, color and shape, the movement at each timestep) of each type of component. These characteristics determine how it every instance of each component is displayed in the **Netlogo** programming environment Wilensky and Rand (2015), an environment for simulating natural and social phenomena. The step after that is to identify and describe the interactions (steps) in the cascade. The interactions describe what happens (changes in state) when an instance of one component is adjacent to an instance of another component. The modeling program limits interactions to binary interactions - all multi-component interactions can be reduced to a combination of binary interactions. The simulation does not model time per se only the sequential nature of interactions so all spontanous decays/transformations are eliminated. Instead, the multiple steps are collapsed into a single step such that the transformations occur instantaneously. Each interaction specification is composed of (1) interaction name, (2) participating components, (3) any products generated (and the number of each product), (4) all preceding interactions for each component that enable it to participate in this interaction, (5) all preceding interactions for each component that prevent it from participating in this interaction. After all the components and interactions of the cascade have been specified, the system will check that the model is correct by checking the validity of the connected graph - all nodes are connected and there is only one graph. The checked model is then used to generate a **Netlogo** program that can be used to simulate the model.

Figure 1 shows a part of the TLR4-Myd88 cascade model as implemented in **Netlogo** . The simulation world is divided into four zones:

a) the zone external to the cell, *i.e.,* outside the cell membrane

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

b) on the cell membrane,

c) the cytoplasm - space between the membrane and the nucleus,

d) inside the nucleus

There is a receptor (circle) on the cell membrane and there is a ligand (oval) outside the cell. There are two types of proteins (rectangles and hexagons) in the cytoplasm. When a ligand binds with a receptor (step (1)) by coming in direct contact, it then phosphorylates a protein (step (2)). The protein can then bind with another type of protein (step (3)). This step is repeated many times with other proteins, phosphates etc., until a protein enters the nucleus (step (4)).

Each component in the simulation can only exist in one space and cannot travel between the zones. If a component moves between two zones, *e.g.,* between the cytoplasm and the nucleus, this is simulated by having two components, one that exists in the cytoplasm and the second in the nucleus. Usually a component will travel to another zone, *e.g.,* the nucleus as a result of an event. This is implemented by removing the component in the first zone and creating a new (different) component in the second zone as a consequence of the event.

The **Netlogo** system takes an agent based approach to modeling and each component is represented as an agent. At the beginning of the simulation instances of each component are created and randomly placed in the appropriate zones. At each time step, each instance moves a distance in a direction (specified in the component specification) and if two instances become adjacent to one another and if there is an interaction that includes components of these two instances, then the specified state changes will occur. The movements can include a randomized step size and direction. The behavior of the system "emerges" out of the behavior of the individual agents and not as system specification.

Constructing a Model

The **CASCADE** system takes in a specification of a cascade and generates a simulation program in **Netlogo** . The **Netlogo** system was chosen because it (1) is widely used for simulations, (2) has a visual component which makes it appealing to biologists, (3) has a large library of software to support simulations and the collection and analysis of the resulting data. A cascade specification is composed of two parts - the components that are involved in the cascade and the interactions between components.

Each component corresponds to a biological component and can represent a protein, enzyme, molecule, *etc.,* . The component is specified by (1) name, (2) initial population, (3) location, (4) visual characteristics such as color, shape and size. The visual characteristics are used to display the components interacting in a visual simulation and is mostly used for debugging purposes. Each interaction corresponds to an event that occurs when two components make contact with one another. The interaction is specified by (1) the names of the participating components, (2) the state of each component prior to the event, (3) the state of each component after the event, (4) the names and number of any components produced by the event.

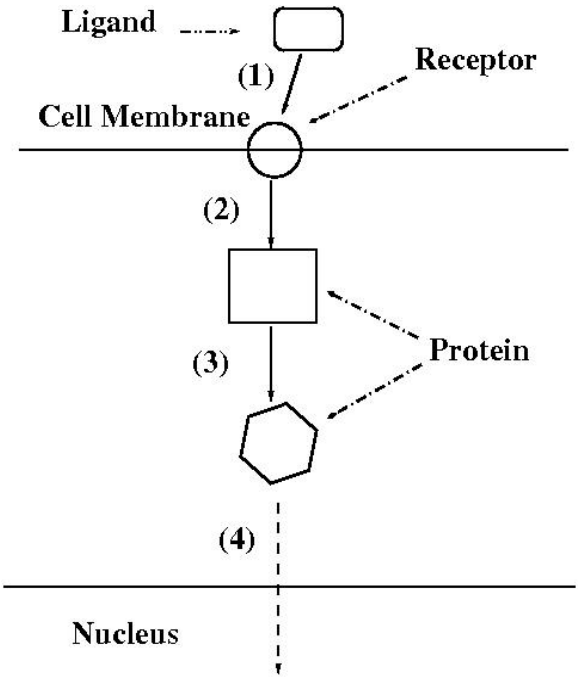

Figure 1: An Example Model Showing 4 Different Steps - Part of the TLR4 Cascade. (1) Binding ligand to protein, (2) phosphorylation, (3) binding between proteins, and (4) protein moving into the nucleus.

The virtual world is a 3D world with dimensions of $30 \times 30 \times 30$. The components are setup with a radius of 0.5 (this can be changed). The modeler has to select the shape and color that represents each component so that they can be distinguished from other components.

Modeling the TLR4-Myd88 Cascade

We used the **CASCADE** system to model the TLR4-Myd88 cascade as part of an effort to better understand the cell response to breast cancer. To study the relationship between TLR signaling and cancer we focused on the TLR signaling cascade in 4T1 murine mammary carcinoma which is often used as a model for stage IV disease in patients with breast cancer (Aslakson and Miller (1992)). The TLR4-Myd88 cascade is a signaling pathway from TLR4 through nuclear translocation of NFkB culminating in CCL2 transcription. CCL2 was of interest because studies have shown that CCL2 expression correlates with breast cancer progression (Saji et al. (2001); Lebrecht et al. (2004); Ueno et al. (2000)). Bone marrow derived CD11c+ dendritic cells (DC),

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

which elicit a prototypical TLR agonist response, were used as a control. Assessing CCL2 allowed us to compare how much CCL2 was generated by the model versus how much was produced in the lab experiments. We had significant amounts of lab data from our previous research on studying responses to Toll-like receptor (TLR) agonists. All of our experiments (both lab and simulation) focused on the levels of expression of CCL2.

Previously we reported that varying the dose, length, or frequency of LPS treatment led to different TLR signaling responses in 4T1 and DC, and that this may be related to differential expression of TLR4, CD14, Myd88 and TRAM (Palha De Sousa et al. (2010)). Subsequently, we explored the effects of reducing TLR4 and Myd88 in mammary carcinoma and found that it led to slower tumor progression, and a decrease in CCL2 and CCL5 expression (Egunsola et al. (2012)). Because of the complexity of the TLR signaling cascade and because we have also found that, depending upon the conditions, TLR signaling can either enhance or suppress tumor growth (Palha De Sousa et al. (2010)), we set out to create a computer model of the TLR4-Myd88 signaling cascade that could be used to gain a greater understanding of how this signaling cascade impacts breast cancer progression. The model was developed by incorporating qRT-PCR data of the relative mRNA concentrations from each cell type. The experimental values were then used as the starting concentration of proteins in the model. The proteins modeled in this signaling cascade were described in a review by Brown et al. (Brown et al. (2011)).

Figure 2 shows four different views of the TLR4-Myd88 model in the **Netlogo** environment. Figure **a** (top left) shows a perspective view with small populations of components. The top (yellow) represents the cell membrane with some components outside the membrane while the bottom (purple) shows the nuclear membrane. Figure b (bottom left) is a side view showing the 4 regions and the components in the regions. Figure **c** (top right) is a magnified side with larger populations of the components. Figure **d** (bottom right) is a full side view with large populations of components in the cytoplasm with small populations outside the cell and inside the nucleus.

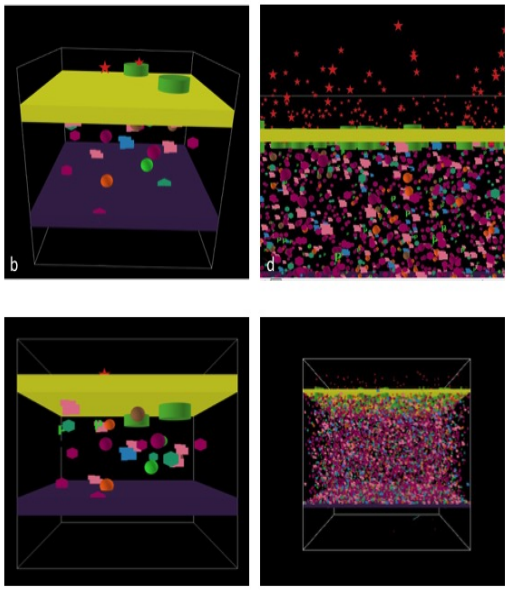

Figure 2: The TLR4-Myd88 Cascade - 4 Views: (a) perspective, (b) side with small populations, (c) magnified side, (d) full side

Experimental Results and Evaluation

The **CASCADE** system was evaluated by students in a class and by research students and faculty in our lab. In the summer of 2016 a group of students involved in research used the **CASCADE** system to create simulation models to better understand the cascades that they were studying. The students were able to (1) create correct specifications of models, (2) run the generated models, (3) compare the results with laboratory data, (4) revise the models and ultimately (5) arrive at a useful simulation model. The process was eased by having one of the **CASCADE** system developers present when the models were being created and revised. The most difficulty encountered by the students was in mapping biological terms to the terms used by the system. Once the vocabulary was understood, use of the system was straightforward. In general the students ended up with computer models that they could visualize and use and it helped their understanding of the cascades.

In the fall of 2016, 24 undergraduate students enrolled in an immunology course were divided into teams of four students. The six groups of students used the program to successfuly create six models of biochemical cascades, all of which are unique subsets of a much larger TLR cascade. Only one of the students had any programming experience and the rest of the class was limited to general applications. Their experience was similar to that of the students from the summer.

Our research group used the **CASCADE** system to study the TLR4 cascade. One of our students had initially constructed a simple model by hand (Phuong (2016)), i.e., directly working with **Netlogo** to construct a model. This approach was time consuming and many bugs were introduced, discovered and fixed. Once we had built a working model that did not incorporate feedback, we were able to do a meta-analysis of the model and the requirements of other cascades. We realized that the basic requirements were similar and that it resembled a graph. We used the knowledge to create the **CASCADE** system and then used it for all subsequent modeling work in our research and in the classroom.

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

Name	# of Components	# of Interactions	# of Feedback
AKT	12	7	0
AP1	15	11	0
CCL3	9	6	0
CXCL10	10	5	0
CXCL12	15	11	0
GPCR	15	10	0
IL8	16	10	0
JAKRAS	14	11	0
JAKSTAT	10	6	0
MAPERK	12	9	0
NFKB	18	10	0
TDAG8	8	4	0
TLR4-Myd88	17	11	2

Table 1: BioChemical Cascade Models created with the **CASCADE** system

The greatest difficulty lay in extending the system incorporate feedback (a directed graph with cycles). Incorporating feedback is important because many biochemical cascades use feedback as a regulatory mechanism. In fact within a cascade there can be multiple feedback loops, *i.e.,* multiple cycles in the graph.

Table 1 shows the models that were successfully created form cascades and the complexity of each model as judged by the number of components, interactions and feedback loops. All but one of the cascades was modeled by students who were directed to simplify the models by eliminating feedback. TLR4-Myd88 is the only cascade in the table that has feedback loops and is the most complex cascade that we modeled. We have used the simulation model to determine the impact of the different feedback mechanisms and have validated the simulation results with lab experiments. We found that although 4T1 cells had low levels of many TLR signaling components, they had better TLR signaling capabilities than professional antigen presenting cells, and that this was not solely attributed to elevated levels of NFkB. Although genes encoding most of the TLR signaling proteins were expressed at significantly greater levels by the DC, the model predicted that the 4T1 tumor cells would produce more CCL2 than DC, results that were validated with experimental data. These results will be reported in a forthcoming paper.

Cascades - Real and Virtual

This section describes some of the differences between biochemical cascades and the simulation models that we created with the **CASCADE** system and how they affect the modeling process. The major issues center around (1) the relative populations of the components and (2) time. The underlying goal in the simulation is to find behaviors of the system and components that match the results obtained in the laboratory. The behavior of individual instances of components do not have to match the behavior of the biological equivalents as long as the behavior of the whole matches. Using an agent based approach means that we do not have to specify values such as diffusion constants or reaction rates. If we know these values, it helps us to specify the characteristics of the components and converge more quickly on a system where the overall simulation behavior matches the biological behavior.

The simulation models created by the **CASCADE** system treat all components as if they moved with equal velocities and had approximately the same size. In reality the components of biological cascades have sizes that vary greatly and their velocities can differ by large amounts and thus the frequency at which events/interactions occur. Part of the initial setup is experimenting with the initial populations of the components to determine the ratios that can reproduce similar results to that of laboratory experiments. Picking a good ratio will replicate the frequency at which the interactions occur, We refer to this step as *normalizing* the model. Keeping the actual numbers low is key to controlling the length of time taken by the simulation. In our models, the population of a single component can be as low as 2 or as high as 1500. The main goal of normalizing the model is to find the parameter values such that *events in the simulation world occur at the same relative frequency as they do in the biological world* indicating the model might accurately represent the actual cascade. Once the model is normalized (this can take a long time), we can then study the importance of the various components by changing their populations and observing the effects. These experiments can then point to directions to increase or decrease the rates of desired reactions.

The specification of the components includes the movement each instance makes in each time step but there is no actual correspondence between a time step and its equivalent in the biological (real) world. One of the parameters that must be determined is how long to run the simulation so that the period of simulation matches the time period of the corresponding lab experiment. Figure 3 shows the results of simulating the TLR4-Myd88 model for 50,000 time steps with populations of DNA (in the nucleus) varying from 25 to 100. The plots from left to right are for DNA levels of 100, 75, 50 and 25 with the DNA level of 25 generating almost a flat line at the bottom of the plot. These results show that higher levels of DNA generate higher levels of CCL2 and faster. If we were to use these results, we would discard the configurations with DNA levels of 25 (the flat line at the bottom) because it appears to be too low to generate any significant response. However things change when we consider the results shown in Figure 4 where the same model is run for 500,000 time steps, *i.e.,* increasing the simulation length by a factor of 10. In this set of experiments, we kept the

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

levels of DNA constant at 25 and measured the responses (expression of CCL2) for 4T1 and for DC. The shape of the responses - DC responds faster than 4T1, over time 4T1 generates higher levels of CCL2 - best matches the results from our lab experiments. The experiments for DNA levels of 50, 75 and 100 when extended to 500,000 time steps resulted in greater CCL2 expression for DC.

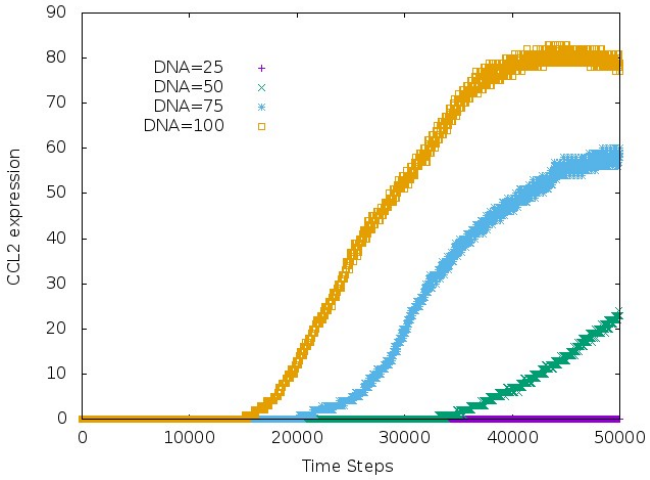

Figure 3: Simulation of TLR4-Myd88 cascade for 50,000 time steps - 4 simulations with different populations of DNA (orange:100, light blue:75, red: 50, dark blue:25) showing the expression of CCL2 over time for 4T1

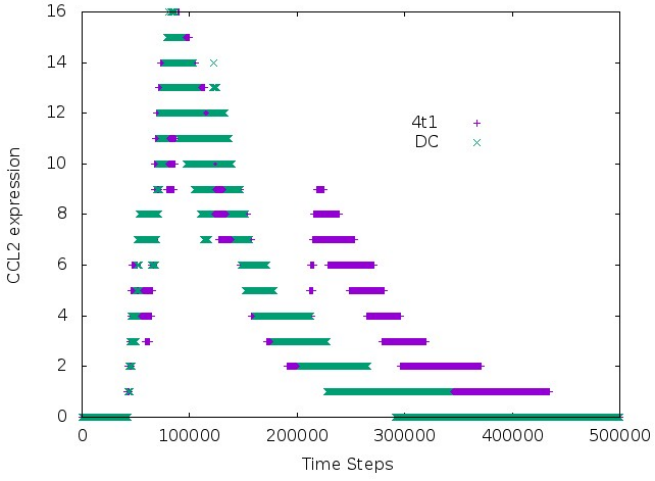

Figure 4: Simulation of TLR4-Myd88 cascade for 500,000 time steps - 2 simulations showing the expression of CCL2 for 4T1 (blue) and DC (red)

Comparing and validating the results from laboratory and computational experiments is a non-trivial task. Figures 5 amd 6 show the results from the lab and simulation experiments for the TLR4-Myd88 cascade. It is not immediately obvious that the two match or are even similar. The elapsed time interval is different and it would be difficult to match up the ranges and scales. The lab experiment terminates after 72 hours and it would be extremely difficult to obtain data beyond that point due to interactions with other parts of the cell and the environment. However this is the best match that we could find and it matches because of the following criteria - (1) CCL2 is expressed earlier in DC, (2) CCL2 expression in DC and 4T1 increases and then decreases, and (3) CCL2 expression in 4T1 is higher. There may be better combinations of values (initial populations) that would create a better match but we have not found them.

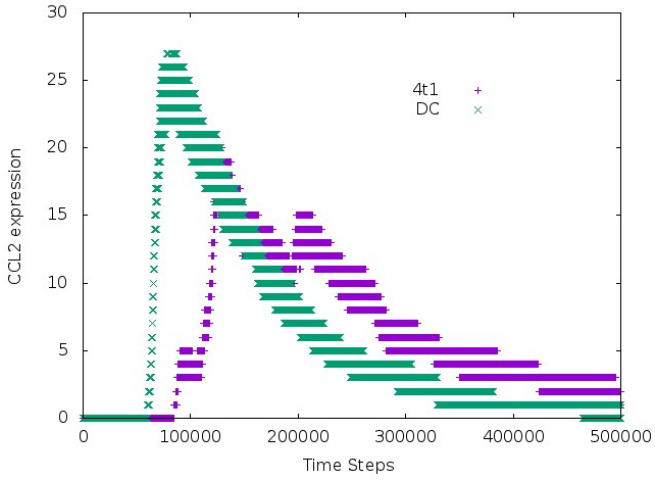

Figure 5: Results from computational experiments for TLR4-Myd88 (500,000 steps) showing levels of CCL2 expression - 4T1: blue, DC: red

There are some other adjustments that biologists have to make when using the **CASCADE** system. The two main items are (1) the restriction of each component to a single space, and (2) the limitations of only having binary interactions. Both of these issues do not limit the generated simulation model but do require that the modeler adjust their model. Firstly, for each component that travels between spaces the modeler must create a different equivalent component for each space. As described earlier, the movement between spaces is of interest only if the component is part of interactions in both spaces. In that case we modify the interactions such that when the relevant event takes place, the component in the originating space is removed while a corresponding component in the destination space is created

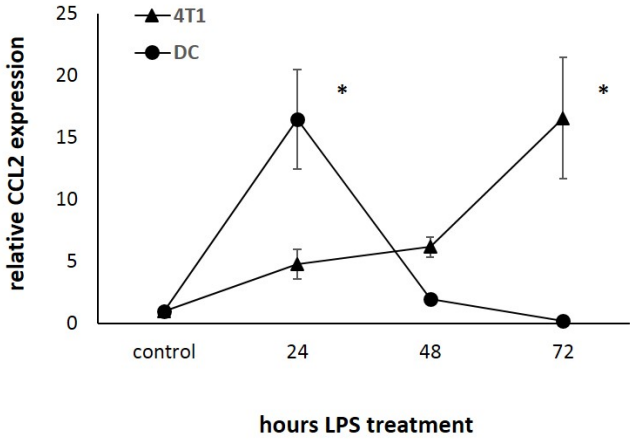

Figure 6: Results from laboratory experiments for TLR4-Myd88 (72 hours) showing levels of CCL2 expression

People	Role	# Models	Time
5	Student Researcher	5	5 weeks
6 groups of 4	Student in class	6	8 weeks
2	Faculty Researcher	1	4 months

Table 2: Productivity Data for using the **CASCADE** system.

to participate in subsequent interactions in the new space. Similarly multiple component interactions can be converted into an equivalent series of binary interactions. The modeler has to create new interactions such that the series of interactions is equivalent to the original multi-component interaction. She may also have to create and use additional components to participate in these interactions.

Productivity

An evaluation of the **CASCADE** system would be incomplete without some discussion about the productivity gains from using the tool since one claim is that the tool will make it easier and faster to create models. The data that we present here is by no means to be considered representative since we do not have enough points. But it does give us some hint that the **CASCADE** system can be very useful and helpful for some faculty and students. The data is shown in Table 2.

All the students involved in this study (except for one) had no prior experience with either computer programming or modeling. The one thesis student who designed and implemented a simulation model of the TLR4-Myd88 cascade (the most complex model we have) from scratch had minimal programming experience and took approximately 15 months to complete the model. In contrast by learning from her experiences, we designed and implemented the **CASCADE** system in 4 months. We then used the system to generate a model for the TLR4-Myd88 and it took us another 4 months to settle upon a valid model and 90% of that time was spent in finding a valid set of parameter values that would generate data comparable to that of the lab experiments. The other students (who had less programming experience) were able to create models in a short period of time.

Future Work

The **CASCADE** system is by no means complete. The current system interfaces with the modeler through console i/o and we are in the process of developing a web interface that will allow users to create models through a browser. The resulting model can then be downloaded and run on their computer. The system is also being enhanced with analytical models that will check the models for general correctness to help users develop correct model specifications. Examples of common errors include missing interactions and components, mis-spelled components and interactions.

We are enhancing the tool to help modelers create new larger models by combining existing models like those that were developed in the immunology course. Finding a set of initial populations of components that produce simulation results matching those of the lab experiments is difficult and time consuming. The complexity and time required grows exponentially as the number of interactions grows. Therefore as the models grown larger, it will become infeasible for a modeler to find valid parameter values by hand. Our solution is to add an optimizer that will search the parameter space to find a valid set of values. The optimizer is based on one that we have previously used to find solutions to other biological modeling problems (Liew (2004); Liew et al. (2007); Long Jr. et al. (2010)) and uses a genetic algorithm to effectively and efficiently search high dimensional spaces with a gradient descent approach. The extensions will (1) create a new larger model from the constituent model, and (2) find the best set of parameter values that match the known lab results. These two extensions should greatly help modelers to create more complex models.

Conclusion

This paper has described our general technique for creating models of biochemical cascades using an agent based modeling approach. The approach has been implemented in the **CASCADE** system and successfully used by students and faculty to create models of twelve biochemical cascades. The **CASCADE** system uses a cascade description to generate a simulation model in **Netlogo**. Our initial experience with use of the tool by both students and faculty are very encouraging and we believe that this tool could accelerate our understanding of biochemical cascades. The **CASCADE** system represents a fundamental improvement in how we create simulation models since it can create instances of a family of cascades and does not require programming knowledge on the part of the modeler. In addition

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).

we used the **CASCADE** system to develop a model of the TLR4-Myd88 cascade and it predicted an unexpected outcome where 4T1 tumor cells would express more CCL2 than DC. These results were validated in a subsequent lab experiment and shows the power of having a simulation model to explore the space of interactions between the components.

Acknowledgements

We are grateful for the contributions of the students in our research lab particularly Tiffany Phuong and Shira Wein and the students enrolled in the Immunology class. This research was funded in part by the Lafayette Excel scholars program and by a grant from the Howard Hughes Medical Institute (HHMI).

References

Aslakson, C. J. and Miller, F. R. (1992). Selective events in the metastatic process defined by analysis of the sequential dissemination of subpopulations of a mouse mammary tumor. *Cancer research*, 52(6):1399–1405.

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 99 (suppl 3), pages 7280–7.

Bray, D., Bourret, R. B., , and Simon, M. I. (1993). Computer simulation of the phosphorylation cascade controlling bacterial chemotaxis. *Molecular Biology of the Cell*, 4(5):469–482.

Brown, J., Wang, H., Hajishengallis, G. N., and Martin, M. (2011). Tlr-signaling networks: an integration of adaptor molecules, kinases, and cross-talk. *Journal of Dental Research*, 90(4):417–427.

Egunsola, A. T., Zawislak, C. L., Akuffo, A. A., Chalmers, S. A., Ewer, J. C., Vail, C., Lombardo, J. C., Perez, D. N., and Kurt, R. A. (2012). Growth, metastasis, and expression of ccl2 and ccl5 by murine mammary carcinomas are dependent upon myd88. *Cellular Immunology*, 272(2):220–229.

Klipp, E. and Liebermeister, W. (2006). Mathematical modeling of intracellular signaling pathways. *BMC Neuroscience*, 7((Suppl 1)):S10.

Lebrecht, A., Grimm, C., Lantzsch, T., Ludwig, E., Hefler, L., Ulbrich, E., and Koelbl, H. (2004). Monocyte chemoattractant protein-1 serum levels in patients with breast cancer. *Tumour Biology*, 25(1-2):14–7.

Liew, C. (2004). Using a genetic algorithm to optimize the gape of a snake jaw. In *Proceedings of Twenty Fourth ACM Symposium on Applied Computing*, pages 975–979.

Liew, C. W., Root, R. G., Long Jr., J. H., and Koob, T. J. (2007). Using artificial organisms to study evolution. In *Proceedings of the 1st IEEE Symposium on Artificial Life*, pages 108–114.

Long Jr., J. H., Porter, M. E., Root, R. G., and Liew, C. W. (2010). Go reconfigure: how fish change shape as they swim and evolve. *Integrative and Comparative Biology*, 50:1120–1139.

Palha De Sousa, C., Blum, C. M., Sgroe, E. P., Crespo, A. M., and Kurt, R. A. (2010). Murine mammary carcinoma cells and cd11c+ dendritic cells elicit distinct responses to lipopolysaccharide and exhibit differential expression of genes required for tlr4 signaling. *Cellular Immunology*, 266(1):67–75.

Phuong, T. (2016). Modeling tlr signaling in murine mammary carcinoma with negative feedback. Honors Thesis submitted to the Department of Biology, Lafayette College.

Saji, H., Koike, M., Yamori, T., Saji, S., Seiki, M., Matsushima, K., and Toi, M. (2001). Significant correlation of monocyte chemoattractant protein-1 expression with neovascularization and progression of breast carcinoma. *Cancer*, 92(5):1085–1091.

Ueno, T., Toi, M., Saji, H., Muta, M., Bando, H., Kuroi, K., Koike, M., Inadera, H., and Matsushima, K. (2000). Significance of macrophage chemoattractant protein-1 in macrophage recruitment, angiogenesis, and survival in human breast cancer. *Clinical Cancer Research*, 6(8):3282–3289.

Wilensky, U. and Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with Netlogo*. MIT Press.

Yang, J., Meng, X., , and Hlavacek, W. S. (2010). Rule-based modeling and simulation of biochemical systems with molecular finite automata. *IET Systems Biology*, 4(6):453–466.

Carole Knibbe et al, eds., Proceedings of the ECAL 2017, Lyon, France, 4-8 September 2017,
(Cambridge, MA: The MIT Press, ©2017 Massachusetts Institute of Technology).