An Agent-Based Modeling and Simulation Environment for Dynamic Biological Systems

Steven Phung[1], Rajdeep Singh[2], Jamie Lawson[2], Michael Hultner[2], Briam Peck[2], Ross Henderson[3], Dennis Hsu[2], Desmond Lun[4,5], Vijayaraj Nagarajan[3], Mariam Quiñones[3], Darrell Hurt[3], Yentram Huyen[3], Mike Tartakovsky[3]

[1]Autodesk, Inc., The Landmark @One Market, San Francisco, California 94105

[2]Lockheed Martin, Information Systems and Global Solutions, 4770 Eastgate Mall, San Diego, California 92121, USA

[3]Office of Cyber Infrastructure and Computational Biology, National Institute of Allergy and Infectious Diseases, NIH Bethesda, Maryland 20817, USA

[4]Center for Computational and Integrative Biology and Department of Computer Science, Rutgers University, Camden, NJ 08102, USA

[5]Phenomics and Bioinformatics Research Centre and School of Mathematics and Statistics, University of South Australia, Mawson Lakes, SA 5095, Australia

ABSTRACT

Motivation: Understanding emergent behaviors of complex biological systems requires modeling and simulation of large and detailed models. Models must be both expressive and scalable to capture the size and complexity of molecular and cellular networks.

Results:

In this report we present GRANITE (Genetic Regulatory Analysis of Networks Investigational Tools Environment), an agent-based modeling (ABM) and multi-agent simulation (MAS) approach to modeling large, complex, and dynamic systems. We have demonstrated the GRANITE capability on metabolic networks: specifically the mycolic acid biosynthesis pathway of the *Mycobacterium tuberculosis*. The agent-based model has been compared to Flux Balance Analysis (FBA) and shown to be able to emulate the internal and external properties of the system as modeled by FBA. We show that the approach is scalable and computationally efficient to allow researcher interaction with a dynamically evolving simulation. The GRANITE tool enables the researcher to propose and test systems-level hypotheses and make predictions for laboratory experiments to validate or refute these hypotheses.

Availability and Implementation:

The GRANITE software is open-source and available from the corresponding author, Ross Henderson. Please indicate GRANITE in the subject line of correspondence.

Contact: rh@nih.gov *

1 INTRODUCTION

Living systems are complex systems. As such, they have emergent behaviors: input-response properties that can be observed but not predicted by first order knowledge of the functions of the system's components. Systems biology is an approach to understand the general principles of living systems by elucidating the relationships between the components of a system and its emergent behaviors.

Only through understanding living things as systems can one hope to understand the mechanisms of cellular and molecular biology. These systems are formed from the many interactions between molecules within the cell and between cells. Examples include metabolic networks, signal transduction networks, gene regulatory networks, and other epigenetic networks. The interplay between these systems creates another level of complexity that makes the modeling and simulation of living systems a serious computational challenge.

Much of the focus of effort in systems biology involves the development of models for biological function at the systems level. To be useful these models must be expressive, computationally tractable, and should yield predictions that can be tested with laboratory experiments. Our initial gap analysis indicated the need for an interactive M&S (Modeling & Simulation) tool that allows for real-time interaction with the simulation. Since Cytoscape (http://www.cytoscape.org/) has limited ability to allow real-time dynamic interaction, we identified two other tools that study dynamics of biological networks and evaluate perturbation hypotheses. FERN (Erhard, *et al*, 2008) allows visualization of the dynamics but it does not allow for real-time interaction with the simulation. Even then, our attempts to integrate GRANITE with FERN proved cumbersome due to limitations of the available interfaces. Perturbation Analyzer tool (Fei Li, *et al*, 2009) was developed to investigate specifically the effects of single or combinatorial concentration perturbations by comparing two different steady states using law of mass action (LMA) in real-time and uses Cytoscape

*To whom correspondence should be addressed.

for visualization. In this report we present a modeling and simulation approach, GRANITE, that is expressive enough to capture any kind of interaction network, can modularly use any kinetic model, is computationally tractable and scalable, and allows researchers to interact and dynamically perturb the system at different hierarchical levels to learn its rules for emergent behavior.

2 METHODS

The Genetic Regulatory Analysis of Networks Investigational Tools Environment (GRANITE) software consists of:

- A simulation environment where software agents can be organized into dynamic models,
- A domain specific language (DSL) for expressing biological function, and
- A graphical user interface (GUI) for dynamic interaction with the simulation.

The agent based modeling and simulation components, and the DSL are implemented in Scala and the GUI is implemented in Java. The GRANITE software is available upon request from the corresponding author.

2.1 Agent-based Modeling (ABM) and Multi-agent Simulation (MAS)

The evolution of assemblies of biological components is often modeled as a system of ordinary differential equations (ODEs) that can be solved using numerical methods. Alternatively, the ABM approach (Eric, 2002) creates an assembly of computational components (agents) that would be governed by the same system of ODEs, but instead of explicitly solving the ODEs using classical numerical analysis, we simply allow the computational components to evolve directly in a MAS environment. This in effect solves the ODEs approximately in a distributed manner. The process of creating and running a system model for an experiment in the GRANITE context is as follows: A metabolic network model for the mycolic acid biosynthesis pathway (MAP) is instantiated from an SBML (Systems Biology Markup Language) model (Raman, *et al*, 2005). A set of reaction agents and their associated metabolites are created by parsing the model, instantiating the agents, the environments, and populating the environment with metabolites. Simple Michaelis-Menten kinetics are used to model the agent reaction kinetics; GRANITE facilitates the use of other kinetic models by providing a generic agent-environment interaction interface. Similarly, the non-agent entities (e.g. metabolites and enzymes) are added to the environment and given initial conditions. The agents are then placed in the simulation framework with a set of parameters. A simulation scheduler strategy (deterministic or stochastic) is chosen and the progress of the simulation is measured in interaction time. For non-interactive simulations the simulation is allowed to evolve until it reaches a steady state. For interactive simulations the simulation evolves under control of the researcher via the Glimpse-GRANITE GUI.

2.2 Scalability

Agents interact with one another only indirectly using the environment as a mediator. This decoupled approach leads to modularity and scalability. The approach is modular because it uses agents to encapsulate biological function, and scalable because it avoids combinatorial interactions and therefore results in an efficient simulation. The performance of the GRANITE system has been benchmarked using the MAP network. We have shown that the computation scales linearly with the number of reaction agents. In the MAS framework, we employed a scheduling capability that provides control of the computational demands by modulating the simulation fidelity. A GRANITE simulation is configured to employ a deterministic or a stochastic scheduler. The deterministic scheduler evolves the simulation using all agent-environment interactions at all times based on the kinetic models of the agents; i.e., their strategies for turning reactants into products. However, an agent's interaction with the environment may not always lead to significant changes in the environment. For example, at very low substrate concentrations, the continuity assumption for the rate law does not hold and the reaction may not be moving forward at all times. Stochastic scheduling exploits this constraint and allows agents (reactions) to interact only if their interaction is significant; see Fig. 1. The uncoupled agents and the scheduling of their interactions with the environment produce simulations that scale linearly with the agent population size.

Let $\boldsymbol{c}$ be the continuity threshold and let the maximum saturation rate of a reaction agent, 'i', be given $\boldsymbol{M_i}$. Let the relative rate of the reaction, $\boldsymbol{r}$, at any time $\mathbf{t}$, $\mathbf{r_{i,t} = v_{i,t}/M_i}$. The mixed strategy used by the agent' 'i' for deciding on whether to interact or not at any given time is as follows:

- If $\boldsymbol{r_{i,t} > c}$, the agent can interact with the environment at time $\boldsymbol{t}$. Let the set of all agents in this category be denoted as $\boldsymbol{A_I}$.
- Else, let $\boldsymbol{r_{i,t} = r_{i,t} / \sum r_{i,t}}$ define a normalized distribution, $\underline{\boldsymbol{r}}$. We then choose a user defined percentage of the agents from the set $\boldsymbol{A_I}$ using roulette wheel selection based on $\underline{\boldsymbol{r}}$.

An important measure of scalability of MAS is the time it takes to evolve to some steady state: the settling time. Factors contributing to settling time include the number of agents participating in the simulation and the fidelity of the underlying kinetic model of each agent (fidelity impacts how closely the computational components can evolve to the solution prescribed by the ODEs).

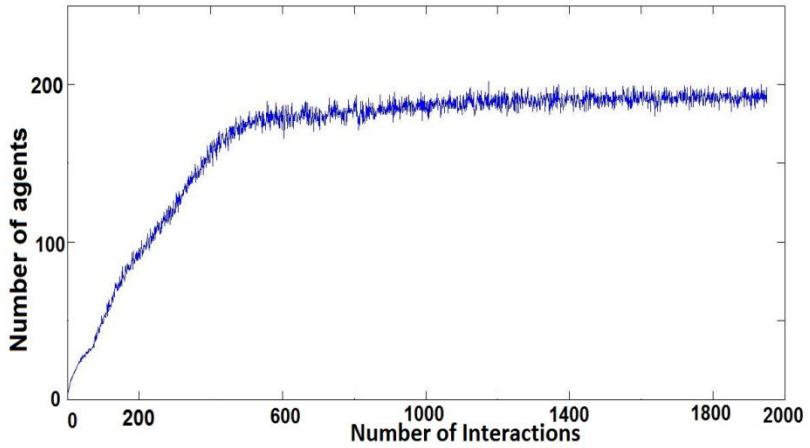

Fig. 1. Agent Population Size vs. Simulation Time. Stochastic Scheduler needs on average 75% agent-environment interactions (166 agents interact on average) and at most 190 agents, 87%, interact at any given time compared to Deterministic Scheduler (all 219 agents interact all the time).

The simulation framework manages this complexity by determining, at each time step, a trust region for each agent in which the agent can make a reliable contribution to environment evolution. The validity of this trust region is determined by the interactions of all the agents with the environment, driven by the scheduler. Figure 2 shows results from empirical experiments, demonstrating that a relatively coarse model of the trust region is sufficient to avoid very large settling times. We use the steady state flux to compare GRANITE to FBA where the GRANITE flux plots are scaled to compare with FBA on a gene by gene basis. The scaling approach we used is very straightforward and intuitive. We chose to group all reactions associated with each gene '*i*' (Raman et al, 2005). Let f_j be the FBA flux and g_j be the GRANITE flux for reaction *j*, and S_i be the set of reactions influenced by gene '*i*'. The affine scaling for all reactions in S_i is then computed as follows:

$$k = \frac{\max_{j \in S_i}[f_j] - \min_{j \in S_i}[f_j]}{\max_{j \in S_i}[g_j] - \min_{j \in S_i}[g_j]}$$

$$o = \max_{j \in S_i}[f_j] - k * \max_{j \in S_i}[g_j]$$

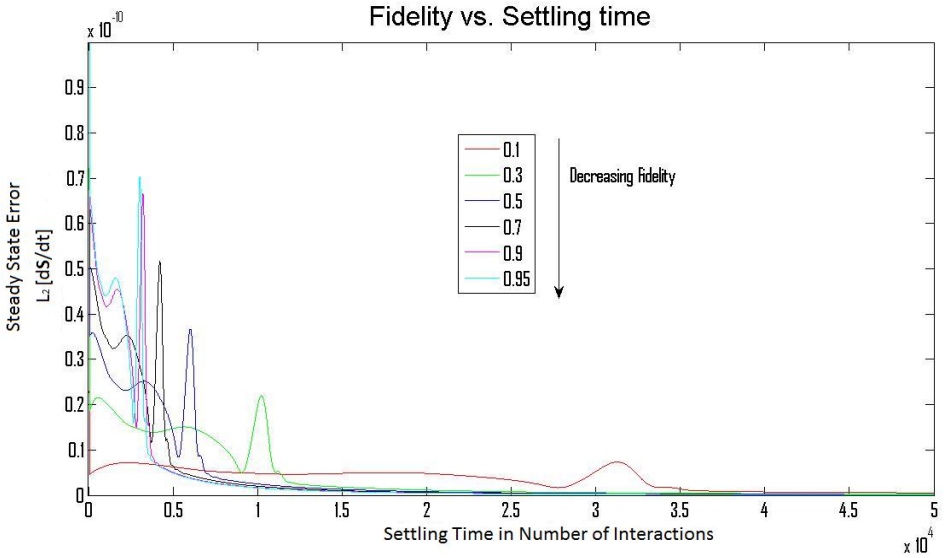

Fig. 2. Fidelity vs. Settling Time. Higher fidelity, trust parameter =0.1, moves the simulation slower to the steady state but the interactions are more accurate (the first order linear approximator defines the trust region using more support points in the same interval). Coarsest fidelity, trust parameter =0.99, corresponds to only 2 support points (the end points of the closed interval) and advances the simulation faster although with less accuracy.

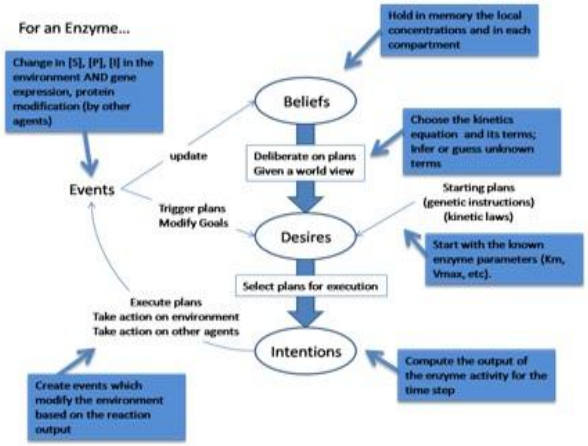

Fig. 3. The Belief-Desires-Intentions strategy is used to control the behavior of agents in a multi-agent simulation. For an agent that represents an enzymatic reaction, the Beliefs are the inputs to the reaction available from the environment and the entity properties; Desires are specified in one or more kinetic models for the reaction, and Intentions are the actions made by the agent onto the environment at each update step.

All of the GRANITE reaction fluxes associated with a gene are then scaled with the scale found for that gene. The correlation between the flux profiles improves significantly with this scaling. Note that one can apply a feedback loop by incorporating these scales into the catalyst concentration values to drive the GRANITE simulation.

3 RESULTS AND DISCUSSION

In this report we present a software framework for ABM of biological entities and a MAS environment for simulation of biological systems. This framework provides a means to create complex models of molecular networks that can evolve in an interactive simulation environment.

3.1 Agency

We employed classic ABM (Axelrod, 1997) to express units of biological function. Using the Belief-Desires-Intentions (BDI) model (Rao, 1995; and Weiss, 2000), as shown in Fig. 3, we created a framework for expressing biological agents that can be composed into complex systems. We discovered that this pattern works very well when agents represent biological function and specific entities. For example, an enzyme is represented by an agent that models its enzymatic reaction. Beliefs in the BDI model represent the world-view of the agent: the inputs to the agent from the environment, such as the state of mutable properties of substrates, enzymes, inhibitors, and other effectors. Desires represent the agent's goals such as the conversion of substrates to products, governed by stoichiometry and kinetic models for a reaction. Intentions are the actual steps an agent takes to affect its desires on the environment; the rate model for a reaction, for instance. The environment is then an arena where different agents compete through their intentions to achieve their desires.

This approach to modeling units of biological function is both expressive and modular. There are no limits placed on the techniques for expressing a functional response to environmental conditions. Thus, alternative assumptions and models can be incorporated into the simulation and tested.

3.2 Simulation

We employed a multi-agent simulation with scheduling strategies to create a computationally tractable and scalable modeling and simulation capability. Agents compete with one another to achieve their goals in one or more environments. The simulation framework's job is therefore to manage the changes to the environment(s) resulting from agent activities scheduled in the system.

3.3 Domain Specific Language (DSL) for Dynamic Biological Systems

The feature that ties modeling and simulation together is a novel Domain Specific Language that enables the systems biologist to express agents and simulation context in a simple and concise form that they can relate to. Where SBML can express state, GRANITE DSL can express state, coordination, and activity. As such, the DSL can describe all of the dynamics of the system, i.e. the system's overall behavior with respect to time. The DSL is an extension of the popular Scala programming language, which is designed for domain specific extensions, and benefits from all of the tools and documentation developed in the Scala community. In addition to GRANITE's ability to use SBML models as inputs, the DSL facilitates creation of biological models in a more natural yet formal way which is biologist friendly. Consider units of measure as an example. Above, we stated that the GRANITE user can supply their own kinetic models. In fact, different reaction agents may employ different kinetic models as appropriate for the reaction. In order to maintain consistency among the different kinetic models, their units must be compatible. This is a hard bookkeeping problem, made harder when different models are developed by different people in different organizations. The GRANITE DSL provides "guardrails" for the user by supplying syntax for defining the units

of measure for the rate constants, or for any other values. The GRANITE system also supplies implicit conversions so that if one model assumes concentrations in moles/liter and another model assumes millimoles/liter, the GRANITE system will automatically make the appropriate conversions. When incompatible or unknown units are combined, GRANITE alerts the user rather than producing meaningless results. Adding two values in units of molarity produces an error because concentrations are not addable. Corresponding volumes are needed for that operation to make sense, and so the GRANITE DSL prevents it.

We discuss some simple steps to illustrate the use of DSL in the context of a metabolic network. The first step defines a meme called "a". Memes are first class modeling objects that have mutable and immutable properties. An immutable property, like molecular weight, always has the same value. A mutable property, like concentration, may vary at different times and in different environments.

```
val a = Species called "a" build
```

The second step defines a simulation; interaction models that will be bound to an environment using a simulation context are created. In the example below, the interaction model is a metabolic network containing two reactions. Reaction ***r1*** produces ***b*** and consumes ***a***, whereas reaction ***r2*** produces ***c*** and consumes ***b*** using given stoichiometric coefficients, kinetic laws, rate parameters, and a deterministic scheduler (in this example) to decouple and synchronize the agent interactions.

```
def metabolicNetwork = CreateMetabolicNetwork of (
   Reaction called "r1" of (1*a) -> (1*b)
         using (MichaelisMenten withSpecificityConstants(a->0.1)
               catalyzedBy(p) withCatalyticConstant(0.1)),
   Reaction called "r2" of (1*b) -> (1*c)
         using (MichaelisMenten withSpecificityConstants(b->0.1)
               catalyzedBy(p) withCatalyticConstant(0.1))
) scheduledBy DeterministicMetabolicModelScheduler(0.01)
```

The specificity constant and the catalyst constant are typically denoted in the Michaelis-Menten kinetics as K_m and K_o respectively, and may be referenced as such in the DSL.

The third step creates the simulation contexts; this involves the creation of environments and the assignment of interaction models affecting those environments. The environment is defined using a *containing* clause which specifies memes and associated properties. Specifying which interaction models to use is accomplished by a *using* clause. Below is an example of defining a simulation context where memes a, b, c, and p are associated with concentration properties which use a metabolic network interaction model.

```
val sc1 = SimulationContext called "sc1" containing (
   a where ConcentrationIs(1000.0),
   b where ConcentrationIs(0.0),
   c where ConcentrationIs(0.0),
   p where ConcentrationIs(1.0)
) using metabolicNetwork
```

Finally, a simulation is constructed by defining which simulation contexts are part of the simulation. Below is an example of defining a simulation. The conciseness reflects the power of the DSL.

```
Simulation of sc1
```

3.4 Validation

As the use of agents is a departure from traditional methods of modeling biological systems, we performed a set of experiments designed to validate the approach. Our basis for validation criteria was the ability to emulate results from established systems, as well as from accepted modeling or simulation methods. For this study we chose to model a metabolic network, the mycolic acid biosynthesis pathway of the *Mycobacterium tuberculosis* which involves 197 metabolites, 219 reactions, and 28 enzymes driving these reactions. The pathway has been defined (Barry, 1998) and models exist in the SBML format (Raman, *et al.*, 2005). Furthermore, systems-level analyses exist in the literature that provide metrics about the internal states of the system against which we can compare the states of the agents and the environment. We chose to compare the ABM-MAS results to a Flux Balance Analysis of the mycolic acid pathway using the same SBML model of MAP as used by Raman *et al.* Instantiating the MAP model into a set of reaction agents with the same stoichiometric parameters, we attempted to emulate the internal and external states of the metabolic pathway at steady-state using Michaelis-Menten kinetics. We examined the ability of the ABM-MAS system to emulate the output of the pathway in terms of the observed proportions of mycolic acids and the flux profiles of the reactions in the network. Initial results showed that we could either emulate the mycolate ratios or the flux profile (Table 1). Using group scaling based on gene-reaction associations, and an optimized set of parameters, the ABM-MAS system was able to reproduce both the observed mycolic acid proportions and the reported flux profiles (Fig. 4). The method for optimizing and discovering the system parameters involves a novel use of genetic algorithms (Lawson, Singh, *et al.*) that will be published separately.

3.5 Dynamic Systems

Agent systems are particularly useful in modeling dynamic systems in a manner that allows the modeler to directly interact with the evolving system. The modeler can make changes to the system and immediately observe the response in real-time. We assert this is a novel technique for proposing and testing hypotheses at the systems level of molecular biology.

Table 1. A comparison of the observed and simulated mycolate ratios.

	methoxy-mycolate to alpha-mycolate	keto-mycolate to alpha-mycolate	trans to cis forms of methoxy-mycolate and keto-mycolate
Observed	0.54	0.49	0.14
Randomly Chosen Parameters	0.49	0.47	1.0
Manually Chosen Parameters	0.36	0.28	0.15
Learned Parameters	0.54	0.49	0.14

The first row presents the published output of the myolic acid pathway (Watanabe, 2001). Initial experiments with randomly chosen parameters were able to approximate the ratios except for the cis:trans bias (row 2). Altering the specific activity levels of the methylases MmaA1 and MmaA4 did improve the cis:trans bias but reduced the fidelity of the other mycolates (row3). Learned parameters using a genetic algorithm approach (Lawson, Singh, *et al.*) were ultimately used to create a model that produced the desired mycolate ratios (row 4).

The Glimpse-GRANITE tool was developed to provide that capability to systems biologists. This GUI, see Fig. 5, provides a command-and-control interface to the simulation that includes the ability to create an agent system, start a simulation, observe the internals and externals of the simulation environment, and configure all aspects of the system and the simulation. The temporal aspect of the simulation is measured in number of interactions within the system. Since the agents are uncoupled from the simulation interactions, a change to an agent is immediately reflected in the simulation – no re-compilation or re-start is necessary. Furthermore, the state of the simulation can be check-pointed, or saved, such that if perturbations of the system destroy the integrity of the steady-state model, the simulation can be brought back to a stable state and new perturbations can be tested.

3.6 Predictive Power of ABM-MAS

In addition to the expressivity, scalability, and evolutionary properties of the ABM-MAS method, it also has the capability of making and testing predictions. Outcomes of the multi-agent simulations are not determined by a global objective or control function. Thus, the system, as a function of initial conditions, will evolve dynamically into a steady-state, an oscillating state, or possibly degenerate into a chaotic state that is not sustainable. The observable features of the system state(s) are important components in measuring the predictive power of the model. If a change to the system model

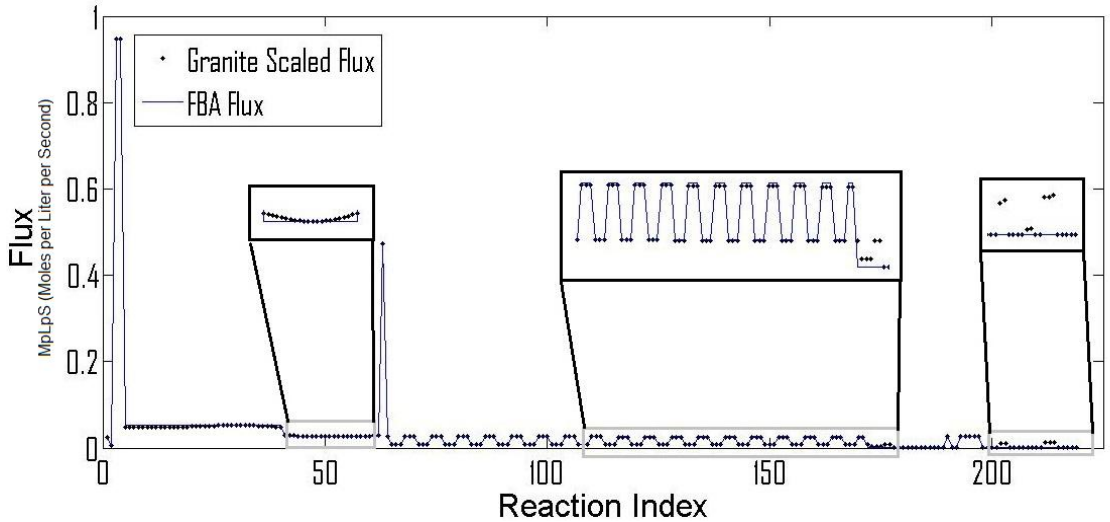

Fig. 4. Comparison of the FBA and ABM-MAS flux profiles. The reaction flux across each reaction point in the mycolic acid pathway (MAP) was compared in this plot. The x axis values represents the numbered reactions in the MAP SBML model while the y axis values represents the flux value calculated using FBA (blue line) and the ABM simulation (black dot). The inset shows the comparison of the high-complexity region of the Flux plot. The comparison shows that the ABM approach is able to emulate the internal flux properties of the FBA analyses with a correlation of 0.99

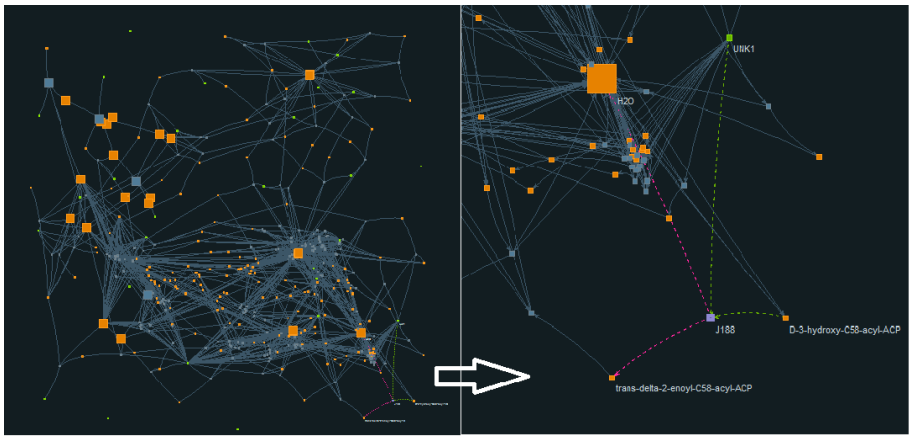

Fig. 5. The GRANITE tool includes a visualization application that allows direct interaction with the simulation. The GUI view displays a directed graph in which nodes are GRANITE memes and directed edges are the relationships between them (the right hand plot is zoomed in on a specific reaction). An edge from a meme to a reaction node implies that the meme is a reactant; an edge to a meme from a reaction node implies that the meme is a product of that reaction. Node color and size are configured based on the interaction model. In a metabolic network, color represents meme type such as reaction or metabolite. The size of a node represents reaction flux or concentration. Researchers can easily identify highly active reactions (agents) and select a subset of memes to compare their property values in real time in a chart view. Users can also perturb the system by changing some meme properties at any time and observe the effects of those perturbations on the system evolution.

(initial conditions inclusive) results in a new system-state with new observable features that can be recreated in the lab, the perturbation is informative and the predictive capability of the model increases for the next round of *in silico* experimentation. Iterations on this hypothetico-deductive cycle promise to build more accurate predictive models and reveal the general principles of the biological system under study. Thus GRANITE is an expressive, scalable, and predictive environment for modeling and simulating biological systems that enables bench researchers to integrate existing system descriptions with current hypotheses, and construct *in silico* experiments that lead to predictions which can be tested in the laboratory. The results of those experiments can inform refinements to the system model that improve the prediction capability and focus lab experimentation. We intend to apply this technique to other interaction networks and integrated systems of metabolic, gene regulatory, and signal transduction networks, that are of interest to systems biology researchers and developers.. Ongoing work is being directed toward establishing a GRANITE user community, so that comprehensive systems-level *in silico* simulations of biological and biochemical networks can be collaboratively designed, created, and developed.

ACKNOWLEDGEMENTS

The authors would like to thank the sponsoring team at NIH, led by Dr. J.J. McGowan, for providing insightful comments on this manuscript. Brian Peck at Lockheed Martin deserves a special acknowledgement for his recent development efforts on GRANITE.

REFERENCES

Erhard., *et al.*(2008) FERN – a Java framework for stochastic simulation and evaluation of reaction networks, *BMC Bioinformatics*

Fei Li., *et al. (2009)* PerturbationAnalyzer: A tool for investigating the effects of concentration perturbation on protein interaction networks, BioInformaitcs, Oxford University Press.

Bonabeau, Eric (2002) Agent-based modeling: methods and techniques for simulating human systems. Proc. National Academy of Sciences 99(3): 7280-7287

Raman, K. *et al.* (2005) Flux Balance Analysis of Mycolic Acid Pathway: Targets for Anti-Tubercular Drugs. PLoS Comput Biol., 1(5) e46. doi:10.1371/journal.pcbi.0010046

Axelrod, Robert (1997) *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton: Princeton University Press. ISBN 978-0-691-01567-5

A. S. Rao and M. P. Georgeff. (1995) BDI-agents: From Theory to Practice, In Proceedings of the First International Conference on Multiagent Systems (ICMAS'95), San Francisco.

Weiss, G. ed. (2000) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press.

Barry CE III, Lee RE, Mdluli K, Simpson AE, Schroeder BG, et al. (1998) Mycolic acids: Structure, biosynthesis and physiological functions. Prog Lipid Res 37: 143–179.

Watanabe M, Aoyagi Y, Ridell M, Minnikin D (2001) Separation and characterization of individual mycolic acids in representative mycobac- teria. Microbiology 147: 1825–1837.

J. Lawson, R. Singh, *et al (submitted 2012)*, A Retrodiction Approach Using a Genetic Algorithm to Analyze the Mycolic Acid Pathway, PLoS