

PROJECT 2 – NEW TOOLS FOR SPATIAL MODELING IN SYSTEMS BIOLOGY OF PROTEIN NETWORKS

This project is a collaboration between Core 1 Project 1 and Project 2 of the UCHC center and the TCNP center at Carnegie Mellon University. Most of the work will be performed by the UCHC team. A letter of support from Robert F. Murphy from the Carnegie Mellon center is attached.

Overview and Aims

A majority of systems biology and computational cell biology research is still confined to the world of compartmental models. Organelles, subcellular regions, or entire cells are often approximated as well-mixed compartments with uniform concentrations of signaling molecules, metabolites, proteins etc. However, it is apparent that in many, if not most, eukaryotic cells, spatial distributions (location and concentration) of many molecules involved in signal transduction cannot be ignored. They play a key role in determining the outcome of various stimulations, and generally in the overall regulatory mechanisms of various networks and pathways. Unfortunately, many of the necessary tools and technologies to effectively study these effects, both experimental and computational, are lacking or not sufficiently developed. Both of the two TCNP centers involved in this proposed collaboration are actively targeting this technology area. The computational and modeling component of the center for Polarity in Networks and Pathways at UCHC has been developing new technologies and resources for experimental data representation, modeling, and simulation for the study of intracellular networks in the specific context of spatially resolved geometries. In particular, we have focused on quantitative microscopy aspects, and technologies such as the Virtual Microscopy framework are now at the stage of functioning prototype. This work has been done by building on the Virtual Cell software platform (VCell, <http://vcell.org>), also developed at UCHC. VCell is publicly available and has the unique capability of automatically running simulations using real geometries of tissues, cells, or subcellular structures – most often derived from fluorescence microscopy data. The informatics component of the Fluorescent Probes and Imaging for Networks and Pathways TCNP led by Bob Murphy at CMU has developed new technologies to record, analyze, and generate subcellular location features (SLFs) that encode specific pattern information for intracellular structures and proteins. Some of this information is now available as a public resource, the Protein Subcellular Location Image Database (PSLID, <http://pslid.cbi.cmu.edu>). Going to the next level, an exciting new technology was developed - creating generative models. These models are built using the data and algorithms from the machine-learning step (of identifying SLFs) and have the capability to generate *de novo* realistic geometries that are essentially equivalent to geometries from collected experimental data – mimicking location patterns, quantification, and sample variability.

We propose to create a new set of components for VCell-based applications to support integration with the database and software developed at CMU. Developing programmatic interfaces for VCell tools to use these resources and algorithms would bring significant and immediate benefits. It will enable researchers to access additional resources for experimental data input, to directly analyze the impact of classifiable patterns of molecular distributions on the activity of intracellular networks, and to create complex simulations of networks where a large number of components are being studied using realistic spatial distributions (which otherwise could not be obtained experimentally at the same time in the same cell).

Aim 1 – PSLID interface for VCell

The strategy is to use web service (WS) based protocols to enable VCell users to extract cellular geometries and protein distribution data from the public PSLID database, and the Open Microscopy Environment (OME) for local installations.

Aim 2 – Virtual geometries

The strategy is to integrate the Matlab-based code from CMU to create a Geometry service for VCell applications which would enable users to create virtual geometries (e.g. cell and organelle shape) as

computational domain for simulations and/or virtual molecular distributions (e.g. protein locations and concentrations) as initial conditions for simulations (independently of, or alongside real experimental data).

Aim 3 – SBML-compatible standard for generative models

The strategy is to develop the XML-based representation of generative models in conjunction with the Systems Biology Markup Language (SBML) Level 3 spatial extensions to allow for exchange of model information with other tools that support the SBML standard.

Several fortuitous developments greatly facilitate this work and create an excellent timing for the proposed collaboration, as described below.

Plan of Work

Aim 1 – PSLID interface for VCell

We will create an additional interface module for the VCell applications to query and retrieve data from PSLID, which would be automatically converted into segmented geometries and/or distribution field data for use in simulations. Two developments will be performed. On the one hand, we plan to make available the PSLID service to the public client-server web-based version of VCell. The PSLID database is currently available for public access manually or programmatically via a web portal. Dr. Murphy's team is implementing a service-oriented application programming (SOAP) approach to enable access through standard web services (WS) mechanisms (they already have implemented a SOAP interface for their SLIF database). The UCHC team is currently introducing WS-based communications between the VCell distributed back-end services. We have started to refactor and decompose several modules of VCell (e.g. VCML translator, math generator, PDE solver, stiff ODE solver) into a set of VCell tools that can be accessed programmatically via WS, so that other compatible software can access these services. On the other hand, this requires WS client implementation in our main VCell server instance, to access these services. This same architecture can then be extended so that the main server would use the new WS-based communication to access a new Geometry service deployed at a PSLID instance (at CMU or at an UCHC mirror). The main VCell server will then route all query and upload requests from the remote Java VCell client interface.

The advantage of the architecture described above is that it decouples the user interface from the service implementation. This would enable us to easily implement a similar support for PSLID for standalone VCell applications, such as the Virtual Microscopy application. This will also be facilitated by the fact that we have chosen the Open Microscopy Environment (OME) for the next-level data source for the Virtual Microscopy application (currently only accepts individual local files). Coincidentally, the CMU team is currently developing OME support for local PSLID installations, which will be the common platform for local storage.

Aim 2 – Virtual geometries

The generative modules are implemented by the CMU team as Matlab code. The UCHC team has recently developed a JNI-based component of VCell to invoke native Matlab libraries for a different purpose (parameter estimation algorithms – we used Matlab and the optional Optimization Toolkit for rapid prototyping). In order to ease the transition from the Matlab prototype to deployment within VCell, we evaluated two different strategies. The simplest approach utilized a library to send commands and collect results from Java (VCell) to a local Matlab installation using the JMatLink library (jmatlink.sourceforge.net). This approach required the user to have both Matlab and the Optimization Toolkit to be installed on his machine and made interactive feedback difficult. The other approach was to use the Matlab Compiler which was more involved but provided for much tighter integration. In this approach Matlab functions were written to invoke the Matlab capabilities, and then compiled into native libraries that were interfaced with Java. This approach is not currently deployed for parameter estimation, due to its substantial complexity and the requirement of a very large Matlab runtime on the local machine. However, this approach is perfectly suited for server-based installations of running the

generative model code from CMU. Furthermore, for less interactive processes Matlab integration either via the Matlab Compiler or JMatLink can be a practical solution. We will develop an additional user interface that will utilize this component to invoke the generative module code to create “virtual” geometries and distributions for use in VCell simulations, independently of, or alongside, experimental input data.

Aim 3 – SBML-compatible standard for generative models

While Matlab is a fairly common tool used by modelers, especially in the physical sciences, portability of algorithms to other simulators developed for the biology community is difficult. The CMU team has created an XML-based representation of generative models with the goal of incorporating it into the current de facto standard for interoperability: the Systems Biology Markup Language (SBML), an XML language developed as a community effort. The UCHC team has been using an XML-based data and model representation for several years (VCML). We propose to build upon Dr. Murphy’s XML specification for generative models to incorporate it into SBML, and to include specific requirements in the forthcoming spatial extensions for the L3 standard. This work does not present major technical hurdles, but is somewhat complicated due to the highly specific nature of the encoding requirements for generative models, as well as of other software tools and abstractions that relate to spatial modeling. XML is a self-descriptive language, which brings a lot of flexibility, but allows for its dialects to be completely incompatible. An approach that we have used successfully over the last few years, since we introduced the VCML representation for all VCell abstractions and data structures, is to carefully define a specification with a private namespace that can be included as proprietary annotations in SBML (which provides the Annotation class specifically for this purpose). As a first step, after completing Aims 1 and 2 above, we will integrate CMU’s XML representation into VCML. This will be refined and tested as the WS communication layer will be upgraded to rely solely on the extended VCML for communicating with PSLID and the Geometry service described above. This would also allow use to use the VCML/SBML translator for generative model representations, which we will then refine to create a mapping from proprietary descriptors to SBML standard descriptors. This would allow easier integration of current and future capabilities of the CMU generative model code, both into VCell tools and other computational biology software. This integration step will require interactions with the rest of the community during the development of the Level 3 extensions. One particular problem is that the generative models encode information relating to both shape and molecular distribution, which fall under two distinct aspects of the Level 3 extensions: encoding of geometry and support of the actual spatial model. Coincidentally, the UCHC team has been a member of the SBML effort from its inception, and is currently the lead for developing the spatial extensions for SBML Level3. Therefore, we are ideally positioned to resolve these potential problems.