Enhancing CompuCell3D with Workflows and Data Provenance

Randy Heiland
Open Systems Laboratory
Indiana University
Bloomington IN 47405, USA

Julio Belmonte
Biocomplexity Institute and
Department of Physics
Indiana University
Bloomington IN 47405, USA

Maciek Swat
Biocomplexity Institute and
Department of Physics
Indiana University
Bloomington IN 47405, USA

Claudio Silva
Scientific Computing and
Imaging Institute
University of Utah
Salt Lake City UT 84112, USA

Andrew Lumsdaine
Open Systems Laboratory
Indiana University
Bloomington IN 47405, USA

James A. Glazier
Biocomplexity Institute and
Department of Physics
Indiana University
Bloomington IN 47405, USA

We describe a project composed of a union of distinct open source software packages, resulting in valuable new functionality with surprisingly little additional effort. The software of primary scientific interest is CompuCell3D, an open source multi-cell modeling framework. We will provide an overview of this package and its underlying Glazier-Graner-Hogeweg algorithm. To enhance CompuCell3D, we adopt another open source application, VisTrails, to provide an easy-to-use workflow tool and data provenance management. Additional open source packages accompanying this project include Globus (for some minimal Grid utilities), the Visualization Toolkit (VTK), and the Python scripting language. Both VTK and Python come bundled in VisTrails. The overarching goal of our project is to make it easier to explore the parameter space for any CompuCell3D model. We will briefly review an earlier use case involving cell sorting, but the use case model of primary interest is somitogenesis.

CompuCell3D (www.compucell3d.org) is a multi-cell modeling *framework* [1]. That is to say, users are able to develop many different types of models, perhaps from different science domains even – physics and biology being the most prevalent. CompuCell3D uses what has come to be called the Glazier-Graner-Hogeweg (GGH) algorithm. GGH follows an energy-minimization philosophy and has as its primitive unit a *generalized cell*. In the current implementation, the spatial domain is a regular lattice (2D or 3D; square or hexagonal). A typical simulation consists of multiple *cell types* evolving in time – interacting with neighboring cells (stochastically) and adhering to various model-defined constraints. Additional fields, e.g. chemotaxis, can also be defined that may affect the dynamics of the generalized cells.

CompuCell3D can be run as an interactive application (Fig 1), displaying results as they are computed in a GUI, or run as a batch program that writes raw data that can be post-processed. A CompuCell3D model is defined textually using either an XML file or a Python script. To explore the effects of changing one or more parameters in a model, e.g. via parameter sweep(s), one would typically do so in a rather ad-hoc and serial fashion. It is this functionality that we wish to address here.

VisTrails (www.vistrails.org) is an open source scientific workflow and provenance management system [2]. Workflows are expressed as dataflows (pipelines) – a set of modules, with links connecting input and output ports, that get executed, in the simplest case, from top to bottom. VisTrails also provides constructs for functional loops and conditionals. As a standalone application, VisTrails provides an interactive Builder window in which one constructs a pipeline (via drag and drop), sets parameters and executes it. Figure 2 illustrates a very simple pipeline that performs an arithmetic binary operation and outputs the result to a console panel. On the left side of the Builder window, we have categories of built-in modules from which to drag and drop. The primary center panel is used to graphically build the pipeline and the smaller picture-in-picture panel in the upper-right is the (graph) history associated with this pipeline. When a user tries to connect an output port of one module to an input port of another, VisTrails will enforce datatype matching. Figure 3 depicts the user interface for extending VisTrails. Provenance data associated with a workflow (a *vistrail*) is maintained via XML files or a relational database. As mentioned before, VisTrails is capable of performing visualizations. These appear in a separate interactive Spreadsheet window. Fortunately for us, VTK is used by both CompuCell3D and VisTrails (Fig 4).

Graphical dataflow networks are not a new concept. The scientific visualization community was exposed to a variety of them about two decades ago, e.g. AVS, SGI

Explorer, and IBM Data Explorer. Workflow systems in use today are typically more broadly defined and offer increased functionality. VisTrails is obviously not the only system available. Two other popular open source scientific workflow systems include Taverna and Kepler. And Microsoft Research now offers its Trident scientific workflow system. We chose VisTrails for this project primarily because of the commonality we saw between it and CompuCell3D, namely, the provision of VTK and the support of Python scripting. It was only later that we fully appreciated its data provenance capability.

For an initial use case, we show previous results from a biological cell sorting model. Cell sorting is a well-known biological process and can be described (in the simplest case) as the reorganization of a random mix of two cell types. The two cell types differ in their cell adhesivities (stickiness). A canonical outcome of a cell sorting simulation is that cell types with lower adhesivity (noncondensing) will engulf cell types with higher adhesivity (condensing). However, by constructing a workflow that automatically sweeps a range of parameters, we can obtain a variety of solutions. We use VisTrail's *Cross* module (in the Control Flow category) to take the cross-product of two lists, offering a convenient method for performing parameter sweeps. By daisy-chaining *Cross* modules together, we can easily build up sets of parameter values. The *Map* module (also in the Control Flow category) will apply a generic function to a given input list, resulting in a sequence of results. The generic function used here 1) edits the input model, inserting the new parameters, and 2) executes CompuCell3D in batch mode. The workflow can quickly become both computationally and data intensive. For the cell sorting workflow, 72 sets of parameters were explored, resulting in a variety of qualitatively distinct outcomes (Fig 5).

To run the entire parameter study simultaneously, we install the Globus client software on the laptop running VisTrails and (using a TeraGrid account) submit all jobs (*globus-job-run*) to Indiana University's Big Red cluster (a TeraGrid resource). Output files from the simulations are written to GPFS. These files are then retrieved (*globus-url-copy*) with another workflow that also performs the visualization using VTK. All metadata (provenance) associated with the workflows are maintained by VisTrails as XML files (optionally, in a database).

Once the basic workflow mechanism was in place for this simpler CompuCell3D model, we addressed a far more complex model – somitogenesis. During embryonic development in a vertebrate, somites (primitive segments) form as spheroids along both sides of the neural tube (Fig 6). Somitogenesis is a complex interaction of multiple cell types, multiple genetic pathways, and the cell clock cycle. It is not our intent to describe the model here; however, more details will be given in the presentation. Suffice it to say that we use a VisTrails workflow in the same manner as described for the cell sorting use case. As of now, we have simply targeted a single parameter in the model to explore – a parameter that seems to be critical in normal somite formation (Fig 7). We will present our preliminary findings from the somitogenesis parameter study and discuss plans for future integration of CompuCell3D and VisTrails.

[1] Swat, M. H., Hester, S. D., Balter, A. I., Heiland, R. W., Zaitlen, B. L., and Glazier, J. A. 2009. Multicell simulations of development and disease using the CompuCell3D simulation environment. In Systems Biology, I. V. Maly, Ed. volume 500 of Methods in Molecular Biology, pages 361--428. Humana Press, Clifton, N.J.

[2] Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. 2006. VisTrails: visualization meets data management. In Proceedings of the 2006 ACM SIGMOD international Conference on Management of Data (Chicago, IL, USA, June 27 - 29, 2006). SIGMOD '06. ACM, New York, NY, 745-747. DOI= http://doi.acm.org/10.1145/1142473.1142574

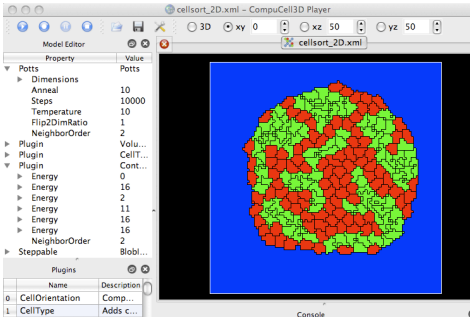

Fig 1. Interactive CompuCell3D application

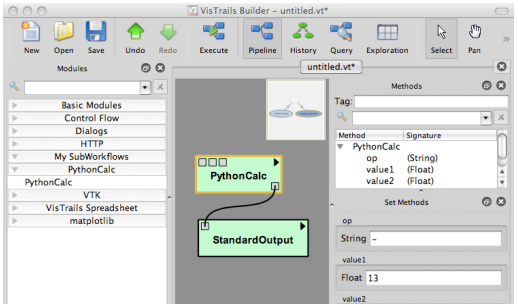

Fig 2. VisTrails application with trivial workflow

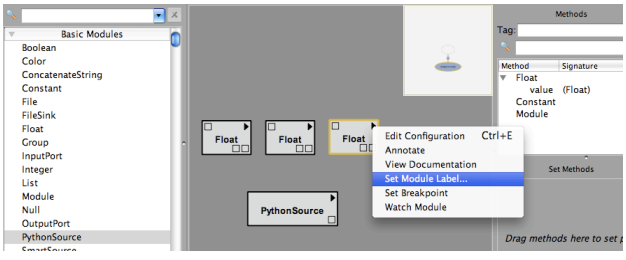

Fig 3. Extensibility of VisTrails

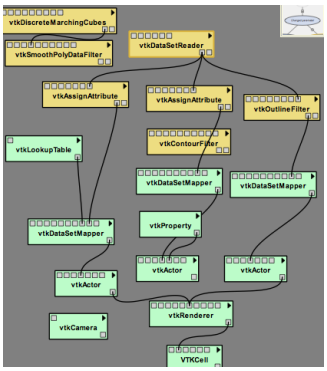

Fig 4. VisTrails VTK workflow

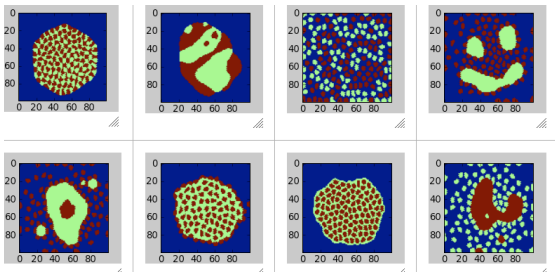

Fig 5. Results from cell sorting workflow

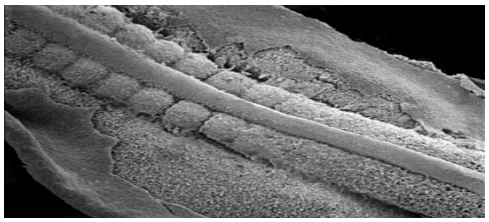

Fig 6. Somites in vertebrate embryo

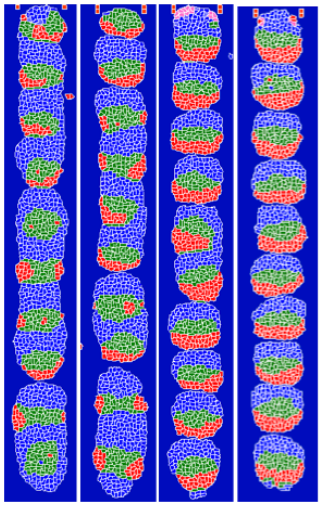

Fig 7. Results from somitogenesis workflow (worst to best)