Cytoscape Springs Forward: Re-architecture for Version 3.0

Allan Kuchinsky[1], Keiichiro Ono[2], Michael Smoot[2], Trey Ideker[2],Annette Adler[1]
[1] Agilent Technologies, 5301 Stevens Creek Blvd., Santa Clara, CA 95051 USA
[2] Department of Genetics, University of California, San Diego, CA 92093 USA
allan_kuchinsky@agilent.com

Website: http://cytoscape.org

Cytoscape is an open source bioinformatics software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other data. The Cytoscape core distribution provides a basic set of features for data integration and visualization. Additional features are available as plugins. Plugins are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. Plugins may be developed by anyone using the Cytoscape open API based on Java™ technology and plugin community development is encouraged.

Over the past several years, Cytoscape has become a standard resource in academia and industry for bio-molecular network analysis. Cytoscape usage has increased at a rate of ~50% per year. To date, Cytoscape has been downloaded over 57,000 times, with a current download rate of 2500/month. There are 74 registered plugins, freely available from the Cytoscape Web site.

This success has engendered some growing pains. While powerful and feature-rich, Cytoscape is a monolithic application presenting a large and complex interface to developers. Like other open source projects with large developer communities, parts of the software have evolved organically with contributions from many programmers. Application components have become tightly-coupled and increasingly fragile. Small changes in one subsystem can have unintended consequences in another, making it difficult to add new features. Moreover, there are currently no mechanisms for plugins to communicate with each other or for supporting different versions of libraries.

To solve these problems, we are re-architecting Cytoscape to be more scalable and flexible and to ease the task of writing plugins, with the following design requirements:

- Organize the source code into clearly defined modules, each with clearly defined interfaces and function.
- separate interfaces (API) from implementation. This enables us to replace actual implementations without breaking existing plugins.
- Simplify the public plugin API. Currently, the Cytoscape public API contains >5000 classes.
- Function across various computing contexts. Such contexts include using Cytoscape to serve as a backend processor for a web service; to run network analyses within a GRID or cluster computing environment including effective use of modern multi-core CPUs; and control via a command line for batch processing.
- The modularization process must ensure that Cytoscape does not lose current functionality or performance.

We are re-architecting Cytoscape using the suite of tools and interfaces provided by the Open Services Gateway Initiative (OSGi, http://osgi.org) and the Spring Dynamic Modules framework (http://www.springsource.org/osgi). The goal is to provide a service architecture which allows modules to register services to be used by other modules and advertised based only on interfaces, so that implementation details are hidden. This re-architecture is the basis of the next major release of Cytoscape, version 3.0. As we proceed, we are addressing several tough design issues.

- Event handling: how do we optimize performance when large amounts of events are being fired, e.g. how to avoid unnecessary network redrawing without introducing dependencies between event producer and consumer.
- Model vs. view: what view information, such as nodes coordinate positions, really should be persisted.
- Handling multiple networks: when to share vs. copy attribute values for equivalent nodes/edges .
- How simple an API? If too simple, do we risk proliferation of ad hoc solutions for common operations?
- The use of new technologies/frameworks will increase the learning curve for plugin developers. We are trying to solve this by creating templates (Maven archetype) and writing tutorial documents.

I will discuss these design issues during my presentation. I will also illustrate these points from the perspective of a plugin, the Cytoscape network editing tool, which brings these issues to bear. I will be interested to learn how fellow participants may have addressed some of these issues.