GTA: Graph Tool Analyzer

F.Ganci, A.Giacomini, E.Monaco, A.Rastelli

July 22th, 2014

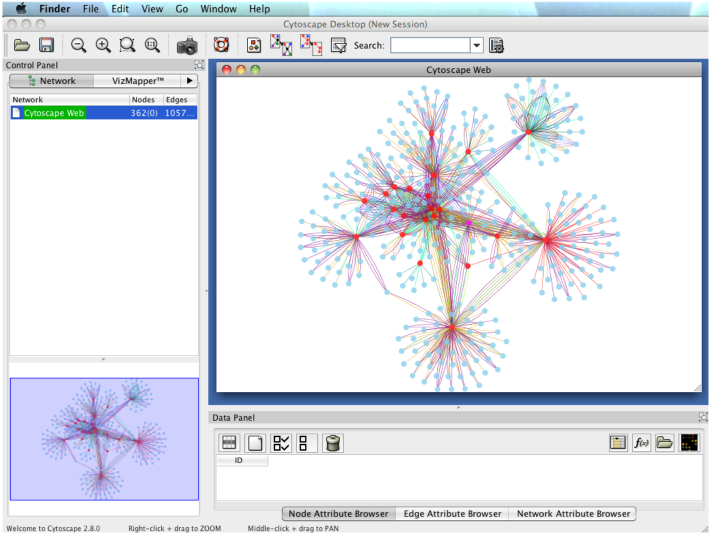
Overview

- In this project we have developed GTA (Graph Tool Analyzer), a plugin for Cytoscape. The plugin is able calculate shortest paths and compute several network centrality algorithms on edge weighted directed graphs
- Used technologies
  - Eclipse
  - Java SE Development Kit 7
  - Cytoscape 2.x

What is Cytoscape?

- Cytoscape is an open source software platform for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data
- Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization
- Cytoscape core distribution provides a basic set of features for data integration, analysis, and visualization. Additional features are available as Apps (formerly called Plugins)

Cytoscape Screenshot



Network	Nodes	Edges
Cytoscape Web	362(0)	1057…

Welcome to Cytoscape 2.8.0 Right-click + drag to ZOOM Middle-click + drag to PAN

Input Graph

- Directed edge weighted graph
- Formats
    - SIF
    - NNF
    - GML
    - XGMML
    - SBML
    - BioPAX
    - PSI-MI Level 1 and 2.5
    - Delimited text
    - Excel Workbook (.xls)

Utilized Pattern

- Model View Controller (MVC) is a software architectural pattern for implementing user interfaces
- It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user. The model view controller design defines the interactions between them
- Well structured code and easily extensible

Methods

- Topologic centrality indexes implemented:
    1. Degree
    2. Stress
    3. Betweenness
    4. Closeness
- Thanks to these algorithms, networks can be studied from different point of views: each node can be viewed in the network's totality or in its neighborhood

Algorithms: Degree

- *Degree centrality* is based on the idea that important nodes are those with the largest number of ties to other nodes in the graph. It is often interpreted in terms of the immediate involvement of nodes in relationships established through the network. Let w_{us} be the weight of the arc connecting node u with node s, the degree centrality of a node u is:
  $C_d(u) = \sum_{u,s \in E} w_{us}$
  according to which the higher is the degree value, the more important (connected) is the node.

Algorithms: Stress

- *Stress centrality* was designed to count the number of shortest paths that contain an element u, gives an approximation of the amount of work or stress the node has to sustain in the network. It results from the sum:
  $C_s(u) = \sum_{s \neq u \neq t \in V} \sigma_{s,t}(u)$
  The higher $C_s(u)$ is, the more central (and busy) is u.

Algorithms: Betweenness

- *Betweenness centrality* measures the influence a node has over the spread of information through the network. Betweenness, in its basic version, is computed as the fraction of shortest paths between node pairs that pass through the node of interest. Its mathematical expression is:
  $C_b(u) = \sum_{s \neq u \neq t \in V} \frac{\sigma_{s,t}(u)}{\sigma_{s,t}}$
  where $\sigma_{s,t}$ is the number of shortest paths from s to t, and $\sigma_{s,t}(u)$ is the number of shortest paths from s to t that pass through a vertex u.

Algorithms: Closeness

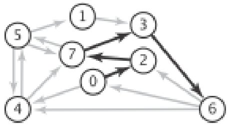
- *Closeness centrality* is defined in a metric space where nodes are ranked because of their geodesic distances or proximity to other nodes of the graph. Indeed, an important node is typically close to, and can communicate quickly with the other nodes in the graph. In other words, it can be regarded as a measure of how important is a node in relationship to the reachability of any other node. Closeness is computed as:
  $C_c(u) = \frac{1}{\sum_{t \in V \setminus u} d_G(u,t)}$

Princeton University Package

- In order to compute the shortest paths needed for the centrality algorithms, a Princeton University package has been used

edge-weighted digraph

4->5	0.35
5->4	0.35
4->7	0.37
5->7	0.28
7->5	0.28
5->1	0.32
0->4	0.38
0->2	0.26
7->3	0.39
1->3	0.29
2->7	0.34
6->2	0.40
3->6	0.52
6->0	0.58
6->4	0.93



shortest path from 0 to 6

0->2	0.26
2->7	0.34
7->3	0.39
3->6	0.52

An edge-weighted digraph and a shortest path

Initial testing

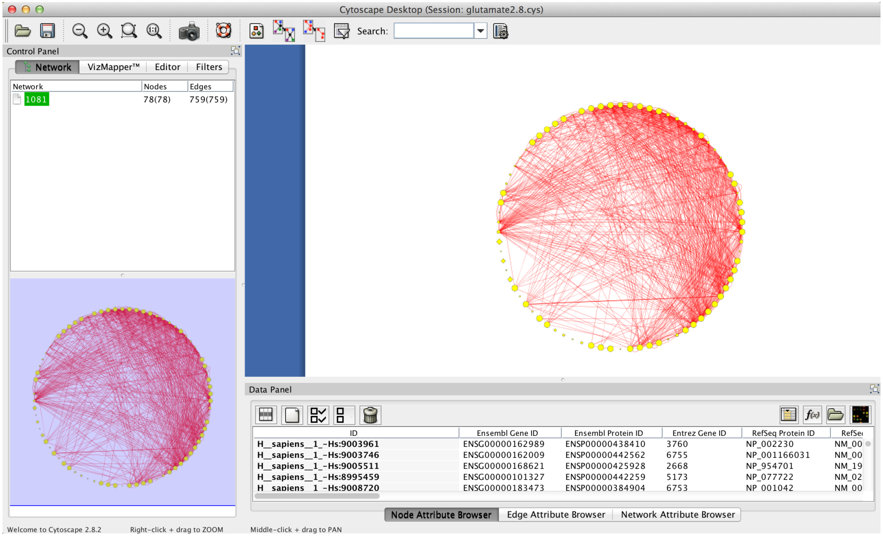
- "Dummy" Graphs used in the first phase of testing
  1. Completely disconnected graph (worst case scenario)
  2. Linear graph (A -> B -> . . .)
  3. Complete graph (best case scenario)
  4. Random graph

Use Case

In the following example of how to use our plugin we are going to use a biological network of the glutamate receptors and their interactions with its neighbors

- **Data set**: the information in the network is taken from BioGRID (Database of Protein and Genetic Interactions)
- **Objective**: determine “key players” of the network using our plugin GTA

Network Screenshot



Cytoscape Desktop (Session: glutamate2.8.cys)
Search:
Control Panel
Network
VizMapper™
Editor
Filters
Data Panel
Node Attribute Browser
Edge Attribute Browser
Network Attribute Browser
Welcome to Cytoscape 2.8.2
Right-click + drag to ZOOM
Middle-click + drag to PAN

Network	Nodes	Edges
1081	78(78)	759(759)

ID	Ensembl Gene ID	Ensembl Protein ID	Entrez Gene ID	RefSeq Protein ID	RefSe
H_sapiens_1_-Hs:9003961	ENSG00000162989	ENSP00000438410	3760	NP_002230	NM_00
H_sapiens_1_-Hs:9003746	ENSG00000162009	ENSP00000442562	6755	NP_001166031	NM_00
H_sapiens_1_-Hs:9005511	ENSG00000168621	ENSP00000425928	2668	NP_954701	NM_19
H_sapiens_1_-Hs:8995459	ENSG00000101327	ENSP00000442259	5173	NP_077722	NM_02
H_sapiens_1_-Hs:9008720	ENSG00000183473	ENSP00000384904	6753	NP_001042	NM_00

Live Demo

- Live demo of our plugin

Future Developments

Thanks to the MVC pattern and a well structured code our plugin can be easily extended:

- Adding other Centrality algorithms
- Improve the plugin's performance on very big networks
- Developing a more user friendly interface and more functionalities with the aim to allow its usage to a wider audience
- Porting the plugin to future Cytoscape versions

Tasks

- Back end: Fabio Ganci, Eder Monaco
- Front end: Alessandro Giacomini, Alessandro Rastelli
- Algorithms: equally divided by the members of the group