Biological Network Exploration Tool: BioNetXplorer

Carlos Roberto Arias Arévalo
Universidad Tecnológica Centroamericana, Tegucigalpa, Honduras, cariasa@unitec.edu

ABSTRACT

BioNetXplorer is a standalone application that allows the integration of more than twenty topological and biological properties of the nodes of a biological network, and that displays them in a intuitive, easy to use interface. Along this functionality the application can also perform graphical shortest paths analysis and shortest paths scoring of genes, due to the intensive computational requirement of the later it has the potential to connect to a Hadoop cluster for faster computation.

Keywords: Biological Network, Topological Analysis

RESUMEN

BioNetXplorer es una aplicación de escritorio que permite la integración de más de veinte propiedades topológicas y biológicas de los nodos de una red biológica, mostrando esta información en un interfaz intuitivo y fácil de usar. Además de esta funcionalidad la aplicación también realiza análisis de caminos más cortos de manera gráfica, y la evaluación de puntuación de genes por medio de caminos más cortos. Debido a las necesidades intensivas de computación de esta última funcionalidad el software tiene el potencial de conectarse a un clúster de Hadoop para computación más rápida.

Palabras claves: Red Biológica, Análisis Topológico

1. INTRODUCTION

With the increasing amount of available biological network information, network analysis is becoming more popular among researchers in the bioinformatics field, therefore there is an increasing need for diverse network analysis tools that facilitate the study of biological networks, analysis tools that integrate both topological and biological data, and that can display the most information in an integrated graphical interface. Another "must-have" capability is the ability to export the annotated networks in formats that can be processed by other available tools like Cytoscape (Shannon, et al., 2003) for graphical analysis or RapidMiner (Rapidminer, 2014) for data mining. Several tools for this purpose have been developed over the years, like Centiscape (Scardoni, Petterlini, & Laudanna, 2009) and NetworkAnalyzer (Assenov, Ramirez, Schelhorn, Lengauer, & Albrecht, 2008) that compute at most seventeen network topological properties, or like VisANT (Hu, Mellor, Wu, Yamada, Holloway, & DeLisi, 2005) that is used for biological analysis based on Gene Ontology (Gene Ontology Consortium, 2010); however, they do not integrate available topological properties along with biological information like Gene Ontology annotations in a single interface, or only compute some subset of topological properties. For this reasons we have devoloped BioNetXplorer, a tool with a user friendly and intuitive interface, with a rich documentation that will allow users to easily start analyzing their own networks. Our main contribution is the capacity to compute several network and node topological properties, retrieve biological related data from external sources and display all of these information in a single interface, furthermore the application can do graphical shortest path analysis and has a gene prioritization utility that is able to connect to an external cluster to improve the time performance of the computation.

2. BACKGROUND

A graph is a data structure that represents a set of relationships between elements or objects. Formally a graph G is a pair defined by G = (V, E), where V is a set of elements that represent the nodes or vertices of the graph, the vertices may or may not hold information, for the purpose of this paper the information that is held in the vertices is dependent on the especic problem that is being discussed, some applications hold natural numbers on them, but some others hold the name of an entity like a gene or protein. E is the set of edges, where each edge represent a relation between two vertices, an edge is defined by $E = \{(u, v)| u,v \in V\}$, this edge may hold additional information as weight. The edges may represent direction, where $(u, v) \neq (v, u)$, in which case the graph is called directed graph, and when direction is not important, the graph is called undirected graph. It is denoted that n is the number of vertices in the graph, formally n = |V|, and e is the number of edges, e = |E|.

Following the properties used in BioNetXplorer are presented. First vertex properties are introduced, as some of these definitions will be later used in the network properties discussion.

2.1 VERTEX INVARIANTS

Degree

This measure could be expanded to In Degree and Out Degree for directed graphs. This property measures how many connections a vertex has, another way of saying how many edges converge to this vertex or how many vertices are neighbors of this vertex. This property will be referred as InDegree(v) and OutDegree(v). For undirected networks the following identities are used: OutDegree(v) = Degree(v) and InDegree(v) = -1.

Neighbor Connectivity Index

This measure was introduced by Tian et al. (Tian & Patel, 2008) and its magnitude tells how the neighbors of a vertex are connected to each other. Basically this invariant counts how many edges are between the neighbors of a given vertex. For illustration look at Figure 1.

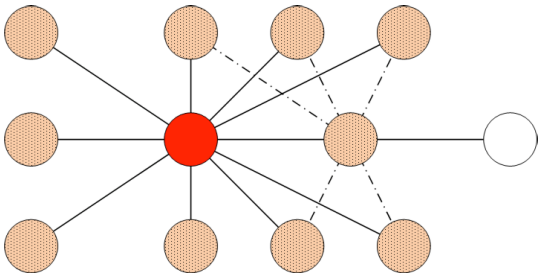

Figure 1: Network Connectivity Index

Notice the dashed edges; they are the edges that connect the immediate neighbors of the darker node. In this particular case the Neighborhood Connectivity Index value of the darker node is 5 as the number of the dashed edges. BioNetXplorer has two very similar invariants. One counts all edges including the self edges, in the case there is an edge going to the same vertex, and this invariant is called nbc1(v). Another one does not count the self edges of the vertices, this one is called nbc2(v). This invariant only counts Out Degree in the case of directed networks.

Singles Count

The singles count invariant shows how many of the neighbor vertices of a vertex have degree = 1. This property will be referred as SingleCount(v).

Single
It is a boolean value that stores takes the value of true when the vertex has only one incoming connection, making this vertex a "single vertex". This property will be referred as isSingle(v).

Closeness Centrality
This vertex invariant represents how close a vertex is to all the others (Freeman, 1979). It is defined by:

$$closeness(v)\frac{1}{\sum_{t\in(V-v)} d_G(v,t)}$$

This is one of the centrality measures, these measures try to represent how central a vertex is in the whole network. The closer this number is to 1 the more central it is, and the easier is for a vertex to reach the others much faster. In Social Networks a vertex representing a person with a high value of closeness would be interpreted as a person that can easily reach most people in the network, either directly or with very few hops through other persons. Notice one important aspect about this invariant, and is that it looks correlated to degree, but there is not always a direct relationship between closeness centrality and degree.

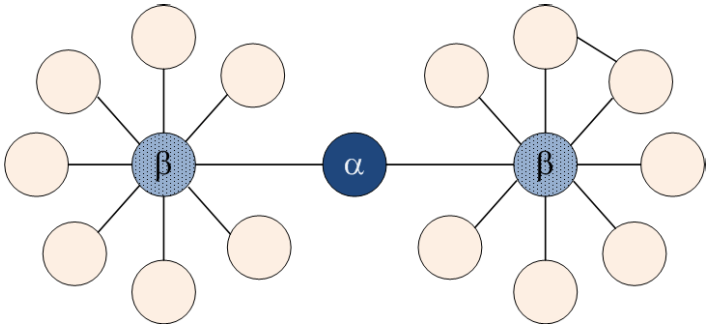

Figure 2: Closeness Centrality

In Figure 2 is clear that the β nodes are the ones with highest degree, but their closeness centrality value is 0.0321, and the α node, although its degree is only two, has the highest closeness centrality value, 0.0333. This happens because the α node is more "central" than the β nodes.

Self Edge
This property will be called hasSelfEdge(v). This is a simple boolean invariant that is true when the node has an edge that goes to itself: (u,u) ∈ E.

Eccentricity
This property of the node stores an integer representing the longest shortest path from this node. It measures how far this node is from the center of the network, so for values close to the diameter of the network would be interpreted that that vertex is on the "edge" of the network, and the lower this value the more central the node is.

$$eccentricity(v) = \max\{shortest\_path(v_i)\}$$

Articulation Point
This is another simple boolean property that is true if the node is an articulation point in the graph. An articulation point, also referred as Cut Vertex, is a vertex v such that when it is eliminated from the graph along with all its incident edges V' = V - {v} and E' = E - {(v,u) | u ∈ V} a new graph G' = (V',E') is created, and this new graph is divided in two or more connected components (Aho, Hopcroft, & Ullman, 1983). If all articulation points of a graph are found, then we would have found all the maximal biconnected components of the graph, which would form a tree whose elements are all the articulation points and the biconnected components. This property will be referred as Articulation_Point(v).

Star Center
This is a boolean property that becomes true when the vertex is central to "many" single vertices. Many is defined by more than 50% of the adjacent vertices are single vertices. So, this property is true when the following relationship holds:

$$isStarCenter(v) = \frac{single\_count(v)}{degree(v)} > 0.5$$

Central
This is a boolean property that becomes true for all v ∈ V that hold the following relationship:

$$isCentral(v) = eccentricity(v) = radius(G)$$

Peripheral
This is a boolean property that becomes true for all v \in V that hold the following relationship:

$$isCentral(v) = eccentricity(v) = diameter(G)$$

Clustering Coefficient
The clustering coefficient of a node describe how close the node and its neighbors are to become a full graph: a clique. If the node and its neighbors form a clique, then the clustering coefficient takes on the value 1. This property is defined by the following formula:

$$clustering(v) = \frac{2 * ec}{nc(nc-1)}$$

Where ec is the number of edges in the subgraph made only of node v and its neighbors, and nc is the number of nodes of that same subgraph.

Entropy
This a vertex invariant this is defined by the following formula (Simonyi, 1995) (Shetty & Adibi, 2005):

$$entropy(v) = p\big(degree(v)\big)\ln(p(degree(v))$$

Where p(k) is the ratio of vertices that have degree k. This value tries to quantify the expected value of the information contained in each node, this is based on the degree of the nodes. In the case of a directed network two values are computed, one for In Degree and another for Out Degree.

Degree Centrality
The degree centrality is defined by:

$$degree\_centrality(v) = \frac{degree(v)}{n-1}$$

Like entropy, this invariant is computed for both In and Out Degree for directed networks. Basically represents how much connected a vertex is to its neighbors. It is the simplest of the centrality measures, and it measures the the number of incident links to a node in relation to the number of nodes in the whole network.

JUNG Vertex Invariants
JUNG stands for Java Universal Network / Graph Framework (O'Madadhain, Fisher, White, & Boey, 2003), it is an open source software library for Java that provides tools for modeling, analysis and visualization of data that can be represented as a network or graph.

JUNG integrates very well with the Java programming environment, in addition it has extensive examples provided with the library. These two reasons made the integration of JUNG into BioNetXplorer. Along the way there were other two advantages that were discovered, one is that it has a very powerful customization mechanism, that allows many graphical visualization operations to be specified by the user of this Application Programming Interface (API). The other advantage is that the visualization of biological networks was better than the usual Cytoscape visualization tool (Shannon, et al., 2003).

The properties that we use from the JUNG library are described next, the descriptions are taken from the JUNG API (http://jung.sourceforge.net/doc/api/index.html) description and complemented with the indicated references.

- **Page Rank**. This is an eigenvector-based algorithm. The score for a given vertex can be seen as the fraction of time spent `visiting' that vertex in a random walk. It modifies the usual random walk by adding to the model a probability. This score was originally proposed by Larry Page and is used by the Google search engine. This algorithm tries to score all the vertices, or web pages as seen by Google, so that higher scoring vertices are more important or relevant.
- **Distance Centrality**. Assigns a value to the vertices based on the distances to each other vertex in the graph. If the value is normalized by averaging then this property is equivalent to closeness centrality.
- **Barycenter**. It is much like Distance Centrality but this is not an averaged value.
- **Closeness Centrality**. Is a vertex score based on the mean distance to each other vertex.
- **HITS**. Assigns hub and authority scores to each vertex depending to the network topology. The idea is that a vertex becomes a hub as long as it has links to authorative vertices, and is an authority vertex if it links to hub vertices.
- **Betweenness Centrality**. This measure scores vertex in such a way that vertices that appear in more shortest paths will have higher betweenness centrality score. It is defined by the following equation:

$$betweenness\_centrality(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

  Where σ_{st} is the number of shortest paths that go from s to t and $\sigma_{st}(v)$ is the number of shortest paths that go from s to t but pass through v. The implementation in JUNG makes reference to (Brandes, 2001), in there it claims that the algorithm is optimized, but experiments proved that this property computation is very time consuming.
- **Eigenvector Centrality**. This property measures the fraction of time that a random walker will spend at the specific vertex, over an infinite time horizon. PageRank is a variant of this score. JUNG API assumes that the graph is strongly connected.
- **Random Walk Betweenness Centrality**. Instead of using the shortest paths to compute the Betweenness Centrality this property uses the expected number of times a node is traversed by a random walk averaged over all pairs of nodes.

2.2 NETWORK INVARIANTS

There are many properties that can be computed to describe certain characteristics about a graph. The most trivial ones are the number of nodes of a network, or the degree of a node. Nevertheless there are other properties that help researchers to understand the nature of the network. Based on the specification of simply the number of nodes and degrees of nodes, one can easily have a lot of networks that fall into the same category, then more properties would help classify or identify networks and vertices more accurately. Therefor BioNetXplorer computes several properties of a graph, which are called network and vertex invariants (as discussed in the previous section), to help the user conduct a more precise identification of the intrinsic properties of the network. The network invariants we studied are:

Nodes

The most simple invariant of all, is the number of nodes in the network.

Edges

It is the number of edges in the network, this is computed from the degree of each node.

Directed

It is a boolean value that shows if the network is directed or not.

Strongly Connected

It is a boolean value that indicates whether the network is strongly connected or not, this is if given any vertex of the network we can reach all the other vertices.

Radius
The network radius is defined by:

$$radius(G) = \min_{v_i \in V}\{eccentricity(v_i)\}$$

Diameter
The diameter is defined by:

$$radius(G) = \min_{v_i \in V}\{eccentricity(v_i)\}$$

This represent the longest shortest path in the whole network, that is why it has the sense of diameter, as it is the longest shortest distance from any two points in the graph.

3. SYSTEM OVERVIEW

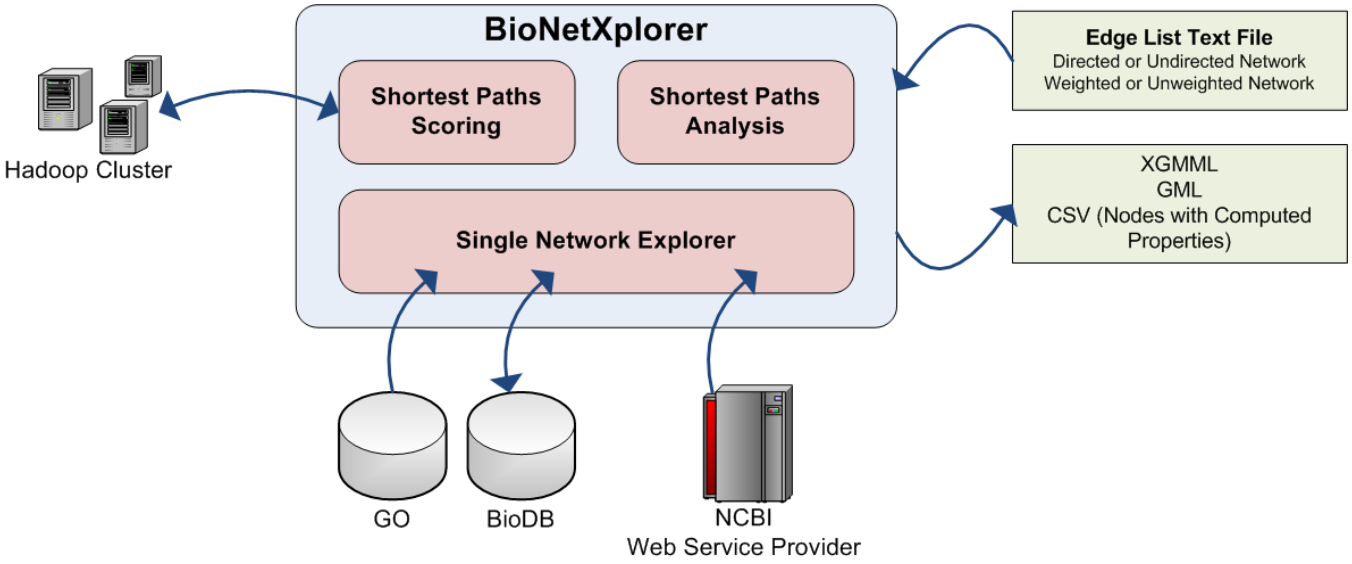

Figure 3: System Overview

BioNetXplorer has three main functionalities: Single Network Exploration, Shortest Paths Analysis and Shortest Paths Scoring, see Figure 3. The *Single Network Exploration*, serves to survey structural and topological properties of the network and individual nodes, it also has the capability of retrieving biological data from the nodes by connecting to a Gene Ontology database and to the National Center of Biotechnology Information (NCBI) Gene Entrez Web Service (Maglott, Ostell, Pruitt, & Tatusova, 2005). When the user opens a network he can select the topological properties that will be computed, the properties that may be selected are: Neighbor Connectivity (NBC) (Tian & Patel, 2008), single count, closeness centrality, network diameter (and radius), articulation points, clustering coefficient, entropy and degree centrality; besides these properties that are computed by our own network library, we have also incorporated the computation of other properties by the Java Universal Network/Graph (JUNG) (O'Madadhain, Fisher, & Nelson, 2010) library, they are: page rank, closeness centrality, eigencentrality, HITS, barycenter, distance centrality, betweenness centrality and random walk betweenness. Once a network is loaded, and the selected properties are calculated the interface allows the user to be able to see all of the node properties in the same window, given that the user has a computer screen with

enough resolution; for lower resolution screens we made a "tabbed" version, where all structural properties are grouped together in one tab, and all biological properties are grouped in another tab. Additionally, this exploration interface allows the user to load graphs in edge list file formats where the networks can be directed or undirected, have weights or not; save the network with computed properties, retrieve and store the network in a MySQL database (Local BioDB), export all vertex properties in comma separated values (CSV) format, export in standard XML format with the specification XGMML (Punin & Krishnamoorthy, 2010) and export in GML (Himsolt, 1996) format. The exportation of the vertex properties in CSV format permits the use of this information as feature vectors and process them in external tools like RapidMiner or Weka, to apply machine learning or data mining algorithms in these data. XGMML format was selected because is standard XML consequently can hold all the properties that are shown in our application. What is more, GML and XGMML are both formats that are importable from Cytoscape, thus permitting the user to do further analysis in Cytoscape.

This interface also allows the user to see a list of the vertices sorted by for different properties: Degree, Closeness Centrality, Articulation Point, Importance. The importance parameter is defined as a node that has a high degree value, and also has high closeness centrality value; so this vertex besides having a lot of local connections also is central to the network. The important vertices are sorted based on whether they are articulation points of the network or not and on their closeness centrality value.

The Shortest Paths Analysis is a utility that allows the user to create subnetworks based on the nodes that are on the shortest path between several user given pairs of vertices, the interface permits the graphical display of the created subnetwork, and to export this subnetwork to XGMML for further analysis in external tools.

The Shortest Paths Scoring, is a functionality that is used to do gene prioritization, where the user can load a weighted network and a training set of seed genes that will serve as sample for prioritizing the rest of the genes on the network using a shortest paths gene scoring method. This utility requires the computation of all the shortest paths on the network which is a time consuming part of the scoring, therefore we incorporated to the application an interface to compute the score in a Hadoop[1] cluster, dramatically improving the time performance of this computation.

Figure 4 shows how BioNetXplorer interacts with external services, and with local databases. In adiction, shows the third party libraries that it uses:

- jsch, for SSH protocol handling, helps with the communication with the hadoop cluster.
- EUtilsLib, for the web services that connect to the NCBI Gene Entrez
- JUNG, to compute additional topological properties
- mysql-connector, to connect to local databases: MyGO a replica of the Gene Ontology Full Database, and BioDB a database to store the networks with the computed properties for later retrieval and analysis
- GO4J is a planned add on to access the Gene Ontology database directly
- NetworkLib, the in house developed library and framework where core computations are handled

The reason to compute all vertex invariants is to be able to explore the network from different perspectives, and also these properties help identify structurally important nodes. This information combine with the biological data helps biological network researchers to visualize interesting patterns thus localizing relevant genes in genetic networks. Figure 5 shows how all data is displayed in a single interface for the convenience of the researcher. BioNetXplorer does not have strong visualization tools, but does export in formats to use the powerful visualization tool Cytoscape.

[1] http://hadoop.apache.org/

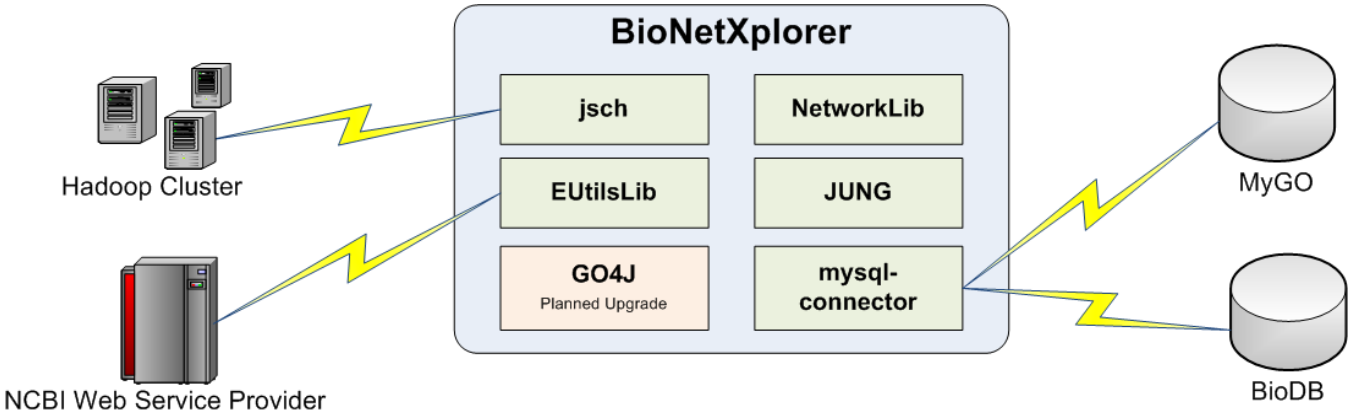

Figure 4: System Overview

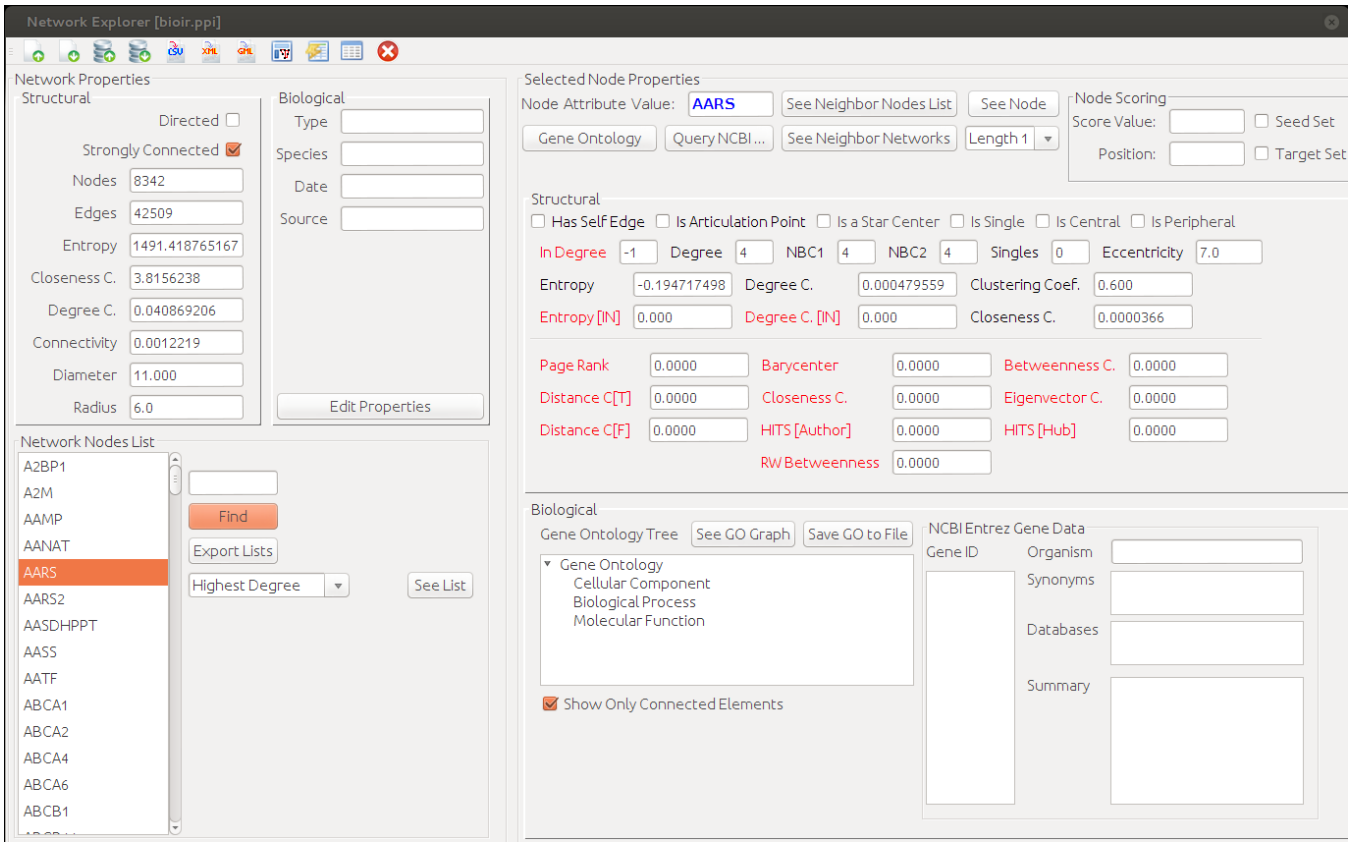

Figure 5: Network Exploration Main Interface

4. REAL WORLD APPLICATIONS

BioNetXplorer capabilities have been applied for diverse topological and biological analysis in past research, for instance (Yeh, Liu, Yeh, & Soo, 2010) mined the nodes structural properties in order to discover important or relevant nodes within subnetworks in their prostate cancer study, (Yeh, 2010) also employed this application to compute all available structural node properties with the purpose of enriching with data the networks in his study,

moreover he also used the shortest path subnetwork display in his pathways analysis to extract significant cancer related subnetworks. Another application of this tool is in the Disease Gene Prioritization study for prostate cancer project, where we have applied with results the Shortest Path Scoring functionality of this tool. The results where coherent with previously found cancer related genes, and provided some new genes for further exploration according to our published work (Arias, Yeh, & Soo, Disease Gene Prioritization, 2011) (Arias, Yeh, & Soo, 2012). An overview of this application can be found in: http://www.youtube.com/user/BioNetExplorer.

5. CONCLUSIONS

BioNetXplorer has proven to be a useful tool for biological networks analysis in related ongoing and past research projects, having the advantange that can export annotated networks in XGMML format, thus allowing these networks to be further analized using Cytoscape or any other external tools. Since BioNetXplorer is a standalone application it could be ran without having other applications in memory, allowing a more efficient use of computer resources. Furthermore, besides having the Single Network Exploration capability where the user can explore both structural and biological properties of the nodes of the network, the application provides Shortest Paths Analysis, and even a Gene Scoring functionality; the later possessing a built in faculty to connect to a Hadoop Cluster improving the time performance of the gene scoring computation.

REFERENCES

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). *Data Structures and Algorithms.* Massachusetts, MA: Addison-Wesley.

Arias, C. R., Yeh, H.-Y., & Soo, V.-W. (2012). Biomarker identification for prostate cancer and lymph node metastasis from microarray data and protein interaction network using gene prioritization method. *The Scientific World Journal* .

Arias, C. R., Yeh, H.-Y., & Soo, V.-W. (2011). Disease Gene Prioritization. In X. Xia, *Bioinformatics.* Rijeka, Croatia: INTECH.

Assenov, Y., Ramirez, F., Schelhorn, S.-E., Lengauer, T., & Albrecht, M. (2008). Computing topological parameters of biological networks. *Bioinformatics* , 282-284.

Brandes, U. (2001). A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology , 25* (2), 163-177.

Freeman, L. C. (1979). *Social Networls , 1* (3), 215-239.

Gene Ontology Consortium. (2010, 7). Gene Ontology Database. http://www.geneontology.org/GO.downloads.database.shtml.

Himsolt, M. (1996). *GML: Graph Modelling Language.* Retrieved from Projects of Theoretische-Informatik: http://www.fim.uni-passau.de/fileadmin/files/lehrstuhl/brandenburg/projekte/gml/gml-documentation.tar.gz

Hu, Z., Mellor, J., Wu, J., Yamada, T., Holloway, D., & DeLisi, a. C. (2005). VisANT: data-integrating visual framework for biological networks and modules. *Nucleic Acids Research* .

Maglott, D., Ostell, J., Pruitt, K., & Tatusova, T. (2005). Entrez Gene: gene-centered information at NCBI. *Nucleic Acid Research* .

O'Madadhain, J., Fisher, D., & Nelson, T. (2010). Java Universal Network/Graph Framework [Software]. *JUNG* . http://jung.sourceforge.net/.

O'Madadhain, J., Fisher, D., White, S., & Boey, Y.-B. (2003). *The jung (java universal network/graph) framework.* University of California, School of Information and Computer Science, Irvine.

Punin, J., & Krishnamoorthy, M. (2010). XGMML (eXtensible Graph Markup and Modeling Language). *XGMML* . http://cgi5.cs.rpi.edu/research/groups/pb/punin/public_html/XGMML/draft-xgmml.html.

Rapidminer. (2014). Rapidminer [Software]. http://www.rapidminer.com/.

Scardoni, G., Petterlini, M., & Laudanna, C. (2009). Analyzing biological network parameters with CentiScaPe. *Bioinformatics* , 2857-2859.

Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., et al. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *GENOME RESEARCH* , 2498--2504.

Shetty, J., & Adibi, J. (2005). Discovering important nodes through graph entropy the case of enron email database. *LinkKDD '05: Proceedings of the 3rd international workshop on Link discovery* (pp. 74-81). NY: ACM.
Simonyi, G. (1995). Graph Entropy: A survey. *Combinatorial Optimization , 20*, 399-441.
Tian, Y., & Patel, J. M. (2008). TALE: A Tool for Approximate Large Graph Matching. *International Conference on Data Engineering* (pp. 963--972). Los Alamitos, CA, USA: IEEE Computer Society.
Yeh, H.-Y. (2010). *Identification of Protein Complexes and Biological Regulation and Signal Networks Using Multiple Biological Databases and Microarray Data.* Hsinchu: National Tsing Hua University.
Yeh, H.-Y., Liu, Y.-Y., Yeh, C.-Y., & Soo, V.-W. (2010). Identifying Prostate Cancer-Related Networks from Microarray Data. *BioInformatics and BioEngineering (BIBE), 2010 IEEE International Conference on*, (pp. 302--303).

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.