NetAna, an application for undirected network analysis

Aretas Gaspariunas, Faculty of Biology, Medicine and Health, The University of Manchester, 2017.

Abstract

The NetAna application is a software tool for network analysis. The application allows the user to perform some of the very principle and basic methods of network analysis. The NetAna application was built using a general-purpose computer programming language Java 8 and implements a user-friendly Graphical User Interface (GUI) for a sensible and straightforward communication between the user and the program.

Introduction

Networks are used to represent many different types of biological data and are formed between a set of interconnected points (nodes) known as edges (Ma'ayan, 2011).

As part of the Java project to perform a network analysis, the NetAna application was developed. The application is capable performing the principle methods of network analysis including: finding a hub, calculating degree distribution, average degree and adding new interaction to the network. The development of project was partly inspired by SNAVI desktop application (Ma'ayan et al., 2009)

Capabilities of the NetAna

The NetAna software tool allows the user to performs multiple methods of network analysis. In this section functions and properties of each of the methods are covered.

Creating a Network

The program allows uploading a simple text (.txt) input file containing a network given the layout of the text inside the file is correct. Each line in the file must represent an interaction between two nodes and the names of the nodes must be separated by tabulation. NetAna assumes that interactions in the network are undirected.

If the network from a file is created successfully a message about the status is displayed in the bottom left corner of the program and a list view on the left side is updated with all new interactions (Figure 1).

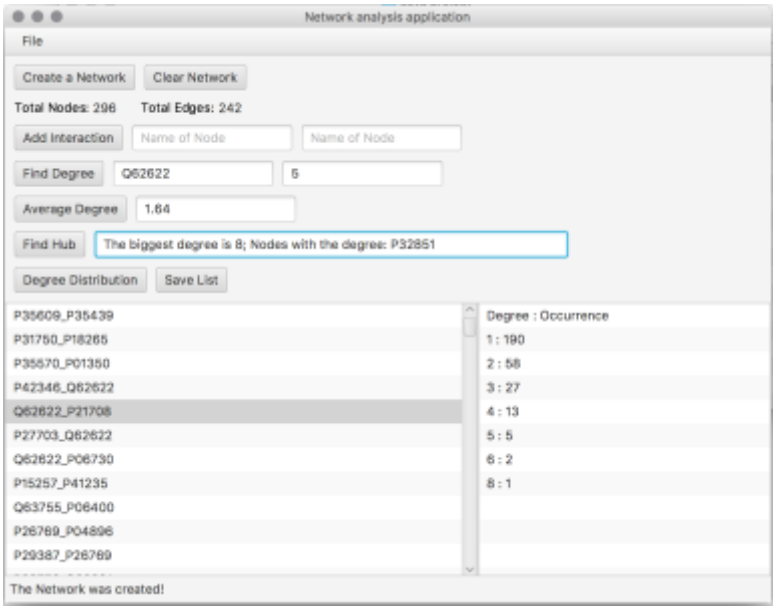

Figure 1. An example of the application in action. The left list view displays the list of all interaction found in the network; the right list view shows the degree distribution of the network; the bottom left corner displays the information about the status of the executed method and the program.

Clear Network

The Clear Network function of the program allows the user to clear the selection of the network and to start a new session if necessary without the need to restart the application. This button also clears both list views and creates a new empty network which can also be used to add new interaction at will.

Add/Remove an Interaction

The Add Interaction function allows the user to add a new interaction to an already existing network. Upon addition, the network is updated and the added interaction should be visible in the interaction list window on left. The method prevents the addition of an empty name or duplicates and displays an error message at the bottom left corner. The same description applies for the Remove Interaction function.

Find Degree

The Find Degree function allows the user to enter a name of the node of interest to find a degree number associated with it.

Average Degree

The Average Degree function when used inform about the total average degree of the network. The average degree is rounded to allow only two numbers after a decimal point.

Find Hub

The Find Hub function searches for a node or nodes with the highest degree in the network and prints out the number of that degree and all the hubs associated with it.

Degree Distribution

The Degree Distribution function calculates the degree distribution of the network and displays it for inspection on the right-side of the list view table.
A Save List button if necessary allows saving the degree distribution in a selected directory.

Save Network to a file

Save Network to a file function can be found under the File section in the Menu bar. This method permits saving of the network in a selected directory and is a useful feature allowing to save the work done on network analysis for future uses.

Algorithmic approach to building NetAna

NetAna was built using the Object-oriented programming approach as opposed to the classical top-to-bottom design. Originally the program consisted of 4 classes: Node, Edge and Network classes; their methods were used to create the objects of those classes. A node is simply a name of the protein or any other one unit of the network. An edge is an interaction between two nodes (Lu et al., 2007); A network consists of a list of all nodes and a list of all interaction between them with no duplications allowed (Pavlopoulos et al., 2011). All the analysis functions performed by NetAna application are methods of the network class. The 4^{th} class in the main program which runs the methods and performs the desired methods on the network. At the last stages of the application development, a GUI was created for a user-friendly experience and general sensibility. This resulted in an additional 5^{th} class and .fxml file. The 5^{th} class contains instructions for how the program should interact with GUI whereas the .fxml file stores information related to the layout and design of the GUI. After GUI was finalised the 4^{th} class was redesigned to launch the actual application window instead of running from the command-line interface.

Methods

NetAna was built using Java 8 programming language in Eclipse Integrated Development Environment (IDE) (Version: Neon.2 Release (4.6.2) Build id: 20161208-0600).
The results of the NetAna were compared to a Python 3 package NetworkX (Hagberg et al).
Tools JavaFX packages coupled with JavaFX Scene Builder 2.0 (Version: 2.0-b20) were utilised to design and create the GUI.
The final application was created by exporting the application package in Eclipse as a .jar type file.

Conclusion

NetAna is a Java built application which utilises several methods to perform an analysis of the network and interact with the user by a simple and sensible GUI design. This is in contract to a traditional command-line interface which often causes confusion when used by inexperienced users in the field of computer science.

Supplementary Material

Supplementary Material with the NetAna application and the full package with Java source code is provided together with this report.

Acknowledgements

NetAna is built and maintained by Aretas Gaspariunas at the University of Manchester. I am grateful for the support and advice provided while learning to code in Java from Dr Jean-Marc Schwartz and the rest of PGT Bioinformatics and Systems Biology group.

References

A Hagberg, D Schult, P Swart, *Exploring Network Structure, Dynamics, and Function using NetworkX* in *Proceedings of the 7th Python in Science conference (SciPy 2008)*, G Varoquaux, T Vaught, J Millman (Eds.), pp. 11-15

Lu, L., Sboner, A., Huang, Y., Lu, H., Gianoulis, T., Yip, K., Kim, P., Montelione, G., and Gerstein, M. (2007). Comparing classical pathways and modern networks: towards the development of an edge ontology. Trends In Biochemical Sciences *32*, 320-331.

Ma'ayan, A., Jenkins, S., Webb, R., Berger, S., Purushothaman, S., Abul-Husn, N., Posner, J., Flores, T., and Iyengar, R. (2009). SNAVI: Desktop application for analysis and visualization of large-scale signaling networks. BMC Systems Biology *3*, 10.

Ma'ayan, A. (2011). Introduction to Network Analysis in Systems Biology. Science Signaling *4*, tr5-tr5.

Pavlopoulos, G., Secrier, M., Moschopoulos, C., Soldatos, T., Kossida, S., Aerts, J., Schneider, R., and Bagos, P. (2011). Using graph theory to analyze biological networks. Biodata Mining *4*.