Applied Mathematics, 2014, 5, 3127-3134
Published Online November 2014 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2014.519296

Scientific Research

An Automatic Cross-System File Generation System for Biological Network Visualization Tools

Tianhan Zhang, Xun Lu*

School of Computer Science & Technology, Soochow University, Suzhou, China
Email: *luxun@suda.edu.cn

Received 24 August 2014; revised 20 September 2014; accepted 10 October 2014

Abstract

In the field of bioinformatics, the size of a biological network is usually very big. Without any help, it's extremely hard to analyze the network. If it is shown as a visualized picture, things will be easier. So it's very important to convert the biological network into a picture. However, there are a lot of software tools to be used to visualize the network. They use different file formats and do not support the transfer from one format to another. Sometimes it's really hard to deal with it. So I analyzed three text file formats of them (".dl", ".net" and ".vna") and developed a program to do this work automatically. The result of execution is very well and the efficiency is also impressive.

Keywords

Bioinformatics, Network, Visualization, Biological Tool

1. Introduction

A biological network is any network that applies to biological systems. A network is any system with sub-units that are linked into a whole, such as species units linked into a whole food web. Biological networks provide a mathematical analysis of connections found in ecological, evolutionary, and physiological studies, such as neural networks [1].

Nowadays, the research on the biological networks is very hot. However, a biological network usually has a large amount of nodes and edges and a frustratingly complex structure. With the larger and larger amount of data, the analysis of the networks is harder and harder. Facing them, no one will have any idea where to start. So if we can directly convert the data into a picture of the network, the analysis will be easy to perform. There are many

*Corresponding author.

How to cite this paper: Zhang, T.H. and Lu, X. (2014) An Automatic Cross-System File Generation System for Biological Network Visualization Tools. *Applied Mathematics*, **5**, 3127-3134. http://dx.doi.org/10.4236/am.2014.519296

tools are used to analyze the network, such as, Ucinet, Pajek, NetDraw and so on. They can process the data quickly, filter the information and combine the information to a well-organized web displayed on the screen [2].

Nevertheless, there are so many tools to be used to visualize network. Each of them has its advantages and disadvantages. It is really a hard decision for researchers to choose which one to use with. Sometimes we need to use software tools together to analyze one biological network, but none of the software supports the conversion between each format. It is not very convenient. If we can find a way to read original data, choose the format we want to generate and the program can automatically make the specific file, that will really help us save a lot of time. Here, my work is to analyze these software file format (only three of them here) and build such a program to analyze the original data and to create specific files we want. In brief, our program can export a certain file a time according to different software tools we want to use to analyze the biological network [3].

2. Software

2.1. Ucinet

Ucinet is a good tool to visualize and analyze biological networks. It can process up to 32,767 nodes in one network. But it's not free and when the amount of nodes is too large, it processes rather slow. Here we just use its file format rather than the software.

There are several formats for this kind of software and each of them is ok. We only talk about the "edgelist1" format below because it's more suitable for our situation here.

This kind of ".dl" format file is always started with a keyword "dl". It must be the first word of the file. Its structure is shown in **Figure 1** [4].

Following the word "dl" is the total amount of edges. For instance, "n = 5" means there are 5 edges listed at the end of the file. It could also be written as "n 5" or "n, 5".

The "format" indicates how the data is arranged at the end of the file. Here we use "edgelist1" format. It means the data is arranged as an edge list. There are various kinds of supported formats. The default value of "format" is "full matrix" as the data is arranged as a full matrix. And if it is "full matrix", we can leave out the word "format = full matrix". The value of this property could also be "linked list", "lower half matrix" and so on [5].

Then follows the word "labels:", which reflects the words below are a sequence of the node names. They are referred as their numbers in "data:" section [6].

At last, the word "data:" comes, which means the rest of the file is the data of an edge list. So this section could not be written before any other keywords and the sequence of the keywords should be paid enough attention. Besides, every value should be separated by at least one space or enter. In the "edgelist1" format, each line of data is an ordered pair of actors, optionally followed by a value indicating the strength of the relationship (or the possibility of the edge's existence). Each line of the data is an existing edge. The two terminal nodes of the edge are written as numbers which indicate the names below the "labels:" [7].

If "labels:" and its values are replaced by "labels embedded", that means we will define the node names in the "data:" part. So the numbers below the "data:" should accordingly be replaced by the names of the nodes [8].

Since we have known the specifications, we can start programming. We followed the above specifications and succeeded in creating a file with our program and data. We opened it in NetDraw and the result is shown in **Figure 2** [9].

2.2. Pajek

Pajek is a program, for Windows, for analysis and visualization of large networks having some thousands or

```
dl n=<the amount of edges>
format=<format to be used>
labels:
<label 1>, <label 2>, <label 3>, …
data:
<one node of a pair><another node of a pair>
<one node of a pair><another node of a pair>
<one node of a pair><another node of a pair>
…
```

Figure 1. File format of ".dl" file.

even millions of vertices. It's available for free at its home page. It's a part of Ucinet and it supports biological networks. It has some main functions listed in **Figure 3** [9].

Here we also just use its file format rather than the software.

There are also several formats supported by this kind of software. We only talk about one of them below again because it's more suitable for us.

This kind of ".net" file consists of two sections. One is "*Vertices" while the other is "*Edge". Its structure is shown in **Figure 4**.

A number follows the word "*Vertices" after a space. It indicates the number of nodes in this network. Then follows a list of node names. The first column is the No. of the name. The second column is the name of a node. There are also a lot of optional specific properties to set from the third column on. Here we don't talk about any of them because we don't need them [11].

The second section is a list of edges under the word "*Edges". Each line has a pair of node No. Each number refers the name associated with it in the "*Vertices" section. Like the ".dl" file, the third column can be the strength of the relationship (or the possibility of the edge's existence) if there is.

If there isn't any node name in the network, then we can leave out the whole node list.

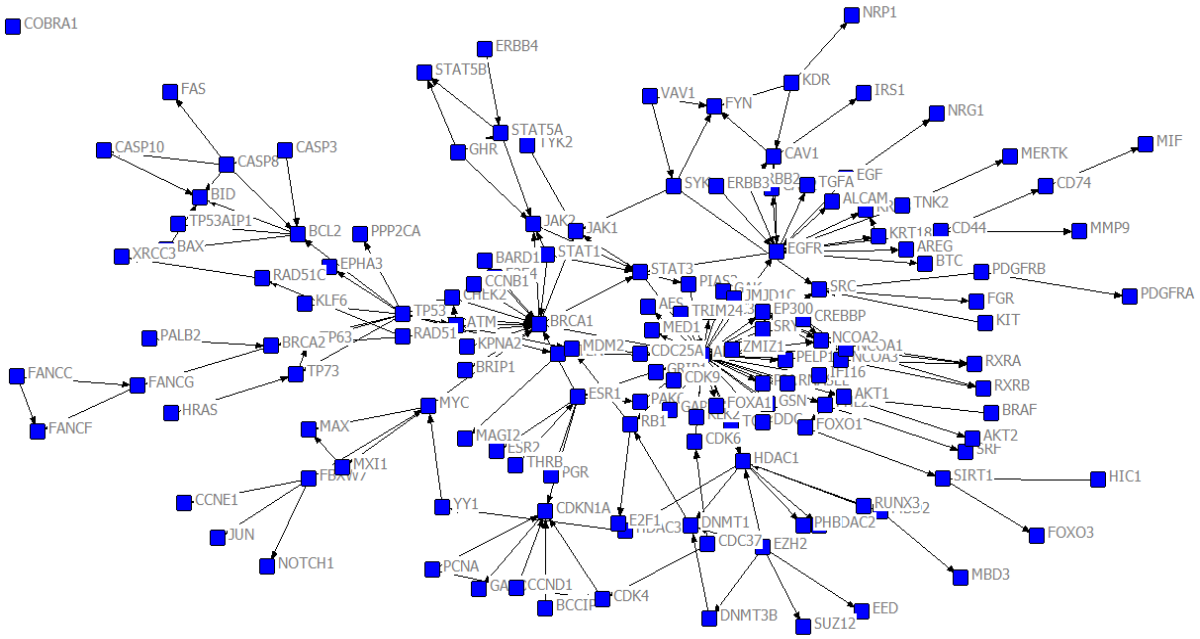

Figure 2. Result of Ucinet dl text file.

- support abstraction by (recursive) decomposition of a large network into several smaller networks that can be treated further using more sophisticated methods
- provide the user with some powerful visualization tools
- implement a selection of efficient (subquadratic) algorithms for analysis of large networks

Figure 3. Some main functions of Pajek.

```
*Vertices <number of vertices>
<No. of vertex 1><name of vertex 1>
<No. of vertex 2><name of vertex 2>
<No. of vertex 3><name of vertex 3>
…
*Edges
<one node No. of a pair><another node No. of a pair>
<one node No. of a pair><another node No. of a pair>
<one node No. of a pair><another node No. of a pair>
…
```

Figure 4. File format of ".net" file.

Since we have known the specifications, we can start programming. We followed the above specifications and succeeded in creating a file with our program and data. We opened it in NetDraw and the result is shown in **Figure 5** [12]. We use different color to mark the different kind of nodes in **Figure 5**.

2.3. NetDraw

NetDraw is a light-weight software to analyze biological networks. It's a part of Ucinet as well. We can use it to visualize a biological network and efficiently analyze them. It provides only a little function to analyze but it is free. What's more, it can process Ucinet dl text file, Pajek text file and its own vna text file. Here we use this software to test our program [13].

NetDraw's own ".vna" text file format also has some restrictions to obey. First, it is a text format file. So it makes everything easier to be done. Then comes the format requirements. Now we start with the overview of the data organization.

This kind of file has four sections: "*Node data", "*Node properties", "*Tie data" and "*Tie properties". There is no sequence requirement for these four sections.

In "*Node data" section, we can specify node information of the network like in **Figure 6**.

The word "*Node data" identifies the section as containing node data. The line following "*Node data" is a property name line. Among the properties, we must have one to act as a unique identifier to differ one node from another. Then comes several lines of node information. Not all the information should be written. All the values can be numeric or text and be enclosed in full quotes or not. But if there are spaces in between the text value, the text must be enclosed in full quotes.

All following lines are assumed to be node data until a new star command is read or the end of the file is reached, the same to other sections.

This section is not necessary.

In "*Node properties" section, we can specify node properties of the network like in **Figure 7**.

As before, the word "*Node properties" identifies the section as containing node properties. And the format requirements are similar to the "*Node data" section except that the first column name must be the ID defined in the "*Node data" section.

This section is also not necessary [14].

In "*Tie data" section, we can specify the relations between any pair of nodes in the network like in **Figure 8**.

It's also similar to the "*Node data" section. However, the first two column names must be "from" and "to"

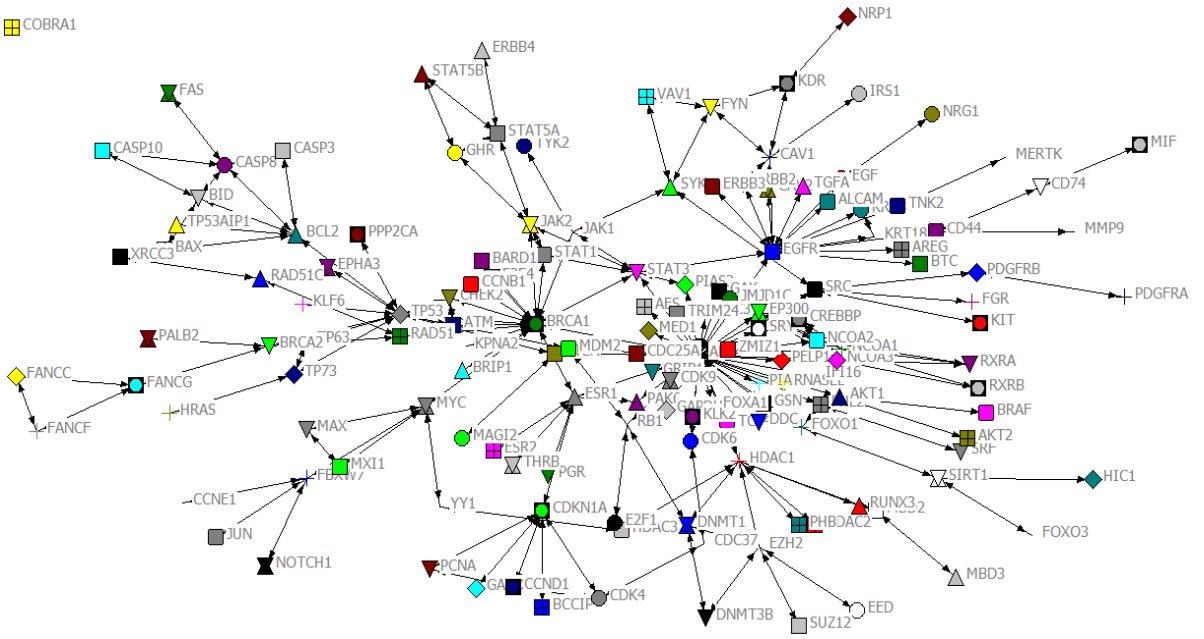

Figure 5. Result of Pajek text file.

```
*Node data
<property 1>, <property 2>, <property 3>, …
<value 1 of node 1><value 2 of node 1><value 3 of node 1> …
<value 1 of node 2><value 2 of node 2><value 3 of node 2> …
<value 1 of node 3><value 2 of node 3><value 3 of node 3> …
…
```

Figure 6. File format of ".vna" file ("*Node data" section).

```
*Node properties
<ID property>, <property 1>, <property 2>, …
<name of node 1><value 1 of node 1><value 2 of node 1> …
<name of node 2><value 1 of node 2><value 2 of node 2> …
<name of node 3><value 1 of node 3><value 2 of node 3> …
…
```

Figure 7. File format of ".vna" file ("*Node properties" section).

```
*Tie data
from to <relation1><relation2><relation 3> …
<name of one node><name of another node><value of relation 1><value of relation 2><value of relation 3> …
<name of one node><name of another node><value of relation 1><value of relation 2><value of relation 3> …
<name of one node><name of another node><value of relation 1><value of relation 2><value of relation 3> …
…
```

Figure 8. File format of ".vna" file ("*Tie data" section).

and the following column name(s) is the names of the relations. They use numerical values. The zero value of a relation represents the edge does not have this relation.

This section is a necessary part.

In "*Tie properties" section, we can add more information to the relations between any pair of nodes in the network. It's also an unnecessary part and here we don't need to use it. So no more explanation about it.

Since we have known the specifications, we can start programming. We followed the above specifications and succeeded in creating a file with our program and data. We opened it in NetDraw and the result is shown in **Figure 9** [15].

3. System Design

3.1. Preprocess

We should do some preprocess to make the file-creating part easier. It includes reading the original data and dividing the data into different graphs.

Input: the original data (a list of edges with the names of the two terminal nodes as the first two column and their possibilities of existence as the third column each line).

Output: a group of graphs.

Algorithm 1: see in **Figure 10**.

3.2. Create Ucinet dl Text File

According to the format requirements of ".dl" files and the preprocessed data, output the desired and well-formed file.

Input: a graph and a name of the generated file.

Output: a required file.

Algorithm 2: see in **Figure 11**.

3.3. Create Pajek Text File

According to the format requirements of ".net" files and the preprocessed data, output the desired and well-formed file.

Input: a graph and a name of the generated file.

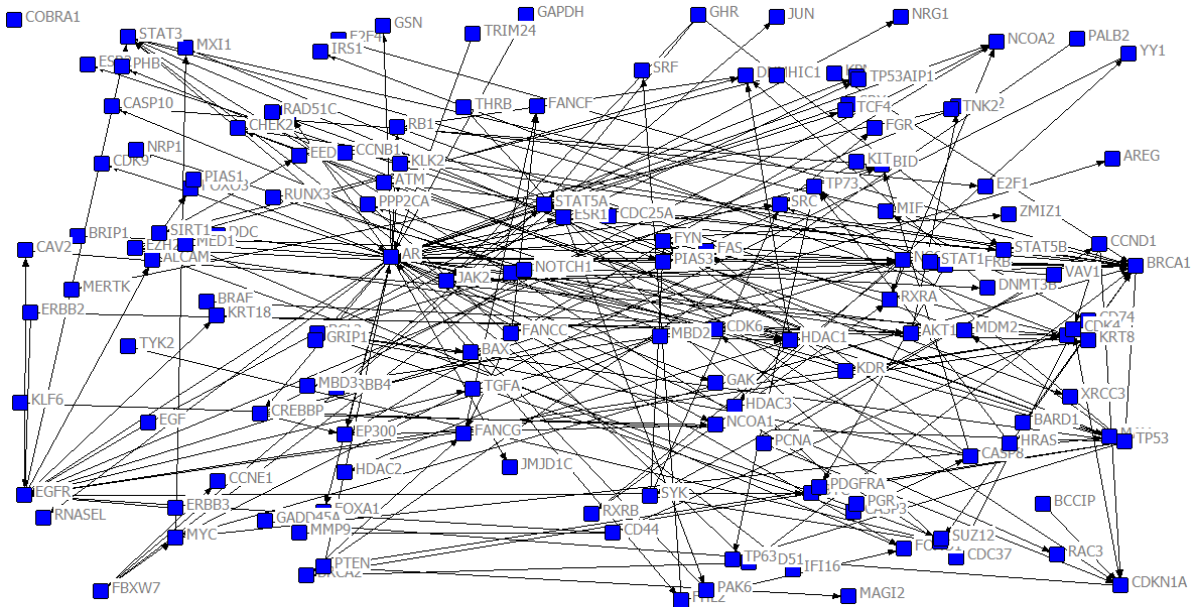

Figure 9. Result of vna text file.

Step 1: Open and read each line of the data file. Considering some possibilities are not an integer, enlarge all the possibilities to integers.
Step 2: Search in the list of edges. Classify them into different separate graphs.

Figure 10. Algorithm 1.

Step 1: Input the file name that the user want to use as the result file name.
Step 2: Create a file with the name.
Step 3: Write "dl n=" and the size of the list after it.
Step 4: Write "format = edgelist1\nlabels embedded:\ndata:".
Step 5: Write each edge of the required graph into the new ".dl" file with their possibilities.
Step 6: Save and close the file.

Figure 11. Algorithm 2.

Output: a required file.

Algorithm 3: see in **Figure 12**.

3.4. Create vna Text File

According to the format requirements of ".vna" files and the preprocessed data, output the desired and well-formed file.

Input: a graph and a name of the generated file.

Output: a required file.

Algorithm 4: see in **Figure 13**.

3.5. Organizing All the Parts

Organize all the algorithms above into a complete program.

Input: the original data, a choice of which graph to be turned into a file, a choice of which file format to be used and the name of the file.

Output: a required file.

Algorithm 5: see in **Figure 14**.

The system is designed by C++ language, and the environment is Windows 7 Home Basic x64 + Intel Core i3 2330 M 2.20 GHz + 4G RAM + Visual Studio 2005 + NetDraw v2.118. If chooses the ".vna" file format, we can get the result shown in **Figure 15** with our system.

Step 1: Input the file name that the user want to use as the result file name.
Step 2: Create a file with the name.
Step 3: Write all the nodes into the "*Vertices" section with the number of nodes written after the word "*Vertices". Note the format requirements.
Step 4: Write each edge of the required graph into the "*Edge" section with their possibilities. Note the format requirements and note that the names of the nodes should be replaced by the No. of the nodes.
Step 5: Save and close the file.

Figure 12. Algorithm 3.

Step 1: Input the file name that the user want to use as the result file name.
Step 2: Create a file with the name.
Step 3: Write "*Tie data\n from to possibility\n" (here we only need to write this section).
Step 4: Write each edge of the required graph into the new ".vna" files with their possibilities.
Step 5: Save and close the file.

Figure 13. Algorithm 4.

Step 1: Execute Algorithm 1.
Step 2: Input the user's choice among the graphs created.
Step 3: Input the user's choice about the desired file format.
Step 4: If the user chooses the ".dl" file format, then execute Algorithm 2. If not, go to Step 5.
Step 5: If the user chooses the ".net" file format, then execute Algorithm 3. If not, go to Step 6.
Step 6: If the user chooses the ".vna" file format, then execute Algorithm 4. If not, exit.

Figure 14. Algorithm 5.

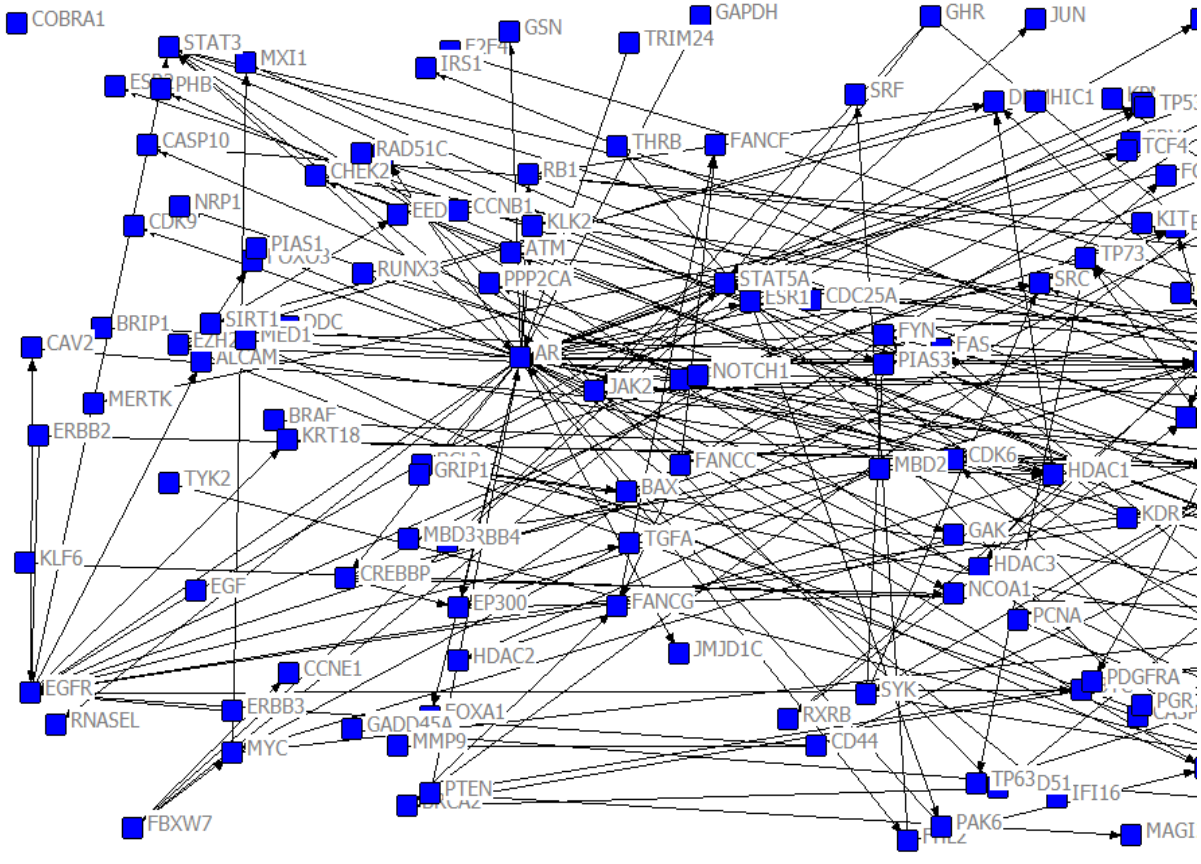

Figure 15. Result of vna text file.

4. Conclusions

Finally, we succeed in creating three different formats of files. From the pictures above, we can see that Pajek text file here can't specify the nodes' shape and colour because different nodes have different shapes. It's not the desired result. What's more, the display process of Ucinet dl text file is the best and the clearest while that of vna text file is the worst.

In future, I will do some further research on these formats, do some research on the display of these biological networks as well as continue doing something in the field of biological network analysis.

Acknowledgements

Tianhan Zhang, student ID Number: 1127403104, currently is an undergraduate student of Computer Science and Technology School of Soochow University. This work was directed by Xun Lu.

References

[1] Oliveira, M. and Gama, J. (2012) An Overview of Social Network Analysis. *Wiley Interdisciplinary Reviews*: *Data Mining and Knowledge Discovery*, **2**, 99-115.

[2] Brent, L.J.N., Lehmann, J. and Ramos-Fernández, G. (2011) Social Network Analysis in the Study of Nonhuman Primates: A Historical Perspective. *American Journal of Primatology*, **73**, 720-730. http://dx.doi.org/10.1002/ajp.20949

[3] Sueur, C., Jacobs, A., Amblard, F., Petit, O. and King, A.J. (2011) How Can Social Network Analysis Improve the Study of Primate Behavior? *American Journal of Primatology*, **73**, 703-719. http://dx.doi.org/10.1002/ajp.20915

[4] Jacoby, D.M.P., Brooks, E.J., Croft, D.P. and Sims, D.W. (2012) Developing a Deeper Understanding of Animal Movements and Spatial Dynamics through Novel Application of Network Analyses. *Methods in Ecology and Evolution*, **3**, 574-583. http://dx.doi.org/10.1111/j.2041-210X.2012.00187.x

[5] Thompson, R.M., Poulin, R., Mouritsen, K.N. and Thieltges, D.W. (2013) Resource Tracking in Marine Parasites: Going with the Flow? *Oikos*, **122**, 1187-1194. http://dx.doi.org/10.1111/j.1600-0706.2012.00245.x

[6] Del Rosso, C. (2009) Comprehend and Analyze Knowledge Networks to Improve Software Evolution. *Journal of Software Maintenance and Evolution*: *Research and Practice*, **21**, 189-215. http://dx.doi.org/10.1002/smr.408

[7] Lusseau, D., Wilson, B., Hammond, P.S., Grellier, K., Durban, J.W., Parsons, K.M., Barton, T.R. and Thompson, P.M. (2006) Quantifying the Influence of Sociality on Population Structure in Bottlenose Dolphins. *Journal of Animal Ecology*, **75**, 14-24. http://dx.doi.org/10.1111/j.1365-2656.2005.01013.x

[8] Calero Medina, C.M. and van Leeuwen, T.N. (2012) Seed Journal Citation Network Maps: A Method Based on Network Theory. *Journal of the American Society for Information Science and Technology*, **63**, 1226-1234. http://dx.doi.org/10.1002/asi.22631

[9] Ashton, W. (2008) Understanding the Organization of Industrial Ecosystems. *Journal of Industrial Ecology*, **12**, 34-51. http://dx.doi.org/10.1111/j.1530-9290.2008.00002.x

[10] Terrell, J.E. (2010) Social Network Analysis of the Genetic Structure of Pacific Islanders. *Annals of Human Genetics*, **74**, 211-232. http://dx.doi.org/10.1111/j.1469-1809.2010.00575.x

[11] Ashton, W.S. and Bain, A.C. (2012) Assessing the "Short Mental Distance" in Eco-Industrial Networks. *Journal of Industrial Ecology*, **16**, 70-82. http://dx.doi.org/10.1111/j.1530-9290.2011.00453.x

[12] Kanngiesser, P., Sueur, C., Riedl, K., Grossmann, J. and Call, J. (2011) Grooming Network Cohesion and the Role of Individuals in a Captive Chimpanzee Group. *American Journal of Primatology*, **73**, 758-767. http://dx.doi.org/10.1002/ajp.20914

[13] Salter-Townshend, M., White, A., Gollini, I. and Murphy, T.B. (2012) Review of Statistical Network Analysis: Models, Algorithms, and Software. *Statistical Analysis and Data Mining*, **5**, 243-264. http://dx.doi.org/10.1002/sam.11146

[14] Belaire, J.A., Dribin, A.K., Johnston, D.P., Lynch, D.J. and Minor, E.S. (2011) Mapping Stewardship Networks in Urban Ecosystems. *Conservation Letters*, **4**, 464-473. http://dx.doi.org/10.1111/j.1755-263X.2011.00200.x

[15] Márquez-Serrano, M., González-Juárez, X., Castillo-Castillo, L.E., González-González, L. and Idrovo, A.J. (2012) Social Network Analysis to Evaluate Nursing Interventions to Improve Self-Care. *Public Health Nursing*, **29**, 361-369. http://dx.doi.org/10.1111/j.1525-1446.2012.01014.x

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or Online Submission Portal.