Check for updates

SOFTWARE TOOL ARTICLE

Selecting relevant nodes and structures in biological networks. BiNAT: a new plugin for Cytoscape [version 1; peer review: 1 approved, 1 approved with reservations, 1 not approved]

Fabio Cumbo, Giovanni Felici, Paola Bertolazzi

Institute for Systems Analysis and Computer Science "Antonio Ruberti", Italian National Research Council - CNR, Rome, 00185, Italy

V1 First published: 21 Nov 2014, 3:287
<https://doi.org/10.12688/f1000research.5753.1>

Latest published: 21 Nov 2014, 3:287
<https://doi.org/10.12688/f1000research.5753.1>

Abstract

Summary: In order to understand a network function, it's necessary the understanding of its topology, since the topology is designed to better undertake the function, and the efficiency of network function is influenced by its topology. For this reason, topological analysis of complex networks has been an intensely researched area in the last decade.

Results: Here we propose BiNAT, a Cytoscape [1] plugin able to perform network analysis, providing a full set of useful tools to discover the most significant nodes and structures in a network.

Conclusions: The plugin has been approved on the official Cytoscape plugins repository and it is downloadable directly from this site: <http://dmb.iasi.cnr.it/binat.php> where a full guide is also available.

Keywords

graph theory; biological networks; network analysis; cytoscape plugin

Open Peer Review

Approval Status

	1	2	3
version 1
21 Nov 2014	view	view	view

1. **Emidio Capriotti**, University of Alabama at Birmingham, Birmingham, USA
2. **Hagen Blankenburg**, European Academy Bozen/Bolzano (EURAC), Bolzano, Italy
3. **Vivek Anantharaman**, National Institutes of Health, Bethesda, MD, USA

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Fabio Cumbo (fabio.cumbo@hotmail.it)

Competing interests: No competing interests were disclosed.

Grant information: FC thanks the National Research Council of Italy (CNR) for the financial support.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2014 Cumbo F *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: Cumbo F, Felici G and Bertolazzi P. **Selecting relevant nodes and structures in biological networks. BiNAT: a new plugin for Cytoscape [version 1; peer review: 1 approved, 1 approved with reservations, 1 not approved]** F1000Research 2014, 3:287 <https://doi.org/10.12688/f1000research.5753.1>

First published: 21 Nov 2014, 3:287 <https://doi.org/10.12688/f1000research.5753.1>

Background

In this section we briefly introduce the fundamental concepts that are needed to understand the principles of the software described in this article. We introduce the main theoretical concepts used to describe and analyze networks, most of which come from graph theory that is a large field containing many results and we describe only a small fraction of those here, focusing on the ones that are relevant to the study of real-world networks.

Graphs and their representation

To represent a graph there are many different ways in literature. A graph is usually represented as an *adjacency matrix*. The adjacency matrix A of a simple graph is the matrix with elements a_{ij} , where $a_{ij} = 1$ if an edge between vertices i and j is present, and 0 otherwise. We note that a) for a graph with no self-edges the elements in the matrix diagonal are all zero, b) the adjacency matrix is symmetric (if there is an edge between i and j then there is an edge between j and i). In some situations it is useful to associate a weight to the edges; such *weighted* (or *valued*) *graphs* can be represented by giving the elements values of the adjacency matrix equal to the weights of the corresponding connections.

Centrality definition

Network centrality measures allow categorizing nodes for their relevance in the network structure. The literature on biological networks are typically interested in the global properties of the network; in the case of centrality it is often considered from a global point of view, as for example analyzing degree or centralities distribution²⁻⁶. In BiNAT were implemented the most used network centrality measures i.e. Degree, Betweenness, Eccentricity, Closeness and Stress centrality.

Software overview

The increasing availability of large network datasets, along with the progress in experimental high-throughput technologies, have promoted the need for tools that allow an easy integration of experimental data with data derived from network computational analysis. In order to enrich experimental data with network topological parameters, it has been developed the Cytoscape plugin BiNAT (Biological Network Analysis Tool).

The plugin computes several network centrality parameters, and allows the user to analyze these computational results in textual and graphical format. BiNAT identifies the nodes that are relevant from the experimental and the topological viewpoint. BiNAT is one of the few Cytoscape plugins that computes several centrality indices at once. In BiNAT the centrality measures can be easily correlated with each other, in order to identify the most significant nodes according to topological properties. Functional to this capability is the scatter plot options, which allows an easy correlation of node centralities.

Equally to other Cytoscape plugins, BiNAT (and its dependencies) must reside in the *plugin* directory in the Cytoscape root folder. BiNAT needs to be executed with administrative privileges to work properly. After first running, BiNAT creates three folders (*binatData*, *binatLog* and *binatTemp*) in the Cytoscape root directory:

- in *binatLog* folder application log files reside in two formats (.html and .txt), both useful to monitor the proper running of all software operations.
- in *binatData* folder generated application data reside such as: a) *stats.xlsx* file that contains network information (centrality measures for every nodes, network diameter, network density, node average degree, network maximal clique degree, etc). b) *complexesIntersectionMatrix.xlsx* file that represents the complexes adjacency matrix computed according to the number of shared nodes between complexes. c) *complexesShortest-PathsMatrix.xlsx* file that represents the complexes adjacency matrix computed according to the shortest path among all shortest paths between two nodes that are members of the same complex. d) *shortestPaths.txt* file that contains all pairs shortest paths in the network, useful to accelerate many other plugin features.
- in *binatTemp* some plugin temporary files reside such as: a) *sp_*.txt* files that contain all pairs shortest paths in the network. The maximal size of every *sp_*.txt* file is 50MB and their creation may take a long time depending on the network size. b) *toResume.txt* file that contains two indexes necessary to resume the process of calculating shortest paths if it has been stopped.

Implementation

The idea behind the developed interface consists of a clear separation from the visualization of results and the Java classes that contain the processing logic. The structure is comparable to the MVC (Model - View - Controller) design pattern that is based on the separation of roles between the software components that interpret three major roles:

- **Model:** application data and rules.
- **View:** it can be any output representation of data, such as a chart or a diagram.
- **Controller:** it mediates input, converting it to commands for the model or view.

The *Controller* is the key class that guides the input information flow (commands and parameters) towards the right actions to be performed.

Implemented algorithms

In this section the algorithms implemented in BiNAT are introduced through a pseudocode and a comprehensive description for each algorithms. The following algorithms have been implemented: Dijkstra algorithm, Betweenness centrality, Closeness centrality, Degree centrality, Eccentricity centrality, Stress centrality and Clique Finder (Bron-Kerbosch) algorithm.

Dijkstra algorithm

Dijkstra's algorithm⁷ solves the single-source shortest path problem when all edges have non-negative weights. It is a greedy algorithm, similar to Prim's algorithm, but the two solve different types

of problems and the properties are computed in different ways. Algorithm starts at the source vertex s and grows a tree T that ultimately spans all vertices reachable from s . Vertices are added to T in order of distance i.e., first s , then the vertex closest to s , then the next closest, and so on. The following implementation (see Listing 1) assumes that the graph G is represented by adjacency lists. As already mentioned, all centrality measures available in BiNAT, excluded the Degree centrality, depend on the Dijkstra's shortest path algorithm.

Listing 1. Pseudocode of the Dijkstra's shortest path algorithm.

```
dijkstra (G, w, s)
1. initialize_single_source (G, s)
2. S ← { } // S will ultimately contains
   vertices of final shortest-path weights
   from s
3. initialize_property_queue Q
4. Q ← V[G]
5. while Q is not empty do
6.   u ← extract_min(Q) // pull out new
   vertex
7.   s ← S ∪ {u} // perform relaxation for
   each vertex v adjacent to u
8.   foreach v in Adj[u] do
9.     relax(u, v, w)
```

Clique Finder (Bron-Kerbosch) algorithm

The Clique Finder algorithm that we use was developed by Bron & Kerbosch (1973)⁸. This algorithm combines a recursive backtracking procedure with a branch and bound technique to eliminate searches that cannot lead to a clique. The recursive procedure is self-referential: finding a clique of length n is accomplished by finding a clique of length $n-1$ and another node that is connected to all the nodes in that clique. The branch and bound technique makes use of rules that allow us to determine in advance certain cases for which possible combinations of nodes and edges will never lead to a clique. There are three sets that are essential for this algorithm:

- “*potential-clique*”: this is a set of nodes where every node is connected to every other node. Each recursive call will either extend this set by one node or reduce it by one node.
- “*candidates*”: this is the set of nodes that are eligible for addition to the “*potential-clique*” set.
- “*already-found*”: this is the set of nodes that have already served as an extension to the present configuration of “*potential-clique*” and are now explicitly excluded. That is, all possible extensions of “*potential-clique*” containing any point in this set have already been generated.

The algorithm operates recursively on each of the sets by generating all extensions of a given configuration of “*potential-clique*” that use given set of “*candidates*” and that do not contain any of the nodes in “*already-found*”, as described in the simplified pseudocode represented in Listing 2. Initially, the set “*candidates*” contains all the nodes in the graph and the set of “*potential-clique*” and “*already-found*”

are empty. Bron & Kerbosch adopt a clever strategy to select the nodes: to choose nodes with the largest number of edges, in order to reach the branch and bound condition as soon as possible. This leads to the larger cliques being found first and sequentially generates cliques having a large common intersection. More details of this algorithm, including a more detailed pseudocode, implementation are given by⁸.

Listing 2. Pseudocode of the Bron-Kerbosch algorithm.

```
bron_kerbosch_clique_finder (potential-clique,
                             candidate, already-found)
1. if a node in already-found is connected to
   all nodes in candidates then
2.   no clique can ever be found (branch and
   bound step)
3. else
4.   foreach candidate-node in candidates do
5.     move candidate-node to
     potential-clique
6.     create new-candidates by removing
     nodes in candidates not connected to
     candidate-node
7.     create new-already-found by removing
     nodes in already-found not connected to
     candidate-node
8.     if new-candidate and
     new-already-found are empty
9.       potential-clique is a maximal-clique
10.    else
11.      bron_kerbosch_clique_finder
      (potential-clique, new-candidates,
      new-already-found)
12.    endif
13.    move candidate-node from
     potential-clique to already-found
14.  endfor
15. endif
```

Supported features

One of the most important commands in BiNAT is the one for the creation of the principal output file (in MS Excel or CSV format), in which all network nodes with all their own centrality measures defined before are listed: the **ranking** measure has been introduced to find a correlation between centrality measures that are in conflict with each other; it is a value ranging from 0 to 10, where highest value are received by nodes that are candidates to be considered hubs. This file creation step passes through almost all stages of operational calculus which constitute the heart of the software. The first step consists is the input of a network. The network must be in TXT format according to the TAB2 standard. The format supported until now is that adopted by BioGRID. To compute all nodes and network centrality measures, it is needed to type the “makestatistics” command in the plugin text field (please see the software manual on the official site for more information about available commands). BiNAT will then performs the following steps:

- First of all BiNAT computes all pairs shortest paths in the network using Dijkstra algorithm⁷ (it is needed to compute almost all centrality measures);
- Once obtained all shortest paths, BiNAT computes all supported centrality measures for each node in the network.
- Then BiNAT assigns a ranking value to each node; such ranking is obtained calculated as the arithmetic average value of all centrality measure for each node. Once completed, all centrality measures for all nodes in the network, BiNAT computes other global measures such as the network density, the network diameter, the maximal clique degree and much more.

Data is ready to be returned in output now. BiNAT creates the output file in the folder specified at the beginning of the operation. To create XLSX files, BiNAT uses the Apache POI libraries: Java API for Microsoft Documents. The aim of the Java POI is to manipulate various file format based upon the Office Open XML standards (OOXML) and Microsoft's OLE 2 Compound Document format (OLE2). In short, the programmer is able to read and write MS Excel, MS Word and MS PowerPoint files using Java.

For a list of all supported command line options, please take a look at the User Manual on the plugin official site. BiNAT also provides a *server mode* that allows the user to fully remotely control the plugin. When the *server mode* is enabled, user can send commands remotely using *BiNAT client* available as desktop and mobile application (for Android devices only).

BiNAT at work

We tested BiNAT on *Saccharomyces Cerevisiae* (Figure 1) protein-protein interaction network. PPI network was extracted from BioGRID database and consist of 6,096 nodes and 214,235 interactions.

It was analyzed with information about yeast complexes provided by the Wodak Lab (Molecular Structure & Function program, Hospital for Sick Children, Toronto, ON, Canada). We first used BiNAT to compute all the centrality measures. A first overview of the global topological properties comes from the values of all centralities along with the diameter and the nodes average degree (Figure 2).

With this data and with the help of a graphical layout, we can deduce a highly connected network. The computation of network centrality measures allow us a first ranking of this network. Further analysis bases on a Gene Ontology database search, or by adding functional annotation data, may allow a deeper functional exploration of the network. The combination of BiNAT with other bioinformatics tools, such as *CentiScaPe* or *Network Analyzer* or *MCODE*, may help to analyze high-throughput genomic and or proteomic experimental data and facilitate the analysis process. It's recommended to see the plugin User Manual for a step-by-step guide on how to use BiNAT and all its features.

Figure 1. *S. Cerevisiae* represented in Cytoscape with biological layout.

#	Betweenness	Closeness	Degree	Eccentricity	Stress
POR2	3.69E-4	4.22E-5	12	0.16	11020
BNA7	1.67E-4	Infinity	7	Infinity	5016
POR1	9.95E-4	5.65E-5	79	0.2	29716
BNA6	4.31E-4	5.21E-5	18	0.2	12890
BNA5	4.19E-4	5.28E-5	29	0.2	12513
TFP1	0.003	7.32E-5	358	0.2	103752
ADR1	6.75E-4	6.12E-5	84	0.2	20176
TFP3	3.70E-4	4.16E-5	26	0.16	11055
OXA1	8.88E-4	5.41E-5	41	0.2	26546
OAF1	4.59E-4	5.57E-5	43	0.2	13707
...	...	...	...	...	...

Nodes Number:	6096
Maximal Clique Degree	34
Network Density	0.005
Network Diameter	8
Nodes Average Degree	70.287

Figure 2. Overview of the global topological properties for *S. Cerevisiae* PPI network.

Conclusions

The Cytoscape plugin BiNAT was designed to provide the user a powerful tool for an accurate analysis of networks centrality. The plugin interface is simple as a shell; a practical user manual can be downloaded from the official web site <http://dmb.iasi.cnr.it/binat.php>.

BiNAT plugin has been accepted by the Cytoscape community and is actually available for the download on the official plugins repository for the 2.8.3 version of Cytoscape.

Software availability

Software access

Software available from Cytoscape Plugins repository: http://chi-anti.ucsd.edu/cyto_web/plugins/ and search for *binat*

Latest source code

<http://dmb.iasi.cnr.it/binat.php>

Archived source code as at the time of publication

<http://dx.doi.org/10.5072/zenodo.12771>⁹

License

GNU General Public License v2.0

Author contributions

PB conceived research. FC developed software. FC, PB and GF analyzed data and results. All authors wrote the paper.

Competing interests

No competing interests were disclosed.

Grant information

FC thanks financial support from the National Research Council of Italy (CNR).

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

- Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks**. *Genome Res.* 2003; **13**(11): 2498–2504.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Jeong H, Tombor B, Albert R, *et al.*: **The large-scale organization of metabolic networks**. *Nature.* 2000; **407**(6804): 651–654.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Joy MP, Brock A, Ingber DE, *et al.*: **High-betweenness proteins in the yeast protein interaction network**. *J Biomed Biotechnol.* 2005; **2005**(2): 96–103.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Wagner A, Fell DA: **The small world inside large metabolic networks**. *Proc Biol Sci.* 2001; **268**(1478): 1803–1810.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Wuchty S, Stadler PF: **Centers of complex networks**. *J Theor Biol* 2003; **223**(1): 45–53.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Yamada T, Bork P: **Evolution of biomolecular networks: lessons from metabolic and protein interactions**. *Nat Rev Mol Cell Biol.* 2009; **10**(11): 791–803.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Dijkstra EW: **A note on two problems in connexion with graphs**. *Numerische Mathematik.* 1959; **1**(1): 269–271.
[Publisher Full Text](#)
- Bron C, Kerbosch J: **Algorithm 457: finding all cliques of an undirected graph**. *Commun ACM.* 1973; **16**(9): 575–577.
[Publisher Full Text](#)
- Cumbo F, Felici G, Bertolazzi P: **BiNAT: new plugin for Cytoscape**. *Zenodo.* 2014.
[Data Source](#)

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 18 February 2015

<https://doi.org/10.5256/f1000research.6152.r7588>

© 2015 Anantharaman V. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Vivek Anantharaman

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA

I reviewed this plugin as an end-user and as a biologist. I have created and analyzed many networks using Cytoscape. I installed the BiNat plugin, but unfortunately was not able to test its functionality because of a flaw I encountered in the initial step. I was unable to load my networks on to the plugin.

1. The author gives two formats that are compatible. My main problem with this is that I am already using cytoscape and I have uploaded a network on Cytoscape. I should be able to use the network I have already uploaded and working with, rather than loading it a second time on to the plugin. At the least it should take the same formats that I was able to load onto Cytoscape (which did not work) or the *.cys format that cytoscape saves. The "import network from table" feature of cytoscape takes tab delimited files. The plugin was unable to take the very same files. This is a major drawback and a fatal flaw in my opinion.
2. The second issue is the English in the paper and the manual. It needs extensive correction and reworking to be legible.

This may very well be a useful and effective plugin. But I was unable to test its functionality because of a) poor instructions in the paper/manual, b) inability in using existing cytoscape networks and formats (This is a major drawback, and the utility of a plugin which cannot act on a network loaded in the main cytoscape window or at least makes use of the same format is very limited).

I recommend that the author rework these issues and make the plugin more robust.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Reviewer Report 21 January 2015

<https://doi.org/10.5256/f1000research.6152.r7310>

© 2015 Blankenburg H. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hagen Blankenburg

Center for Biomedicine, European Academy Bozen/Bolzano (EURAC), Bolzano, Italy

The manuscript presents BiNAT, a Cytoscape plugin for computing a number of network centrality parameters. In the last years, topological network analyses have yielded interesting insights, thus this is an important area of research. However, I see a number of major issues with both the software and the accompanying manuscript that should be addressed.

- Similar plugins:

As the authors state, *"BiNAT is one of the few Cytoscape plugins that computes several centrality indices at once."* The other tools should not just be named at the end of the manuscript but should be properly cited and described in the introduction. [NetworkAnalyzer](#) is included as a core plugin in every Cytoscape installation; [CentiScaPe](#) is available via the Cytoscape App store and has been published in [F1000Research](#) as part of the [Cytoscape App collection](#). Although novelty is not a criterion for publication in [F1000Research](#), without a thorough comparison to these tools it is hard to tell what BiNAT can do that these tools cannot (e.g. if there are particular centrality measures that only BiNAT can compute).

- Cytoscape 2/3 installation procedure:

I had some difficulties trying to install BiNAT. CentiScaPe can be installed with a single click in the Cytoscape App store; why is this elegant option not available for BiNAT? I could only test BiNAT with the outdated Cytoscape 2.8, as in Cytoscape 3.2 for MacOS I could not find a "plugin" folder (and creating the folder with the BiNAT files inside did not have an effect). This made me wonder if BiNAT is only compatible with Cytoscape 2 and not with the recent Cytoscape 3 (the compatibility entry in the [plugin directory](#) and the missing entry in the [App store](#) suggest this)? While of course I would like to see compatibility with Cytoscape 3, it should at least be clearly stated if this is not the case in order to prevent confusion.

- Usability / user interface:

While the authors state that *"[t]he plugin interface is simple as a shell"*, a user that is familiar with the intuitive, few click interfaces of NetworkAnalyzer and CentiScaPe might find BiNAT's command line interface quite challenging and unintuitive. If there is a clear benefit of this command line over a graphical and clickable interface this should be described with good examples (e.g. does it allow to script and save a complex analysis?). As a minimum example, the commands required to compute the Yeast network mentioned in the manuscript could be provided. However, for the user that does not feel overly comfortable typing commands into a shell (and I assume that Cytoscape is used by quite a few of those), a graphical, clickable, and intuitive interface to BiNAT seems to be essential.

- File formats:

Why does BiNAT actually need the ability to read network files? Input file handling and network creation should be completely handled by Cytoscape, which has a general table import function that can handle any kind of plain text file, not just BioGRID Tab 2.0; plugins like BiNAT should work on the final networks and not replicated core functionality.

- Manuscript:

In its current form the manuscript contains too many technical descriptions. Implementation details like the model-view-controller design pattern, the description of BiNAT's internal directory structure and file organization, or what Apache module is used for writing Microsoft Excel files might be interesting for a developer but are irrelevant for the reader that is a potential user. All this text could be moved to the author's website, freeing plenty of space for a comparison to the other available network analysis tools and for practical use-cases and examples. In a final step, a language check should be performed to make sure the text is fully comprehensible.

- Figures:

Figure 1 does currently not contain much relevant information and should be replaced with a figure that is somehow connected to BiNAT. A classic approach would be to use the computed centrality parameters in Cytoscape's VizMapper to change the color/size of important nodes, ideally focusing on a subnetwork of the Yeast interactome. The [Cytoscape Tumblr](#) might work as a great source of inspiration for useful and aesthetically pleasing Cytoscape representations. In addition, Figure 2 should probably be Table 1 instead, as it only contains tabular text.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 04 December 2014

<https://doi.org/10.5256/f1000research.6152.r6765>

© 2014 Capriotti E. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Emidio Capriotti

Division of Informatics, Department of Pathology, University of Alabama at Birmingham, Birmingham, AL, USA

The paper briefly describes BINAT, a new Cytoscape plugin to calculate several network features. The plugin provides an interface to export the results of the calculation in an excel file.

The authors should have better described the advantages of using their plugin showing, for example, the computing time required to perform calculation on networks with different size.

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research