RESEARCH ARTICLE

CatbNet: A Multi Network Analyzer for Comparing and Analyzing the Topology of Biological Networks

Ehsan Pournoor, Naser Elmi and Ali Masoudi-Nejad*

Laboratory of Systems Biology and Bioinformatics (LBB), Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran

ARTICLE HISTORY

Received: August 11, 2018
Revised: October 19, 2018
Accepted: December 02, 2018

DOI:
10.2174/1389202919666181213101540

Abstract: ***Background*****:** Complexity and dynamicity of biological events is a reason to use comprehensive and holistic approaches to deal with their difficulty. Currently with advances in omics data generation, network-based approaches are used frequently in different areas of computational biology and bioinformatics to solve problems in a systematic way. Also, there are many applications and tools for network data analysis and manipulation which their goal is to facilitate the way of improving our understandings of inter/intra cellular interactions.

Methods**:** In this article, we introduce CatbNet, a multi network analyzer application which is prepared for network comparison objectives.

Result and Conclusion**:** CatbNet uses many topological features of networks to compare their structure and foundations. One of the most prominent properties of this application is classified network analysis in which groups of networks are compared with each other.

Keywords: Network biology, Topological features, Bioinformatics, Python, Network comparing, Biological researches.

1. INTRODUCTION

Nowadays, the great technological advances in sequencing methods make a ubiquitous shift of biological researches from traditional low-throughput methods to massively parallel techniques in the fields such as genomics and proteomics. But, at the same time the booming amount of generated data raises some problems in extracting and interpretation of meaningful information out of a wealth of raw data. On the other hand, it is a well-established fact that using network structure for representing the huge amount of biological data is not only useful but also vital in many cases. Generally, using networks to represent and analyze biological data is called network biology. In network biology, vertices of the network can be different elements such as genes, proteins, metabolites, disease and *etc*. Also edges between different nodes can be based on a wide spectrum of relations such as correlation, physical interaction, reaction and *etc*. For instance, in disease interactome network, nodes are proteins which are associated to that disease and edges between proteins are physical interaction between them.

Finding the most influential or important node in a huge biological network has been a vital question for many researchers in this field. It is noteworthy that, different terms such as hubs, key players, essentiality, lethality and centrality have been used to discuss this concept. Historically, the origin of this concept brings back to social science, but it has been extended to biology since previous decades [1-3]. Up to now, many different centrality measures has been generated to calculate the importance of each node within the network from different perspectives. For instance, in a great article by Jalili *et al.* [4] a wide variety of centrality measures has been gathered and described in a web server (CentiServer) which is accessible for everyone and also includes the most recent centrality measures. In fact, using this server or the R package of CentiServer, different centrality measures can be calculated for each network.

Due to the importance and ever-increasing usage of centrality measures in different applications such as biomarker discovery, disease pathogenesis discovery and drug repurposing, many packages and software have been developed to compute centrality measures for networks in the recent years. Undoubtedly, the java based software Cytoscape [5] is among the most prevalent and user-friendly software among biologists and bioinformaticians. Until now, many plugins have been generated for Cytoscape to broaden its ability for applications of network biology in computational biology. For example, Cytoscape plugin NetworkAnalyzer [6], computes and visualizes a comprehensive set of topological parameters such as network diameter, density, heterogeneity, clustering coefficient, shortest path lengths and *etc*. Another Cytoscape plugin is cytoHubba [7] which ranks nodes in a network by their features. Another plugin is CytoNCA [8] which supports eight different centrality measures and accept both weighted and unweighted biological networks as input. It is also possible to upload biological information for nodes/edges and integrate them by topological parameters of the network to detect specific nodes. In addition to these

*Address correspondence to this author at the Laboratory of Systems Biology and Bioinformatics (LBB), Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran; Tel: +98-21-6695-9256; Fax: +98-21-6640-4680; E-mail: amasoudin@ut.ac.ir

1389-2029/19 $58.00+.00

Cytoscape plugins, some standalone java programs such as CentiBiN [9] have been generated for ranking vital nodes of biological networks based on some centrality measures and visualizing the network and centrality measures distribution. Also, due to the pervasive usage of R language all over the world, many R packages have been developed in recent years to calculate network topological parameters and visualize networks in the great fashion. For instance, CentiServer [4] R package which has been introduced in 2015, can calculate 55 centrality measures in R environment or using its webserver (http://www.centiserver.org). It is worth mentioning that this webserver includes valuable and comprehensive information about more than a hundred centrality measures and available packages/tools to visualize or analyze networks. Recently another R package CINNA [10] was created to decipher the central informative nodes in the network by integrating different centrality measures. Newly, a comprehensive webtool which is called Network Analysis Provider (NAP) has been developed using R and Shiny to automate network construction and intra/inter network topology comparison between a pair of networks [11]. This webtool can rank nodes and edges based on different centrality measures and also it can be used to compare two networks and extract their intersection and provide high quality plots.

NetworkX [12], A library of Python is among the most powerful libraries to analyze big networks in considerable time. This package originally was born in 2002 by Hagberg *et al.*, but the first public release was in 2005. NetworkX is a package for creations, manipulation, analysis of the structure, dynamic and function of complex networks. Using this package, importing a wide variety of standard/nonstandard data formats can be carried out. Furthermore, generating different types of random and classic networks, analyzing their structure, designing new network algorithms, drawing networks and *etc*. are easily possible. Here, to develop CatbNet, we used Python language and version 1.9.1 of Networkx library for network manipulation and analysis.

As it has been mentioned until now, one of the important questions in network biology is how to find the most prominent or vital node. But, this is not the most essential application of network biology. In fact, different biological networks have different topological features which are the cornerstone of the new term in systems biology which is called network medicine [13, 14]. In network medicine, the features of biological networks such as human proteins interactome [15], human disease interactome network [16], PPI or gene co-expression network of disease stages can be widely used to segregate them to meaningful sets. For instance, researchers showed that, network features between case and control samples or different stages of tumors were significantly different [17-19]. Hence, some new studies have been mainly focused on applications of network mining and integrating it with machine learning methods for early diagnosis and also the prognosis the disease. For instance, Jalili *et al.* [20] used some centrality measures to discriminate Alzheimer's disease patients from healthy controls based on their EEG network using machine learning algorithms. More recently, in a great review [21], the applications of network-based measures and machine learning techniques for precision oncology has been studied. It has been stated that how graph theory algorithms in integration with omics data and machine learning techniques can be used for decoding tumor specific molecular mechanism, finding candidate targets and drug repositioning. In another study, Integration of network features, sequence features and functional features with classification algorithms carried out to identify novel Alzheimer genes [22].

Certainly, for acquiring an acceptable performance and accuracy for machine learning approaches, a large number of samples and intellectually selected features are needed. Hence, there is an unavoidable need for a network manipulation and analyzing tools to analyze a number of networks automatically and generate some statistical measures to select appropriately discriminating features among numerous number of topological features. Furthermore, that is not covered to anybody that, all of the researchers in this field are not mainly programming and network science experts; hence there is a clear need for a tool with a user friendly graphical user interface to facilitate usage of network biology power in biological problems in the omics era. To cover such requirements in network biology, we created a graph mining tool which is able to compute most important network topological features, from node-based measures such as: different centralities to whole network-based measures such as: network density, diameter, for analytical and comparative purposes. We developed CatbNet which contains graphical user interface for biologists with options to choose from input data to output result data. With the aim of CatbNet it is possible for users to analyze and compare multiple networks at once with further statistical and graphical analyses.

2. METHODS AND IMPLEMENTATION

2.1. Programming Language and Utilized Packages

Nowadays, most statistical tools and utilities are created using Very High-Level Programming Languages (VHLL). VHLLs are languages with strong abstraction of computer details and architecture. Programs written with VHLLs are mostly independent of a particular type of computer and easy to read, write and maintain. They are similar to human natural language and suitable for rapid application development. These languages are mainly used for specific tasks and purposes and often called domain-specific languages. In computational biology and bioinformatics researches, based on the goal and conditions, languages such as R, Perl and Python are used frequently to create packages and tools in the context of biology. Python (https://www.python.org/) is a very high-level programming language which is used in both academic and commercial applications. It is free, open source, platform independent, clean syntax, easy to learn, object-oriented and fast scripting language with large packages and libraries available for different areas of science. All these capabilities led us to use Python as our primary language and its libraries: Networkx [12], Numpy [23], Matplotlib [24], Pandas [25], Scipy [26], PyQt4 (https://pypi.python.org/pypi/PyQt4) and pandas to develop the CatbNet.

Networkx is a powerful library created in python for complex network generation, analysis and manipulation. We used Networkx for network handling operations such as network load and topological calculations. Scipy is a python package for mathematics, science and engineering. For statistical and mathematical operations like One-way ANOVA test we used Numpy and Scipy libraries. Matplotlib is another package which is developed for graphical visualization tasks. It is used for 2-D plotting of different charts, histograms and diagrams. In our project, to visualize network features boxplots we used matplotlib integrated with Pandas data structures. Also, for creating the graphical user interface of our application, PyQt4 is used. PyQt4 is a comprehensive set of Python bindings for Digia's Qt cross platform GUI toolkit.

2.2. Load and Initialization

In this application, we made the ability to load network data from four well-known network file formats; GML, GraphML, Edge list and Pajek. It is possible for users to import their data in these formats for further analysis. Imported networks could be weighted, directed or undirected networks, but in every runtime, it is recommended to import networks with the same properties (for example all networks be weighted). However, if there be multiple network types, only common features will be available in output results. In the load step, users must choose the directory in which all network files are stored. All files should be in the same file format and a user must select the format. In this phase, user must specify that networks are weighted, directed or not.

In bioinformatics and systems biology tasks, it may occur that users have groups of networks. For example, one group could be patients' reconstructed co-expression networks and the second could be networks of healthy control group. If user wants to explore these two groups, we created the capability of loading grouped networks. In this case, the option 'network files are classified in groups' must be checked. Furthermore, if grouped files are presented, the file naming convention should be observed. The file names must include the class name continued by three underscore characters and the network name (ex. Class1___network1, class2 ___network1). If file names do not obey this rule, networks will be considered alone with no grouping condition. Based on the files which are selected by the user, network data will be loaded and the analysis task will be initialized.

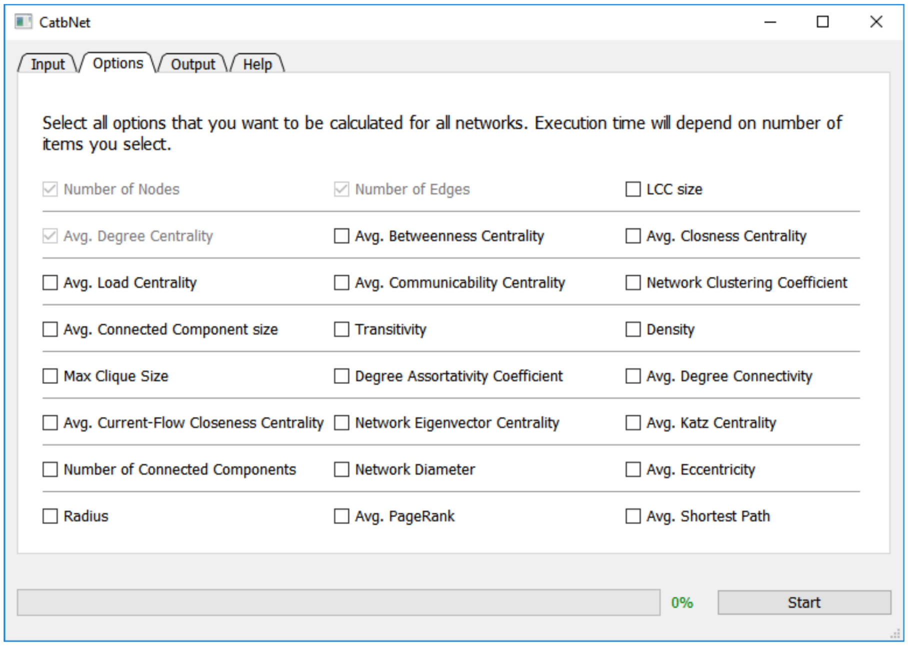
2.3. Topological Features of Networks

CatbNet calculates 24 network topological features (Fig. **1**) for every network and uses the result for its comparison. However, users can select among them for calculation based on prior knowledge or interest. Here we used many meaningful features in biological networks: Number of Nodes, Number of Edges, Largest Connected Component (LCC) size, Avg. Degree centrality, Avg. Closeness Centrality, Avg. Betweenness Centrality, Avg. Load Centrality, Avg. Communicability Centrality, Network Clustering Coefficient, Avg. Connected Component Size, Transitivity, Density, Max Clique size, Degree Assortativity Coefficient, Avg. Degree Connectivity, Avg. Edge Betweenness Centrality, Network Edge Connectivity, Avg. Katz Centrality, Number of Connected Components, Network Diameter, Avg. Eccentricity, Radius, Avg. PageRank and Avg. Shortest Path.

As mentioned before, if there be networks in different types (for instance: Multigraph, Directed Graph, Undirected Graph, Bipartite Graph and *etc*.) only common features will be computed and compared for these networks. Also, there may be cases in which some properties will not be available (for example, it is possible that network be disconnected and network diameter be infinite). In such cases, remainder networks which have valid values will be accounted for.

2.4. Examinations

The main goal of CatbNet is to compare many networks with each other and bold their differences. To achieve that, in this application, two different analyses are provided; ordinary and group analyses (next two subsections). Furthermore, there are two types of measures mentioned in the previous section: network-based measures (ex. Network den-



Select all options that you want to be calculated for all networks. Execution time will depend on number of items you select.

☑ Number of Nodes ☑ Number of Edges ☐ LCC size
☑ Avg. Degree Centrality ☐ Avg. Betweenness Centrality ☐ Avg. Closness Centrality
☐ Avg. Load Centrality ☐ Avg. Communicability Centrality ☐ Network Clustering Coefficient
☐ Avg. Connected Component size ☐ Transitivity ☐ Density
☐ Max Clique Size ☐ Degree Assortativity Coefficient ☐ Avg. Degree Connectivity
☐ Avg. Current-Flow Closeness Centrality ☐ Network Eigenvector Centrality ☐ Avg. Katz Centrality
☐ Number of Connected Components ☐ Network Diameter ☐ Avg. Eccentricity
☐ Radius ☐ Avg. PageRank ☐ Avg. Shortest Path

Fig. (1). All network features that are prepared in CatbNet. From the Options tab, the user can choose the desired network topological features.

sity, diameter) and node-based measures (ex. Node betweenness centrality, node degree). To compare networks with each other, all measures including network-based and node-based measures should be computed.

2.4.1. Ordinary Analysis

In the ordinary analysis, each network is compared with all other networks. To compare two networks, all measures should be calculated for both, then compare their values. Network-based measures are already comparable, but node-based measures are not. For this reason, we convert those to network-based measures by calculating the mean of node based features by arithmetic average. Suppose that *m* is a node-based measure. For network *N*, the resulted network-based measure m_N will be:

$$m_N = \frac{\sum_{1}^{n} m_i}{n}$$

Where m_i is node-based measure value for node *i* and *n* is the number of nodes in network *N*. By these average values, now it is possible to compare networks.

2.4.2. Group Analysis

If we import grouped data files and consequently have groups of networks (for example, group of patients PPI network and group of healthy PPI networks) and we want to check if there are differences in any of the measures between these two groups, we can use group analysis. In this case, we compare measures for one group against the other. Therefore, for network-based measures we should consider average of network values and for node-based values we consider average of average values. Consider *G* as a group including *p* number of networks, for network-based measure of *m*:

$$m_G = \frac{\sum_{i=1}^{p} m_i}{p}$$

Where m_i is network-based feature value of network *i* in group *G*. For the node-based measure *m'*, the mean within group of average node-based measure of the network must be calculated such as following:

$$m'_G = \frac{\sum_{j=1}^{p} \frac{\sum_{i=1}^{q} m'_{ij}}{q}}{p}$$

Where m'_{ij} is node-based value for node *i* in network *j, q* is the number of nodes in network *j* and *p* is the number of networks in group *G*. It is worth mentioning that the standard deviation for node-based and network based measures can be calculated simply.

2.5. Output

After CatbNet runtime, depending on the user selection, result files will contain: 1- a *.tsv (tab separated values) file for common node-based measures in all networks (Fig. **1**). In this file, for all nodes in all networks, node-based features values are presented. 2- Another *.tsv file for all features of networks including network-based and mean of node-based topological measures. In this file, for every network, all common measures are represented (Fig. **2**). It is possible to have both directed and undirected networks in the input file simultaneously. Therefore, to overcome inconsistency problems we calculate common attributes of networks. Using these files, users can compare networks. 3- Boxplots for each measure in common node-based features (Fig. **3**). CatbNet creates a boxplot for each feature in common node-based features of all networks. Boxplot is a meaningful graphical tool to show data dispersion of a feature. 4- Boxplots for all measures of groups. For every measure, data dispersion of group networks is plotted in a boxplot. 5- One-way ANOVA test for each node-based measure. For example, to examine if closeness centrality difference for networks is statistically significant or not, CatbNet provides a text file in which for all measures of node-based features One-way ANOVA test results are presented. 6- One-way ANOVA test for group comparison. Note that, before application execution, a user can specify which outputs are useful and interesting for further studies.

In order to clarify the steps in CatbNet, a schematic of it has been depicted in Fig. (**6**) as the following.

3. APPLICATIONS

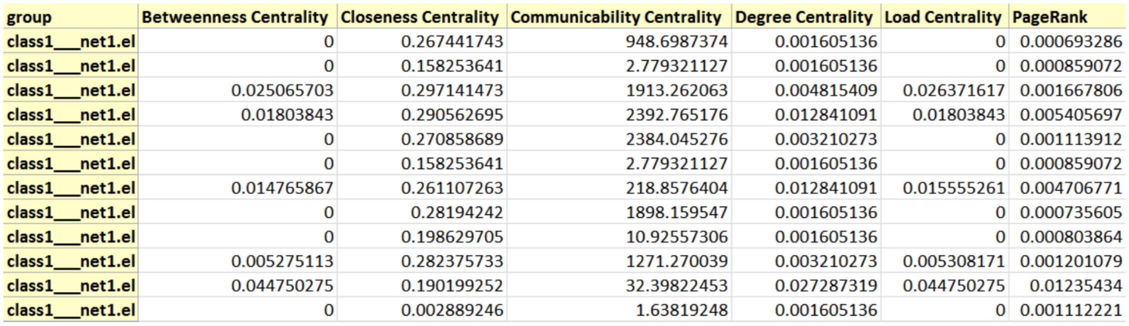
For case studies, the disease interactome network was utilized from [16] which contains subnetworks for diseases in the human interactome network. This set of network contains a diverse range of networks with different sizes which all are in the edge list format. For instance, the number of nodes was from 245 up to 3746 and the number of edges from 238 to 6548. These disease networks were divided into three distinct classes with different class numbers. This set of data is downloadable from the GitHub page of this application (https://github.com/LBBSoft/CatbNet/blob/master/test-data.rar.)

3.1. Extracting Topological Features for a Set of Networks

Through importing the input file, which contained a set of networks, and selecting the understudy list of features from the options tab of CatbNet (Fig. **1**), and running the program, the calculated features magnitudes attained. For our case study, a small part of the result has been illustrated in Figs. (**2** and **3**).

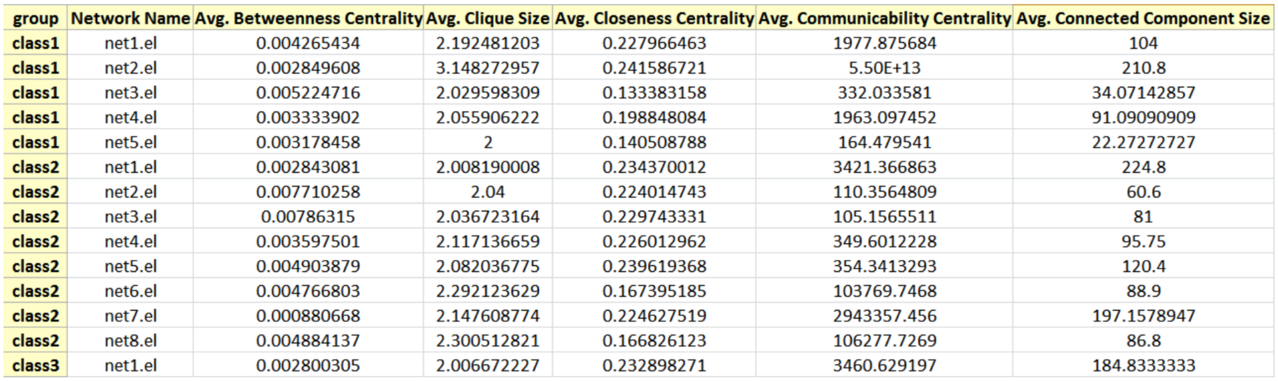
3.2. Comparing Different Networks or Network Groups based on Topological Feature

For our case study, in Fig. (**4**) the closeness centrality dispersion of 24 different networks was used to depict a boxplot. Based on this figure, closeness centrality distribution was different for those networks and the significance of the difference was tested by one-way ANOVA test of the CatbNet. Also, the distribution the four groups is plotted in Fig. (**5**) *via* boxplot. It is obvious that Class 4 has bigger values for Avg. Closeness Centrality.



group	Betweenness Centrality	Closeness Centrality	Communicability Centrality	Degree Centrality	Load Centrality	PageRank
class1___net1.el	0	0.267441743	948.6987374	0.001605136	0	0.000693286
class1___net1.el	0	0.158253641	2.779321127	0.001605136	0	0.000859072
class1___net1.el	0.025065703	0.297141473	1913.262063	0.004815409	0.026371617	0.001667806
class1___net1.el	0.01803843	0.290562695	2392.765176	0.012841091	0.01803843	0.005405697
class1___net1.el	0	0.270858689	2384.045276	0.003210273	0	0.001113912
class1___net1.el	0	0.158253641	2.779321127	0.001605136	0	0.000859072
class1___net1.el	0.014765867	0.261107263	218.8576404	0.012841091	0.015555261	0.004706771
class1___net1.el	0	0.28194242	1898.159547	0.001605136	0	0.000735605
class1___net1.el	0	0.198629705	10.92557306	0.001605136	0	0.000803864
class1___net1.el	0.005275113	0.282375733	1271.270039	0.003210273	0.005308171	0.001201079
class1___net1.el	0.044750275	0.190199252	32.39822453	0.027287319	0.044750275	0.01235434
class1___net1.el	0	0.002889246	1.63819248	0.001605136	0	0.001112221

Fig. (2). Node-based features of networks nodes in sample execution. In this file for each node in every network, common node-based features between all networks are represented (in the snapshot just some nodes of class1___net1 are observable).



group	Network Name	Avg. Betweenness Centrality	Avg. Clique Size	Avg. Closeness Centrality	Avg. Communicability Centrality	Avg. Connected Component Size
class1	net1.el	0.004265434	2.192481203	0.227966463	1977.875684	104
class1	net2.el	0.002849608	3.148272957	0.241586721	5.50E+13	210.8
class1	net3.el	0.005224716	2.029598309	0.133383158	332.033581	34.07142857
class1	net4.el	0.003333902	2.055906222	0.198848084	1963.097452	91.09090909
class1	net5.el	0.003178458	2	0.140508788	164.479541	22.27272727
class2	net1.el	0.002843081	2.008190008	0.234370012	3421.366863	224.8
class2	net2.el	0.007710258	2.04	0.224014743	110.3564809	60.6
class2	net3.el	0.00786315	2.036723164	0.229743331	105.1565511	81
class2	net4.el	0.003597501	2.117136659	0.226012962	349.6012228	95.75
class2	net5.el	0.004903879	2.082036775	0.239619368	354.3413293	120.4
class2	net6.el	0.004766803	2.292123629	0.167395185	103769.7468	88.9
class2	net7.el	0.000880668	2.147608774	0.224627519	2943357.456	197.1578947
class2	net8.el	0.004884137	2.300512821	0.166826123	106277.7269	86.8
class3	net1.el	0.002800305	2.006672227	0.232898271	3460.629197	184.8333333

Fig. (3). All common features of networks. Network-based features are calculated and node-based features are indicated as average values of nodes (for example Avg. Betweenness Centrality).

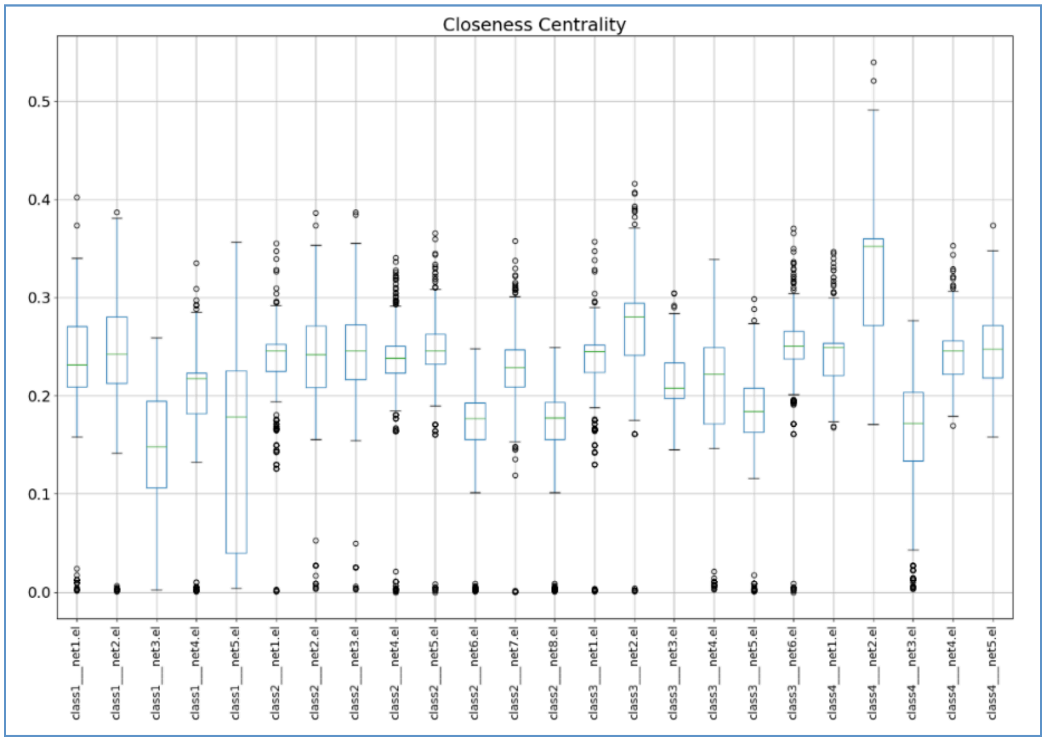


Fig. (4). Different networks data dispersion comparison for closeness centrality using boxplot representation. CatbNet creates such boxplot charts for every common node-based feature.

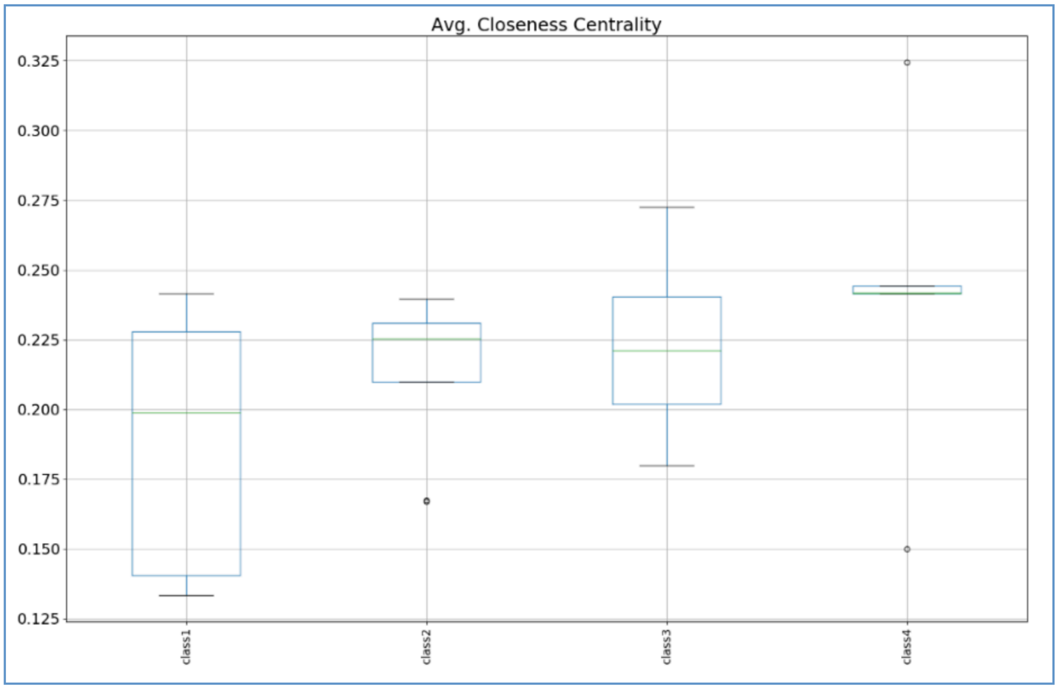


Fig. (5). Group comparison for measure Avg. Closeness Centrality. If loaded network data be classified, CatbNet will provide group comparison boxplot. In this case, all network-based and node-based attributes will be compared.

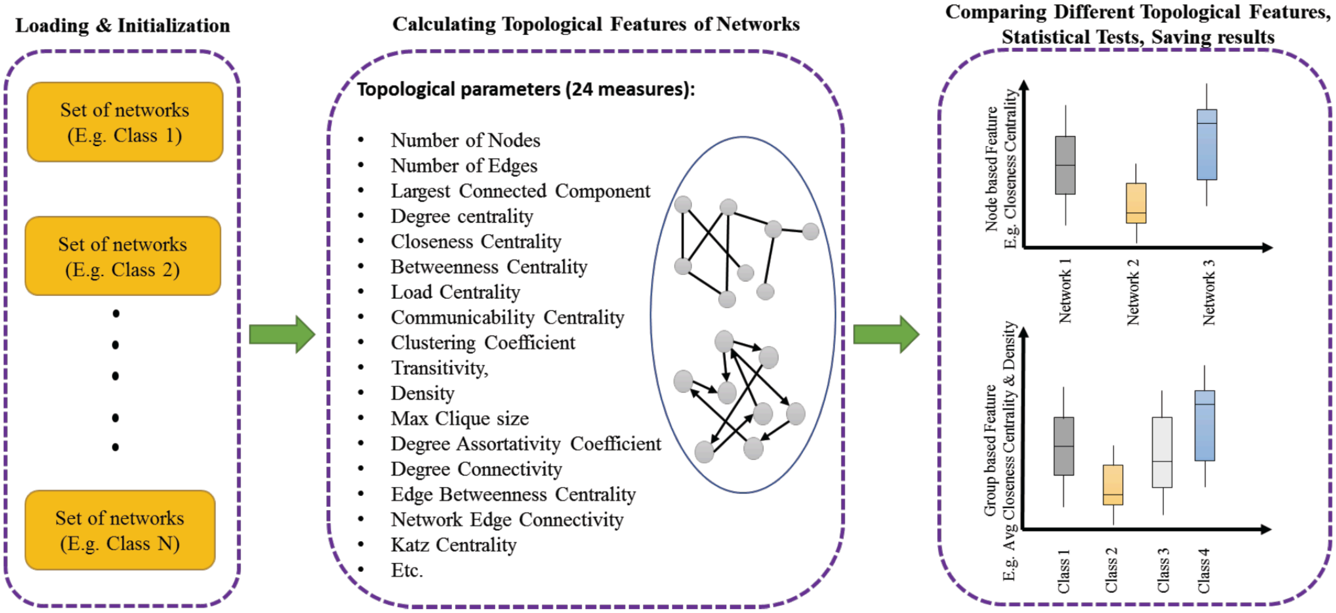

Loading & Initialization
Set of networks (E.g. Class 1)
Set of networks (E.g. Class 2)
Set of networks (E.g. Class N)
Calculating Topological Features of Networks
Topological parameters (24 measures):
Number of Nodes
Number of Edges
Largest Connected Component
Degree centrality
Closeness Centrality
Betweenness Centrality
Load Centrality
Communicability Centrality
Clustering Coefficient
Transitivity,
Density
Max Clique size
Degree Assortativity Coefficient
Degree Connectivity
Edge Betweenness Centrality
Network Edge Connectivity
Katz Centrality
Etc.
Comparing Different Topological Features, Statistical Tests, Saving results
Node based Feature E.g. Closeness Centrality
Network 1
Network 2
Network 3
Group based Feature E.g. Avg Closeness Centrality & Density
Class 1
Class 2
Class 3
Class 4

Fig. (6). Schematic of steps in CatbNet, from loading data to saving the results.

CONCLUSION

CatbNet is a user-friendly multi network analyzer application. It has been developed using a set of python packages for network analysis and comparison purposes. This application has a graphically GUI to help researchers of different fields to analyze networks especially computational biologists. It calculates 24 centrality measures and topological features for a set of networks simultaneously. Furthermore, it accepts various network formats including GML, GraphML, edge list and Pajek and different types of networks, such as directed/undirected and weighted networks. There is no doubt that, there are numerous valuable packages and applications for calculating different topological features and centrality measures, but to the best of our knowledge, there is no package to simultaneously get a number of networks or a group of networks as input and calculate their features. Another prominent novelty of this application is comparing different networks or group of networks by each other based on the different features within its feature list and calculating statistical significance using ANOVA test and calculating *P*-value for each comparison. The significance level of each feature in separating networks or network classes can be used as a guide to choose appropriately segregating features among numerous features to distinguish them accurately. In fact, these unique properties made CatbNet a very suitable and useful application for network mining, network medicine and machine learning applications in systems biology.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

Not applicable.

HUMAN AND ANIMAL RIGHTS

No Animals/Humans were used for studies that are the basis of this research.

CONSENT FOR PUBLICATION

Not applicable.

CONFLICT OF INTEREST

The authors declare no conflict of interest, financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Freeman, L.C. Going the wrong way on a one-way street: Centrality in physics and biology. *J. Soc. Struct*., **2008**; *9*(2), 1-15.

[2] Jeong, H. Lethality and centrality in protein networks. *Nature*., **2001**, *411*(6833), 41-2.

[3] He X, Zhang J. Why do hubs tend to be essential in protein networks? *PLoS Gene*., **2006**, *2*(6), e88.

[4] Jalili, M.; Salehzadeh-Yazdi, A; Asgari. Y; Arab, S.S.; Yaghmaie, M.; Ghavamzadeh A.; Alimoghaddam, K . CentiServer: a comprehensive resource, web-based application and R package for centrality analysis. *PloS. One*., **2015**, *10*(1), e0143111.

[5] Shannon, P.; Markiel, A.; Ozier, O.; Baliga, N.S.; Wang, J.T.; Ramage, D.; Amin, N.; Schwikowski, B.; Ideker, T. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome. Res*., **2003**, *13*(11), 2498-504.

[6] Assenov, Y.; Ramírez, F.; Schelhorn, S.E.; Lengauer, T.; Albrecht, M. Computing topological parameters of biological networks. *Bioinformatics*., **2007**, *24*(2), 282-4.

[7] Chin, C.H.; Chen, S.H.; Wu, H.H.; Ho, C.W.; Ko, M.T.; Lin, C.Y. cytoHubba: Identifying hub objects and sub-networks from complex interactome. *BMC. Sys. Biol*., **2014**, *8*(4), S11.

[8] Tang, Y.; Li, M.; Wang, J.; Pan, Y.; Wu, F.X. CytoNCA: A cytoscape plugin for centrality analysis and evaluation of protein interaction networks. *Biosystems*, **2015**, *127*(2015), 67-72.

[9] Junker, B.H.; Koschützki, D.; Schreiber, F. Exploration of biological network centralities with CentiBiN. *BMC. Bioinformatics*., **2006**, *7*(219), 193-201.

[10] Ashtiani, M.; Jafari, M. CINNA: Deciphering central informative nodes in network analysis. *BioRxiv*, **2017**, *27*(168757), 1-9.

[11] Theodosiou, T.; Efstathiou, G.; Papanikolaou, N.; Kyrpides, N.C. Bagos, P.G.; Iliopoulos, I.; Pavlopoulos, G.A. NAP: The network analysis profiler, a web tool for easier topological analysis and comparison of medium-scale biological networks. *BMC. Res*., **2017**, *10*(1), 278.

[12] Hagberg, A; Swart, P.S.; Chult, D. Exploring network structure, dynamics, and function using NetworkX. Los Alamos National Laboratory (LANL), **2008**.

[13] Barabasi, A.L; Gulbahce, N.; Loscalzo, J. Network medicine: A network-based approach to human disease. *Nat. Rev. Genet*., **2011**, *12*(1), 56-68.

[14] Loscalzo, J. *Network Medicine*. Harvard University Press, **2017**.

[15] Rolland, T.; Taşan, M.; Charloteaux, B.; Pevzner, S.J.; Zhong, Q. Sahni, N.; Yi, S.; Lemmens, I.; Fontanillo, C.; Mosca, R.; Kamburov, A.; Ghiassian, S.D.; Yang, X.; Ghamsari, L.; Balcha, D.; Begg, B.E.; Braun, P.; Brehme, M.; Broly, M.P.; Carvunis, A.R.; Convery-Zupan, D.; Corominas, R.; Coulombe-Huntington, J. Dann, E.; Dreze, M; Dricot, A.; Fan, C.; Franzosa, E.; Gebreab, F.; Gutierrez, B.J.; Hardy, M.F.; Jin, M.; Kang, S.; Kiros, R.; Lin, G.N.; Luck, K.; MacWilliams, A.; Menche, J.; Murray, R.R.; Palagi, A.; Poulin, M.M.; Rambout, X.; Rasla, J.; Reichert, P.; Romero, V.; Ruyssinck, E.; Sahalie, J.M.; Scholz, A.; Shah, A.A.; Sharma, A.; Shen, Y.; Spirohn, K.; Tam, S.; Tejeda, A.O.; Trigg, S.A.; Twizere, J.C.; Vega, K.; Walsh, J.; Cusick, M.E.; Xia, Y.; Barabási, A.L.; Iakoucheva, L.M.; Aloy, P.; De Las Rivas, J.; Tavernier, J.; Calderwood, M.A.; Hill, D.E.; Hao, T.; Roth, F.P.; Vidal, M. A Proteome-scale map of the human interactome network. *Cell*, **2014**, *159*(5), 1212-1226.

[16] Guney, E.; Menche, J.; Vidal, M.; Barábasi, A.L. Network-based *in silico* drug efficacy screening. *Nat. Comm*., **2016**, 7: 10331, 1-13.

[17] Cheng, F.; Liu, C.; Shen, B.; Zhao, Z. Investigating cellular network heterogeneity and modularity in cancer: A network entropy and unbalanced motif approach. *BMC. Syst. Biol*., **2016**, *10* (65), 11-16.

[18] West, J.; Bianconi, G.; Severini, S.; Teschendorff, A.E. Differential network entropy reveals cancer system hallmarks. *Sci. Rep*., **2012**, *2*(1), 802.

[19] de Anda-Jáuregui, G.; Velázquez-Caldelas, T.E.; Espinal-Enríquez, J.; Hernández-Lemus, E. Transcriptional network architecture of breast cancer molecular subtypes. *Front. Physiol*., **2016**, *7*(1), 568.

[20] Jalili, M. Graph theoretical analysis of Alzheimer's disease: Discrimination of AD patients from healthy subjects. Information *Sciences*., **2017**, *384*, 145-56.

[21] Zhang, W.; Chien, J.; Yong, J.; Kuang, R. Network-based machine learning and graph theory algorithms for precision oncology. npj *Precis. Oncol*., **2017**, *1*(1), 25.

[22] Jamal, S.; Goyal, S.; Shanker, A.; Grover, A. Integrating network, sequence and functional features using machine learning approaches towards identification of novel Alzheimer genes. *BMC. Genome*. **2016**, *17*(1), 807.

[23] Walt, S.; Colbert, S.C.; Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng*., **2011**, *13*(2), 22-30.

[24] Hunter, J.D.; Matplotlib: A 2D graphics environment. *Compt. Sci. Eng*., **2007**; *9*(3), 90-5.

[25] McKinney, W. Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference: SciPy, Texas, U.S.; **2010**; pp. 51-6.

[26] Jones, E.; Oliphant T.; Peterson P. {SciPy}: Open Source Scientific Tools for Python **2014**.