Article id: jbg.2017.1.3.3
Journal of Bioinformatics and Genomics, 2017, 1-3
doi: 10.18454/jbg.2017.1.3.3
Advance Access Publication Date: 15 May 2017

NOVEL COMPUTATIONAL TOOLS AND DATABASES

CELNETANALYZER: HIGH-PERFORMANCE JAVA PACKAGE FOR THE TOPOLOGICAL ANALYSIS OF CELLULAR NETWORKS

Funding

This work was supported by the Belarusian Republican Foundation for Fundamental Research [grant number M12-071].

Conflict of Interest

None declared.

Vasily Grinev[1*], Dmitry Kushal[1], Vitaly Charapovich[1]

[1]Department of Genetics, Faculty of Biology, Belarusian State University, Nezavisimosti Avenue-4, 220030, Minsk, Republic of Belarus

*To whom correspondence should be addressed.

Associate editor: Giancarlo Castellano
Received on 27 April 2017, revised on 02 May 2017, accepted on 12 May 2017.

Abstract

***Summary*:** *A simple-to-use Java-based software package CelNetAnalyzer was developed. CelNetAnalyzer is managed through a graphical user interface and it returns a comprehensive list of the topological indices including compositional complexity, degree and neighbourhood, clustering, distance, centrality and heterogeneity indices as well as simple cycles and Shannon information entropy of undirected networks. Comparative studies have shown that due to parallelization and use of enhanced and newly developed algorithms, CelNetAnalyzer calculates these parameters significantly faster than competitors.*

***Availability and Implementation*:** *CelNetAnalyzer is an open-source project and free distributed for non-commercial use. Software package, source code, test network and the results of the topological analysis can be downloaded from website of the Department of Genetics at Belarusian State University (http://bio.bsu.by/genetics/grinev_software.html).*

Supplementary information: *Supplementary data are available at Journal of Bioinformatics and Genomics online.*

Keywords: *cellular molecular networks, topological analysis, software.*

Contact: *grinev_vv@bsu.by*

The theory of complex networks is one of the fastest growing areas of the modern science (Newman, 2010). The progress made in this field has been widely applied in the analysis of structural organization, functionality and robustness of various types of networks including engineering, social and biological networks. Physics, sociology and computer sciences are the disciplines most actively utilizing advances in the theory of complex networks. It is obvious that in all these cases, the practical application of theoretical achievements became possible due to the successful development of appropriate software enabling fast and comprehensive analysis of large networks.

Recent advances in the molecular biology suggested a network principle of organization as a basis for the robust functioning of the cell (MacNeil and Walhout, 2011). Application of the network analysis for molecular biology purposes became possible mainly due to the widespread of OMICS-technologies as well as methods to reverse engineering of cellular networks. Whereas the methods of cellular networks reconstruction are developing quite rapidly, the special methods of structural analysis of such networks are delayed. Therefore, among various available computer programs we studied only NetworkAnalyzer (Assenov et al., 2008) is representing a software tool directed to the analysis of cellular networks. In PubMed, one of the most frequently used programs in the study of social networks Pajek package (Batagelj and Mrvar, 1998) is cited only 24 times of which only two references are related to the analysis of cellular networks. The complexity of the software for biological needs, difficulties in interpretation of topological indices and time-consuming analysis of large cellular networks are the main possible reasons of limited application of such network packages. To avoid these limitations, we propose a novel software product – a Java package CelNetAnalyzer.

One of the key features of the package is highly integrated and interrelated calculation of network indices. In graph theory, there exists a set of indices that describe the topology of the networks. However, only some of them can be interpreted from a biological point of view. In this context, we carefully analyzed of the biological relevance of such metrics and se-

lected only 46 global and local indices. The final list of indices includes compositional complexity, degree and neighbourhood, clustering, distance, centrality and heterogeneity indices, simple cycles and Shannon information entropy (see Supplementary Data for further details). In our software, the most complex is the algorithm for count of five-membered cycles which is implemented in three steps (see Supplementary Data for further explanations). Next on the complexity are algorithms for calculation of the betweenness centrality and count of six-membered cycles.

The second key feature of the package is parallel computation. This feature is one of the main factors of the high performance of our software. Additional feature of the package is that it is cross-platform. In particular, the package has been successfully tested in Windows 7 Ultimate, Ubuntu Linux 9.04 and Mac-OS-X-Mavericks 10.9 environment.

The software can work in two dynamically linked modes. Small networks are handled in the basic mode, which does not require large amounts of RAM. However, if the number of shortest paths from one node to another is greater than 2.15×10^9 then such network is classified as large and it automatically switches to the processing of the network using enhanced mode of operation. This mode requires significantly more RAM comparing to the basic one. The critical amount of RAM for proper program functioning can be calculated based on the number and size of the program-generated arrays (see Supplementary Data for further details).

Functionality of the CelNetAnalyzer package was tested on two types of cellular networks. The first one was inferred by ARACNE2 algorithm (Margolin et al., 2006) from publicly available microarray data of 106 t(8;21)-positive human acute myeloid leukemia samples (Supplementary Table 1). This leukemia network contains 9773 nodes and 199974 edges. The second network was built on the basis of information on protein-protein and protein-gene interactions in human cells deposited in databases BioGRID v.3.1.91 (Chatr-Aryamontri et al., 2013) and STRING v.9.0 (Szklarczyk et al., 2011). The resulting reference network includes 17342 nodes and 1522080 edges.

CelNetAnalyzer was compared to four other programs: NetworkAnalyzer v.2.8.3 (Assenov et al., 2008), Pajek v.4.01 (Batagelj and Mrvar, 1998), NetworkX v.1.9.1 (Aric et al., 2008) and igraph v.0.7.1 (Csardi and Nepusz, 2006). The comparison of the network tools was based on three criteria: simplicity of software use, list of cellular network-oriented topological indices, and software performance. All basic tests were conducted with gene regulatory network from leukemia cells on a desktop computer equipped by Intel® Core™ i5-4670K 3.40 GHz CPU, 8.00 GB RAM and 64-bit operation system Windows 7 Ultimate.

All of the selected programs are free of charge for non-commercial usage; they are cross-platform and support a wide range of network formats. However, only first two of the above-mentioned programs are stand-alone with GUI like CelNetAnalyzer, whereas the last two are the libraries and their use requires special programming skills in Python or R respectively.

NetworkX v.1.9.1 and igraph v.0.7.1 has the broadest capabilities for the analysis of networks: lists of indices include 177 and 103 items, respectively. However, only some of these indices can be interpretable from a biological standpoint. The remaining two programs offer a more intuitive for biologists list of network metrics. During development of CelNetAnalyzer, we selected only biologically relevant indices. Furthermore, CelNetAnalyzer was equipped by effective algorithms for calculating the simple cycles in cellular networks and none of the comparators exhibit such ability for network analysis.

Software performance was evaluated by using two approaches: 1) the time required for the calculation of a representative topological index; 2) the time required for the calculation of all topological indices. The local topological index betweenness centrality was selected for first approach. All the compared programs may calculate this index. Moreover, algorithms for calculation of the betweenness centrality are among the most time-consuming algorithms. The results of this study show a great performance of the CelNetAnalyzer (Supplementary Table 2). As for second approach, the closest in design CelNetAnalyzer and NetworkAnalyzer v.2.8.3 were compared by this way. CelNetAnalyzer analyzes the test network in 237-fold faster in none parallel mode and in 828-fold faster in four threads mode than NetworkAnalyzer v.2.8.3. Herewith it should be noted that the NetworkAnalyzer does not search for simple cycles in networks, which takes majority of the computing time during work of CelNetAnalyzer.

Thus, CelNetAnalyzer package provides a powerful tool for the topology analysis of large undirected networks. This software uses lightweight graphical interface and contains state-of-the-art algorithms to perform fast calculation of wide range topological indices directed on structure elucidation of cellular networks. Simple format of the obtained results of the topological analysis (spreadsheet *.txt tab-delimited format) makes the results easy to subsequent use.

Appendix

Supplementary Table 1 – List of the publicly available microarrays deposited in the repository NCBI GEO and used in this study

GEO series	GEO samples
GSE13159	GSM330387, GSM330388, GSM330389, GSM330390, GSM330391, GSM330392, GSM330393, GSM330394, GSM330395, GSM330396, GSM330397, GSM330398, GSM330399, GSM330400, GSM330401, GSM330402, GSM330403, GSM330404, GSM330405, GSM330406, GSM330407, GSM330408, GSM330409, GSM330410, GSM330411, GSM330412, GSM330413, GSM330414, GSM330415, GSM330416, GSM330417, GSM330418, GSM330419, GSM330420, GSM330421, GSM330422, GSM330423, GSM330424, GSM330425, GSM330426
GSE14468	GSM158712, GSM158716, GSM158719, GSM158721, GSM158722, GSM158750, GSM158752, GSM158781, GSM158799, GSM158810, GSM158814, GSM158861, GSM158873, GSM158878, GSM158899, GSM158905, GSM158906, GSM158911, GSM158912, GSM158925, GSM158927, GSM158966, GSM158970, GSM158982, GSM158984, GSM158985, GSM159005, GSM159032, GSM159037, GSM159050, GSM159068, GSM159081, GSM159097, GSM159105, GSM159110
GSE17855	GSM445983, GSM445989, GSM445990, GSM446010, GSM446017, GSM446025, GSM446034, GSM446123, GSM446125, GSM446128, GSM446129, GSM446139, GSM446146
GSE22056	GSM445922, GSM445935, GSM445938, GSM445943, GSM445950, GSM445959, GSM445964, GSM445970, GSM445972, GSM446051, GSM446054, GSM446055, GSM446057, GSM446066, GSM446067
GSE29883	GSM740083, GSM740087, GSM740088

Supplementary Table 2. Features of the CelNetAnalyzer and its key counterparts.

Features	Software				
	NetworkAnalyzer v.2.8.3	**Pajek v.4.01**	**NetworkX v.1.9.1**	**igraph v.0.7.1**	**CelNetAnalyzer**
Ease of use					
Type of software	Stand-alone	Stand-alone	Library	Library	Stand-alone
Platform	Cross-platform	Cross-platform	Cross-platform	Cross-platform	Cross-platform
License	GNU LGPL	Free for non-commercial use	BSD License	GNU GPL	Free for non-commercial use
Requirements for programming skills	No	No	Yes	Yes	No
Graphical user interface	Yes	Yes	No	No	Yes
Structural properties of network					
Number of calculated network parameters	19	46	177	103	46
Simple cycles	No	No	No	No	Yes
Output format of network statistics	.netstats	.vec, .txt (tab delimited)	.csv, .xml	.txt (space delimited)	.txt (tab delimited)
Performance					
Programming language	Java	C	Python	R	Java
Parallelization	No	No	No	No	Yes
Performance, min [1]	144. 55 ± 11.04 [2]	3.19 ± 0.01 [3]	31.73 ± 2.31 [3]	0.37 ± 0.002 [3]	0.13 ± 0.002 (0.39 ± 0.005) [2], [4], [5]

Notes.

(1)Software performance was evaluated in a test on calculation of the local topological index betweenness centrality. This index was selected on two criteria: 1) all the compared programs may calculate this index; 2) algorithms for calculation of the betweenness centrality are among the most time-consuming algorithms. The test was carried out on a desktop computer equipped by Intel® Core™ i5-4670K 3.40 GHz CPU, 8.00 GB RAM and 64-bit operation system Windows 7 Ultimate. The gene regulatory network from t(8;21)-positive acute myeloid leukemia was used for that. The table shows the time (in minutes) required to calculation of the selected index. Results are expressed as arithmetic mean plus/minus standard deviation from three independent runs.

(2)A relevant part of source code was used for calculation of the selected index.

(3)These software tools permit to calculate each topological index independently.

(4)A set of common auxiliary arrays is generated during code execution. These arrays are used not only to calculate the betweenness centrality index but in the calculation of the most of other indices. During calculation of the betweenness centrality index, preparation of auxiliary arrays takes 83.8 % of CPU time and the rest time is spent on the calculation of the index itself.

(5)CelNetAnalyzer was tested in two modes: with parallelization (four threads) and in none parallel mode (one thread). The time taken for the calculation of betweenness centrality in none parallel mode is indicated in parentheses.

Supplementary Text 1. Key limitations in the functioning of CelNetAnalyzer software.

1) Size of network.

The CelNetAnalyzer can work in two dynamically linked modes. Small networks are processed in the basic mode. If the number of shortest paths from one node to another exceeds 2.15×10^9 the network is classified as large and calculation switches to the enhanced mode of operation. If the number of shortest paths passing through any node in network outreaches 4.61×10^{18}, the program will generate an error. The program also reports an error if the number of shortest paths from one node to another exceeds 9.22×10^{18}. Another limitation of the program is the overall number of nodes and edges in the network, which should not be more than 1×10^9.

2) Allocated RAM.

The critical amount of RAM for proper program functioning can be calculated based on the number and size of the program-generated arrays. A running program creates the following arrays: 1) adjacency array $N \times N$ with elements of type *int* to store the shortest distances between nodes, where N is the number of nodes in the network; 2) array $N \times N$ with elements of type *int* for small networks or *long* for large ones, which stores the number of shortest paths in network and it is also used in the calculation of the centralities; 3) two additional arrays $N \times N$ with elements of type *int* for calculation of cycles with unique order of nodes connections; 4) jagged array of size $N \times N_f$ for storing neighbours for each node, where N_f is the number of first neighbours of node. Herewith the size of the element type *int* is 4 bytes and 8 bytes for type *long*. Program generates also other auxiliary arrays, for example, a number of one-dimensional arrays $1 \times N$ or array $R \times N$, where R is a radius of network. However, these arrays are small in size comparing to arrays $N \times N$.

3) The time required to complete the analysis.

The time required to complete the analysis of network mostly depends on the complexity of the algorithms. The most complex is the algorithm for search of five-membered cycles which is implemented in three steps. The complexity of the first step is equal to $N \times \langle N_f \rangle^2 \times \langle N_s \rangle \times P_2 \times P_3$, where N is the number of nodes in the network, $\langle N_f \rangle$ and $\langle N_s \rangle$ are the average number of neighbours of the first and second order, respectively, P_2 is the probability that the neighbour node of the node of second order is node of first order and P_3 is the probability that the neighbour node of the node of second order is also node of second order. The complexity of second step is proportional to $N \times \langle N_f \rangle^3 \times \langle N_s \rangle \times P_1 \times (P_2)^2$, where P_1 is the probability that two neighbours of the target node are connected. Finally, the complexity of the third step is $N \times \langle N_f \rangle^4 \times (P_1)^3$. Next on the complexity, there are algorithms for calculation of the stress and betweenness centralities $N^2 \times \langle N_f \rangle$ and search of six-membered cycles $N \times \langle (N_f)^3 \rangle$.

4) Number of threads.

One of the key features of the package is parallel computation. The program should operate stably using at least eight threads. We have not tested the performance on machines with more cores or on clusters of computers.

Supplementary Text 2. Description of the network parameters which are calculated by CelNetAnalyzer.

Compositional complexity.

Compositional complexity C_{com} of network can be measure by following formula:

$$C_{com} = C_{compl}(N + E),$$

where C_{compl} is a coefficient of network completeness (also known as connectedness or network density), N and E is number of nodes and edges in a given network, respectively. Coefficient of network completeness can be calculated as:

$$C_{compl} = \frac{E}{E_{max}} = \frac{2E}{N(N+1)},$$

where E_{max} is a number of edges in complete network with auto-loops.

The C_{compl} is a value between 0 and 1. It shows how densely the network is populated with edges (or how close a given network is to complete network). A network which contains no edges and solely isolated nodes has a density of 0. In contrast, the density of a complete network is 1.

From two above equations the final formula for compositional complexity of network is:

$$C_{com} = \frac{2E(N+E)}{N(N+1)}.$$

Degree and neighbourhood indices (Assenov *et al.*, 2008; Bonchev *et al.*, 2005; Diestel, 2005; Maslov, Sneppen, 2002; Stelzl *et al.*, 2005).

In undirected networks, the node degree a_i of a node N_i is the number of edges linked to N_i:

$$a_i = \sum_{j=1}^{N} a_{ij}.$$

A self-loop of a node is counted like one edge for the node degree. The node degree distribution gives the number of nodes with degree a for $a = 0, 1, \ldots$.

The sum of all node degrees in a network defines its total adjacency A:

$$A = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} = \sum_{i=1}^{N} a_i.$$

The network (global or average) node degree <*a*> is the average of the degrees for all nodes in the network:

$$< a >= \frac{\sum_{i=1}^{N} a_i}{N} = \frac{A}{N}.$$

The neighbourhood of a given node N_i is the set of its neighbours. The connectivity $conn_{N_i}$ of a node N_i is the number of its neighbours (or size of its neighbourhood) and it is equal to a for nodes without auto-loops or $a - 1$ for nodes with auto-loops. The neighbourhood connectivity NC_{N_i} of a node N_i is defined as the average connectivity of all neighbours of N_i:

$$NC_{N_i} = \frac{\sum_{f=1}^{n} conn_{N_f}}{conn_{N_i}},$$

where N_f is first neighbours of node N_i and $n = a_i$ (or $a_i - 1$ for nodes with auto-loops).

The neighbourhood connectivity distribution gives the average of the neighbourhood connectivities of all nodes N with k neighbours for $k = 0, 1, \ldots$.

The topological coefficient T_{N_i} of a node N_i with k_n neighbours is computed as follows:

$$T_{N_i} = \frac{avg(J(N_i,\ N_j))}{k_n}.$$

Here, $J(N_i, N_j)$ is defined for all nodes N_j that share at least one neighbour with N_i. The value $J(N_i, N_j)$ is the number of neighbours shared between the nodes N_i and N_j, plus one if there is a direct link between N_i and N_j.

The topological coefficient is a relative measure for the extent to which a node shares neighbours with other nodes. Nodes that have one or no neighbours are assigned a topological coefficient of 0.

Clustering indices (Assenov *et al*., 2008; Barabási, Oltvai, 2004; Bonchev *et al*., 2005; Soffer *et al*., 2005; Watts, Strogatz, 1998).

For undirected networks, the standard definition of local clustering coefficient C_{N_i} of a node N_i is:

$$C_{N_i} = \frac{2E_i}{a_i(a_i - 1)},$$

where E_i is the number of edges between the first neighbours of node N_i and a_i is the degree of this node.

In according to this definition, the clustering coefficient of a node N_i is the number of triangles that pass through this node, relative to the maximum number of 3-loops that could pass through the node. The clustering coefficient of a node is always a number between 0 and 1. Here, nodes with less than two neighbours are assumed to have a clustering coefficient of 0.

The global (network) clustering coefficient <*c*> is the average of the clustering coefficients for all nodes in the network:

$$< c >= \frac{\sum_{i=1}^{N} C_{N_i}}{N}.$$

This is sometimes also called as transitivity of network.

The degree-correlation bias insensitive local clustering coefficient $\tilde{C}_{N_i}$ of a node N_i can be calculated as following:

$$\tilde{C}_{N_i} = \frac{E_i}{\omega_i},$$

where ω_i is the maximum number of edges that can be drawn among the a_i neighbours of a node N_i, given the degree sequence of its neighbourns $\{a_1, \ldots, a_n\}$(n = a_i).

The global clustering coefficient <$\tilde{c}$> in that case will be equal:

$$< \tilde{c} >= \frac{\sum_{i=1}^{N} \tilde{C}_{N_i}}{N}.$$

Distance indices (Assenov *et al*., 2008; Bonchev *et al*., 2005).

A path in the network is a sequence of adjacent edges between two nodes without traversing any intermediate node twice. The length of a path is the number of edges forming it. There may be multiple paths connecting two given nodes. The shortest path length, also called "distance", between two nodes N_i and N_j is denoted by $d(N_i, N_j)$ (or simply d_{ij}). The sum of all shortest paths for a given node is node distance d_i:

$$d_i = \sum_{j=1}^{N} d_{ij}.$$

The average of distances <d_i> for node N_i (also known as the average shortest path length or the characteristic path length) gives the expected distance between two connected nodes and can be calculated as following:

$$< d_i >= \frac{d_i}{N-1},$$

or

$$< d_i >= \frac{d_i}{N}$$

for node with auto-loop.

The distance distribution gives the number of node pairs (N_i, N_j) with $d(N_i, N_j) = k$ for $k = 1, 2, \ldots$.

The sum of all shortest paths for a given network is network distance D:

$$D = \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij} = \sum_{i=1}^{N} d_i.$$

The average node distance <*d*> is the relation of network distance to number of nodes in network:

$$< d >= \frac{D}{N}.$$

The average network distance <*D*> (average path length or average degree of node-node separation) can be calculated by following formula:

$$< D >= \frac{D}{N^2 - N + M},$$

where M is number of auto-loops.

Node eccentricity e_i is the maximum distance between node N_i and any of the remaining network nodes. The largest node eccentricity is termed network diameter *NetD*. The diameter can also be described as the largest distance between two nodes. The minimal eccentricity is termed network radius *NetR*. The node(s) with minimum eccentricity is defined as network centre *NetC*.

Centrality indices (Assenov *et al*., 2008; Brandes *et al*., 2001; Dong, Horvath, 2007; Freeman, 1977; Freeman, 1978; Newman *et al*., 2005; del Rio *et al*., 2009; Yoon *et al*., 2006).

The stress centrality $C_s(N_i)$ of a node N_i is the number of shortest paths passing through N_i:

$$C_s(N_i) = \sum_{N_s \neq N_i \in N} \sum_{N_t \neq N_i \in N} \sigma_{st}(N_i),$$

where N_s and N_t are nodes in the network different from N_i and $\sigma_{st}(N_i)$ is the number of shortest paths from N_s to N_t that N_i lies on.

The stress centrality value for each node N_i is normalized by dividing by the sum of centralities of the all nodes of the network:

$$C_s(N_i)norm = \frac{C_s(N_i)}{\sum_{j=1}^{N} C_s(N_j)},$$

where N is the total number of nodes in the network.

A node has a high stress if it is traversed by a high number of shortest paths. This parameter is defined only for networks without multiple edges. The stress centrality distribution gives the number of nodes with C_s for different values of C_s.

The betweenness centrality $C_b(N_i)$ of a node N_i is computed as follows:

$$C_b(N_i) = \sum_{N_s \neq N_i \in N} \sum_{N_t \neq N_i \in N} \frac{\sigma_{st}(N_i)}{\sigma_{st}},$$

where σ_{st} denotes the number of shortest paths from N_s to N_t.

Betweenness centrality is computed only for networks that do not contain multiple edges. The betweenness value for each node N_i is normalized by dividing by the number of node pairs excluding N_i:

$$C_b(N_i)norm = \frac{2C_b(N_i)}{N^2 - 3N + 2},$$

where N is the total number of nodes in the connected component that N_i belongs to.

Thus, the betweenness centrality of each node is a number between 0 and 1. The betweenness centrality of a node reflects the amount of control that exerts this node over the interactions of other nodes in the network. This measure favours nodes that join communities (dense subnetworks), rather than nodes that lie inside of a community.

The closeness centrality $C_c(N_i)$ of a node N_i is an inverse of node distance d_i and is computed as follows:

$$C_c(N_i) = \frac{1}{\sum_{i=1,j=1}^{N} d_{ij}}.$$

Normalized value of closeness centrality for given node N_i is reciprocal value to characteristic path length and is computed as follows:

$$C_c(N_i)norm = \frac{N-1}{\sum_{i=1,j=1}^{N} d_{ij}} = \frac{1}{< d_i >}.$$

The closeness centrality of each node is a number between 0 and 1. The closeness centrality of isolated nodes is equal to 0. Closeness centrality is a measure of how fast information spreads from a given node to other reachable nodes in the network.

The centralization of any network is a measure of how central its most central node is in relation to how central all the other nodes are. To calculate of network centralization, the sum of differences in centrality between the most central node in a network and all other nodes is calculated in first then this quantity is divided by the theoretically largest such sum of differences in any network of the same degree. Thus, every centrality measure can have its own centralization measure. Defined formally, if $C_x(N_i)$ is any centrality measure of node N_i, if $C_x(N_n)$ is the largest such measure in the network, and if

$$max \sum_{i=1}^{N} C_x(N_n) - C_x(N_i)$$

is the largest sum of differences in node centrality C_x for any network of with the same number of nodes, then the centralization of the network is:

$$C_{cen} = \frac{\sum_{i=1}^{N} C_x(N_n) - C_x(N_i)}{max \sum_{i=1}^{N} C_x(N_n) - C_x(N_i)}.$$

Networks whose topologies resemble a star have a centralization close to 1, whereas decentralized networks are characterized by having a centralization close to 0.

For normalized values of centralities, network stress centralization can be calculated as follow:

$$C_{s.cen} = \frac{\sum_{i=1}^{N} C_s(N_n)norm - C_s(N_i)norm}{N-1}.$$

Similarity, network betweenness centralization can be calculated by following way:

$$C_{b.cen} = \frac{\sum_{i=1}^{N} C_b(N_n)norm - C_b(N_i)norm}{N-1}.$$

As for network closeness centralization, this index can be calculated by next formula:

$$C_{c.cen} = \frac{\sum_{i=1}^{N} C_c(N_n)norm - C_c(N_i)norm}{(N^2 - 3N + 2)/(2N-3)}.$$

In finally, the combined centrality score $C_{cs}(N_i)$ for each gene in the network can be calculated according to the following formula:

$$C_{cs}(N_i) = \frac{\sum_{i=1}^{m} \frac{C_x(N_i) - minC_x(N_n)}{maxC_x(N_n) - minC_x(N_n)}}{m},$$

where $C_x(N_i)$ is any centrality measure of node N_i in a given network, $maxC_x(N_n)$ and $minC_x(N_n)$ define the maximum and minimum score obtained for x^{th}-centrality in a given network, respectively, and m refers to number of combined centralities (for instance, m = 2 for groups of 2 centralities, m = 3 for groups of 3 centralities etc.).

Combined centrality score estimates how close to the largest observed centrality measures are the centralities of the gene analysed. Thus, the higher the combined score is, the higher the individual centrality measures are.

Heterogeneity indices (Assenov *et al.*, 2008; Dong, Horvath, 2007; Hu *et al.*, 2008).

In undirected networks, two nodes are connected if there is a path of edges between them. Within a network, all nodes that are pairwise connected form a connected component. The number of connected components indicates the connectivity of a network – a lower number of connected components suggest a stronger connectivity.

The network heterogeneity reflects the tendency of a network to contain hub nodes. In complex scale-free networks, this parameter has significant impact on network performance, such as robustness and attack tolerance. To calculate of network heterogeneity index H, Gini coefficient-based approach is the most appropriate since it permits to quantify the heterogeneity of a network with any degree distribution and quantitatively compares the heterogeneity of networks with different types of degree distribution. In according to this approach, the H considers the difference of every two degree values in a degree sequence. Actually H is equal to on-half of the relative mean difference, i.e. the arithmetic average of the absolute values of the differences between all possible pairs of node degrees:

$$H = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |a_i - a_j|}{2N^2 < a >}.$$

The heterogeneity index of any given network is a number between 0 and 1 and provides a measure of the average degree inequality in a network: larger index implies higher level of heterogeneity and vice versa. For any real network *H* is always less than 1, and *H* may approach 1 only for infinite networks.

Shannon entropy.

In information theory, information entropy is a measure of unpredictability or uncertainty in a random variable. Information entropy is often called the Shannon entropy in honor of Claude E. Shannon, who in 1948 developed the first mathematical theory of entropy (Shannon, 1948). This theory links the complexity of the system, the amount of information that is needed to describe such a system, and uncertainty that arises from the transfer of information on this system through a noisy communication channel. In general, the more complex the system, the greater the amount of information needed to describe it, and the more information uncertainty that occurs when sending a message of such a system.

Consider a system *S*, described by the random variable *X*, which can take on the values x_1, x_2, …, x_n with probabilities p_1, p_2, …, p_n, respectively. Distribution of the values of such a variable is subject to a simple probability law:

$$P_X(x_i) = p_i, p_i \geq 0, i = 1, 2, \dots, n$$

$$\sum_{i=1}^{n} p_i = 1$$

The amount of information I_S needed to describe a system *S* in which the a priori probability of i^{th} event is equal to $1/n$ is given by formula of R. Hartley (Hartley, R. V. L., 1928):

$$I_S = -log_2 \frac{1}{n} = log_2 n$$

Shannon entropy H_S of such system is equal to the amount of information. However, Shannon entropy of a system becomes smaller than the amount of information when the probabilities of occurrence of i^{th} events are not equal. In this case, Shannon entropy can be calculated as follow:

$$H_s = \sum_{i=1}^{n} p_i(x) log_2 \left(\frac{1}{p_i(x)}\right) = -\sum_{i=1}^{n} p_i\,(x) log_2 p_i(x)$$

When studying cellular networks, probability *p* of i^{th} event (e.g., the likelihood that the node N_i will have a degree *a*) is unknown, but given only the value x_i of random variable *X* (in the above example it is value of node degree determined in the topological analysis). In this case, the computation of the information entropy is preceded normalization of such data – the calculation of the theoretical probabilities of single events. Suppose that $a_i^{(j)}$ is a value of the degree of the i^{th} node in j^{th} network (i = 1, 2, ..., N; j = 1, 2, ..., m). Then:

$$p_i^{(j)} = \frac{a_i^{(j)}}{\sum_{i=1}^{N} a_i^{(j)}}, i = 1, 2, \dots, N; j = 1, 2, \dots, m$$

At identical (equiprobable) values of degree for each node of the given network probability *p* of i^{th} event is:

$$p_i = \frac{1}{N}, i = 1, 2, \dots, N$$

Since different cellular networks may differ in a size of *N*, normalized Shannon entropy H_S^{norm} should be used in comparative analysis:

$$H_S^{norm} = -\frac{\sum_{i=1}^{N} p_i^{(j)}(x) log_2 p_i^{(j)}(x)}{I_S}, i = 1, 2, \dots, N; j = 1, 2, \dots, m$$

References

Aric, A.A., Schult, D.A. and Swart, P.J. (2008). Exploring network structure, dynamics, and function using NetworkX. In: Varoquaux G., Vaught T., Millman J. (eds). *Proceedings of the 7th Python in Science Conference (SciPy 2008)*. Pasadena, USA, pp. 11-15.

Assenov, Y., Ramírez, F., Schelhorn, S.E., Lengauer, T. and Albrecht, M. (2008). Computing topological parameters of biological networks. *Bioinformatics*, 24, 282-284. doi: 10.1093/bioinformatics/btm554

Barabási, A. L. and Oltvai, Z. N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genetic*., 5, 101–113.

Batagelj, V. and Mrvar, A. (1998). Pajek – program for large network analysis. *Connections*, 21, 47-57.

Bonchev, D. and Buck, G. A. (2005) Quantitative measures of network complexity. In: Bonchev, D. and Rouvray, D. H. (eds). Complexity in chemistry, biology and ecology. *Springer*, New York, pp. 191–235.

Brandes, U. (2001) A faster algorithm for betweenness centrality. // *J. Math. Sociol*., 25, 163–177.

Chatr-Aryamontri, A., Breitkreutz, B.J., Heinicke, S., Boucher, L., Winter, A., Stark, C., Nixon, J., Ramage, L., Kolas, N., O'Donnell, L., Reguly, T., Breitkreutz, A., Sellam, A., Chen, D., Chang, C., Rust, J., Livstone, M., Oughtred, R., Dolinski, K. and Tyers, M. (2013). The BioGRID interaction database: 2013 update. *Nucleic Acids Research*, 41, D816-D823. doi: 10.1093/nar/gks1158

Csardi, G. and Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695.

Del Rio, G. *et al*. (2009) How to identify essential genes from molecular networks? *BMC Sys. Biol*., 3, 102. DOI:10.1186/1752–0509–3–102.

Diestel, R. (2005) Graph theory. *Springer-Verlag*, Heidelberg, ISBN 3–540–26182–6.

Dong, J. and Horvath, S. (2007) Understanding network concepts in modules. *BMC Sys. Biol*., 1, 24. DOI:10.1186/1752–0509–1–24.

Freeman, L. C. (1977) A set of measures of centrality based on betweenness. *Sociometry*, 40, 35–41.

Freeman, L. C. (1978) Centrality in social networks. Conceptual clarification. *Soc. Networks*, 79, 215–239.

Hartley, R. V. L. (1928) Transmission of information. *Bell Sys. Tech. J*., 7, 535–563.

Hu, H.–B. and Wang, X.–F. (2008) Unified index to quantifying heterogeneity of complex networks. *Physica A*, 387, 3769–3780.

MacNeil, L.T. and Walhout, A.J.M. (2011). Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome Research*, 21, 645-657. doi: 10.1101/gr.097378.109

Margolin, A.A., Wang, K., Lim, W.K., Kustagi, M., Nemenman, I. and Califano, A. (2006). Reverse engineering cellular networks. *Nature Protocols*, 1, 663-672. doi: 10.1038/nprot.2006.106

Maslov, S. and Sneppen, K. (2002) Specificity and stabil-

ity in topology of protein networks. *Science*, 296, 910–913.

Newman, M. E. J. (2005) A measure of betweenness centrality based on random walks. *Soc. Networks*, 27, 39–54.

Newman, M.E.J. (2010). *Networks. An introduction.* Oxford University Press, USA, 784 pp.

Shannon, C. E. (1948) A mathematical theory of communication. *Bell Sys. Tech. J.*, 27, 379–423, 623–656.

Soffer, S. N. and Vazquez, A. (2005) Network clustering coefficient without degree-correlation biases. *Phys. Rev.*, 71, 057101–1–057101–4.

Stelzl, U. *et al.* (2005) A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122, 957–968.

Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., Doerks, T., Stark, M., Muller, J., Bork, P., Jensen, L.J. and von Mering, C. (2011). The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39, D561-568. doi: 10.1093/nar/gkq973

Watts, D. J. and Strogatz, S. H. (1998) Collective dynamics of 'small-world' networks. *Nature*, 393, 440–442.

Yoon, J. *et al*. (2006) An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics*, 22, 3106–3108.

CELNETANALYZER: ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ JAVA-ПАКЕТ ДЛЯ ТОПОЛОГИЧЕСКОГО АНАЛИЗА КЛЕТОЧНЫХ СЕТЕЙ

Финансирование

Эта работа была поддержана Белорусским республиканским фондом фундаментальных исследований [номер гранта М12-071].

Конфликт интересов

Не указан.

Василий Гринев[1*], Дмитрий Кушель[1], Виталий Черепович[1]

[1] Кафедра генетики биологического факультета Белорусского государственного университета, Минск, Республика Беларусь

*Корреспондирующий автор

Редактор: Джанкарло Кастельяно

Получена 27 Апреля 2017, доработана 02 Мая 2017, принята 12 Мая 2017.

Аннотация

Разработан простой в использовании Java-пакет CelNetAnalyzer, предназначенный для топологического анализа клеточных молекулярных сетей. Пакет CelNetAnalyzer управляется через графический интерфейс пользователя и обеспечивает расчет разнообразных топологических индексов для ненаправленных сетей. Сравнительные исследования показали, что CelNetAnalyzer рассчитывает эти индексы существенно быстрее, чем аналогичные компьютерные программы. Высокая скорость работы программы обеспечивается благодаря многопоточности проводимых расчетов, а также улучшению существующих и использованию новых алгоритмов вычислений топологических индексов. Программа CelNetAnalyzer является открытым проектом и свободно распространяется для некоммерческого использования. Программный пакет, исходный код, тестовые сети и результаты их топологического анализа, а так же руководство пользователя программы доступны на сайте кафедры генетики Белорусского государственного университета (http://bio.bsu.by/genetics/grinev_software.html).

Ключевые слова: *клеточные молекулярные сети, топологический анализ, программное обеспечение.*