PROFEAT Tutorial

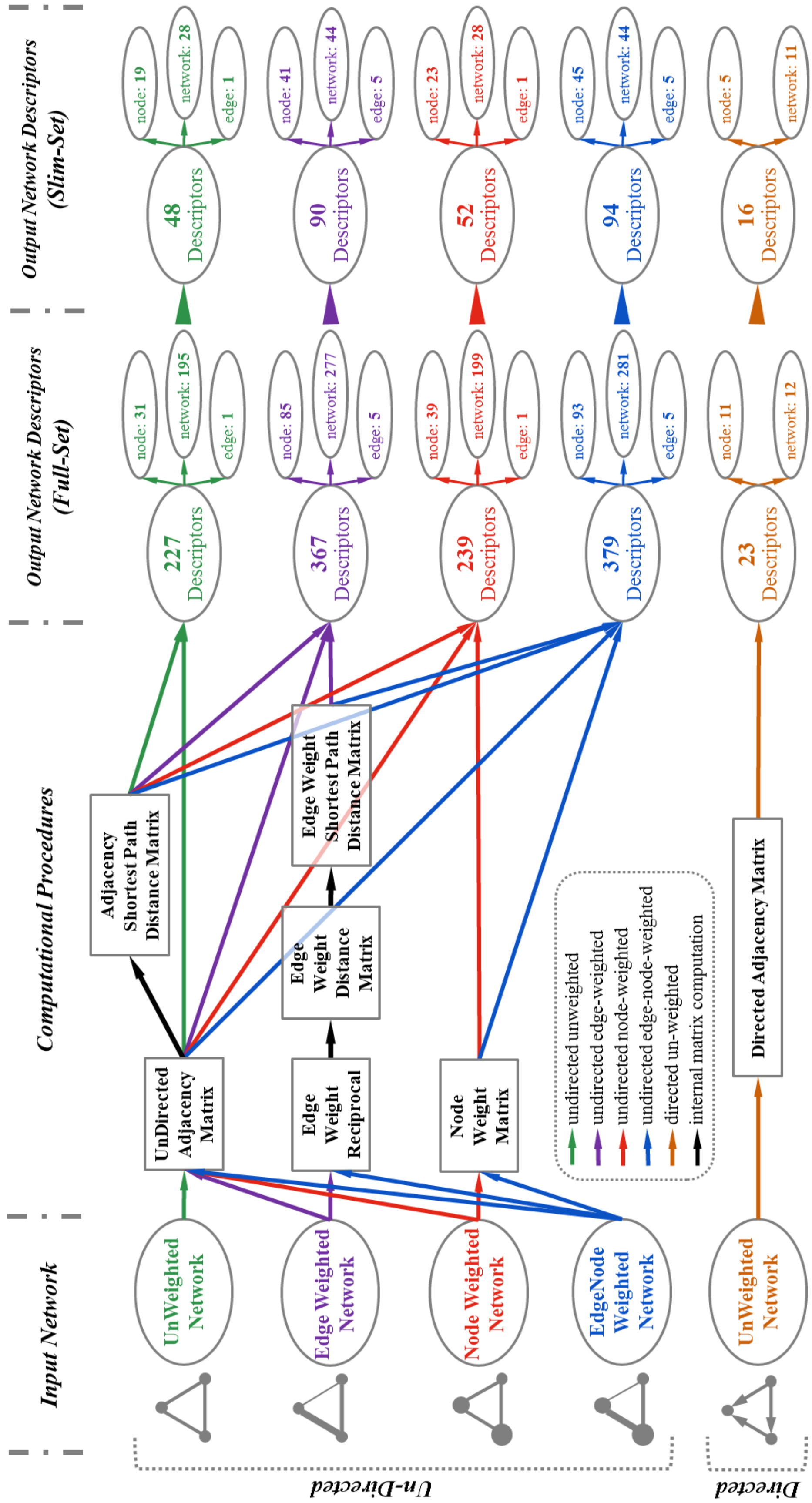
Biological Network Descriptor

Table of Contents

Prepared by: Zhang Peng (zhangpeng1202@gmail.com), 31 Dec 2016

(A) Computational Flowchart

Figure 1 | Computational flowchart for PROFEAT network descriptors



(B) List of Network Descriptors

Based on feature group indexing in PROFEAT, each network descriptor was indexed as (*X, Y, Z*), where node-level descriptors were indexed by *X=G10*, network-level descriptors were indexed by *X=G11*, and edge-level descriptors were indexed by *X=G12*. Next, each descriptor was labelled as un-weighted, edge-weighted, node-weighted, or directed by *Y=1, 2, 3, 4* respectively. The properties calculated by the normalized weight was labelled by an extra "*N*" in position of *Y*. Lastly, *Z* represented the descriptor ID# in the following **Table 1**, **2** and **3**.

For example:

- (*G10, 1, 7*): the node-level un-weighted neighbourhood connectivity
- (*G10, 2, 25*): the node-level edge-weighted betweenness centrality
- (*G10, 4, 49*): the node-level directed local clustering coefficient
- (*G11, 2N, 196*): the network-level normalized edge-weighted transitivity
- (*G11, 3, 202*): the network-level node-weighted global clustering coefficient
- (*G12, 2N, 2*): the edge-level normalized edge-weighted edge betweenness

In the tables below, all descriptors were grouped into different categories according to their definitions and algorithms, and each column listed the computed descriptors for each network type. Some descriptors could be defined by either un-weighted connectivity information or weighted information. Therefore, some notations were given: "○" (*Y=1*) represents the features calculated based on un-weighted network adjacency, "▬" (*Y=2*) represents the features calculated based on edge weight, "●" (*Y=3*) represents the features calculated based on node weight, and "➚" (*Y=4*) represents the features calculated based on directed information.

Additionally, a slim set of network descriptors were selected, which is a cut-down version of the PROFEAT network descriptors that have been particularly applied in studying systems biology. The descriptors in slim set were marked by "★" in the ID column.

Table 1 | List of the node-level descriptors covered in PROFEAT

ID	(G10) Node-Level Network Descriptor	Network Type				
		Un-Directed				Directed
		Un-Weighted	Edge Weighted	Node Weighted	EdgeNode Weighted	Un-Weighted
Connectivity/Adjacency-based Properties						
★1	Degree	○	○	○	○	
2	Scaled Connectivity	○	○	○	○	
★3	Number of Selfloops	○	○	○	○	➚
★4	Number of Triangles	○	○	○	○	➚
5	Z Score	○	○	○	○	
★6	Clustering Coefficient	○	○	○	○	
★7	Neighborhood Connectivity	○	○	○	○	
★8	Topological Coefficient	○	○	○	○	
★9	Interconnectivity	○	○	○	○	
★10	Bridging Coefficient	○	○	○	○	
★11	Degree Centrality	○	○	○	○	
Shortest Path Length-based Properties						
★12	Average Shortest Path Length	○	○ ━	○	○ ━	
13	Distance Sum	○	○ ━	○	○ ━	
★14	Eccentricity	○	○ ━	○	○ ━	
15	Eccentric	○	○ ━	○	○ ━	
16	Deviation	○	○ ━	○	○ ━	
17	Distance Deviation	○	○ ━	○	○ ━	
★18	Radiality	○	○ ━	○	○ ━	
★19	Closeness Centrality (avg)	○	○ ━	○	○ ━	
20	Closeness Centrality (sum)	○	○ ━	○	○ ━	
★21	Eccentricity Centrality	○	○ ━	○	○ ━	
22	Harmonic Closeness Centrality	○	○ ━	○	○ ━	
23	Residual Closeness Centrality	○	○ ━	○	○ ━	
★24	Load Centrality	○	○ ━	○	○ ━	
★25	Betweenness Centrality	○	○ ━	○	○ ━	
26	Normalized Betweenness	○	○ ━	○	○ ━	
★27	Bridging Centrality	○	○ ━	○	○ ━	
28	CurrentFlow Betweenness	○	○ ━	○	○ ━	
29	CurrentFlow Closeness	○	○ ━	○	○ ━	
Eigenvector-based Centrality Indices						
★30	Eigenvector Centrality	○	○	○	○	
★31	Page Rank Centrality	○	○	○	○	
Edge-Weighted Properties						
★32	Strength		━		━	
★33	Assortativity		━		━	
34	Disparity		━		━	
35	Geometric Mean of Triangles		━		━	
36	Barrat's Local Clustering Coefficient		━		━	
37	Onnela's Local Clustering Coefficient		━		━	
38	Zhang's Local Clustering Coefficient		━		━	
39	Holme's Local Clustering Coefficient		━		━	
★40	Edge-Weighted Interconnectivity		━		━	
Node-Weighted Properties						
★41	Node Weight			●	●	
42	Node Weighted Cross Degree			●	●	
43	Node Weighted Local Clustering Coeff.			●	●	
★44	Node-Weighted Neighbourhood Score			●	●	

	Directed Properties					
★45	In-Degree					↗
46	In-Degree Centrality					↗
★47	Out-Degree					↗
48	Out-Degree Centrality					↗
★49	Directed Local Clustering Coefficient					↗
50	Neighbourhood Connectivity (only in)					↗
51	Neighbourhood Connectivity (only out)					↗
52	Neighbourhood Connectivity (in & out)					↗
53	Average Directed Neighbour Degree					↗

Table 2 | List of the network-level descriptors covered in PROFEAT

ID	**(G11) Network-Level Network Descriptor**	**Network Type**				
		Un-Directed				**Directed**
		Un-Weighted	**Edge Weighted**	**Node Weighted**	**EdgeNode Weighted**	**Un-Weighted**
	Connectivity/Adjacency-based Properties					
★1	Number of Nodes	○	○	○	○	○
★2	Number of Edges	○	○	○	○	○
★3	Number of Selfloops	○	○	○	○	↗
★4	Maximum Connectivity	○	○	○	○	
★5	Minimum Connectivity	○	○	○	○	
★6	Average Number of Neighbours	○	○	○	○	
7	Total Adjacency	○	○	○	○	
★8	Network Density	○	○	○	○	↗
★9	Average Clustering Coefficient	○	○	○	○	
★10	Transitivity	○	○	○	○	
★11	Heterogeneity	○	○	○	○	
★12	Degree Centralization	○	○	○	○	
13	Central Point Dominance	○	○	○	○	
14	Degree Assortativity Coefficient	○	○	○	○	
	Shortest Path Length-based Properties					
15	Total Distance	○	○ —	○	○ —	
★16	Network Diameter	○	○ —	○	○ —	
★17	Network Radius	○	○ —	○	○ —	
18	Shape Coefficient	○	○ —	○	○ —	
★19	Characterisitc Path Length	○	○ —	○	○ —	
20	Network Eccentricity	○	○ —	○	○ —	
★21	Average Eccentricity	○	○ —	○	○ —	
22	Network Eccentric	○	○ —	○	○ —	
23	Eccentric Connectivity	○	○ —	○	○ —	
24	Unipolarity	○	○ —	○	○ —	
25	Integration	○	○ —	○	○ —	
26	Variation	○	○ —	○	○ —	
27	Average Distance	○	○ —	○	○ —	
28	Mean Distance Deviation	○	○ —	○	○ —	
29	Centralization	○	○ —	○	○ —	
★30	Global Efficiency	○	○ —	○	○ —	
	Topological Indices					
31	Edge Complexity Index	○	○	○	○	
★32	Randic Connectivity Index	○	○	○	○	

33	Atom-Bond Connectivity Index	○	○	○	○	
34	Zagreb Index 1	○	○	○	○	
35	Zagreb Index 2	○	○	○	○	
36	Zagreb Index Modified	○	○	○	○	
37	Zagreb Index Augmented	○	○	○	○	
38	Zagreb Index Variable	○	○	○	○	
39	Narumi-Katayama Index	○	○	○	○	
40	Narumi-Katayama Index (log)	○	○	○	○	
41	Narumi Geometric Index	○	○	○	○	
42	Narumi Harmonic Index	○	○	○	○	
43	Alpha Index	○	○	○	○	
44	Beta Index	○	○	○	○	
45	Pi Index	○	○	○	○	
46	Eta Index	○	○	○	○	
★47	Hierarchy	○	○	○	○	
★48	Robustness	○	○	○	○	
49	Medium Articulation	○	○	○	○	
50	Complexity Index A	○	○ —	○	○ —	
51	Complexity Index B	○	○ —	○	○ —	
★52	Wiener Index	○	○ —	○	○ —	
53	Hyper-Wiener	○	○ —	○	○ —	
54	Harary Index 1	○	○ —	○	○ —	
55	Harary Index 2	○	○ —	○	○ —	
56	Compactness Index	○	○ —	○	○ —	
57	Superpendentic Index	○	○ —	○	○ —	
58	Hyper-Distance-Path Index	○	○	○	○	
★59	BalabanJ Index	○	○ —	○	○ —	
60	BalabanJ-like 1 Index	○	○ —	○	○ —	
61	BalabanJ-like 2 Index	○	○ —	○	○ —	
62	BalabanJ-like 3 Index	○	○ —	○	○ —	
63	Geometric Arithmetic Index 1	○	○	○	○	
64	Geometric Arithmetic Index 2	○	○ —	○	○ —	
65	Geometric Arithmetic Index 3	○	○ —	○	○ —	
66	Szeged Index	○	○ —	○	○ —	
67	Product Of Row Sums	○	○ —	○	○ —	
68	Product Of Row Sums (log)	○	○ —	○	○ —	
69	Schultz Topological Index	○	○ —	○	○ —	
70	Gutman Topological Index	○	○ —	○	○ —	
71	Efficiency Complexity	○	○ —	○	○ —	
	Entropy-based Complexity Indices					
★72	Information Content (Degree Equality)	○	○	○	○	
73	Information Content (Edge Equality)	○	○	○	○	
74	Information Content (Edge Magnitude)	○	○	○	○	
75	Information Content (Distance Degree)	○	○	○	○	
76	Information Content (Distance Degree Equality)	○	○	○	○	
★77	Radial Centric Information Index	○	○	○	○	
78	Distance Degree Compactness	○	○	○	○	
79	Distance Degree Centric Index	○	○	○	○	
80	Graph Distance Complexity	○	○	○	○	
81	Information Layer Index	○	○	○	○	
★82	Bonchev Information Index 1	○	○	○	○	
★83	Bonchev Information Index 2	○	○	○	○	
★84	Bonchev Information Index 3	○	○	○	○	
85	Balaban-like Information Index 1	○	○	○	○	
86	Balaban-like Information Index 2	○	○	○	○	

Eigenvalue-based Complexity Indices						
★87	Graph Energy	○	○	○	○	
★88	Laplacian Energy	○	○	○	○	
89	Spectral Radius	○	○	○	○	
90	Estrada Index	○	○	○	○	
91	Laplacian Estrada Index	○	○	○	○	
92	Quasi-Weiner Index	○	○	○	○	
93	Mohar Index 1	○	○	○	○	
94	Mohar Index 2	○	○	○	○	
95	Graph Index Complexity	○	○	○	○	
96	Adjacency Matrix HM (S=1)	○	○	○	○	
97	Adjacency Matrix SM (S=1)	○	○	○	○	
98	Adjacency Matrix ISM (S=1)	○	○	○	○	
99	Adjacency Matrix PM (S=1)	○	○	○	○	
100	Adjacency Matrix IPM (S=1)	○	○	○	○	
101	Laplacian Matrix HM (S=1)	○	○	○	○	
102	Laplacian Matrix SM (S=1)	○	○	○	○	
103	Laplacian Matrix ISM (S=1)	○	○	○	○	
104	Laplacian Matrix PM (S=1)	○	○	○	○	
105	Laplacian Matrix IPM (S=1)	○	○	○	○	
106	Distance Matrix HM (S=1)	○	○	○	○	
107	Distance Matrix SM (S=1)	○	○	○	○	
108	Distance Matrix ISM (S=1)	○	○	○	○	
109	Distance Matrix PM (S=1)	○	○	○	○	
110	Distance Matrix IPM (S=1)	○	○	○	○	
111	Distance Path Matrix HM (S=1)	○	○	○	○	
112	Distance Path Matrix SM (S=1)	○	○	○	○	
113	Distance Path Matrix ISM (S=1)	○	○	○	○	
114	Distance Path Matrix PM (S=1)	○	○	○	○	
115	Distance Path Matrix IPM (S=1)	○	○	○	○	
116	Aug. Vertex Degree Matrix HM (S=1)	○	○	○	○	
117	Aug. Vertex Degree Matrix SM (S=1)	○	○	○	○	
118	Aug. Vertex Degree Matrix ISM (S=1)	○	○	○	○	
119	Aug. Vertex Degree Matrix PM (S=1)	○	○	○	○	
120	Aug. Vertex Degree Matrix IPM (S=1)	○	○	○	○	
121	Extended Adjacency Matrix HM (S=1)	○	○	○	○	
122	Extended Adjacency Matrix SM (S=1)	○	○	○	○	
123	Extended Adjacency Matrix ISM (S=1)	○	○	○	○	
124	Extended Adjacency Matrix PM (S=1)	○	○	○	○	
125	Extended Adjacency Matrix IPM (S=1)	○	○	○	○	
126	Vertex Connectivity Matrix HM (S=1)	○	○	○	○	
127	Vertex Connectivity Matrix SM (S=1)	○	○	○	○	
128	Vertex Connectivity Matrix ISM (S=1)	○	○	○	○	
129	Vertex Connectivity Matrix PM (S=1)	○	○	○	○	
130	Vertex Connectivity Matrix IPM (S=1)	○	○	○	○	
131	Random Walk Markov HM (S=1)	○	○	○	○	
132	Random Walk Markov SM (S=1)	○	○	○	○	
133	Random Walk Markov ISM (S=1)	○	○	○	○	
134	Random Walk Markov PM (S=1)	○	○	○	○	
135	Random Walk Markov IPM (S=1)	○	○	○	○	
136	Weighted Struct. Func. IM1 HM (S=1)	○	○	○	○	
137	Weighted Struct. Func. IM1 SM (S=1)	○	○	○	○	
138	Weighted Struct. Func. IM1 ISM (S=1)	○	○	○	○	
139	Weighted Struct. Func. IM1 PM (S=1)	○	○	○	○	
140	Weighted Struct. Func. IM1 IPM (S=1)	○	○	○	○	

141	Weighted Struct. Func. IM2 HM (S=1)	○	○	○	○	
142	Weighted Struct. Func. IM2 SM (S=1)	○	○	○	○	
143	Weighted Struct. Func. IM2 ISM (S=1)	○	○	○	○	
144	Weighted Struct. Func. IM2 PM (S=1)	○	○	○	○	
145	Weighted Struct. Func. IM2 IPM (S=1)	○	○	○	○	
146	Adjacency Matrix HM (S=2)	○	○	○	○	
147	Adjacency Matrix SM (S=2)	○	○	○	○	
148	Adjacency Matrix ISM (S=2)	○	○	○	○	
149	Adjacency Matrix PM (S=2)	○	○	○	○	
150	Adjacency Matrix IPM (S=2)	○	○	○	○	
151	Laplacian Matrix HM (S=2)	○	○	○	○	
152	Laplacian Matrix SM (S=2)	○	○	○	○	
153	Laplacian Matrix ISM (S=2)	○	○	○	○	
154	Laplacian Matrix PM (S=2)	○	○	○	○	
155	Laplacian Matrix IPM (S=2)	○	○	○	○	
156	Distance Matrix HM (S=2)	○	○	○	○	
157	Distance Matrix SM (S=2)	○	○	○	○	
158	Distance Matrix ISM (S=2)	○	○	○	○	
159	Distance Matrix PM (S=2)	○	○	○	○	
160	Distance Matrix IPM (S=2)	○	○	○	○	
161	Distance Path Matrix HM (S=2)	○	○	○	○	
162	Distance Path Matrix SM (S=2)	○	○	○	○	
163	Distance Path Matrix ISM (S=2)	○	○	○	○	
164	Distance Path Matrix PM (S=2)	○	○	○	○	
165	Distance Path Matrix IPM (S=2)	○	○	○	○	
166	Aug. Vertex Degree Matrix HM (S=2)	○	○	○	○	
167	Aug. Vertex Degree Matrix SM (S=2)	○	○	○	○	
168	Aug. Vertex Degree Matrix ISM (S=2)	○	○	○	○	
169	Aug. Vertex Degree Matrix PM (S=2)	○	○	○	○	
170	Aug. Vertex Degree Matrix IPM (S=2)	○	○	○	○	
171	Extended Adjacency Matrix HM (S=2)	○	○	○	○	
172	Extended Adjacency Matrix SM (S=2)	○	○	○	○	
173	Extended Adjacency Matrix ISM (S=2)	○	○	○	○	
174	Extended Adjacency Matrix PM (S=2)	○	○	○	○	
175	Extended Adjacency Matrix IPM (S=2)	○	○	○	○	
176	Vertex Connectivity Matrix HM (S=2)	○	○	○	○	
177	Vertex Connectivity Matrix SM (S=2)	○	○	○	○	
178	Vertex Connectivity Matrix ISM (S=2)	○	○	○	○	
179	Vertex Connectivity Matrix PM (S=2)	○	○	○	○	
180	Vertex Connectivity Matrix IPM (S=2)	○	○	○	○	
181	Random Walk Markov HM (S=2)	○	○	○	○	
182	Random Walk Markov SM (S=2)	○	○	○	○	
183	Random Walk Markov ISM (S=2)	○	○	○	○	
184	Random Walk Markov PM (S=2)	○	○	○	○	
185	Random Walk Markov IPM (S=2)	○	○	○	○	
186	Weighted Struct. Func. IM1 HM (S=2)	○	○	○	○	
187	Weighted Struct. Func. IM1 SM (S=2)	○	○	○	○	
188	Weighted Struct. Func. IM1 ISM (S=2)	○	○	○	○	
189	Weighted Struct. Func. IM1 PM (S=2)	○	○	○	○	
190	Weighted Struct. Func. IM1 IPM (S=2)	○	○	○	○	
191	Weighted Struct. Func. IM2 HM (S=2)	○	○	○	○	
192	Weighted Struct. Func. IM2 SM (S=2)	○	○	○	○	
193	Weighted Struct. Func. IM2 ISM (S=2)	○	○	○	○	
194	Weighted Struct. Func. IM2 PM (S=2)	○	○	○	○	
195	Weighted Struct. Func. IM2 IPM (S=2)	○	○	○	○	

Edge-Weighted Properties						
★196	Weighted Transitivity		▬		▬	
197	Barrat's Global Clustering Coefficient		▬		▬	
198	Onnela's Global Clustering Coefficient		▬		▬	
199	Zhang's Global Clustering Coefficient		▬		▬	
200	Holme's Global Clustering Coefficient		▬		▬	
Node-Weighted Properties						
201	Total Node Weight			●	●	
202	Node Weighted Global Clustering Coeff			●	●	
Directed Properties						
★203	Average In-Degree					➚
★204	Maximum In-Degree					➚
★205	Minimum In-Degree					➚
★206	Average Out-Degree					➚
★207	Maximum Out-Degree					➚
★208	Minimum Out-Degree					➚
★209	Directed Global Clustering Coefficient					➚
210	Directed Flow Hierarchy					➚

Table 3 | List of the edge-level descriptors covered in PROFEAT

ID	(G12) Edge-Level Network Descriptor	Network Type				
		Un-Directed				Directed
		Un-Weighted	Edge Weighted	Node Weighted	EdgeNode Weighted	Un-Weighted
★1	Edge Weight		○ ▬		○ ▬	
★2	Edge-Betweenness	○	○ ▬	○	○ ▬	

(C) Sample Input & Output

C.1 Overview of Input File Format

Currently, PROFEAT supports both SIF and NET network file format, where SIF is compatible with the majority of the network software (including Cytoscape [1], Gephi [2], GraphWeb [3], NAViGaTOR [4], PINA [5], SpectralNET [6]), and NET format is used in Pajek [7].

SIF Network File Format

SIF, namely Simple Interaction File, is tab-delimited, specifying the two linked nodes in each line, with the relationship type in between. The following example illustrates the unweighted SIF file, where the biological binary interaction network could be protein-protein interaction network, gene regulatory network, gene co-expression network, drug-target network, etc.

*[**Node A**] tab [**Relationship**] tab [**Node B**]*

Edge-weighted SIF is defined by extending the fourth column for numerical edge weight between the two connected nodes. In biological networks, the edge weight could be PPI kinetics constant, PPI binding affinity, gene co-expression association, interaction confidence level, etc.

*[**Node A**] tab [**Relationship**] tab [**Node B**] tab [**Edge Weight**]*

Directed SIF format is the same as the original SIF format, with the added direction information. For the two nodes in each line, the earlier one points to the latter one. Here, the example of unweighted SIF means that *Node A* points to *Node B* (*A* → *B*). Biologically, the directed network usually represents the oriented process map (e.g. signalling pathway, metabolic reaction, etc.).

NET Network File Format

NET format, developed by Pajek, mainly includes 3 sections (**vertices*, **edges*, and **arcs*) in this file structure, where (1) **vertices* section lists all the nodes; (2) **edges* section lists all the undirected interactions between two nodes, with an optional edge weight in the third column; and (3) **arcs* section lists all the directed interactions, pointing from the earlier node to the later node.

****vertices***
[Node A]
[Node B]
[Node C]
****edges***
[Node A]** tab **[Node C]
****arcs***
[Node B]** tab **[Node C]

The above example means there are 3 nodes (**vertices*) A, B, C in the network, where there is an undirected interaction (**edge*) between *A* and *C*, and a directed interaction (**arcs*) from *B* to *C*.

TXT Node Weight File Format

The node weight file is separated from the network file. It follows the tab-delimited txt format, specifying the node ID and its numerical node weight, while the node ID must be matched with the network file. Biologically, node weight represents the molecular level (e.g. gene expression).

[Node ID]** tab **[Node Weight]

Table 4 | Required file(s) for each input network type

Input Network Type	**Required File(s)**			
	Unweighted Network File	Edge-Weighted Network File	Node-Weight Text File	Directed Network File
Undirected Un-Weighted Network	✓			
Undirected Edge-Weighted Network		✓		
Undirected Node-Weighted Network	✓		✓	
Undirected Edge-Node-Weighted Network		✓	✓	
Directed Un-Weighted Network				✓

C.2 Overview of Output File Format

For an un-weighted input network, PROFEAT provides only the un-weighted descriptors. For a (edge, node, or both) weighted input network, PROFEAT will compute the un-weighted features, the original weighted features, and the normalized weighted features.

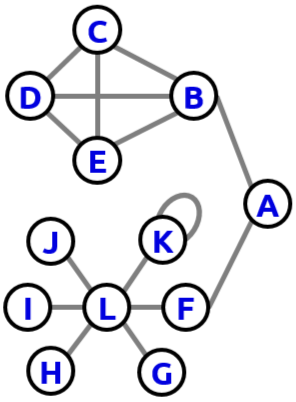
The output file is well organized in text format, by giving (1) a header started with "!" including the input network file name, the total number of networks, the total number of nodes, and the total number of edges; (2) the node-level descriptors; (3) the network-level descriptors; and (4) the edge-level descriptors respectively.

If there are multiple separated networks in the single input file, PROFEAT will automatically detect them, rank them, rename them, and compute the descriptors for each individual network accordingly. This function is embedded in all types of input networks. For such case study, please refer to the later section "Multiple Networks in Single Input File" for the details.

After submitting the job, a unique network id (*net-x*) and a URL will be given. Users could save the URL (*e.g.* *http://bidd2.nus.edu.sg/cgi-bin/profeat2016/network/profeat-result.cgi?uid=net-x*) for accessing the results later, in case that it may take some time to finish computing the large networks.

C.3 Undirected Un-Weighted Network

Table 5 | Sample input & output of an undirected un-weighted network

Sample Input		
Network Graphics	**Network in SIF**	**Network in NET**
	A interact B B interact C B interact D B interact E C interact D D interact E C interact E A interact F L interact F L interact K K interact K L interact J L interact I L interact H L interact G	*vertices A B C D E F G H I J K L *edges A B B C B D B E C D D E C E A F L F L K K K L J L I L H L G *arcs

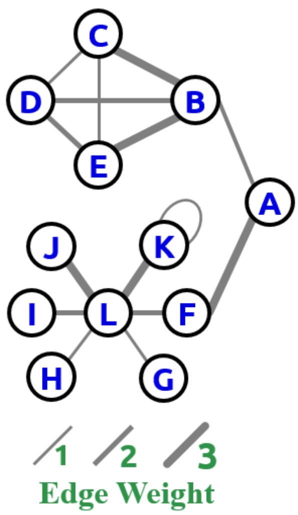
Sample Output

```
! Input Network File Name: sample_network.sif
! Total Number of Networks: 1
! Total Number of Nodes: 12
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:  sample_network.sif {12 Nodes; 15 Edges}
# # Node-Level Descriptors
[G10.0.0]        Node Label:                        A    B    …    L
[G10.1]          Un-Weighted Features
[G10.1.1]        Degree:                            2    4    …    6
      …                    …                        …    …    …    …
# # Network-Level Descriptors
[G11.1]          Un-Weighted Features
[G11.1.1]        Number of Nodes:                  12
[G11.1.2]        Number of Edges:                  15
      …                    …                        …
# # Edge-Level Descriptors
                                                  [G12.1.2]
NodeLabel 1      NodeLabel 2                      Edge Betweenness
A                B                                0.49
      …                    …                      …
```

C.4 Undirected Edge-Weighted Network

Table 6 | Sample input & output of an undirected edge-weighted network

Sample Input		
Network Graphics	**Network in SIF**	**Network in NET**
	A interact B 2 B interact C 3 B interact D 2 B interact E 3 C interact D 1 D interact E 2 C interact E 1 A interact F 3 L interact F 2 L interact K 3 K interact K 1 L interact J 3 L interact I 2 L interact H 1 L interact G 1	*vertices A B C D E F G H I J K L *edges A B 2 B C 3 B D 2 B E 3 C D 1 D E 2 C E 1 A F 3 L F 2 L K 3 K K 1 L J 3 L I 2 L H 1 L G 1 *arcs

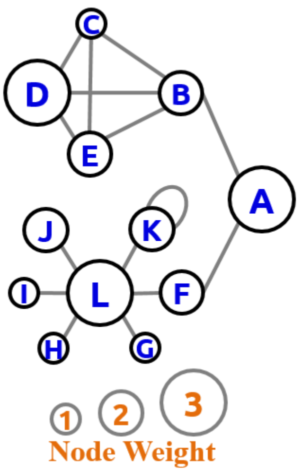
Sample Output

```
! Input Network File Name: sample_network_edge_weighted.sif
! Total Number of Networks: 1
! Total Number of Nodes: 12
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:   sample_network_edge_weighted.sif {12 Nodes; 15 Edges}
# # Node-Level Descriptors
[G10.0.0]        Node Label:                              A    B    ...   L
[G10.1]          Un-Weighted Features
[G10.1.1]        Degree:                                  2    4    ...   6
     ...                          ...                     ...  ...  ...   ...
[G10.2]          Orginal Edge-Weighted Features
[G10.2.11]       Edge-Weight Avg Shortest Path Length:   1.06 1.24  ...   1.0
     ...                          ...                     ...  ...  ...   ...
[G10.2N]         Normalized Edge-Weighted Features
[G10.2N.11]      N. Edge-Weight Avg Shortest Path Length: 0.46 0.56 ...  0.46
     ...                          ...                     ...  ...  ...   ...
# # Network-Level Descriptors
[G11.1]          Un-Weighted Features
[G11.1.1]        Number of Nodes:                         12
[G11.1.2]        Number of Edges:                         15
     ...                          ...                     ...
[G11.2]          Original Edge-Weighted Features
[G11.2.14]       Edge-Weight Total Distance:             93.0
     ...                          ...
[G11.2N]         Normalized Edge-Weighted Features
[G11.2N.14]      N. Edge-Weight Total Distance:          45.2
     ...                          ...                     ...
# # Edge-Level Descriptors
                                                         [G12.2.1]     ...  [G12.1.2]
NodeLabel 1      NodeLabel 2                             Edge Weight   ...  Edge Betweenness
A                B                                       2             ...  0.485
     ...                          ...                     ...          ...     ...
```

C.5 Undirected Node-Weighted Network

Table 7 | Sample input & output of an undirected node-weighted network

Sample Input			
Network Graphics	**Network in SIF**	**Network in NET**	**Node Weight**
	A interact B B interact C B interact D B interact E C interact D D interact E C interact E A interact F L interact F L interact K K interact K L interact J L interact I L interact H L interact G	*vertices A B C D E F G H I J K L *edges A B B C B D B E C D D E C E A F L F L K K K L J L I L H L G *arcs	A 3 B 2 C 1 D 3 E 2 F 2 G 1 H 1 I 1 J 2 K 2 L 3

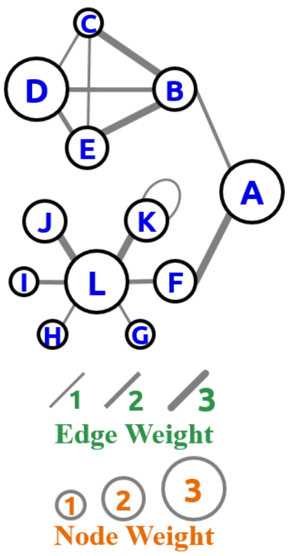
Sample Output

```
! Input Network File Name: sample_network.sif
! Input Node Weight File Name: sample_network_node_weighted.txt
! Total Number of Networks: 1
! Total Number of Nodes: 12
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:  sample_network.sif {12 Nodes; 15 Edges}
# # Node-Level Descriptors
[G10.0.0]        Node Label:                      A    B    ...   L
[G10.1]          Un-Weighted Features
[G10.1.1]        Degree:                          2    4    ...   6
    ...                 ...                       ...  ...  ...   ...
[G10.3]          Original Node-Weighted Features
[G10.3.38]       Node Weight:                     3    2    ...   3
    ...                 ...                       ...  ...  ...   ...
[G10.3N]         Normalized Node-Weighted Features
[G10.3N.38]      N. Node Weight:                  1    0.5  ...   1
    ...                 ...                       ...  ...  ...   ...
# # Network-Level Descriptors
[G11.1]          Un-Weighted Features
[G11.1.1]        Number of Nodes:                 12
[G11.1.2]        Number of Edges:                 15
    ...                 ...                       ...
[G11.3]          Original Node-Weighted Features
[G11.3.150]      Total Node Weight:               23
    ...                 ...                       ...
[G11.3N]         Normalized Node-Weighted Features
[G11.3N.150]     N. Total Node Weight:           5.53
    ...                 ...                       ...
# # Edge-Level Descriptors
                                                 [G12.1.2]
NodeLabel 1      NodeLabel 2                     Edge Betweenness
A                B                               0.49
    ...                 ...                      ...
```

C.6 Undirected Edge-Node-Weighted Network

Table 8 | Sample input & output of an undirected edge-node-weighted network

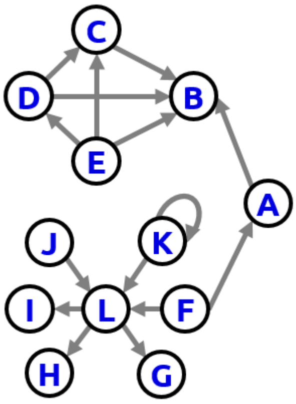
<table>
<tr><th colspan="4">Sample Input</th></tr>
<tr><th>Network Graphics</th><th>Network in SIF</th><th>Network in NET</th><th>Node Weight</th></tr>
<tr>
<td>
</td>
<td>A interact B 2
B interact C 3
B interact D 2
B interact E 3
C interact D 1
D interact E 2
C interact E 1
A interact F 3
L interact F 2
L interact K 3
K interact K 1
L interact J 3
L interact I 2
L interact H 1
L interact G 1</td>
<td>*vertices
A
B
C
D
E
F
G
H
I
J
K
L
*edges
A B 2
B C 3
B D 2
B E 3
C D 1
D E 2
C E 1
A F 3
L F 2
L K 3
K K 1
L J 3
L I 2
L H 1
L G 1
*arcs</td>
<td>A 3
B 2
C 1
D 3
E 2
F 2
G 1
H 1
I 1
J 2
K 2
L 3</td>
</tr>
<tr><th colspan="4">Sample Output</th></tr>
</table>

```
! Input Network File Name: sample_network_edge_weighted.sif
! Input Node Weight File Name: sample_network_node_weighted.txt
! Total Number of Networks: 1
! Total Number of Nodes: 12
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:   sample_network_edge_weighted.sif {12 Nodes; 15 Edges}
# # Node-Level Descriptors
[G10.0.0]         Node Label:                           A    B    …    L
[G10.1]           Un-Weighted Features
[G10.1.1]         Degree:                               2    4    …    6
      …                       …                         …    …    …    …
[G10.2]           Orginal Edge-Weighted Features
      …                       …                         …    …    …    …
[G10.3]           Orginal Node-Weighted Features
      …                       …                         …    …    …    …
[G10.2N]          Normalized Edge-Weighted Features
      …                       …                         …    …    …    …
[G10.3N]          Normalized Node-Weighted Features
      …                       …                         …    …    …    …
# # Network-Level Descriptors
[G11.1]           Un-Weighted Features
[G11.1.1]         Number of Nodes:                      12
[G11.1.2]         Number of Edges:                      15
      …                       …                         …
[G11.2]           Original Edge-Weighted Features
      …                       …                         …
[G11.3]           Original Node-Weighted Features
      …                       …                         …
[G11.2N]          Normalized Edge-Weighted Features
      …                       …                         …
[G11.3N]          Normalized Node-Weighted Features
      …                       …                         …
# # Edge-Level Descriptors
                                                       [G12.2.1]       …   [G12.1.2]
NodeLabel 1       NodeLabel 2                          Edge Weight     …   Edge Betweenness
A                 B                                    2               …   0.485
      …                       …                         …              …      …
```

C.7 Directed Un-Weighted Network

Table 9 | Sample input & output of an directed un-weighted network

Sample Input		
Network Graphics	**Network in SIF**	**Network in NET**
	A direct-to B C direct-to B D direct-to B E direct-to B D direct-to C E direct-to D E direct-to C F direct-to A F direct-to L K direct-to L K direct-to K J direct-to L L direct-to I L direct-to H L direct-to G	*vertices A B C D E F G H I J K L *edges *arcs A B C B D B E B D C E D E C F A F L K L K K J L L I L H L G

Sample Output

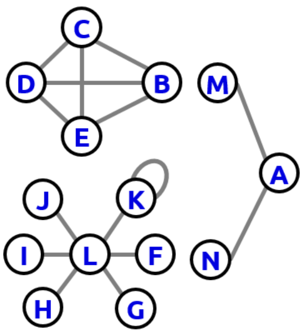
```
! Input Network File Name: sample_network_directed.sif
! Total Number of Networks: 1
! Total Number of Nodes: 12
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:  sample_network_directed.sif {12 Nodes; 15 Edges}
# # Node-Level Descriptors
[G10.0.0]        Node Label:                 A    B    ...   L
[G10.4]          Directed Features
[G10.4.41]       In-Degree:                  1    4    ...   3
[G10.4.42]       Out-Degree:                 1    0    ...   3
      ...                    ...             ...  ...  ...   ...
# # Network-Level Descriptors
[G11.4]          Directed Features
[G11.4.1]        Number of Nodes:            12
[G11.4.2]        Number of Edges:            15
      ...                    ...             ...
```

C.8 Multiple Networks in Single Input File

Quantitative network analysis may get trouble by having mixed networks in data collection. The available tools have not yet provided the function to split the disconnected network from a single input. To illustrate this function, "*sample_network_multiple.sif*", containing 3 separated networks, is inputted. PROFEAT analyses the global adjacency, splits the input into 3 new files, ranks by the number of nodes, and renames by adding suffix. Finally, each network is proceed for descriptor calculation accordingly.

Table 10 | Sample input & output of a single file containing disconnected networks

Sample Input		
Network Graphics	**Network in SIF**	**Network in NET**
	A interact M B interact C B interact D B interact E C interact D D interact E C interact E A interact N L interact F L interact K K interact K L interact J L interact I L interact H L interact G	*vertices A B C D E F G H I J K L M N *edges A M B C B D B E C D D E C E A N L F L K K K L J L I L H L G *arcs

Sample Output

```
! Input Network File Name: sample_network_multiple.sif
! Total Number of Networks: 3
! Total Number of Nodes: 14
! Total Number of Edges: 15
! PROFEAT Network Descriptor: Slim Set

# Network File:   sample_network_multiple_sub_1.sif {7 Nodes; 7 Edges}
# # Node-Level Descriptors
[G10.0.0]         Node Label:                        F    G    …    L
[G10.1]           Un-Weighted Features
       …                           …                 …    …    …    …
# # Network-Level Descriptors
[G11.1]           Un-Weighted Features
       …                           …                 …
# # Edge-Level Descriptors
       …                           …                 …


# Network File:   sample_network_multiple_sub_2.sif {4 Nodes; 6 Edges}
# # Node-Level Descriptors
[G10.0.0]         Node Label:                        B    C    D    E
[G10.1]           Un-Weighted Features
       …                           …                 …    …    …    …
# # Network-Level Descriptors
[G11.1]           Un-Weighted Features
       …                           …                 …
# # Edge-Level Descriptors
       …                           …                 …


# Network File:   sample_network_multiple_sub_3.sif {3 Nodes; 2 Edges}
# # Node-Level Descriptors
[G10.0.0]         Node Label:                        A    M    N
[G10.1]           Un-Weighted Features
       …                           …                 …    …    …
# # Network-Level Descriptors
[G11.1]           Un-Weighted Features
       …                           …                 …
# # Edge-Level Descriptors
       …                           …                 …
```

(D) Concepts and Algorithms of Network Descriptors

For a connected and undirected network, some basic information matrices will be generated:

1. Un-weighted matrix
   1.1. Adjacency matrix "A", with $A_{ij}=A_{ij}=1$, if exists an edge linking node i and node j. Otherwise, $A_{ij}=A_{ij}=0$.
2. Edge-weight matrix
   2.1. Edge weight matrix "EW", assigning $EW_{ij}=EW_{ji}=$ edge weight between node i and j.
   2.2. Normalized edge weight matrix "$NorEW$", defined as below. Here, the constant factor 0.99 in the denominator is to slightly enlarge the domain from the minimum value to the maximum value, ensuring the normalized minimum edge weight will not be zero.

$$NorEW_{ij} = \frac{EW_{ij} - \min\{EW\}}{\max\{EW\} - 0.99 * \min\{EW\}}$$

3. Node-weighted matrix
   3.1. Node weight list "NW", where NW_i = node weight of node i, based on the input data.
   3.2. Normalized node weight list "$NorNW$". Again, the constant 0.99 in the denominator is to ensure the normalized minimum node weight will not be zero.

$$NorNW_i = \frac{NW_i - min\{NW\}}{max\{NW\} - 0.99 * min\{NW\}}$$

For a connected and directed network, directed adjacency matrix will be generated:

4. Un-weighted matrix
   4.1. Directed adjacency matrix "a", where $a_{ij}=1$, if exists a directed link from node i pointing to node j. $a_{ji}=1$ only if exists another directed link from node j pointing to node i.

The network descriptors will be introduced according to their order in **Table 1**, **2**, and **3** given previously. As some descriptors can be derived from either un-weighted adjacency matrix or weighted matrix, we will mainly introduce the un-weighted ones, and the weighted ones can be easily obtained by substituting the algorithm with the weighted matrix.

D.1 Node-Level Descriptors

Feature Category: Connectivity/Adjacency-based Properties

1. Degree

Degree of a node i "deg_i" is the number of edges linked to it.

2. Scaled Connectivity

$$scaledConnect_i = \frac{deg_i}{max\{deg_G\}}$$

3. Number of Selfloops

Selfloops of a node i "$selfloop_i$" is the number of edges linking to itself.

4. Number of Triangles [8]

$$tri_i = \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} A_{ij}A_{ik}A_{jk}$$

5. Z Score [9,10]

Z score is a connectivity index of a node, based on the degree distribution of a network. It has been applied in discovering network motifs in some studies.

$$zscore_i = \frac{deg_i - avg\{deg_G\}}{dev\{deg_G\}}$$

6. Clustering Coefficient [11,12]

The clustering coefficient of a node i is defined as below, where e_i is the number of connected pairs between all neighbours of node i. It is assumed to be 0, if less than two neighbours.

$$cluster_i = \frac{2e_i}{deg_i(deg_i - 1)}$$

7. Neighborhood Connectivity [13]

The connectivity of a node is the number of its neighbours. The neighbourhood connectivity of a node i is defined as its average connectivity of all neighbours.

$$neighbourConnect_i = \frac{\sum_{j=1}^{N} A_{ij} \cdot deg_j}{deg_i}$$

8. Topological Coefficient [14]

In calculating topological coefficient, j represents all the nodes sharing at least one neighbour with i, and $J(i, j)$ is the number of shared neighbours between i and j. If there is a direct edge between i and j, plus an additional 1 to $J(i, j)$. It is a measure to estimate the tendency of the nodes to share neighbours.

$$topology_i = avg\{J(i,j)/deg_i\}$$

9. Interconnectivity [15,16,17]

Firstly, the interconnectivity score is generated for each edge in the network. *N(i)* is the neighbours of node *i*, such that |*N(i)* ∩ *N(j)*| is the number of shared neighbours between node *i* and node *j*.

$$ICN\_edge_{ij} = A_{ij} \cdot \left(\frac{2 + |N(i) \cap N(j)|}{\sqrt{deg_i \cdot deg_j}} \right)$$

Next, the interconnectivity for each node is calculated based on the *ICN_edge* scores.

$$ICN\_node_i = \frac{1}{deg_i} \sum_{j=1}^{N} ICN\_edge_{ij}$$

10. Bridging Coefficient [18]

The bridging coefficient describes how well the node is linked between high-degree nodes.

$$bridge_i = \frac{{deg_i}^{-1}}{\sum_{j=1}^{N} A_{ij} \cdot \frac{1}{deg_j}}$$

11. Degree Centrality [19]

$$centralityDeg_i = \frac{deg_i}{N - 1}$$

<u>**Feature Category: Shortest Path Length-based Properties**</u>

12. Average Shortest Path Length [20]

Shortest path lengths are computed by Dijkstra's algorithm to generate an *NxN* matrix for storing the pairwise shortest path lengths, such that D_{ij} is the shortest path length between node *i* and node *j*. For an unweighted network, the shortest path length is basically the minimum number of edges linking between any two nodes. For an edge-weighted network, the weighted shortest path length could be generated based on the edge weight matrix. Here, $avgSPL_i$ is the average length of shorest paths between node *i* and all other nodes.

$$avgSPL_i = \frac{1}{N - 1} \sum_{j=1}^{N} D_{ij}$$

13. Distance Sum [21]

Distance sum is obtained by adding up all the shortest paths from node *i*.

$$distSum_i = \sum_{j=1}^{N} D_{ij}$$

14. Eccentricity [21]

Eccentricity is the maximum non-infinite shortest path length between node *i* and all the other nodes.

$$eccentricity_i = max\,\{D_{ij}\}$$

15. Eccentric [21]

Different from eccentricity measure, eccentric index is the absolute difference between the nodes' eccentricities and the graph's average eccentricity.

$$eccentric_i = |eccentricity_i - avg\,\{eccentricity_G\}|$$

16. Deviation [21]

Node's deviation measures the difference between the node's distance sum and the graph's unipolarity, where the unipolarity is defined as the minimum of distance sums among all nodes.

$$deviation_i = distSum_i - unipolarity_G$$

17. Distance Deviation [21]

This is the absolute difference between nodes' distance sum and graph's average distance.

$$distDev_i = |distSum_i - distAvg_G|$$

18. Radiality [22]

Radiality is computed by subtracting the average shortest path length of node *i* from the diameter plus 1, and the result is then divided by the network diameter.

High value of radiality implies the node is generally nearer to other nodes, while a low radiality indicates the node is peripheral in the network.

$$radiality_i = \frac{diameter_G\ -\ avgSPL_i\ +\ 1}{diameter_G}$$

19. Closeness Centrality (avg) [22,23,24]

The closeness centrality of a node is defined as the reciprocal of the average shortest path length. It measures how fast information spreads from a given node to other reachable nodes in the network.

$$centralityCloseAvg_i = \frac{1}{\frac{1}{N}\sum_{j=1}^{N} D_{ij}}$$

20. Closeness Centrality (sum)

$$centralityCloseSum_i = \frac{1}{\sum_{j=1}^{N} D_{ij}}$$

21. Eccentricity Centrality

$$centralityEccentricity_i = \frac{1}{max\{D_{ij}\}}$$

22. Harmonic Centrality [25]

The harmonic closeness is the sum of reciprocals of average shortest path lengths for each node.

$$centralityHar_i = \sum_{j=1}^{N} \frac{1}{D_{ij}}$$

23. Residual Centrality [26]

$$centralityRes_i = \sum_{j=1}^{N} \frac{1}{2^{D_{ij}}}$$

24. Load Centrality [22,27]

The load centrality of a node *i* is the fraction of all shortest paths that passing through the node *i*. A node has a high load centrality if it is involved in a high number of shortest paths.

25. Betweenness Centrality [22,28]

The betweenness centrality quantifies the number of times a node serving as a linking bridge along the shortest path between two other nodes. It is computed by the following equation, where *s* and *t* are the nodes different from *i* in the network, $\sigma_{st}(i)$ is the number of shortest paths from *s* to *t* that passing through *i,* and σ_{st} is the number of shortest paths from *s* to *t*. The betweenness centrality reflects the extent of control of that node exerting over the interactions with other nodes in the network.

$$centralityBtw_i = \frac{\sum_{s \neq i \neq t} \sigma_{st}(i)}{\sigma_{st}}$$

26. Normalized Betweenness Centrality

$$centralityBtwNor_i = \frac{centralityBtw_i - \ min\{centralityBtw_G\}}{max\{centralityBtw_G\} - \ min\{centralityBtw_G\}}$$

27. Bridging Centrality [18]

The bridging centrality of a node is the product of the bridging coefficient and the betweenness centrality. A higher bridging centrality means more information flowing through that node.

$$centralityBridge_i = \ bridge_i \cdot centralityBtw_i$$

28. Current Flow Betweenness [29,30,31]

Previously, the betweenness centrality is based on the shortest path length in the network. Here, the current flow betweenness centrality is assumed that information efficiently spreading in the network like an electrical current, as a current flow analog.

Firstly, the resistance *R* of an edge is defined, where $r(e)=1/w(e)$ and $w(e)$ is the weight of an edge *e*. For unweighted networks, $w(e)=1$ for all edges.

Secondarily, a vector *b*, namely supply, is defined where current enters and leaves the network. Since there should be as much current entering as leaving the network, $\sum b(v)=0$.

$$b_{st}(v) = \begin{cases} 1, & \text{v=s} \\ \text{-1}, & \text{v=t} \\ 0, & \text{otherwise} \end{cases}$$

Thirdly, the electrical current *c* is defined and it should follow the law below.

Kirchhoff's Current Law (for every $v \in V$):

$$\sum_{(v,w)\in E} c(v,w) - \sum_{(u,v)\in E} c(u,v) = b(v)$$

Kirchhoff's Potential Law (for every current cycle $e_i \ldots e_k$ in the network):

$$\sum_{i=1}^{k} c(e_i) = 0$$

Lastly, the potential difference *p* is defined by *Ohm's Law*, where $p(e)=c(e)/r(e)$. To calculate the current flow betweenness, throughput $\tau(v)$ of a node *v*, and throughput $\tau(e)$ of an edge *e* are defined:

$$\tau(v) = \frac{1}{2}\left(-|b(v)| + \sum_{e} |c(e)|\right)$$

$$\tau(e) = |c(e)|$$

Therefore, current flow betweenness (sometimes also called random-walk betweenness) is then defined, where τ_{st} denotes the throughput of a *s-t* current, and $N_b = (N\text{-}1)(N\text{-}2)$.

$$CFbetween_i = \frac{1}{N_b} \sum_{s,t\epsilon V} \tau_{st}(i)$$

29. Current Flow Closeness [29,30,31]

The current flow closeness centrality is a variant of the current flow betweenness centrality, by using the analog of shortest path length in electrical networks.

$$CFclose_i = \frac{N_c}{\sum_{s\neq t} p_{st}(s) - p_{st}(t)}$$

Where, $N_c = (N\text{-}1)$, and $p_{st}(s)\text{-}p_{st}(t)$ denotes the effective resistance of *s-t* current, interpreted as an alternative measure of distance between node *s* and node *t*.

Feature Category: Eigenvector-based Centrality Indices

30. Eigenvector Centrality [32,33]

Eigenvector centrality is the eigenvalue-based methods to approximate the importance of each node in a network. It assumes that each node's centrality is the sum of its neighbors' centrality values, which is saying that an important node should be linking to important neighbors.

In algorithm, the eigenvector centralities for all nodes are initialized to 1 at the beginning, and then an eigenvalue-based function is applied to iteratively converge the centrality to a fixed value, by considering the neighbourhood relationships and the neighbors' centrality values. Let $\{\lambda_1, \lambda_2 \ldots \lambda_k\}$ be the non-zero eigenvalues of adjacency matrix of the network, and λ_{max} is the maximum eigenvalue.

$$centralityEigen_i = \frac{1}{\lambda_{max}} \sum_{j=1}^{N} A_{ij} \cdot centralityEigen_j$$

31. PageRank Centrality [34,35,36,37,38,39]

PageRank is an algorithm implemented in Google search engine to rank the websites, according to the webpage connections in the World Wide Web. It is a variant of eigenvector centrality, by initializing the PageRank centralities to an equal probability value *1/N* for all nodes.

The equation will iteratively update the node centrality value by using a constant damping factor *d*, its neighbors' PageRank centrality value, and its degree. The algorithm stops running, when the PageRank centrality converges, and the damping factor *d* is generally assumed to 0.85.

$$pageRank_i = \frac{1-d}{N} + d \cdot \sum_{j=1}^{N} A_{ij} \cdot \frac{pageRank_j}{deg_j}$$

Feature Category: Edge-Weighted Properties

32. Strength [40]

The strength for each vertex is defined as the sum of all the edge weights connected to that vertex.

$$strength_i = \sum_{j=1}^{N} A_{ij} \cdot W_{ij}$$

33. Assortativity [40,41]

In an unweighted graph, assotativity is as the same as the previously defined neighbourhood connectivity. For a weighted graph, it is defined as below.

$$assortativity_i = \frac{1}{strength_i} \sum_{j=1}^{N} W_{ij} \cdot deg_j$$

34. Disparity [42]

$$disparity_i = \sum_{j=1}^{N} \left(\frac{A_{ij} \cdot W_{ij}}{strength_i} \right)^2$$

35. Geometric Mean of Triangles [8]

$$geo\_tri_i = \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} \sqrt[3]{W_{ij} W_{ik} W_{jk}}$$

36. Barrat's Local Clustering Coefficients [43]

$$clusterBarrat_i = \frac{1}{strength_i (deg_i - 1)} \sum_{j=1}^{N} \sum_{k=1}^{N} \left(A_{ij} A_{ik} A_{jk} \cdot \frac{W_{ij} + W_{ik}}{2} \right)$$

37. Onnela's Local Clustering Coefficients [43,44]

$$clusterOnnela_i = \frac{1}{deg_i \cdot (deg_i - 1)} \sum_{j=1}^{N} \sum_{k=1}^{N} \left(\widehat{W_{ij}} \widehat{W_{ik}} \widehat{W_{jk}} \right)^{1/3}$$

$$\widehat{W_{ij}} = \frac{W_{ij}}{max\{W\}}$$

38. Zhang's Local Clustering Coefficients [43,45]

$$clusterZhang_i = \frac{\sum_{j=1}^{N} \sum_{k=1}^{N} \widehat{W_{ij}} \widehat{W_{ik}} \widehat{W_{jk}}}{\left(\sum_{k=1}^{N} \widehat{W_{ij}} \right)^2 - \sum_{k=1}^{N} \widehat{W_{ij}}^{\,2}}$$

39. Holme's Local Clustering Coefficients [43,46]

$$clusterHolme_i = \frac{\sum_{j=1}^{N} \sum_{k=1}^{N} \widehat{W_{ij}} \widehat{W_{ik}} \widehat{W_{jk}}}{max\{W\} \cdot \sum_{j=1}^{N} \sum_{k=1}^{N} \widehat{W_{ij}} \widehat{W_{ik}}}$$

40. Edge-Weighted Interconnectivity [16]

The edge-weighted interconnectivity is defined similarly with (#9) the unweighted interconnectivity. Firstly, the interconnectivity score for each edge is calculated.

$$EW\_ICN\_edge_{ij} = \frac{2W_{ij} + \sum_{u \in N(i) \cap N(j)} W_{iu} W_{ju}}{\sqrt{strength_i \cdot strength_j}}$$

Where, W_{ij} is the weight of the edge linking node i and node j, and the previously defined $strength_i$ is the sum of weights of connected edges to node i.

Next, the edge-weighted interconnectivity for each node is calculated based on *EW_ICN_edge* scores.

$$EW\_ICN\_node_i = \frac{1}{deg_i} \sum_{j=1}^{N} EW\_ICN\_edge_{ij}$$

Feature Category: Node-Weighted Properties

41. Node Weight

The node weight NW_i is directly extracted from the node weight matrix generated.

42. Node Weighted Cross Degree [47]

For analyzing networks with heterogeneous node weights, the next two node-weighted informative measures were derived recently for the economic trading network study. In the definition, *ExtA* is the extended adjacency matrix, where $ExtA_{ij} = A_{ij} + \delta_{ij}$, and δ_{ij} is Kronecker's delta constant.

$$\delta_{ij} = \begin{cases} 0, & if\ i \neq j \\ 1, & if\ i = j \end{cases}$$

$$NWcrossdeg_i = \sum_{j=1}^{N} ExtA_{ij} \cdot NW_i$$

43. Node Weighted Local Clustering Coefficient [47]

This node-weighted local clustering coefficient works, only if the node-weighted cross degree is not zero, otherwise the local clustering coefficient will be assumed as zero.

$$NWcluster_i = \frac{1}{NWcrossdeg_i{}^2} \sum_{j=1}^{N} \sum_{k=1}^{N} ExtA_{ij} \cdot NW_j \cdot ExtA_{ik} \cdot NW_k \cdot ExtA_{jk}$$

44. Node-Weighted Neighbourhood Score [15]

This score is defined in a study of disease-gene networks, by assigning the fold change of gene expression as the node weight. Below, *neighbour(i)* denotes all the neighbours of node *i* in the network.

$$NWneighbourhood_i = \frac{1}{2} NW_i + \frac{1}{2} \cdot \frac{\sum_{j \epsilon neighbour(i)} NW_j}{|neighbour(i)|}$$

Feature Category: Directed Properties

45. In-Degree [1,8]

As previously mentioned, "*A*" represents the undirected adjacency matrix and "*a*" represents the directed adjacency matrix, where a_{ij}=1 means a directed edge has node *i* points to node *j*. In-degree of a node counts the number of directed edges pointing to itself.

$$deg_i{}^{+} = \sum_{j \epsilon N} a_{ji}$$

46. In-Degree Centrality

The in-degree centrality for a node is the fraction of nodes its incoming edges are connected to.

47. Out-Degree [1,8]

Out-degree of a node counts the number of directed edges pointing out of itself.

$$deg_i{}^{-} = \sum_{j \epsilon N} a_{ij}$$

48. Out-Degree Centrality

The out-degree centrality for a node is the fraction of nodes its outgoing edges are connected to.

49. Directed Local Clustering Coefficient [1]

In directed networks, local clustering coefficient is defined slightly different from undirected one.

$$cluster_i{}^{\pm} = \frac{e_i}{(deg_i{}^{+} + deg_i{}^{-})(deg_i{}^{+} + deg_i{}^{-} - 1)}$$

50. Neighbourhood Connectivity (only in) [1]

It is the average out-connectivity of all in-neighbours of node i.

$$neighbourConnectivity_i{}^{+} = \frac{\sum_{j \in N} a_{ji} \cdot deg_j{}^{-}}{\sum_{j \in N} a_{ji}}$$

51. Neighbourhood Connectivity (only out) [1]

It is the average in-connectivity of all out-neighbours of node i.

$$neighbourConnectivity_i{}^{-} = \frac{\sum_{j \in N} a_{ij} \cdot deg_j{}^{+}}{\sum_{j \in N} a_{ij}}$$

52. Neighbourhood Connectivity (in & out) [1]

It is the average connectivity of all neighbours of node i, where the direction is ignored here.

$$neighbourConnectivity_i{}^{\pm} = \frac{\sum_{j \in N} a_{ij} \cdot (deg_j{}^{+} + deg_j{}^{-}) + \sum_{j \in N} a_{ji} \cdot (deg_j{}^{+} + deg_j{}^{-})}{\sum_{j \in N} a_{ji} + \sum_{j \in N} a_{ij}}$$

53. Average Directed Neighbour Degree [8]

$$avgDirectedNeighbourDeg_i{}^{\pm} = \frac{\sum_{j \in N}[(a_{ij} + a_{ji}) \cdot (deg_j{}^{+} + deg_j{}^{-})]}{2 \cdot (deg_j{}^{+} + deg_j{}^{-})}$$

D.2 Network-Level Descriptors

<u>Feature Category: Connectivity/Adjacency-based Properties</u>

1. Number of Nodes

The number of the nodes (or vertices) in the network, noted as *N*.

2. Number of Edges

The number of edges (or links) in the network, noted as *E*.

3. Number of Selfloops

$$selfloops_G = \sum_{i=1}^{N} selfloop_i$$

4. Maximum Connectivity

$$connectivityMax_G = max\{deg_G\}$$

5. Minimum Connectivity

$$connectivityMin_G = min\{deg_G\}$$

6. Average Number of Neighbors

The average of the number of neighbours (or degree, connectivity) for all nodes.

$$neighbourAvg_G = \frac{1}{N}\sum_{i=1}^{N} deg_i$$

7. Total Adjacency [48]

The total adjacency is the half of the sum of the adjacency matrix entries.

$$totalAdjacency_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}$$

8. Network Density [48]

The network density measures the efficiency of the information progression in a network in time. The denominator *N*(N-1)/2* is the maximum number of links if the network is completely connected. For a directed network, the denominator is *N*(N-1)*.

$$density_G = \frac{E}{N(N-1)/2}$$

9. Global Clustering Coefficient [11,12]

Network clustering coefficient is the average of all the node-level clustering coefficients.

$$cluster_G = \frac{1}{N}\sum_{i=1}^{N} cluster_i$$

10. Transitivity [8]

Transitivity is calculated based on the number of triangles for each node in the network.

$$transitivity_G = \frac{2 * \sum_{i=1}^{N} tri_i}{\sum_{i=1}^{N} deg_i(deg_i - 1)}$$

11. Heterogeneity [49]

Heterogeneity measures the variation of degree distribution, reflecting the tendency of a network to have hubs. This index is biologically meaningful, as biological networks are usually heterogeneous with some central nodes highly connected and the rest nodes having few connections in the network.

$$heterogeneity_G = \sqrt{\frac{N \cdot \sum_{i=1}^{N}\left(deg_i{}^2\right)}{\left(\sum_{i=1}^{N} deg_i\right)^2} - 1}$$

12. Degree Centralization [49]

Degree centralization (or connectivity centralization) is useful for distinguishing such characteristics as highly connected networks (e.g. star-shaped) or decentralized networks, which have been used for studying the structural differences of metabolic networks.

$$centralizationDeg_G = \frac{N}{N-2}\left(\frac{connectivityMax_G}{N-1} - density_G\right)$$

13. Central Point Dominance [50]

Central point dominance is defined based on the measure of betweenness centrality.

$$centralDominance_G = \frac{1}{N-1}\sum_{i=1}^{N}(max\{centralityBtw_i\} - centralityBtw_i)$$

14. Degree Assortativity Coefficient [51]

It measures the similarity of degree with respect to each edge in the network, by calculating the standard Pearson correlation coefficient between the degrees of the two connecting vertices of each edge. Its value lies in between -1 and 1, where 1 represents perfect assotativity and -1 indicates perfect dissortativity.

<u>Feature Category: Shortest Path Length-based Properties</u>

15. Total Distance [48]

It is the sum of all the non-redundant pairwise shortest path distances in the network.

$$totalDistance_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij}$$

16. Network Diameter [1]

The network diameter is the largest distance in shorest path length matrix.

$$diameter_G = max\{D_{ij}\}$$

17. Network Radius [1]

The network radius is the smallest distance in shorest path length matrix.

$$radius_G = min\{D_{ij}\}$$

18. Shape Coefficient [52]

The shape coefficient of a network is defined by its radius and its diameter.

$$shapeCoef_G = \frac{diameter_G - radius_G}{radius_G}$$

19. Characterisitc Path Length [1]

The characteristic path length is the average distance in shorest path length matrix.

$$CPL_G = \frac{\sum_{i=1}^{N} avgSPL_i}{N}$$

20. Network Eccentricity [21]

$$eccentricity_G = \sum_{i=1}^{N} eccentricity_i$$

21. Average Eccentricity [21]

$$eccentricityAvg_G = \frac{eccentricity_G}{N}$$

22. Network Eccentric [21]

$$eccentric_G = \frac{1}{N}\sum_{i=1}^{N} eccentric_i$$

23. Eccentric Connectivity [53]

It is defined as the sum of the product of eccentricity and degree of each node, it has been shown the high correlation with regard to physical properties of diverse nature in various datasets.

$$eccentricConnect_G = \sum_{i=1}^{N} eccentric_i \cdot deg_i$$

24. Unipolarity [21]

It measures the minimal distance sum, which is the sum of shortest path lengths for each node.

$$unipolarity_G = min\{distSum_i\}$$

25. Integration [21]

It is the sum of all the nodes' distance sum, where each shortest path is counted once.

$$integration_G = \frac{1}{2}\sum_{i=1}^{N} distSum_i$$

26. Variation [21]

The network variation is defined as the maximum variance in the node-level measures.

$$variation_G = max\{deviation_i\}$$

27. Average Distance [21]

This measures the mean shortest path length by dividing the integration by the number of nodes.

$$distAvg_G = \frac{2 \cdot integration_G}{N}$$

28. Mean Distance Deviation [21]

This mean distance deviation is to average the node-level distance deviation values.

$$distDevMean_G = \frac{1}{N}\sum_{i=1}^{N} distDev_i$$

29. Centralization [21]

This centralization descriptor sums the variance value for all nodes in the network.

$$centrailization_G = \sum_{i=1}^{N} deviation_i$$

30. Global Efficiency [54]

The global efficiency is a measure of the information exchange efficiency across the entire network. It can be used to determine the cost-effectiveness of the network structure.

$$efficiency_G = \frac{1}{N(N-1)}\sum_{i\neq j}^{N} \frac{1}{D_{ij}}$$

<u>**Feature Category: Topological Indices**</u>

31. Edge Complexity Index [48]

The global edge complexity is defined by dividing the total adjacency by N^2.

$$edgeComplexity_G = \frac{totalAdjacency_G}{N^2}$$

32. Randic Connectivity Index [55]

The randic index is a function of the connectivity of edges.

$$randic_G = \sum_{E_{i,j}\in G} \left(deg_i \cdot deg_j\right)^{-\frac{1}{2}}$$

33. Atom-Bond Connectivity Index [56]

The ABC index is a graph-invariant measure, which has been applied to study the stability of chemical structure. Here, it is used to describe the stability of a network structure.

$$ABC_G = \sum_{E_{i,j} \in G} \left(\frac{deg_i + deg_j - 2}{deg_i \cdot deg_j} \right)^{\frac{1}{2}}$$

34. Zagreb Index 1 [57,58,59,60]

There are five Zagreb indices variants are defined based on the nodes' degree.

$$zagreb1_G = \sum_{i=1}^{N} {deg_i}^2$$

35. Zagreb Index 2

$$zagreb2_G = \sum_{E_{i,j} \in G} deg_i \cdot deg_j$$

36. Modified Zagreb Index

$$zagrebModified_G = \sum_{E_{i,j} \in G} \frac{1}{deg_i \cdot deg_j}$$

37. Augmented Zagreb Index

$$zagrebAugmented_G = \sum_{E_{i,j} \in G} \left(\frac{deg_i \cdot deg_j}{deg_i + deg_j - 2} \right)^3$$

38. Variable Zagreb Index

$$zagrebVariable_G = \sum_{E_{i,j} \in G} \frac{deg_i + deg_j - 2}{deg_i \cdot deg_j}$$

39. Narumi-Katayama Index [61]

The NK index is the product of degrees of all nodes. It has been shown the relationships with thermodynamics properties. Additionally, its logged index, geometric index, and harmonic Index are provided as follows. In our program, if Narumi index goes beyond *sys.maxsize*, then Narumi Index and Narumi Geometric Index will be assigned as zero.

$$narumi_G = \prod_{i=1}^{N} deg_i$$

40. Narumi-Katayama Index (log)

$$narumiLog_G = log_2 \left(\prod_{i=1}^{N} deg_i \right)$$

41. Narumi Geometric Index [62]

$$narumiGeo_G = \left(\prod_{i=1}^{N} deg_i\right)^{\frac{1}{N}}$$

42. Narumi Harmonic Index [62]

$$narumiHar_G = \frac{N}{\sum_{i=1}^{N}(deg_i)^{-1}}$$

43. Alpha Index [10]

Alpha index is a connectivity measure to evaluate the number of cycles in a network in comparison with maximum number of cycles, such that the higher alpha index, the more connected nodes. Trees and simple networks have alpha index equal to zero, and a completely connected network have alpha index equal to 1. Sometimes, alpha index is named as Meshedness Coefficient.

$$alpha_G = \frac{E - N}{\frac{N(N-1)}{2} - (N-1)}$$

44. Beta Index [10]

It measures the graph connectivity, by the ratio of the number of edges over the number of nodes. Simple networks have beta value less than 1, and more complex networks have higher beta index.

$$beta_G = \frac{E}{N}$$

45. Pi Index [10]

Pi is the relationship between the total length of the network and its diameter. Namely Pi index, it has a similar meaning with the definition of π, indicating of the shape of the network.

$$pi_G = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}}{diameter_G}$$

46. Eta Index [10]

Eta index is the average adjacency per edge. Adding nodes will result in decreasing of eta index.

$$eta_G = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}}{E}$$

47. Hierarchy [10]

Hierarchy index is the gradient of the linear power-law regression, by fitting $\log_{10}$ (node frequency) over $\log_{10}$ (degree distribution). It usually has the value between 1 and 2, where the low hierarchy indicates the weak hierarchical relationship.

Hierarchy is notated as h in the fitted regression equation $y=ax^h$, where x is the degree distribution and y is the node frequency of that specific degree.

$$y = a \cdot x^{hierachy}$$

48. Robustness [63]

Robustness is to measure the stability of a network under node-removal attacks. By removing each node, the size of the largest fragmented component S is used to define the robustness.

$$robustness_G = \frac{\sum_{k=1}^{N} S_k}{N(N-1)}$$

49. Medium Articulation [64,65,66]

Medium articulation MA is a complexity measure of a network, reaching its maximum with medium number of edges. It is defined based on the redundancy (MA_R) and the mutual information (MA_I).

$$MA_G = MA_R \cdot MA_I$$

Redundancy MA_R is defined as:

$$MA_R = 4\left(\frac{R - R_{path}}{R_{clique} - R_{path}}\right)\left(1 - \frac{R - R_{path}}{R_{clique} - R_{path}}\right)$$

$$R = \frac{1}{E}\sum_{i=1}^{N}\sum_{j>i}^{N} log_{10}\left(deg_i \cdot deg_j\right)$$

$$R_{clique} = 2 \cdot log_{10}(N-1)$$

$$R_{path} = 2 \cdot \frac{N-2}{N-1} log_{10} 2$$

Mutual information MA_I is defined as:

$$MA_I = 4\left(\frac{I - I_{clique}}{I_{path} - I_{clique}}\right)\left(1 - \frac{I - I_{clique}}{I_{path} - I_{clique}}\right)$$

$$I = \frac{1}{E}\sum_{i=1}^{N}\sum_{j>i}^{N} log_{10} \frac{2\,E}{deg_i \cdot deg_j}$$

$$I_{clique} = log_{10}(\frac{N}{N-1})$$

$$I_{path} = log_{10}(N-1) - \frac{N-3}{N-1} log_{10} 2$$

50. Complexity Index A [48]

It is the ratio of total adjacency and the total distance of a network.

$$complexityA_G = \frac{totalAdjacency_G}{totalDistance_G}$$

51. Complexity Index B [48]

It is defined by the ratio of vertex degree and its distance sum for each vertex.

$$complexityB_G = \sum_{i=1}^{N} \frac{deg_i}{distSum_i}$$

52. Wiener Index [67]

The Wiener index measures the sum of the shortest path lengths between all pairs of vertices.

$$wiener_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij}$$

53. Hyper-Wiener Index [68]

$$hyperWiener_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left({D_{ij}}^2 + D_{ij}\right)$$

54. Harary Index 1 [69]

$$harary1_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} {D_{ij}}^{-1}$$

55. Harary Index 2 [69]

$$harary2_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} {D_{ij}}^{-2}$$

56. Compactness [70]

This measure is based on Wiener index, by dividing the Wiener index by *N(N-1)*.

$$compactness_G = \frac{4 \cdot wiener_G}{N(N-1)}$$

57. Superpendentic Index [71]

$$superpendentic_G = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij}\right)^{\frac{1}{2}}$$

58. Hyper-Distance-Path Index [72,73]

This index is consist of two parts: the exactly Wiener index, and the delta number.

$$hyper\_path_G = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij} + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\binom{D_{ij}}{2}$$

59. BalabanJ Index [74]

This BalabanJ index counts into the distance sum of the two end-vertex for each edge. BalabanJ index has been proven to be relevant to the network branching. There are another three differently defined variants of BalabanJ indices are given in the followings.

$$Jm_G = \frac{E}{\mu + 1}\sum_{E_{i,j} \in G}\left(disSum_i \cdot disSum_j\right)^{-\frac{1}{2}}$$

Where, $\mu = E + 1 - N$, which denotes the cyclomatic number of a graph.

60. BalabanJ-Like Index 1 [75]

$$Jm1_G = \frac{E}{\mu + 1}\sum_{E_{i,j} \in G}\left(disSum_i \cdot disSum_j\right)^{\frac{1}{2}}$$

61. BalabanJ-Like Index 2 [75]

$$Jm2_G = \frac{E}{\mu + 1} \sum_{E_{i,j} \in G} \left(disSum_i + disSum_j\right)^{\frac{1}{2}}$$

62. BalabanJ-Like Index 3 [75]

$$Jm3_G = \frac{E}{\mu + 1} \sum_{E_{i,j} \in G} \left(\frac{disSum_i \cdot disSum_j}{disSum_i + disSum_j}\right)^{\frac{1}{2}}$$

63. Geometric Arithmetic Index 1 [58,76]

GA index consists of the geometrical and arithmetic means of the end-to-end degree of an edge.

$$GA1_G = \sum_{E_{i,j} \in G} \frac{2\sqrt{deg_i \cdot deg_j}}{deg_i + deg_j}$$

64. Geometric Arithmetic Index 2 [58,76]

There are 2 extended geometric-arithmetic indices, which make use of the information of the shortest path lengths. In some studies, the geometric-arithmetic indices have shown its power in characterizing the network structure features.

$$GA2_G = \sum_{E_{i,j} \in G} \frac{2\sqrt{n_i \cdot n_j}}{(n_i + n_j)}$$

$$n_i := \left|x \in node(G), D_{xi} < D_{xj}\right|$$

$$n_j := |x \in node(G), D_{xj} < D_{xi}|$$

In the definition of geometric arithmetic index 2 (GA2), x is a node, n_i is the number of nodes closer to node i, and n_j is the number of nodes closer to node j, while the nodes with same distance to node i and node j are ignored.

65. Geometric Arithmetic Index 3 [58,76]

$$GA3_G = \sum_{E_{i,j} \in G} \frac{2\sqrt{m_i \cdot m_j}}{(m_i + m_j)}$$

$$m_i := \left|y \in edge(G), D_{yi} < D_{yj}\right|$$

$$m_j := |y \in edge(G), D_{yi} < D_{yi}|$$

In the definition of geometric arithmetic index 3 (GA3), y is an edge in the graph, the distance between edge y to node i is defined as $D_{yi} = min\ \{D_{pi}, D_{qi}\}$, where p and q are the two ends of edge y. In the context above, m_i is number of edges closer to node i and m_j is the number of edges closer to node j, while the edges with same distance to node i and node j are not counted.

66. Szeged Index [77]

$$szeged_G = \sum_{E_{i,j} \in G} n_i \cdot n_j$$

Where n_i and n_j are as same defined as the previous geometric-arithmetic index 2.

67. Product of Row Sums [78]

If PRS is greater than *sys.maxsize*, it will be assigned as zero in the program.

$$PRS_G = \prod_{i=1}^{N} distSum_i$$

68. Product of Row Sums (log)

$$PRSLog_G = log_2\left(\prod_{i=1}^{N} distSum_i\right)$$

69. Schultz Topological Index [79]

By using adjacency matrix *A*, shortest path distance matrix *D*, and the vertex degree vector *v*, Schultz defined a topological index to described the network structure.

In the equation below, *(D+A)* forms an addictive *NxN* matrix, and this matrix is then multiplied by a *1xN* vector *v*, such that obtaining another *1xN* vector. The sum of all the elements in the resultant vector is called the Schultz topological index.

$$schultz_G = \sum_{i=1}^{N} [v(D+A)]_i$$

70. Gutman Topological Index [80]

Gutman topological index is a further defined Schultz index, where ADA is the matrix multiplication.

$$gutman_G = \sum_{i=1}^{N}\sum_{j=1}^{N} [ADA]_{ij}$$

71. Efficiency Complexity [64,65,66]

The efficiency complexity is motivated in analyzing the weighted networks, as it suggests to measure not only the shortest path lengths but also the cost (number of links).

$$EC_G = 4\left(\frac{E - E_{path}}{1 - E_{path}}\right)\left(1 - \frac{E - E_{path}}{1 - E_{path}}\right)$$

$$E = \frac{2}{N(N-1)}\sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{D(i,j)}$$

$$E_{path} = \frac{2}{N(N-1)}\sum_{i=1}^{N}\left(N - \frac{N-i}{i}\right)$$

Feature Category: Entropy-Based Complexity Indices

72. Information Content (Degree Equality) [81]

This information content measures the probability distribution of vertex degree, where $N^d{}_i$ is the number of nodes having the same degree, and k^d is the maximum of degree.

$$I_{vertexDegree} = -\sum_{i=1}^{k^d} \frac{N^d{}_i}{N} \cdot log_2\left(\frac{N^d{}_i}{N}\right)$$

73. Information Content (Edge Equality) [82]

This measure is based on the probability distribution of edge connectivity, where each edge has an end-to-end connectivity value. Let *(a, b)* and $a \leq b$ be the edge's end-to-end connectivity, such that the edges having the same edge connectivity will be grouped into the same subset.

$$I_{edgeEquality} = -\sum_{i=1}^{k^{edge}} \frac{E_i}{E} \cdot log_2\left(\frac{E_i}{E}\right)$$

Where, E_i is the number of edges having the same end-to-end connectivity, and k^{edge} is the number of different edge subsets.

74. Information Content (Edge Magnitude) [82]

As another measure based on the edge information, it is defined by the connectivity magnitude of each edge, and $randic_G$ is the network-level randic connectivity index introduced previously.

$$I_{edgeMagnitude} = -\sum_{E_{i,j}\in G} \frac{\left(deg_i \cdot deg_j\right)^{-1/2}}{randic_G} \cdot log_2\left(\frac{\left(deg_i \cdot deg_j\right)^{-1/2}}{randic_G}\right)$$

75. Information Content (Distance Degree) [81]

The distance degree of a node *i* is equivalently the distance sum $distSum_i$ defined previously.

$$I_{distanceDegree} = -\sum_{i=1}^{N} \frac{distSum_i}{2 \cdot Weiner_G} \cdot log_2\left(\frac{distSum_i}{2 \cdot Weiner_G}\right)$$

76. Information Content (Distance Degree Equality) [81]

The probability distribution regarding on the nodes' distance degree value gives the definition of the information content on distance degree equality. As below, k^{dd} is the number of node groups in the distribution of distance degree, $N^{dd}{}_i$ is the number of nodes having the same distance degree.

$$I_{distanceDegreeEquality} = -\sum_{i=1}^{k^{dd}} \frac{N^{dd}{}_i}{N} \cdot log_2\left(\frac{N^{dd}{}_i}{N}\right)$$

77. Radial Centric Information Index [81]

It is measuring the probability distribution of vertex eccentricity. In the definition below, $N^e{}_i$ is the number of nodes having the equal eccentricity value *i*, and k^e is the maximum of eccentricity.

$$I_{radialCentric} = -\sum_{i=1}^{k^e} \frac{N^e{}_i}{N} \cdot log_2\left(\frac{N^e{}_i}{N}\right)$$

78. Distance Degree Compactness [83]

This measure is defined based on the distribution of nodes' locations from the centre of a network, where the centre is determined by the closeness centrality score in this case. Here, Q_k is the sum of distance degree of all nodes that located at the same topological distance k from the centre.

$$I_{compactness} = 2Weiner_G \cdot log_2(2Weiner_G) - \sum_k Q_k \cdot log_2(Q_k)$$

79. Distance Degree Centric Index [84]

$$I_{distanceDegreeCentric} = -\sum_{i=1}^{K^c} \frac{N_i}{N} log_2 \frac{N_i}{N}$$

Where N_i is the number of nodes having the same eccentricity and the same degree, K^c is the number of equivalent classes of N_i.

80. Graph Distance Complexity [85]

As a similar definition as $I_{infoLayer}$, this distance complexity includes the nodes' distance sums.

$$I_{distanceComplexity} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{ecc_i} N^i{}_j \cdot \frac{j}{distSum_i} \cdot log_2\left(\frac{j}{distSum_i}\right)$$

81. Information Layer Index [86]

$$I_{infoLayer} = -\sum_{i=1}^{N}\sum_{j=1}^{ecc_i} \frac{N^i{}_j}{N} \cdot log_2\left(\frac{N^i{}_j}{N}\right)$$

Where, ecc_i is the eccentricity value of node i, and N^i_j is the number of nodes in the j^{th} sphere of node i. In other words, N^i_j is the number of nodes in shortest distance j away from node i.

82. Bochev Information Index 1 [87,88]

Bochev indices applies the probability distribution of shortest path lengths to Shannon's entropy formula, and it has three variants as follows. $diameter_G$ is the maximum distance between two nodes in the network, and k_i is the occurrence of distance i in the shortest path length matrix D_{ij}.

$$I_{bochev1} = -\frac{1}{N} \cdot log_2\left(\frac{1}{N}\right) - \sum_{i=1}^{diameter_G} \frac{2k_i}{N^2} \cdot log_2\left(\frac{2k_i}{N^2}\right)$$

83. Bochev Information Index 2 [87,88]

$$I_{bochev2} = -Weiner_G \cdot log_2(Weiner_G) - \sum_{i=1}^{diameter_G} i \cdot k_i \cdot log_2(i)$$

84. Bochev Information Index 3 [87,88]

$$I_{bochev3} = -\sum_{i=1}^{diameter_G} \frac{2k_i}{N(N-1)} \cdot log_2\left(\frac{2k_i}{N(N-1)}\right)$$

85. Balaban-like Information Index 1 [89,90]

Differently from BalabanJ indices, Balaban-like information index 1 & 2 are defined based on the distribution of distance degree in the network. In the equation below, g_k is the number of nodes at distance k from node i, and μ is namely the cyclomatic number.

$$I_{balaban1} = \frac{E}{\mu+1} \sum_{E_{i,j} \in G} [u_i \cdot u_j]^{-1/2}$$

$$u_i = -\sum_{k=1}^{dimeter} \frac{k \cdot g_k}{distSum_k} \cdot log_2\left(\frac{k}{distSum_k}\right)$$

$$\mu = E + 1 - N$$

86. Balaban-like Information Index 2 [89,90]

$$I_{balaban2} = \frac{E}{\mu+1} \sum_{E_{i,j} \in G} [v_i \cdot v_j]^{-1/2}$$

$$v_i = distSum_i \cdot log_2(distSum_i) - u_i$$

<u>**Feature Category: Eigenvalue-Based Complexity Indices**</u>

87. Graph Energy [91]

Given a network, let $\{\lambda_1, \lambda_2 \ldots \lambda_k\}$ be the non-zero eigenvalues of its adjacency matrix, such that k is the number of eigenvalues and λ_{max} is the maximum of the eigenvalues

$$Energy_G = \sum_{i=1}^{k} |\lambda_i|$$

88. Laplacian Energy [91]

Laplacian matrix L_{ij} is generated based on the degree and the adjacency relationships, as below. Such that, Laplacian matrix produces $\mu_i : \{\mu_1, \mu_2 \ldots, \mu_k\}$ as the Laplacian eigenvalues of the network.

$$L_{ij} = \begin{cases} -1 & if\ A_{ij} = 1 \\ deg_i & if\ i = j \\ 0 & otherwise \end{cases}$$

$$LaplacianEnergy_G = \sum_{i=1}^{k} \left|\mu_i - \frac{2E}{N}\right|$$

89. Spectral Radius [92]

$$SpRadius_G = max\{|\lambda_i|\}$$

90. Estrada Index [93]

$$Estrada_G = \sum_{i=1}^{k} e^{\lambda_i}$$

91. Laplacian Estrada Index [94]

$$LaplacianEstrada_G = \sum_{i=1}^{k} e^{\mu_i}$$

92. Quasi-Wiener Index [95]

Quasi-Wiener is defined by Laplacian eigenvalues. As the last eigenvalue μ_k is always zero, it is excluded.

$$quasiWeiner_G = N \sum_{i=1}^{k-1} \frac{1}{\mu_i}$$

93. Mohar Index 1 [73,96]

$$mohar1_G = \frac{1}{N} \cdot quasiWeiner_G \cdot log_2\left(\sum_{i=1}^{k-1} \mu_i\right)$$

94. Mohar Index 2 [73,96]

$$mohar2_G = \frac{4}{N \cdot \mu_{k-1}}$$

95. Graph Index Complexity [64]

$$Cr_G = 4 \cdot cr \cdot (1 - cr)$$

$$cr = \frac{\lambda_{max} - 2\cos\frac{\pi}{N+1}}{N - 1 - 2\cos\frac{\pi}{N+1}}$$

96 - 195. A Set of Eigenvalue-Based Descriptors from Variants of Matrices [97,98]

There are 5 novel eigenvalue-based descriptors recently introduced, namely HM_G, SM_G, ISM_G, PM_G, and IPM_G. Let M be a re-defined matrix based on the given graph G, and $\{\lambda_1, \lambda_2 \ldots \lambda_k\}$ be its non-zero eigenvalues. As the factor "*s*" may have different discrimination power for different networks, we thus provide these eigenvalue-based descriptors at both $s = 1$ and $s = 2$.

$$HM_G = -\sum_{i=1}^{k}\left[\frac{|\lambda_i|^{\frac{1}{s}}}{\sum_{j=1}^{k}|\lambda_i|^{\frac{1}{s}}} log_2\left(\frac{|\lambda_i|^{\frac{1}{s}}}{\sum_{j=1}^{k}|\lambda_i|^{\frac{1}{s}}}\right)\right]$$

$$SM_G = \sum_{i=1}^{k} |\lambda_i|^{\frac{1}{s}}$$

$$ISM_G = \frac{1}{\sum_{i=1}^{k}|\lambda_i|^{\frac{1}{s}}}$$

$$PM_G = \prod_{i=1}^{k} |\lambda_i|^{\frac{1}{s}}$$

$$IPM_G = \frac{1}{\prod_{i=1}^{k}|\lambda_i|^{\frac{1}{s}}}$$

These 5 eigenvalue-based descriptors could be applied to the following 10 differently re-defined matrices, including (1) adjacency matrix, (2) laplacian matrix, (3) distance matrix, (4) distance path matrix, (5) augmented vertex degree matrix, (6) extended adjacency matrix, (7) vertex connectivity matrix, (8) random walk Markov matrix, (9) weighted structure function matrix 1, and (10) weighted structure function matrix 2, which are defined as follows.

Therefore, totally 50 eigenvalue-based descriptors are calculated in this set.

(1) Adjacency matrix A_{ij}, is the initially generated based on the connections of the network.

(2) Laplacian matrix L_{ij}, is introduced previously in the definition of Laplacian energy.

(3) Distance matrix D_{ij}, is the shortest distance between all the nodes.

(4) Distance path matrix DP_{ij}, is derived from the distance matrix, by counting all the internal paths between a pair of nodes, including their shortest paths.

$$DP_{ij} = \binom{D_{ij}+1}{2}$$

(5) Augmented vertex degree matrix AVD_{ij}, is defined by the nodes' degree and distance.

$$AVD_{ij} = \frac{deg_j}{2^{D_{ij}}}$$

(6) Extended adjacency matrix EA_{ij}, is a symmetric matrix based on the nodes' degree.

$$EA_{ij} = \begin{cases} \frac{1}{2}\left(\frac{deg_i}{deg_j} + \frac{deg_j}{deg_i}\right) & \text{if } A_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}$$

(7) Vertex connectivity matrix VC_{ij}, is another symmetric matrix based on the nodes' degree.

$$VC_{ij} = \begin{cases} \frac{1}{\sqrt{deg_i \cdot deg_j}} & \text{if } A_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}$$

(8) Radom walk Markov matrix RWM_{ij}, is a non-symmetric matrix based on the nodes' degree. It is based on the assumption that each neighbour node can be reached from a given node with the same probability, such that the probability of reaching the neighbor of node i is $1/deg_i$. The generated distribution of walks is called the simple random walks.

$$RWM_{ij} = \begin{cases} \frac{1}{deg_i} & \text{if } A_{ij} = 1 \\ 0 & \text{otherwise} \end{cases}$$

(9) Weighted structure function matrix 1 $IM1_{ij}$, is a more complexly defined matrix. In the following definitions, $radius_G$ is the maximum shortest path length in the network, and $|S_d(i)|$ is the number of nodes that are at the shortest distance d away from the node i.

$$f1(i) = \sum_{d=1}^{radius_G} (radius_G + 1 - d) \cdot |S_d(i)|$$

$$pf1(i) = \frac{f1(i)}{\sum_{j=1}^{N} f1(j)}$$

$$IM1_{ij} = 1 - \frac{|pf1(i) - pf1(j)|}{2^{D_{ij}}}$$

(10) Weighted structure function matrix 2 $IM2_{ij}$, is slight differently defined as below.

$$f2(i) = \sum_{d=1}^{radius_G} (radius_G \cdot e^{1-d}) \cdot |S_d(i)|$$

$$pf2(i) = \frac{f2(i)}{\sum_{j=1}^{N} f2(j)}$$

$$IM2_{ij} = 1 - \frac{|pf2(i) - pf2(j)|}{2^{D_{ij}}}$$

Feature Category: Edge-Weighted Properties

196. Weighted Transitivity [8]

$$weighted\_transitivity_G = \frac{\sum_{i=1}^{N} geo\_tri_i}{\sum_{i=1}^{N} deg_i(deg_i - 1)}$$

197. Barrat's Global Clustering Coefficients [43]

$$clusterBarrat_G = \frac{1}{N}\sum_{i=1}^{N} clusterBarrat_i$$

198. Onnela's Global Clustering Coefficients [43,44]

$$clusterOnnela_G = \frac{1}{N}\sum_{i=1}^{N} clusterOnnela_i$$

199. Zhang's Global Clustering Coefficients [43,45]

$$clusterZhang_G = \frac{1}{N}\sum_{i=1}^{N} clusterZhang_i$$

200. Holme's Global Clustering Coefficients [43,46]

$$clusterHolme_G = \frac{1}{N}\sum_{i=1}^{N} clusterHolme_i$$

Feature Category: Node-Weighted Properties

201. Total Node Weight

$$total\_NW_G = \sum_{i=1}^{N} NW_i$$

202. Node Weighted Global Clustering Coefficient [47]

$$NWcluster_G = \frac{1}{N}\sum_{i=1}^{N} NWcluster_i$$

Feature Category: Directed Properties

203. Average In-Degree

$$avg\_deg_G{}^{+} = \frac{1}{N}\sum_{i\epsilon N} deg_i{}^{+}$$

204. Maximum In-Degree

$$max\_deg_G{}^{+} = \max\{deg_i{}^{+}\}$$

205. Minimum In-Degree

$$min\_deg_G{}^{+} = \min\{deg_i{}^{+}\}$$

206. Average Out-Degree

$$avg\_deg_G{}^{-} = \frac{1}{N}\sum_{i\epsilon N} deg_i{}^{-}$$

207. Maximum Out-Degree

$$max\_deg_G{}^{-} = \max\{deg_i{}^{-}\}$$

208. Minimum Out-Degree

$$min\_deg_G{}^{-} = \min\{deg_i{}^{-}\}$$

209. Directed Global Clustering Coefficient [1]

$$cluster_G{}^{\pm} = \frac{1}{N}\sum_{i\in N} cluster_i{}^{\pm}$$

230. Directed Flow Hierarchy [99]

Flow hierarchy is a measurement of the percentage of edges that not involved in any directed cycles in the directed network.

D.3 Edge-Level Descriptors

1. Edge Weight

The edge weight EW_i is directly extracted from the user-provided edge weight list.

2. Edge Betweenness [22,100]

Similarly with the definition of the node-level betweenness centrality. The edge betweenness quantifies the number of times an edge serving as a linking bridge along the shortest path between two nodes. In the following equation, node s and node t are two different nodes in the network, $\sigma_{st}(e)$ is the number of shortest paths from s to t that passing through the edge e, and σ_{st} is the number of shortest paths from node s to node t.

$$edgeBetweenness_e = \frac{\sum_{s \neq t} \sigma_{st}(e)}{\sigma_{st}}$$

(E) Computational Time Cost

The CPU time of PROFEAT was evaluated in computing the slim set of network descriptors for 10 human tissue-specific protein-protein-interaction (PPI) networks in five different network types and various sizes (**Table 11**).

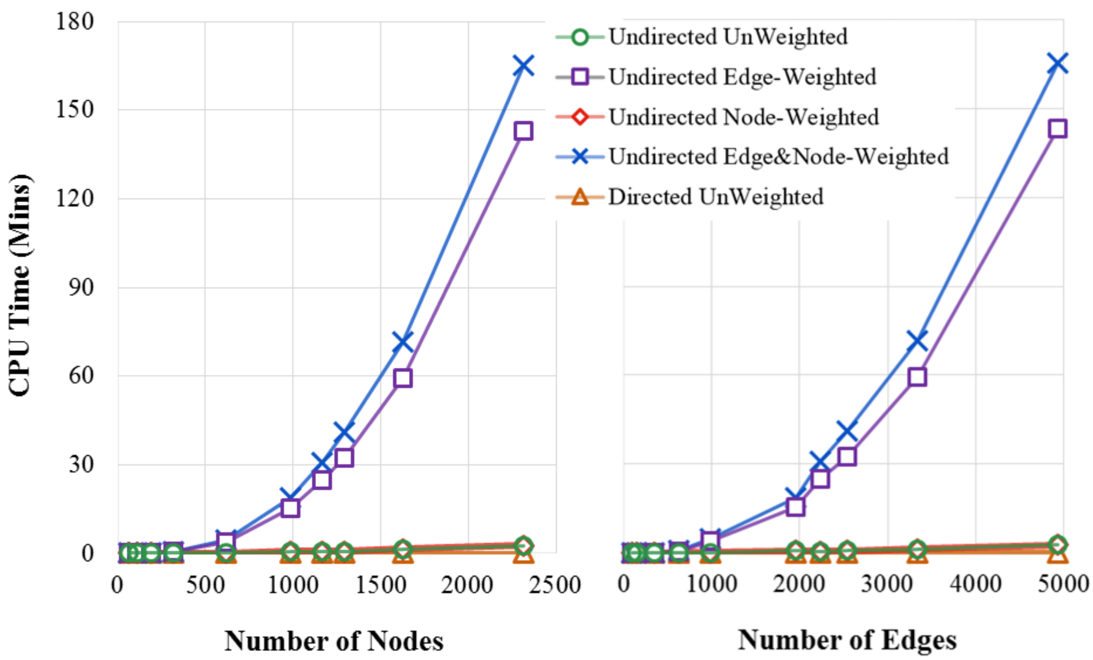
These networks were constructed as follows: firstly 38,131 human PPIs between 9,084 proteins were collected from HPRD (Human Protein Reference Database) [101], secondly 111,152 tissue-protein associations were extracted from HPRD, thirdly the human PPIs were grouped according to their distributed tissues, and lastly the tissue-specific lists of PPIs were processed to find their largest connected components as the human tissue-specific PPI networks. 10 tissue-specific PPI networks were selected with varying number of nodes from 63 to 2317 and varying number of edges from 91 to 4924. Each of the 10 networks was constructed into five different network types. The first four types are undirected unweighted, edge-weighted, node-weighted, and edge-node-weighted networks with their edge-weights or node-weights randomly generated. The fifth type is the directed unweighted network with the direction of each edge tentatively assigned from the left-node to the right-node in the input of SIF Format.

The evaluation of CPU time on running the 10 tested networks was measured on a Dell OptiPlex9010 Intel Core i7-3770 3.4GHz CPU, were summarized in **Table 11** and **Figure 2**. Specifically, the CPU time for the un-weighted network is within 1 minute for a network having no more than 1500 nodes or 3000 edges, the CPU time is less than 3 minutes if the network size is less than 2400 nodes or 5000 edges. On the other hand, the CPU time for the edge-weighted network is less than 30 minutes if the network size is no more than 1,200 nodes or 2,200 edges. The edge-node-weighted network requires the highest computational time, as it costs about 1 hour for the network with 1,600 nodes or 3,300 edges. As there are only 23 descriptors calculated for directed networks, its CPU time is always within 5 seconds for all the testing networks.

Table 11 | CPU time in computing the slim set of PROFEAT network descriptors for ten human tissue-specific PPI networks in five different network types

Tissue	Human Systems	Network Size		CPU Time (Mins) for Different Network Types				
		No of Nodes	No of Edges	Un-Weighted	Edge-Weighted	Node-Weighted	EdgeNode-Weighted	Directed
Lymph Node	Immune	63	91	0.008	0.014	0.009	0.018	0.007
Hippocampus	Nervous	107	146	0.010	0.033	0.011	0.040	0.007
Bone Marrow	Immune	189	348	0.018	0.134	0.018	0.155	0.008
Muscle	Musculoskeletal	315	632	0.044	0.551	0.045	0.660	0.009
Small Intestine	Digestive	616	980	0.189	3.83	0.178	4.66	0.013
Colon	Digestive	988	1951	0.672	15.42	0.653	18.67	0.021
Ovary	Reproductive	1165	2230	0.609	24.87	0.612	30.71	0.026
Spleen	Immune	1292	2543	0.765	32.34	0.761	40.98	0.029
Pancreas	Endocrine	1625	3336	1.38	59.26	1.39	71.45	0.043
Lung	Respiratory	2317	4942	2.68	143.10	2.72	165.33	0.067

Figure 2 | CPU time in computing the slim set of PROFEAT network descriptors for the networks described in **Table 11** with respect to the number of nodes (left) and the number of edges (right)



(F) Typical Applications of Network Descriptors in Systems Biology

Table 12 | List of network descriptors (node-level, network-level, and edge-level) in different categories provided by PROFEAT and the selected systems biological applications

Network Descriptors	Applications in Systems Biology
Connectivity/Adjacency-based Properties	
Node-Level: Degree, Scaled Connectivity, Number of Selfloops/Triangles, Zscore, Clustering Coefficient, Topological Coefficient, Neighborhood Connectivity, Interconnectivity, Degree Centrality, Bridging Coefficient **Network-Level:** Number of Nodes/Edges/Selfloops, Max/Min Connectivity, Average Neighbours, Total Adjacency, Density, Average Clustering Coefficient, Transitivity, Heterogeneity, Degree Centralization, Central Point Dominance, Degree Assortativity Coefficient **Edge-Level:** Unweighted Edge Betweenness	Degree, average neighbours and density implicated the genes in disease network [102]. Neighbourhood connectivity measured the stability of protein/genetic regulatory networks [103]. Interconnectivity prioritized the disease genes [15]. Global clustering coefficient provided molecular characterization in gene co-expression network [104].
Shortest Path Length-based Properties	
Node-Level: Average Shortest Path Length, Eccentric, Eccentricity, Radiality, Distance Sum, Deviation, Distance Deviation, Closeness Centrality, Eccentricity Centrality, Harmonic Centrality, Residual Centrality, Load Centrality, Betweenness Centrality, Bridging Centrality, CurrentFlow Closeness, CurrentFlow Betweenness **Network-Level:** Total Distance, Shape Coefficient, Diameter, Radius, Character. Path Length, Network Eccentricity, Average Eccentricity, Network Eccentric, Eccentric Connectivity, Unipolarity, Integration, Variation, Avg Distance, Mean Distance Deviation, Centralization, Global Efficiency **Edge-Level:** Edge Weight, Weighted Edge Betweenness	Centrality and peripherality (eccentricity, radiality) implicated genes in disease network [102]. Eccentricity and distance deviation identified the metabolic biomarkers [105]. Shortest path length, betweenness, closeness, radiality and integration explored protein-drug interactome for lung cancer [106], identified the hubs and bridging nodes in drug addiction mechanisms [107]. Edge-betweenness facilitated the modularity analysis [28].
Topological Indices	
Node-Level: *N.A.* **Network-Level:** Edge Complexity Index, Randic Connectivity Index, ABC Index, Zagreb Indices, Narumi Indices, Alpha/Beta/Pi/Eta Index, Hierarchy, Robustness, Medium Articulation, Complexity Indices, Wiener Index, Hyper-Wiener, Harary Indices, Compactness, Superpendentic Index, Hyper-Distance-Path Index, BalabanJ, BalabanJ-like Indices, Geometric Arithmetic Indices, Product of Row Sums, Topological Indices, Szeged Index, Efficiency Complexity	Exponent of power-law degree distribution (hierarchy index), provided molecular characterization of cellular state in gene co-expression network [104], characterized the yeast genetic interaction network [108], measured the robustness of protein interaction networks and genetic regulatory networks [103]. Complexity indices and BalabanJ index classified the metabolic networks from 3 domains of life [109].

Table 12 (*continue*) | List of network descriptors (node-level, network-level, and edge-level) in different categories provided by PROFEAT and the selected systems biological applications

Entropy-based Complexity Indices	
Node-Level: *N.A.* **Network-Level:** Entropy on (degree equality/edge equality/edge magnitude/ distance degree/distance degree equality), Radial Centric Information Index, Distance Degree Compactness, Distance Degree Centric Index, Graph Distance Complexity, Info Layer Index, Bonchev Info Indices, Balaban-like Info Indices	Information-theoretic entropy measures identified and ranked the highly discriminating metabolic biomarker candidates for obesity [105]. Radial centric information index, degree equality-information index, and 3 Dehmer's entropy descriptors classified the metabolic networks of 43 organisms from 3 domains of life [109].
Eigenvalue-based Complexity Indices	
Node-Level: Eigenvector Centrality, Page Rank Centrality **Network-Level:** Graph Energy, Laplacian Energy, Spectral Radius, Estrada Index, Laplacian Estrada Index, Quasi-Weiner Index, Mohar Indices, Graph Index Complexity, 50 Dehmer's Entropy by Matrices of (adjacency/laplacian/distance/ distance path/augmented vertex degree/extended adjacency/vertex connectivity/ random walk markov/weighted struct func 1/weighted struct func 2)	PageRank centrality identified prognostic marker genes for pancreatic cancer [110]. The PageRank centrality/degree quotient scored and found the non-hub important nodes in microbial networks from 3 distinct organisms [37].
Edge-Weighted Properties	
Node-Level: Strength, Assortativity, Disparity, Geometric Mean of Triangles, Edge-Weighted Local Clustering Coefficient, Edge-Weighted Interconnectivity **Network-Level:** Weighted Transitivity, Edge-Weighted Global Clustering Coeff	Edge-weighted clustering coefficient identified the gene modules in co-expression network [45]. Edge-weighted interconnectivity ranked the candidate disease genes in biological networks [16].
Node-Weighted Properties	
Node-Level: Node Weight, Node-Weighted Cross Degree, Node-Weighted Local Clustering Coeff, Node-Weighted Neighbourhood Score **Network-Level:** Total Node Weight, Node-Weighted Global Clustering Coefficient	Node-weighted neighbourhood score prioritized the novel disease genes for the prediction of drug targets for a given disease [15].
Directed Properties	
Node-Level: In-Degree, In-Degree Centrality, Out-Degree, Out-Degree Centrality, Directed Local Clustering Coefficient, Neighbourhood Connectivity (in/out/in-&-out), Average Directed Neighbour Degree **Network-Level:** In-Degree (max, avg, min), Out-Degree (max, avg, min), Directed Global Clustering Coefficient	In/out-degree, and clustering coefficient analyzed the gene regulatory networks under different conditions [111]. Directed clustering coefficient and average directed neighbour degree studied the neuro-connectivity networks [8].

(G) Reference

1 Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res*, 13, 2498-2504, (2003).

2 Bastian, M., Heymann, S. & Jacomy, M. Gephi: an open source software for exploring and manipulating networks. *3rd International AAAI Conference on Weblogs and Social Media*, (2009).

3 Reimand, J., Tooming, L., Peterson, H., Adler, P. & Vilo, J. GraphWeb: mining heterogeneous biological networks for gene modules with functional significance. *Nucleic Acids Res*, 36, W452-459, (2008).

4 Djebbari, A. *et al.* NAViGaTOR: large scalable and interactive navigation and analysis of large graphs. *Internet Mathematics*, 7, 314-347, (2011).

5 Wu, J. *et al.* Integrated network analysis platform for protein-protein interactions. *Nat Methods*, 6, 75-77, (2009).

6 Forman, J. J., Clemons, P. A., Schreiber, S. L. & Haggarty, S. J. SpectralNET--an application for spectral graph analysis and visualization. *BMC Bioinformatics*, 6, 260, (2005).

7 Batagelj, V. & Mrvar, A. Pajek – Program for Large Network Analysis. *Connections*, 21, 47-57, (1998).

8 Rubinov, M. & Sporns, O. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52, 1059-1069, (2010).

9 Milo, R. *et al.* Network Motifs Simple Building Blocks of Complex Networks. *Science*, 298, 824-827, (2002).

10 Rodrigue, J. P. *The Geography of Transport Systems*. Third edn, (Routledge, 2013).

11 Barabasi, A. L. & Oltvai, Z. N. Network biology: understanding the cell's functional organization. *Nat Rev Genet*, 5, 101-113, (2004).

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

13 Maslov, S. & Sneppen, K. Specificity and stability in topology of protein networks. *Science*, 296, 910-913, (2002).

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

15 Emig, D. *et al.* Drug target prediction and repositioning using an integrated network-based approach. *PLoS One*, 8, e60618, (2013).

16 Hsu, C. L., Huang, Y. H., Hsu, C. T. & Yang, U. C. Prioritizing disease candidate genes by a gene interconnectedness-based approach. *BMC Genomics*, 12 Suppl 3, S25, (2011).

17 Zhu, C., Kushwaha, A., Berman, K. & Jegga, A. G. A vertex similarity-based framework to discover and rank orphan disease-related genes. *BMC Syst Biol*, 6 Suppl 3, S8, (2012).

18 Hwang, W., Cho, Y., Zhang, A. & Ramanathan, M. Bridging Centrality: Identifying Bridging Nodes In Scale-free Networks. *12th ACM International Conference on Knowledge Discovery and Data Mining*, (2006).

19 Freeman, L. C. Centrality in Social Networks Conceptual Clarification. *Social Networks*, 1, 215-239, (1978).

20 Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269-271, (1959).

21 Skorobogatov, V. A. & Dobrynin, A. A. Metrical analysis of graphs. *MATCH Commun Math Comp Chem*, 23, 105-155, (1988).

22 Brandes, U. A faster algorithm for betweenness centrality. *J Math Sociol*, 25, (2001).

23 Sabidussi, G. The centrality index of a graph. *Psychometrika*, 31, 581-603, (1966).

24 Newman, M. E. J. A measure of betweenness centrality based on random walks. *Social Networks*, 27, (2003).

25 Rochat, Y. Closeness Centrality Extended to Unconnected Graphs: The Harmonic Centrality Index. *ASNA*, (2009).

26 Dangalchev, C. Residual Closeness in Networks. *Physica A: Statistical Mechanics and its Applications*, 365, 556-564, (2006).

27 Shimbel, A. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15, 501-507, (1953).
28 Yoon, J., Blumer, A. & Lee, K. An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality. *Bioinformatics*, 22, 3106-3108, (2006).
29 K. Stephenson, M. Z. Rethinking centrality: methods and examples. *Social Networks*, 11, 1-37, (1989).
30 Newman, M. E. J. Betweenness centrality based on random walks. *Social Networks*, 27, 39-54, (2005).
31 Borgatti, S. P. Centrality and network flow. *Social Networks*, 27, 55-71, (2005).
32 Straffin, P. D. Linear Algebra in Geography: Eigenvectors of Networks. *Mathematics Magazine*, 53, 269-276, (1980).
33 Newman, M. E. J. *Mathematics of Networks*. 2nd edn, (Palgrave Macmillan, 2008).
34 Langville, A. N. & Meyer, C. D. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. (Princeton University Press, 2012).
35 Michael, W. B. & Murray, B. *Understanding Search Engines: Mathematical Modeling and Text Retrieval*. Vol. 8 (Society for Industrial and Applied Mathematics, 1999).
36 Brin, S. & Page, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *7th International World-Wide Web Conference*, (1998).
37 Banky, D., Ivan, G. & Grolmusz, V. Equal opportunity for low-degree network nodes: a PageRank-based method for protein target identification in metabolic graphs. *PLoS One*, 8, e54204, (2013).
38 Gleich, D. F. PageRank beyond the Web. *SIAM Rev*, 57, 321-363, (2014).
39 Ivan, G. & Grolmusz, V. When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics*, 27, 405-407, (2011).
40 Barrat, A., Barthelemy, M., Pastor-Satorras, R. & Vespignani, A. The architecture of complex weighted networks. *Proc Natl Acad Sci U S A*, 101, 3747-3752, (2004).
41 Leung, C. C. & Chau, H. F. Weighted Assortative And Disassortative Networks Model. *Physica A: Statistical Mechanics and its Applications*, 378, 591-602, (2008).
42 Barthélemy, M. Architecutre of Complex Weighted Networks. *Talk Series on Networks and Complex Systems*, (2005).
43 Saramäki, J., Kivelä, M., Onnela, J. P., Kaski, K. & Kertész, J. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75, 027105, (2007).
44 Onnela, J. P., Saramäki, J., Kertész, J. & Kaski, K. Intensity and Coherence of Motifs in Weighted Complex Networks. *Physical Review E*, 71, (2005).
45 Zhang, B. & Horvath, S. A general framework for weighted gene co-expression network analysis. *Stat Appl Genet Mol Biol*, 4, Article17, (2005).
46 Holme, P., Park, S. M., Kim, B. J. & Edling, C. R. Korean University Life in a Network Perspective: Dynamics of a Large Affiliation Network. *Physica A: Statistical Mechanics and its Applications*, 373, 821-830, (2007).
47 Wiedermann, M., Donges, J. F., Heitzig, J. & Kurths, J. Node-weighted interacting network measures improve the representation of real-world complex systems. *EPL (Europhysics Letters)*, 102, 28007, (2013).
48 Bonchev, D. *Complexity in Chemistry, Biology, and Ecology*. (Springer US, 2007).
49 Dong, J. & Horvath, S. Understanding network concepts in modules. *BMC Syst Biol*, 1, 24, (2007).
50 Mangioni, G. *Complex Networks: Results of the 1st International Workshop on Complex Networks*. 1st edn, Vol. 207 (Springer-Verlag Berlin Heidelberg, 2009).
51 Newman, M. E. J. Mixing patterns in networks. *Physical Review E*, 67, (2003).
52 Petitjean, M. Applications of the Radius-Diameter Diagram to the Classification of Topological and Geometrical Shapes of Chemical Compounds. *J. Chem. Inf. Comput. Sci.*, 32, 331-337, (1992).
53 Sharma, V., Goswami, R. & Madan, A. K. Eccentric Connectivity Index: A Novel Highly Discriminating Topological Descriptor for Structure-Property and Structure-Activity Studies. *J. Chem. Inf. Comput. Sci.*, 37, 273-282, (1997).

54 Latora, V. & Marchiori, M. Efficient behavior of small-world networks. *Physical Review Letters*, 87, (2001).

55 Li, X. & Gutman, I. *Mathematical Aspects of Randic-Type Molecular Structure Descriptors*. (University of Kragujevac, 2006).

56 Furtula, B., Graovac, A. & Vukičević, D. Atom-Bond Connectivity Index of Trees. *Discrete Applied Mathematics*, 157, 2828-2835, (2009).

57 Diudea, M. V., Gutman, I. & Lorentz, J. *Molecular Topology*. (Nova Publishing, 2001).

58 Vukičević, D. & Furtula, B. Topological Index Based on The Ratios of Geometrical and Arithmetical Means of End-Vertex Degrees of Ddges. *Journal of Mathematical Chemistry*, 46, 1369-1376, (2009).

59 Khalifeh, M. H., Yousefi-Azari, H. & Ashrafi, A. R. The First and Second Zagreb Indices of Some Graph Operations. *Discrete Appl. Math.*, 157, 804-811, (2009).

60 Gutman, I. Graph theory and molecular orbitals. XII. Acyclic polyenes. *The Journal of Chemical Physics*, 62, 3399, (1975).

61 Narumi, H. & Katayama, M. *Simple Topological Index. A Newly Devised Index Characterizing the Topological Nature of Structural Isomers of Saturated Hydrocarbons*, Hokkaido Univ. Japan, (1984).

62 Narumi, H. New Topological Indices for Finite and Infinite Systems. *MATCH Commun. Math. Chem.*, 22, 195-207, (1987).

63 Piraveenan, M., Uddin, S. & Chung, K. S. K. in *Advances in Social Networks Analysis and Mining, IEEE* 38-45 (IEEE, Istanbul, 2012).

64 Kim, J. & Wilhelm, T. What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, 387, 2637-2652, (2008).

65 Dehmer, M., Grabner, M. & Furtula, B. Structural discrimination of networks by using distance, degree and eigenvalue-based measures. *PLoS One*, 7, e38564, (2012).

66 Dehmer, M., Grabner, M. & Varmuza, K. Information indices with high discriminative power for graphs. *PLoS One*, 7, (2012).

67 Wiener, H. Structural Determination of Paraffin Boiling Points. *Journal of the American Chemical Society*, 69, 17-20, (1947).

68 Klein, D. J., Lukovits, I. & Gutman, I. On the Definition of the Hyper-Wiener Index for Cycle-Containing Structures. *J. Chem. Inf. Comput. Sci.*, 35, 50-62, (1995).

69 Balaban, A. *Topological Indices and Related Descriptors in QSAR and QSPAR*. (CRC Press, 2000).

70 Doyle, J. K. & Graver, J. E. Mean Distance in a Graph. *Discrete Mathematics*, 17, (1977).

71 Gupta, S. Superpendentic Index: A Novel Topological Descriptor for Predicting Biological Activity. *J. Chem. Inf. Comput. Sci.*, 39, 272-277, (1999).

72 Diudea, M. V. Walk Numbers eWM: Wiener-Type Numbers of Higher Rank. *J. Chem. Inf. Comput. Sci.*, 36, 535-540, (1996).

73 Todeschini, R. & Consonni, V. *Handbook of Molecular Descriptors*. (Wiley VCH, 2008).

74 Balaban, A. Highly Discriminating Distance-Based Topological Index. *Chemical Physics Letters*, 89, (1982).

75 Bono Lucic *et al. On the Novel Balaban-Like and Balaban-Detour-Like Molecular Descriptors*. Vol. 5 (Nova Science Publishers, 2013).

76 Zhou, B., Gutman, I., Furtula, B. & Du, Z. B. On Two Types of Geometric–Arithmetic Index. *Chemical Physics Letters*, 482, 153-155, (2009).

77 Khadika, P. V. The Szeged Index and an Analogy with the Wiener Index. *J. Chem. Inf. Comput. Sci.*, 35, 547-550, (1995).

78 Schultz, H. P., Schultz, E. B. & Schultz, T. P. Topological Organic Chemistry. 4. Graph theory, Matrix Permanents, and Topological Indices of Alkanes. *J. Chem. Inf. Comput. Sci.*, 32, 69-72, (1992).

79 Schultz, H. P. Topological Organic Chemistry. 1. Graph Theory and Topological Indices of Alkanes. *J. Chem. Inf. Comput. Sci.*, 29, 227-228, (1989).

80 Gutman, I. Selected properties of the Schultz molecular topological index. *J. Chem. Inf. Comput. Sci.*, 34, 1087-1089, (1994).

81 Leroy, G. *Information Theoretic Indices for Characterization of Chemical Structures*. Vol. 27 (Wiley, 1985).

82 Bonchev, D., Mekenyan, O. V. & Trinajstii, N. Isomer Discrimination by Topological Information Approach. *Journal of Computational Chemistry*, 2, 127-148, (1981).

83 Balaban, A. T., Bertelsen, S. & Basak, S. C. New centric topological indexes for acyclic molecules (trees) and substituents (rooted trees) and coding of rooted trees. *MATCH Communications in Mathematical and in Computer Chemistry*, 30, 55-72, (1994).

84 Dehmer, M., Sivakumar, L. & Varmuza, K. *On Distance-Based Entropy Measures*. Vol. 12 (University of Kragujevac, 2012).

85 Raychaudhury, C., Ray, S. K., Ghosh, J. J., Roy, A. B. & Basak, S. C. Discrimination of Isomeric Structures Using Information Theoretic Topological Indices. *Journal of Computational Chemistry*, 5, 581-588, (1984).

86 Konstantinova, E. V. & Paleev, A. A. Sensitivity of Topological Indices of Polycyclic Graphs. *Vychisl Sistemy*, 136, 38-48, (2006).

87 Bertz, S. H. The First General Index of Molecular Complexity. *J. Am. Chem. Soc.*, 103, 3599-3601, (1981).

88 Bonchev, D. & Trinajstic, N. Information theory, distance matrix, and molecular branching. *The Journal of Chemical Physics*, 67, (1977).

89 Balaban, A. T. New Vertex Invariant and Topological Indices of Chemical Graphs Based on Information on Distances. *Journal of Mathematical Chemistry*, 8, 383-397, (1991).

90 Dehmer, M. & Mowshowitz, A. A history of graph entropy measures. *Information Sciences*, 181, 57-78, (2011).

91 Gutman, I. & Zhou, B. Laplacian energy of a graph. *Linear Algebra and its Applications*, 414, 29-37, (2006).

92 Gradshteyn, I. S. & Ryzhik, I. M. *Tables of Integrals, Series, and Products*. 6th edn, 1115-1116 (Academic Press, 2000).

93 Estrada, E. Characterization of 3D Molecular Structure. *Chemical Physics Letters*, 319, 713-718, (2000).

94 Fath-Tabar, G. H., Ashra, A. R. & Gutman, I. Note on Estrada and L-Estrada Indices of Graphs. *Classe des Sciences Mathematiques et Naturelles, Sciences mathematiques naturelles*, CXXXIX, 1-16, (2009).

95 Karacali, B. Hierarchical Motif Vectors for Prediction of Functional Sites in Amino Acid Sequences Using Quasi-Supervised Learning. *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, 9, 1432-1441, (2012).

96 Mohar B., Babic D. & N., T. A Novel Definition of the Wiener Index for Trees. *J. Chem. Inf. Comput. Sci.*, 33, 153-154, (1993).

97 Dehmer, M., Emmert-Streib, F., Tsoy, Y. R. & Varmuza, K. *Quantifying Structural Complexity of Graphs: Information Measures in Mathematical Chemistry*. (Nova Science Publishers, 2011).

98 Dehmer, M. Uniquely discriminating molecular structures using novel eigenvalue-based descriptors. *MATCH Commun. Math. Comput. Chem.*, 67, 147-172, (2012).

99 Luo, J. & Magee, C. L. Detecting evolving patterns of self-organizing networks by flow hierarchy measurement. *Complexity*, 16, 53-61, (2011).

100 Brandes, U. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30, 136-145, (2008).

101 Keshava Prasad, T. S. *et al.* Human Protein Reference Database--2009 update. *Nucleic Acids Res*, 37, D767-772, (2009).

102 Goh, K. I. *et al.* The human disease network. *Proc Natl Acad Sci U S A*, 104, 8685-8690, (2007).

103 Maslov, S. & Sneppen, K. Specificity and stability in topology of protein networks. *Science*, 296, 910-913, (2002).

104 Carter, S. L., Brechbuhler, C. M., Griffin, M. & Bond, A. T. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20, 2242-2250, (2004).

105 Netzer, M. *et al.* A network-based feature selection approach to identify metabolic signatures in disease. *J Theor Biol*, 310, 216-222, (2012).

106 Peng, X. *et al.* Exploring a structural protein-drug interactome for new therapeutics in lung cancer. *Mol Biosyst*, 10, 581-591, (2014).

107 Pang, K., Cheng, C., Xuan, Z., Sheng, H. & Ma, X. Understanding protein evolutionary rate by integrating gene co-expression with protein interactions. *BMC Syst Biol*, 4, 179, (2010).
108 Tong, A. H. *et al.* Global mapping of the yeast genetic interaction network. *Science*, 303, 808-813, (2004).
109 Mueller, L. A., Kugler, K. G., Netzer, M., Graber, A. & Dehmer, M. A network-based approach to classify the three domains of life. *Biol Direct*, 6, 53, (2011).
110 Winter, C. *et al.* Google goes cancer: improving outcome prediction for cancer patients by network-based ranking of marker genes. *PLoS Comput Biol*, 8, e1002511, (2012).
111 Luscombe, N. M. *et al.* Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*, 431, 308-312, (2004).